

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**A Máquina Geométrica - Modelo
Computacional para Concorrência e
Não-Determinismo Usando como
Estrutura Espaços Coerentes**

por

RENATA HAX SANDER REISER

Tese submetida à avaliação,
como requisito parcial para a obtenção do grau de
Doutor em Ciência da Computação

Prof. Dr. Antônio Carlos da Rocha Costa
Orientador

Prof. Dr. Dalcídio Moraes Claudio
Co-orientador

Porto Alegre, março de 2002.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Reiser, Renata Hax Sander

A Máquina Geométrica - Modelo Computacional para Concorrência e Não-Determinismo Usando como Estrutura Espaços Coerentes / por Renata Hax Sander Reiser. — Porto Alegre: PPGC da UFRGS, 2002.

258 f.: il.

Tese (doutorado) — Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2002. Orientador: Costa, Antônio Carlos da Rocha; Co-orientador: Dalcídio Moraes, Claudio.

1.Espaços Coerentes. 2. Domínios Semânticos. I. Costa, Antônio Carlos da Rocha. II. Dalcídio Moraes, Claudio. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Por acreditar na ciência e nos homens, descobri que a maior força esta na crença em mim mesmo, na compreensão de minhas limitações e na busca em superá-las. Esta tese faz parte desta descoberta, cuja conquista só foi possível pela cooperação e apoio que recebi. Agradeço, de forma especial

- ao Paulo, meu esposo, cujo amor e dedicação foram minha principal motivação e segurança, compartilhando as responsabilidades e alegrias desta conquista;
- às minhas amadas filhas, Carolina e Cristina, que apreenderam desde pequenas a superar a ausência, preservando o amor, o carinho e a amizade que nos une;
- a meus familiares, em especial aos meus pais Ronaldo e Jacy, a minha sogra D. Iria e a querida tia Norma pelo respeito, dedicação e incentivo;
- a minha amiga, grande incentivadora profa. Dra. Graçaliz Pereira Dimuro, que me inicializou na pesquisa científica, cujo exemplo pessoal e profissional sempre me inspirou profunda admiração e cujas orientações me foram valiosas;
- ao meu orientador prof. Dr. Antônio Carlos da Rocha Costa, pela sua capacidade e competência aliada a um grande entusiasmo e incansável dedicação, com sabedoria e muita paciência soube me conduzir, concedendo o tempo necessário para que esta tese fosse construída com muito prazer;
- ao meu co-orientador, prof. Dr. Dalcídio Moraes Claudio, que me introduziu no PPGC, cujos conselhos e confiança têm me guiado desde lá;
- aos colegas de curso e de forma muito carinhosa ao grande amigo prof. Marilton Sanchotene Aguir, pela sua estimada amizade, respeito e apoio técnico;
- ao grupo de Fundamentos da Computação da UCPel, aos meus alunos e afilhados, em especial aqueles que se tornaram grandes amigos e colegas de profissão, Ana Paula e Luis Fernando, Nisia, Leda, Rafael, Fábio e Renata, André e Juliana, Dario, que sempre me incentivaram;
- aos diretores, professores, coordenadores de cursos, e funcionários das Escolas de Educação e de Informática da UCPEL, em especial aos profs. Francisco de Paula M. Rodrigues e Regina T. Xavier e à secretária Elisabete B. Aguiar;
- aos professores, funcionários e membros da reitoria e pró-reitorias da UCPEL;
- ao grupo de pesquisa GMC, aos funcionários da biblioteca e ao corpo docente do PPGC da UFRGS, em especial à profa Dra. Laira Vieira Toscani e aos profs. Dr. Paulo F. Blauth Menezes, Dr. Tiarajú Asmuz Divério e Dr. Alfio Ricardo Martini;
- ao prof. Dr. Edward Hermann Haeusler, pelas importantes sugestões que acompanham este trabalho desde seu início, ainda no mestrado;
- ao amigo Dr. Gustavo Valle pelas proveitosas conversas pessoais;
- às fundações CAPES e FAPERGS, pelos auxílios em forma de bolsa;
- à fé em Deus, que me ajuda a não temer a redenção e buscar sempre renovação.

Sumário

Lista de Abreviaturas	10
Lista de Símbolos	13
Lista de Figuras	14
Lista de Tabelas	16
Resumo	17
Abstract	18
1 Introdução	19
1.1 Conceção da Idéia	19
1.2 Objetivo e Etapas de Desenvolvimento	20
1.3 Estrutura Ordenada do Modelo	21
1.3.1 Espaço Coerente \mathbb{D}_n dos Processos Computacionais	21
1.3.2 Espaço Coerente \mathbb{D}_∞ dos Processos Computacionais.	22
1.3.3 Espaço Coerente $\mathbb{D}_{2\infty}$ dos Processos Transfinitos	23
1.4 Interpretação Semântica do Modelo	24
1.5 Linguagem do Modelo	24
1.6 Estruturação do Texto	25
2 Histórico e Fundamentação	27
2.1 Modelos de Máquinas	27
2.1.1 Máquina de Turing	28
2.1.2 Máquinas de Registradores.	29
2.1.3 Máquina de Grafos	32
2.2 Modelos Semânticos	35
2.3 Semântica Denotacional das Linguagens de Programação Determinísticas	36
2.3.1 Domínios de Scott	37
2.3.2 Contribuições Posteriores	38
2.3.3 Teoria dos Domínios e Matemática Intervalar	38
2.4 Semântica Denotacional das Linguagens de Programação Não-Determinísticas	39
2.4.1 Semântica Algébrica	40
2.4.2 Estruturas de Eventos	41
2.4.3 Domínios-Potência	41
2.4.4 Semântica Parcialmente Aditiva.	42
2.5 Semântica dos Tipos nas Linguagens de Programação	42
2.5.1 Teoria dos Tipos em Programação	42
2.5.2 Tipos Simples como Conjuntos	43
2.5.3 Tipos Simples como Domínios	43
2.5.4 dI-Domínios	44
2.5.5 Espaços Coerentes: Domínios de Girard	44
2.5.6 Isomorfismo de Curry-Howard	46

3 Estados e Processos Computacionais no Modelo de Máquina Geométrica	48
3.1 Estados Computacionais no Modelo MG	49
3.1.1 Estados Computacionais Determinísticos	49
3.1.2 Estados Computacionais	50
3.2 Processos Computacionais no Modelo MG	50
3.2.1 Processo Identidade-Default	51
3.2.2 Processo Computacional Elementar	52
3.2.3 Testes Computacionais	53
3.2.4 Função Posição $\Upsilon_{[S \rightarrow S]}$	54
3.3 Construtores de Processos	54
3.3.1 Construtor Identidade	55
3.3.2 Produto Seqüencial	55
3.3.3 Soma Determinística	56
3.3.4 Produto Paralelo	56
3.3.5 Soma Não-Determinística	57
3.3.6 Construção Indutiva do Conjunto de Processos Computacionais	58
4 Nível $\mathbb{D}_0 - \mathbb{D}_1$ da Construção do Espaço Coerente de Processos Gerados Indutivamente	60
4.1 Espaço Coerente \mathbb{D}_0 dos Processos Elementares.	60
4.1.1 Função Posição em \mathbb{D}_0	61
4.2 Espaço Coerente $\bar{\mathbb{D}}_0$ do PP de Processos Elementares	62
4.2.1 Domínio Plano do PP de Processos Elementares	63
4.2.2 Função Posição em $\bar{\mathbb{D}}_0$	64
4.2.3 Relação de Coerência Estrita em $\bar{\mathbb{D}}_0$	64
4.3 Espaço Coerente $\bar{\mathbb{D}}_0^\perp$ da SND de Processos Elementares	65
4.3.1 Domínio Plano da SND de Processos Elementares	66
4.3.2 Função Posição em $\bar{\mathbb{D}}_0^\perp$	67
4.4 Espaço Coerente \mathbb{P}_0	67
4.4.1 Imersões e Projeções do Espaço Poerente \mathbb{P}_0	70
4.4.2 Função Posição em \mathbb{P}_0	72
4.5 Espaço Coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$	73
4.5.1 Projeções e Imersões do Espaço Coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$	76
4.5.2 Função Posição em $\mathbb{P}_0 \amalg \mathbb{P}_0$	76
4.6 Espaço Coerente $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$	77
4.6.1 Espaço Coerente Plano de Testes Elementares	77
4.6.2 Projeções e Imersões do $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$	80
4.6.3 Função Posição em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$	80
4.7 Espaço Coerente \mathbb{D}_1	81
4.7.1 Imersões e Projeções do Subnível $\mathbb{P}_0 - \mathbb{D}_1$	83
4.7.2 Imersões e Projeções do Primeiro Nível $\mathbb{D}_0 - \mathbb{D}_1$	84
4.7.3 Função Posição em \mathbb{D}_1	86
5 Nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ da Construção da Estrutura Ordenada do Modelo MG	89
5.1 Nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ da Construção	89
5.1.1 Espaço Coerente \mathbb{D}_n	89
5.1.2 Função Posição em \mathbb{D}_n	90
5.1.3 Espaço Coerente $\bar{\mathbb{D}}_n$	90
5.1.4 Espaço Coerente $\bar{\mathbb{D}}_n^\perp$	91

5.2	Imersões e Projeções do Nível $\mathbb{D}_n - \mathbb{D}_{n+1}$	91
5.2.1	Imersões $\pi_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$	91
5.2.2	Projeções $\Pi_n^{(\theta)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$	94
5.2.3	Exemplificação da Aplicação das Funções $\pi_n^{(\theta)}$ e $\Pi_n^{(\theta)}$	95
5.3	Imersões $\pi_n^{\rightarrow}, \pi_n^{\leftarrow}$ e Projeções $\Pi_n^{\rightarrow}, \Pi_n^{\leftarrow}$ em $\mathbb{D}_n - \mathbb{D}_{n+1}$	102
5.3.1	Imersões $\pi_n^{\leftarrow(\alpha)}, \pi_n^{\rightarrow(\alpha)}$	102
5.3.2	Projeções $\Pi_n^{\leftarrow(\alpha)}, \Pi_n^{\rightarrow(\alpha)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$	104
5.4	Imersões e Projeções π_n^{\leftrightarrow} e Π_n^{\leftrightarrow}	107
5.4.1	Imersões $\pi_n^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$	107
5.4.2	Projeções $\Pi_n^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$	109
5.5	Imersões e Projeções $\pi_n^{\rightarrow\leftarrow}$ e $\Pi_n^{\rightarrow\leftarrow}$	116
5.5.1	Imersões $\pi_n^{\rightarrow\leftarrow}$	116
5.5.2	Projeções $\Pi_n^{\rightarrow\leftarrow}$	118
5.6	Imersões e Projeções π_n^{\rightleftarrows} e Π_n^{\rightleftarrows}	121
5.6.1	Imersões π_n^{\rightleftarrows}	121
5.6.2	Projeções Π_n^{\rightleftarrows}	122

6 Completação do Espaço Coerente de Processos Gerados Indutivamente

6.1	Espaço Coerente \mathbb{D}_∞ dos Processos Computacionais	126
6.1.1	Elementos Compactos do Domínio \mathbb{D}_∞	127
6.1.2	Objetos Totais do Domínio \mathbb{D}_∞	127
6.2	Espaço Coerente $\mathbb{D}_\infty^{(\theta)}$	130
6.2.1	Imersões $\pi_{n,\infty}^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{(\theta)}$	130
6.2.2	projeções $\Pi_{\infty,n}^{(\theta)}$	133
6.3	Espaços Coerentes $\mathbb{D}_\infty^{\rightarrow}$ e $\mathbb{D}_\infty^{\leftarrow}$	139
6.3.1	Imersões $\pi_{n,\infty}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\rightarrow}$ e $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftarrow}$	139
6.3.2	Projeções $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \mathbb{D}_n$ e $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty^{\leftarrow} \rightarrow \mathbb{D}_n$	140
6.3.3	Cone Limite $(\mathbb{D}_\infty^{\rightarrow}, \Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \mathbb{D}_n)$	143
6.3.4	Cone Limite $(\mathbb{D}_\infty^{\leftarrow}, \Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty^{\leftarrow} \rightarrow \mathbb{D}_n)$	144
6.4	Espaço Coerente $\mathbb{D}_\infty^{\leftrightarrow}$	145
6.4.1	Imersões $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$	146
6.4.2	Projeções $\Pi_{\infty,n}^{\leftrightarrow}$	147
6.4.3	Cone Limite $(\mathbb{D}_\infty^{\leftrightarrow}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n)$	150
6.5	Espaço Coerente $\mathbb{D}_\infty^{\rightarrow\leftarrow}$	153
6.5.1	Imersões $\pi_{n,\infty}^{\rightarrow\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\rightarrow\leftarrow}$	153
6.5.2	Projeções $\Pi_{\infty,n}^{\rightarrow\leftarrow}$	154
6.5.3	Cone Limite $(\mathbb{D}_\infty^{\rightarrow\leftarrow}, \Pi_{\infty,n}^{\rightarrow\leftarrow} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n)$	156

7 Espaço Coerente $\mathbb{D}_{2\infty}^{\rightarrow}$ dos Processos Computacionais Transfinitos

7.1	Espaço Coerente $\mathbb{D}_{\infty+1}^{\rightarrow}$	161
7.1.1	Imersões e Projeções $\pi_n^{\frac{1}{2}}$ e $\Pi_n^{\frac{1}{2}}$	162
7.1.2	Imersões $\pi_{n,\infty+1}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty+1}^{\rightarrow}$ e Projeções $\Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$	164
7.1.3	Cone Limite $(\mathbb{D}_{\infty+1}^{\rightarrow}, \Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n)$	166
7.1.4	Relacionamento entre $\mathbb{D}_\infty^{\rightarrow}$ e $\mathbb{D}_{\infty+1}^{\rightarrow}$	167
7.1.5	Relacionamento entre os Domínios $\mathbb{D}_{\infty+1}^{\rightarrow}, \mathbb{D}_\infty^{\leftrightarrow}$ e $\mathbb{D}_\infty^{\rightarrow\leftarrow}$	168
7.2	Espaço Coerente $\mathbb{D}_{\infty+m}^{\rightarrow}$	169
7.2.1	Imersões $\pi_n^{\frac{m}{2}}$ e Projeções $\Pi_n^{\frac{m}{2}}$ em $\mathbb{D}_n - \mathbb{D}_{n+1}$	169
7.2.2	Imersões $\pi_{n,\infty+m}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty+m}^{\rightarrow}$ e Projeções $\Pi_{\infty+m,n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$	173

7.2.3	Cone Limite ($\mathbb{D}_{\infty+m}^{\rightarrow}, \Pi_{\infty+m, n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$)	173
7.2.4	Relacionamento entre $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{\infty+m+1}^{\rightarrow}$	174
7.3	Espaço Coerente $\mathbb{D}_{2\infty}^{\rightarrow}$	175
7.3.1	Imersões $\pi_{\infty+m, 2\infty}^{\rightarrow}$ e Projeções $\Pi_{2\infty, \infty+m}^{\rightarrow}$	175
8	Interpretação dos Construtores de Processos no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$ dos Processos Computacionais	179
8.1	Construtores de Processos Modelados pela Construção Espacial do Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$	179
8.1.1	Operador Identidade $Id_{\mathbb{D}_{\infty}^{\rightarrow}}$	179
8.1.2	Operador $\mathbf{F}_{(00)}$	179
8.1.3	Operador $\mathbf{F}_{(10)}$	180
8.1.4	Operador $\mathbf{F}_{(20)}$	182
8.2	Construtores de Processos Modelados pela Construção Temporal do Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$	186
8.2.1	Funções Deslocamento	186
8.2.2	Abordagem Algébrica para Interpretações nos Domínios $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$	191
8.2.3	Interpretação Gráfica para Funções Deslocamento nos Domínios $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$	193
8.2.4	Operador $\mathbf{F}_{(01)(11)}$	195
8.2.5	Operador $\mathbf{F}_{(02)(12)(22)}$	197
8.3	Composição dos Construtores no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$	200
9	Linguagem de Processos Baseada no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$ dos Processos Computacionais	208
9.1	Assinatura da linguagem	208
9.1.1	Termos Bem Formados da Assinatura Σ	209
9.2	Especificação Equacional da Linguagem	209
9.2.1	Função de Representação $\mathfrak{R}^* : \pi_{n, \infty}^{\rightarrow}[\mathbb{D}_n] \subseteq \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \Omega^*$	210
9.2.2	Relação de Ordem em Ω^*	213
9.2.3	Extensão da Função de Representação	213
9.2.4	Aplicação da Função de Representação no Domínio $\mathbb{D}_{\infty}^{\rightarrow}$	216
9.2.5	Propriedades dos operadores em Ω .	217
9.3	Equações Recursivas em Ω	218
10	Interpretação Semântica Baseada no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$ dos Processos Computacionais	222
10.1	Espaço Coerente \mathbb{S} dos Estados Computacionais	222
10.1.1	Espaço Coerente dos Nomes das Variáveis Computacionais	223
10.1.2	Espaço Coerente dos Valores das Variáveis Computacionais	223
10.1.3	Espaço Coerente $\mathbb{I} \multimap \mathbb{V}$	223
10.1.4	Espaço Coerente dos Estados Computacionais	225
10.1.5	Espaço Coerente $\mathbb{S} \multimap \mathbb{S}$	225
10.2	Função de Avaliação	226
11	Considerações Finais	231
11.1	Conclusões	231
11.1.1	Modelo MG	231
11.1.2	Diversidade dos Modelos MG	232
11.1.3	Modelo MGD	233
11.2	Propostas para Continuação dos Trabalhos	233
11.2.1	Relacionamento entre as Memórias Infinita e Transfinita no modelo MG	233
11.2.2	Classificação do Modelo MG na Hierarquia dos Níveis de Abstração para Modelos Semânticos	234

11.2.3	Interpretações para Construções Recursivas	235
11.2.4	Abordagem Semântica Baseada na Teoria dos Tipos	236
11.2.5	Extensão da Linguagem com Representação para Processos Distribuídos . .	236
11.2.6	Aplicações do Modelo na Computação Científica	236
Anexo	Aplicações da Função de Representação	237
Bibliografia	250

Lista de Abreviaturas

MMG	modelo de Máquina Geométrica
<i>cosp</i>	espaço coerente
PS	produto seqüencial
SD	soma determinística
PP	produto paralelo
SND	soma não-determinística
<i>utc</i>	unidade de tempo computacional
<i>Cprefix</i>	construção temporal de processos por prefixação
<i>Cinfix</i>	construção temporal de processos por infixação
<i>Csufix</i>	construção temporal de processos por sufixação

Lista de Símbolos

$\mathit{CospLin}$	Categoria dos espaços coerentes e funções lineares
V	Conjunto dos valores computacionais
I	Conjunto dos indexadores de pontos do espaço geométrico
S	Conjunto dos estados computacionais determinísticos
$\wp(S)$	Conjunto dos estados computacionais
\mathcal{A}_i	Subfamília de ações rotuladas pela posição i em I
\mathcal{A}	Família de ações rotuladas por posições em I
\mathcal{P}	Conjunto dos processos computacionais
\mathbf{p}	Processo computacional
skip	Processo identidade-default em $S \rightarrow S$
\mathbf{skip}	Processo identidade-default em $\wp(S) \rightarrow \wp(S)$
\mathcal{D}	Conjunto dos processos elementares
$d^{(n)}$	Processo elementar que executa a ação d e altera a posição n
$\mathit{Id}_{\mathcal{P}}$	Construtor identidade de processos
\cdot	Construtor PP
$+$	Construtor SD
$ $	Construtor SND
$ $	Construtor PP
\mathbb{D}_0	cosp dos processos elementares
\mathbb{D}_n	cosp dos processos finitos, executados em $2^n \mathit{utc}$
$\mathbb{D}_n - \mathbb{D}_{n+1}$	$n + 1$ etapa da estrutura ordenada do MMG
$\bar{\mathbb{D}}_n$	cosp do PP de processos finitos, executados em $2^n \mathit{utc}$
$\approx \bar{\mathbb{D}}_n$	Relação de coerência em $\bar{\mathbb{D}}_n$
$\bar{\mathbb{D}}_n^\perp$	cosp da SND de processos executados em $2^n \mathit{utc}$
$\approx \bar{\mathbb{D}}_n^\perp$	Relação de incoerência em \mathbb{D}_n
$\bar{\mathbb{D}}_n^{\perp\perp}$	cosp da SND de processos executados em $2^n \mathit{utc}$
$\approx \bar{\mathbb{D}}_n^{\perp\perp}$	Relação de incoerência em \mathbb{D}_n
$\Upsilon_{\mathbb{P}_n}$	Função posição definida em \mathbb{P}_n
$\mathbb{P}_n \amalg \mathbb{P}_n$	cosp do PS entre processos, executados em $2^n \mathit{utc}$
\mathbb{B}	cosp dos testes computacionais
$\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$	cosp da SD de processos executados em $2^n \mathit{utc}$
$\gamma_n^{(0)}$	Imersão de \mathbb{P}_n para \mathbb{D}_{n+1}
$\gamma_n^{(1)}$	Imersão de $\mathbb{P}_n \amalg \mathbb{P}_n$ para \mathbb{D}_{n+1}
$\gamma_n^{(2)}$	Imersão de $\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n$ para \mathbb{D}_{n+1}
$\Gamma_n^{(0)}$	Projeção de \mathbb{D}_{n+1} para \mathbb{P}_n
$\Gamma_n^{(1)}$	Projeção de \mathbb{D}_{n+1} para $\mathbb{P}_n \amalg \mathbb{P}_n$
$\Gamma_n^{(2)}$	Projeção de \mathbb{D}_{n+1} para $\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n$
$\phi_n^{(0)}$	Imersão de \mathbb{D}_n para \mathbb{P}_n
$\phi_n^{(1)}$	Imersão de $\bar{\mathbb{D}}_n$ para \mathbb{P}_n
$\phi_n^{(2)}$	Imersão de $\bar{\mathbb{D}}_n^\perp$ para \mathbb{P}_n
$\Phi_n^{(0)}$	Projeção de \mathbb{P}_n para \mathbb{D}_n
$\Phi_n^{(1)}$	Projeção de \mathbb{P}_n para $\bar{\mathbb{D}}_n$
$\Phi_n^{(2)}$	Projeção de \mathbb{P}_n para $\bar{\mathbb{D}}_n^\perp$
$\psi_n^{(0)}$	Primeira imersão de \mathbb{P}_n para $\mathbb{P}_n \amalg \mathbb{P}_n$
$\psi_n^{(1)}$	Segunda imersão de \mathbb{P}_n para $\mathbb{P}_n \amalg \mathbb{P}_n$

$\Psi_n^{(0)}$	Primeira projeção de $\mathbb{P}_n \prod \mathbb{P}_n$ para \mathbb{P}_n
$\Psi_n^{(1)}$	Segunda projeção de $\mathbb{P}_n \prod \mathbb{P}_n$ para \mathbb{P}_n
$\lambda_n^{(0)}$	Primeira imersão de \mathbb{P}_n para $\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n$
$\lambda_n^{(1)}$	Segunda imersão de \mathbb{P}_n para $\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n$
$\lambda_n^{(2)}$	Terceira imersão de \mathbb{B} para $\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n$
$\Lambda_n^{(0)}$	Primeira projeção de $\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n$ para \mathbb{P}_n
$\Lambda_n^{(1)}$	Segunda projeção de $\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n$ para \mathbb{P}_n
$\Lambda_n^{(2)}$	Terceira projeção de $\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n$ para \mathbb{B}
$\pi_n^{(\theta)}$	Imersão de \mathbb{D}_n para \mathbb{D}_{n+1} relativa a indexação θ
$\Pi_n^{(\theta)}$	Projeção de \mathbb{D}_{n+1} para \mathbb{D}_n relativa a indexação θ
π_n^{\rightarrow}	Imersão de \mathbb{D}_n para \mathbb{D}_{n+1} por <i>Cprefix</i>
π_n^{\leftarrow}	Imersão de \mathbb{D}_n para \mathbb{D}_{n+1} por <i>Csufix</i>
Π_n^{\rightarrow}	Projeção de \mathbb{D}_{n+1} para \mathbb{D}_n por <i>Cprefix</i>
Π_n^{\leftarrow}	Projeção de \mathbb{D}_{n+1} para \mathbb{D}_n por <i>Csufix</i>
π_n^{\leftrightarrow}	Imersão de \mathbb{D}_n para \mathbb{D}_{n+1} por <i>Cinfix</i>
$\pi_n^{\overleftrightarrow{}}$	Projeção de \mathbb{D}_{n+1} para \mathbb{D}_n por <i>Cinfix</i>
$\pi_n^{\rightarrow\leftarrow}$	Imersões de \mathbb{D}_n para \mathbb{D}_{n+1} por <i>Cprefix</i> e <i>Csufix</i>
$\Pi_n^{\leftarrow\rightarrow}$	Projeções de \mathbb{D}_{n+1} para \mathbb{D}_n <i>Cprefix</i> e <i>Csufix</i>
$\pi_n^{\overleftrightarrow{=}}$	Imersão de \mathbb{D}_n para \mathbb{D}_{n+1} por <i>Cprefix</i> , <i>Cinfix</i> e <i>Csufix</i>
$\Pi_n^{\overleftrightarrow{=}}$	Projeção de \mathbb{D}_{n+1} para \mathbb{D}_n por <i>Cprefix</i> , <i>Cinfix</i> e <i>Csufix</i>
D_{∞}^{\rightarrow}	<i>cosp</i> dos processos por <i>Cprefix</i>
D_{∞}^{\leftarrow}	<i>cosp</i> dos processos por <i>Csufix</i>
$D_{\infty}^{\leftrightarrow}$	<i>cosp</i> dos processos pela construção por <i>Cinfix</i>
$D_{\infty}^{\rightarrow\leftarrow}$	<i>cosp</i> dos processos pela construção por <i>Csufix</i> e <i>Cprefix</i>
$D_{\infty}^{\leftarrow\rightarrow}$	<i>cosp</i> dos processos por <i>Csufix</i> e <i>Cprefix</i>
$\mathbb{D}_{\infty}^{\mathbf{c}}$	Conjunto dos objetos compactos do espaço coerente \mathbb{D}_{∞}
\mathcal{RGr}	Categoria dos grafos reflexivos
Set	Categoria dos conjuntos
$\pi_{n,\infty}^{\rightarrow}$	Imersão de \mathbb{D}_n para \mathbb{D}_{∞} por <i>Cprefix</i>
$\pi_{n,\infty}^{\leftarrow}$	Imersão de \mathbb{D}_n para \mathbb{D}_{∞} por <i>Csufix</i>
$\Pi_{\infty,n}^{\rightarrow}$	Projeção de \mathbb{D}_{∞} para \mathbb{D}_n por <i>Cprefix</i>
$\Pi_{\infty,n}^{\leftarrow}$	Projeção de \mathbb{D}_{∞} para \mathbb{D}_n por <i>Csufix</i>
$\pi_{n,\infty}^{\leftrightarrow}$	Imersão de \mathbb{D}_n para \mathbb{D}_{∞} por <i>Cprefix</i> e <i>Csufix</i>
$\Pi_{\infty,n}^{\leftrightarrow}$	Projeção de \mathbb{D}_{∞} para \mathbb{D}_n por <i>Cprefix</i> e <i>Csufix</i>
$\pi_{n,\infty}^{\overleftrightarrow{}}$	Imersão de \mathbb{D}_n para \mathbb{D}_{∞} por <i>Cprefix</i> , <i>Csufix</i> e <i>Cinfix</i>
$\Pi_{\infty}^{\overleftrightarrow{}}$	Projeção de \mathbb{D}_{∞} para \mathbb{D}_n por <i>Cprefix</i> , <i>Csufix</i> e <i>Cinfix</i>
$\pi_{n,\infty}^{\rightarrow\leftarrow}$	Imersão de \mathbb{D}_n para \mathbb{D}_{∞} por <i>Cprefix</i> e <i>Csufix</i>
$\Pi_{\infty,n}^{\leftarrow\rightarrow}$	Projeção de \mathbb{D}_{∞} para \mathbb{D}_n por <i>Cprefix</i> e <i>Csufix</i>
$\mathbb{D}_{2\infty}^{\rightarrow}$	<i>cosp</i> dos processos computacionais transfinitos
\mathcal{M}	Conjunto enumerável de modelos de máquinas geométricas
\mathcal{E}	Espaço geométrico
$\mathbb{D}_{2\infty}^{\rightarrow}$	<i>cosp</i> dos processos transfinitos por <i>Cprefix</i>
$\mathbb{D}_{2\infty}^{\leftarrow}$	<i>cosp</i> dos processos transfinitos por <i>Csufix</i>
$\mathbb{D}_{m\infty}^{\rightarrow}$	<i>cosp</i> dos processos transfinitos construído por prefixo, distribuídos em m máquinas geométricas
$Id_{\mathbb{D}_{\infty}^{\rightarrow}}$	Operador identidade em $\mathbb{D}_{\infty}^{\rightarrow}$
$F_{(00)}$	Operador inclusão em $\mathbb{D}_{\infty}^{\rightarrow}$
$\mathbf{F}_{(10)}$	Operador PP em $\mathbb{D}_{\infty}^{\rightarrow}$
$\mathbf{F}_{(20)}$	Operador SND em $\mathbb{D}_{\infty}^{\rightarrow}$
\mathbf{F}^{\leftarrow}	Função deslocamento esquerda em $\mathbb{D}_{\infty}^{\rightarrow}$

\mathbf{F}^{\rightarrow}	Função deslocamento direita em $\mathbb{D}_{\infty}^{\rightarrow}$
$\mathbf{F}^{(01)(11)}$	Operador PS em $\mathbb{D}_{\infty}^{\rightarrow}$
$\mathbf{F}^{(02)(12)(22)}$	Operador SD em $\mathbb{D}_{\infty}^{\rightarrow}$
\mathcal{FLin}	Conjunto dos morfismos interpretando operadores em \mathcal{P}
Ind_T	Conjunto dos indicadores de testes
Ind_P	Conjunto dos indicadores de processos
FOp	Conjunto dos indicadores de operações
$FRec$	Conjunto dos operadores recursivos
$FRec$	Conjunto dos operadores infinitos
Σ	Assinatura da linguagem de processos induzida por $\mathbb{D}_{\infty}^{\rightarrow}$
Ω	Conjunto das expressões válidas por Σ
Ω^*	Conjunto das expressões finitas válidas por Σ
\mathcal{R}	Função de representação para interpretações em $\mathbb{D}_{\infty}^{\rightarrow}$
\mathcal{R}^*	Função de representação restrita aos objetos em $\pi_{n,\infty}^{\rightarrow}[\mathbb{D}_{\infty}^{\rightarrow}]$
\mathbb{S}	Espaço coerente dos estados computacionais
\mathbb{I}	<i>cosp</i> dos nomes das variáveis computacionais
\mathbb{V}	<i>cosp</i> dos valores das variáveis computacionais
$\mathbb{I} \multimap \mathbb{V}$	<i>cosp</i> dos traços lineares das funções de \mathbb{I} para \mathbb{V}
$ltr(s)$	Traço linear da função s
Lin	Conjunto de todas as funções lineares
$\mathbb{S} \multimap \mathbb{S}$	Domínio semântico para os processos interpretados em $\mathbb{D}_{\infty}^{\rightarrow}$
$\mathbb{B} \multimap \mathbb{S}$	Domínio semântico para os testes interpretados em \mathbb{B}
\mathbf{Bool}	<i>cosp</i> plano dos valores booleanos
$[\mathbb{I} \multimap \mathbb{V}] \multimap \mathbb{V}$	Domínio semântico para as ações em \mathcal{A}
\mathcal{Eval}	Função de avaliação de $\mathbb{D}_{\infty}^{\rightarrow}$ para $\mathbb{S} \multimap \mathbb{S}$
$eval$	Função de avaliação restrita aos objetos em $\pi_{\infty}^{\rightarrow}[\mathbb{D}_n]$
$eval_{bool}$	Função de avaliação de \mathbb{B} para $\mathbb{S} \multimap \mathbf{Bool}$
\mathcal{US}	Categoria dos sistemas unários

Lista de Figuras

FIGURA 3.1 - Modelo MG com Memória Infinita.	48
FIGURA 3.2 - Modelo MG com Processadores Infinitos.	49
FIGURA 3.3 - Processo <i>skip</i>	51
FIGURA 3.4 - Processo Elementar Enumerável Uni-Dimensional $d^{(n)}$	52
FIGURA 3.5 - Processo Elementar $d^{(n)}$	53
FIGURA 3.6 - Teste Computacional b	53
FIGURA 3.7 - Produto Seqüencial $d^{(k)} \cdot e^{(l)}$	55
FIGURA 3.8 - Soma Determinística $d^{(k)} +_b e^{(l)}$	56
FIGURA 3.9 - Produto Paralelo $\parallel _ , d^{(k)}, e^{(l)}, _ \parallel$	57
FIGURA 3.10 - Soma Não-Determinística $\mid _ \mid d^{(k)} \mid e^{(k)} \mid _ \mid$	58
FIGURA 3.11 - Processos como Transformação de Estados Uni-Dimensionais.	59
FIGURA 4.1 - Metodologia de Construção de $\mathbb{D}_0 - \mathbb{D}_1$	60
FIGURA 4.2 - O Espaço Coerente \mathbb{D}_0 dos Processos Elementares.	61
FIGURA 4.3 - O Espaço Coerente $\bar{\mathbb{D}}_0$ do PP de Processos Elementares.	63
FIGURA 4.4 - O Espaço Coerente Plano $\bar{\mathbb{D}}_0$ do PP de Processos Elementares.	63
FIGURA 4.5 - O Espaço Coerente $\bar{\mathbb{D}}_0^\perp$ da SND de Processos Elementares.	65
FIGURA 4.6 - O Domínio Plano \mathbb{D}_0^\perp da SND de Processos Elementares.	67
FIGURA 4.7 - O Espaço Coerente $\mathbb{P}_0 = \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp$ de Processos.	68
FIGURA 4.8 - A Função Projecção $\Phi_{\mathbb{D}_0}^0 : \mathbb{P}_0 \rightarrow \mathbb{D}_0$	70
FIGURA 4.9 - As Projecções e Imersões em $\mathbb{D}_0 - \mathbb{P}_0$	72
FIGURA 4.10 - O Espaço Coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$ do PS de Processos.	74
FIGURA 4.11 - Representação do PS com Interpretação em $\mathbb{P}_0 \amalg \mathbb{P}_0$	75
FIGURA 4.12 - Representações das Imersões e Projecções em $\mathbb{P}_0 \amalg \mathbb{P}_0$	76
FIGURA 4.13 - O Espaço Coerente $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$ da SD de Processos.	78
FIGURA 4.14 - A Construção da Soma Determinística $d^{(k)} +_b e^{(l)}$ em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$	78
FIGURA 4.15 - A União Indexada de Tokens em $\mathbf{P}_0 \amalg_{\mathbf{B}} \mathbf{P}_0$	79
FIGURA 4.16 - Representação Gráfica da Construção do Processo v em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$	79
FIGURA 4.17 - Representações das Imersões sobre $\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0$	81
FIGURA 4.18 - Representações das Imersões sobre \mathbb{D}_1	82
FIGURA 4.19 - Projecções e Imersões em $\mathbb{D}_0 - \mathbb{D}_1$	83
FIGURA 4.20 - Formalização do Relacionamento entre \mathbb{D}_0 e \mathbb{D}_1	85
FIGURA 4.21 - Representação dos Processos Interpretados por $y, z \in \mathbb{D}_2$	88
FIGURA 5.1 - Construção do Espaço Coerente de Processos \mathbb{D}_∞	89
FIGURA 5.2 - Interpretação da Composição de Imersões em $\pi_n^\theta[\mathbb{D}_n] \subseteq \mathbb{D}_\infty$	93
FIGURA 5.3 - Construção do Processo $(d^{(k)} \cdot e^{(l)}) \cdot (f^{(n)})$ em \mathbb{D}_2	97
FIGURA 5.4 - Representação do Processo $(d^{(k)} \cdot (e^{(l)} \cdot f^{(n)}))$	97
FIGURA 5.5 - Construção do Processo $((d^{(k)} \cdot (e^{(l)})) \cdot ((f^{(n)})))$ em \mathbb{D}_3	99
FIGURA 5.6 - Construção do Processo $((d^{(k)} \cdot e^{(l)}) \cdot (f^{(n)}))$ em \mathbb{D}_3	100
FIGURA 5.7 - Representação do Processo P	102
FIGURA 5.8 - Representação da Composição Finita da Imersão $\pi_{0,n-1}^{\leftarrow}[\mathbb{D}_0]$ em \mathbb{D}_n	104
FIGURA 5.9 - Exemplificação das Projecções $\Pi_n^{\rightarrow}, \Pi_n^{\leftarrow}$ sobre \mathbb{D}_n	106
FIGURA 5.10 - Representação de Objetos em $\pi_{0,n}^{\leftarrow}[\mathbb{D}_n]$	110
FIGURA 5.11 - Aplicações das Projecções $\Pi_{n,0}^{\rightarrow}$ sobre Objetos em \mathbb{D}_n	115
FIGURA 5.12 - Representação de Objetos em $\pi_{0,n}^{\rightarrow\leftarrow}[\mathbb{D}_n]$	117
FIGURA 5.13 - Aplicação das Projecções $\Pi_n^{\rightarrow\leftarrow}$ sobre Objetos em \mathbb{D}_n	120

FIGURA 5.14 - Representação da Composição Finita das Imersões π_n^{\rightleftarrows}	122
FIGURA 6.1 - Construção do Espaço Coerente de Processos $\mathbb{D}_{\infty}^{\rightarrow}$	124
FIGURA 6.2 - Representação para Objetos em $\pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \in \mathbb{D}_{\infty}^{(\theta)}$	132
FIGURA 6.3 - Representação para os Processos Q , Q' e Q^{∞}	135
FIGURA 6.4 - Representação do Processo $W00$	136
FIGURA 6.5 - Representação do Processo $W'00$	137
FIGURA 6.6 - Representação do Processo $W001$	137
FIGURA 6.7 - Processo $W'011$	138
FIGURA 6.8 - Representações de Interpretações em $\pi_{n,\infty}^{\leftrightarrow}[\mathbb{D}_n]$	147
FIGURA 6.9 - Representações de Interpretações em $\pi_{n,\infty}^{\leftarrow}[\mathbb{D}_n]$	155
FIGURA 7.1 - Construção do Espaço Coerente \mathbb{D}_{∞^2}	159
FIGURA 7.2 - Construção do Processo Transfinito W em \mathcal{M}	160
FIGURA 7.3 - Aplicações da Imersão $\pi_{n,m}^{\uparrow}$ sobre Objetos em $[\mathbb{D}_n]$	164
FIGURA 7.4 - Representações das Interpretações em $\pi_{\infty+1,n}^{\rightarrow}[\mathbb{D}_n]$	165
FIGURA 7.5 - Relacionamento entre $\mathbb{D}_{\infty+1}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\rightarrow}$	167
FIGURA 7.6 - Os Espaços Coerentes $\mathbb{D}_{\infty+1}^{\rightarrow}$, $\mathbb{D}_{\infty}^{\leftarrow}$ e $\mathbb{D}_{\infty}^{\leftrightarrow}$	169
FIGURA 7.7 - Representações das Imersões $\pi_{m,m+n}^{\rightarrow}[\mathbb{D}_m]$	172
FIGURA 7.8 - Relacionamento entre $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{\infty+m+1}^{\rightarrow}$	175
FIGURA 7.9 - Representações das Interpretações em $\pi_{\infty+m,2\infty}^{\rightarrow}[\mathbb{D}_{\infty+m}^{\rightarrow}]$	176
FIGURA 7.10 - Projeções sobre $\mathbb{D}_{2\infty}^{\rightarrow}$	177
FIGURA 7.11 - Aplicações da Imersão $\pi_{\infty+m,2\infty}^{\rightarrow}$ sobre Objetos em $\mathbb{D}_{\infty+m}^{\rightarrow}$	178
FIGURA 8.1 - As Funções Deslocamento no PP de Processos.	186
FIGURA 8.2 - As Funções Deslocamento na SS de Processos.	187
FIGURA 8.3 - Representação para a Função Deslocamento Direita.	194
FIGURA 8.4 - Representação para Função Deslocamento Esquerda.	194
FIGURA 8.5 - O Processo R	207
FIGURA 10.1 - O Relacionamento entre $\mathbb{D}_{\infty}^{\rightarrow}$, Ω e $\mathbb{S} \rightarrow \mathbb{S}$	222

Lista de Tabelas

TABELA 6.1 -	Categorias Relacionadas com a Construção do Modelo MG.	129
TABELA 8.1 -	Composição Infinita do PS de Processos Elementares.	202
TABELA 8.2 -	Composição Infinita do PS de Processos Paralelos.	204
TABELA 9.1 -	Equações Satisfeitas em Ω	217
TABELA 9.2 -	Processos Elementares.	218
TABELA A.1 -	Representações Finitas com Processo <i>skip</i>	238
TABELA A.2 -	Representações para PS com Processo <i>skip</i>	240
TABELA A.3 -	Representações para SD com Processo <i>skip</i>	242
TABELA A.4 -	Representações Finitas de Construções com Processo $\mathbf{d}^{(k)}$	244
TABELA A.5 -	Representações para PS com Processo $\mathbf{d}^{(k)}$	247

Resumo

O trabalho constitui-se numa investigação teórica da estrutura ordenada e intuitiva dos espaços coerentes, introduzidos por Girard [GIR 86], na definição do modelo de máquina geométrica para construção e interpretação de estados e processos computacionais rotulados por posições de um espaço geométrico. Esta interpretação poderá ser aplicada às construções determinísticas, incluindo dois tipos especiais de paralelismo - o espacial, com memória e processos infinitos definidos por estruturas matriciais, que operam sobre dimensões independentes, de forma sincronizada; e o temporal, na versão genérica do modelo, com memória global transfinita e processos distribuídos num conjunto enumerável de máquinas geométricas, sincronizadas no tempo. O modelo contempla interpretação para computações não-determinísticas e prevê a aplicação de operadores exponenciais na interpretação do espaço funcional. A noção mais intuitiva deste trabalho está na definição da relação de coerência, que define o grafo sobre o qual se constrói este domínio semântico. Sobre o conjunto de pontos compatíveis de tais grafos, a coerência estrita interpreta a condição implícita para modelar o paralelismo - a concorrência entre posições de memória. Na construção dual, justificada pela presença da negação involutiva no grafo complementar, a incoerência interpreta a condição para o não-determinismo - o conflito de acesso à memória. Para os demais construtores, o produto sequencial e a soma determinística, consideram-se os endofuntores produto e soma direta da categoria $CospLin$ dos espaços coerentes e funções lineares. A estrutura ordenada deste modelo é formalizada pelo espaço coerente \mathbb{D}_∞ de todos os processos, construído em níveis a partir do espaço coerente \mathbb{D}_0 dos processos elementares, seguindo a metodologia proposta por Scott [SCO 76]. Neste sentido, cada nível da construção está identificado por um subespaço \mathbb{D}_n que reconstrói todos os objetos do nível anterior, preservando suas propriedades e relações, além de construir os novos objetos. Compatível com a abordagem algébrica, o relacionamento entre os níveis é expresso por funções lineares denominadas imersões e projeções, interpretando os construtores de processos e seus destrutores, respectivamente. Pelo procedimento de completação, assegura-se a existência do menor ponto fixo para equações recursivas definidas pela composição infinita destes morfismos. Além disso, as interpretações para processos infinitos, construídos por prefixação, apresentadas em $\mathbb{D}_\infty^{\rightarrow}$ comprovam que este modelo é compatível com a diversidade dos construtores. O espaço coerente \mathbb{D}_{∞^2} dos processos transfinitos generaliza a construção e define a estrutura ordenada do modelo de máquina geométrica distribuída. Seus objetos são subconjuntos coerentes de tokens rotulados por posições do espaço geométrico e indexados por subconjuntos isomorfos aos ordinais transfinitos. O espaço coerente $\mathbb{S} \multimap \mathbb{S}$ dos traços lineares de funções definidas sobre o espaço coerente \mathbb{S} dos estados computacionais constitui-se no modelo semântico para análise do comportamento associado a cada processo interpretado em \mathbb{D}_∞ . A definição da função de representação introduz um domínio de expressões que formaliza uma linguagem capaz de expressar, de forma mais operacional, as interpretações obtidas neste modelo de máquina. Cada uma das expressões válidas na linguagem é compatível com uma expressão gráfica.

Palavras-Chave: Teoria dos Domínios, Espaços Coerentes, Domínios Semânticos, Concorrência e Não-determinismo, Modelos de Máquinas Paralelas.

TITLE: “THE GEOMETRIC MACHINE - A MODEL FOR CONCURRENCE AND NON-DETERMINISM BASED ON COHERENCE SPACES.”

Abstract

This work presents a theoretical investigation of the constructive, intuitive and ordered structure of the coherence spaces, introduced by Girard, in order to define the geometric machine model for interpretation of computational states and processes labelled by positions of a geometric space. This interpretation can be applied to deterministic process constructions, including two special types of parallelism - the temporal parallelism, with infinite memory and infinite processes defined over array structures, that operate over independent dimensions in a synchronized way; and the spatial parallelism, in a generic version of the model, with a transfinite global memory shared by transfinite processes distributed in an enumerable set of geometric machines, synchronized in the time. The work also provides interpretation to the non-deterministic computations and applies the exponential operators in the interpretation of the functional space. The most basic notion of this work is the definition of the coherence relation as the admissibility of parallelism between basic operations (elementary processes). That relation defines the web over which the coherence space of the whole set of deterministic and non-deterministic processes is step-wise and systematically build. Over the set of the compatible points of such graph, the strict coherence interprets the implicit condition to model parallelism - the true concurrence. In the dual construction, justified by the presence of involutive negation in the complementary graph, the incoherence interprets the condition that models non-determinism - the conflict of memory accesses. The other constructors, the sequential product and the deterministic sum, are defined by the endofunctors in the *CospLin* category of the coherence spaces and linear functions. The ordered structure of this model is formalized by the coherence space \mathbb{D}_∞ of all processes, constructed by levels from the coherence space \mathbb{D}_0 of the elementary processes, following the Scott's methodology [SCO 76]. In this sense, each level is identified by a subspace \mathbb{D}_n , which reconstructs all the objects from the level before, preserving their properties and relations, and drives the construction of the new objects. Compatible with the algebraic-theoretic approach to computational processes, the relationship between the levels is expressed by linear functions called embedding and projection-functions, which interpret constructors and destructors of processes, respectively. The completion procedure guarantees the existence of the least fixed point to the recursive equations, defined by infinite composition of these morphisms. In addition, the interpretation for infinite processes constructed by prefix is presented in $\mathbb{D}_\infty^\rightarrow$, confirms that the ordered structure of these model is compatible with the diversity of constructors. The coherence space \mathbb{D}_{∞^2} of transfinite processes generalizes the construction and defines the ordered structure of the distributed geometric machine model. Its objects are coherent subsets of tokens labelled by the positions of a geometric space and indexed by isomorphic subsets related to the transfinite ordinal numbers. In order to analyze the behavior related to the interpretations in \mathbb{D}_∞ , the coherence space $\mathbb{S} \multimap \mathbb{S}$ of the linear traces of functions, defined over the coherence space \mathbb{S} of the computational states, is introduced. The definition of the representation-function \mathfrak{R} induces the construction of the domain Ω of valid expressions and formalizes a (graphic) language which is able to express, in an more operational way, the interpretations obtained in the geometric machine model.

Keywords: Domain Theory, Coherent Spaces, Semantic Domains, Concurrent and Non-deterministic Computations, Parallel Machines.

1 Introdução

Este trabalho constitui-se no texto que formaliza a tese, requisito para o doutoramento no Programa de Pós-Graduação da UFRGS, submetendo-a à apreciação por pesquisadores em áreas relacionadas. Nesta introdução apresentam-se as principais idéias que fundamentam este trabalho e os objetivos alcançados. Além disso, salienta-se a motivação para a escolha do tema e sua importância dentro do contexto da Ciência da Computação.

Considerando-se que o principal objetivo da Teoria dos Domínios na Semântica Denotacional é prover um modelo matemático para um sistema de tipos, incluindo o espaço de funções, e explicar a noção de computabilidade em relação a estes tipos, esta proposta pode ser entendida como uma tentativa de aplicação destas idéias.

É notório que a Teoria dos Domínios é uma teoria bem consolidada constituindo-se atualmente numa importante área de investigação e de fundamentação teórica para a Ciência da Computação, especialmente para as Linguagens de Programação e para a Teoria dos Modelos de Provas, com uma vasta literatura (algumas das suas principais contribuições serão resumidas no próximo capítulo).

Entretando, os modelos existentes que buscam as interpretações de sistemas dinâmicos como processos computacionais envolvendo paralelismo e não-determinismo ainda não são capazes de satisfazer certos questionamentos por se tratarem de estruturas complexas que em geral se mostram de difícil aplicação. As principais questões são apresentadas e analisadas ao longo do próximo capítulo.

Um dos resultados importantes do presente trabalho poderá ser o caráter simples e intuitivo da estrutura ordenada do modelo de máquina descrito, restrito a uma tipo específico de memória, e tornando explícito um relacionamento intuitivo entre o paralelismo espacial e temporal de processos computacionais.

1.1 Concepção da Idéia

A noção de computação pode ser concebida fundamentalmente de dois modos, o primeiro que está relacionado com a reescrita de expressões, na qual se encaixam o λ -cálculo e as equações recursivas. O outro, que estabelece uma computação como uma transformação de estados, na qual se consideram o fluxo e os recursos, ambos diretamente relacionados com a idéia de sistemas dinâmicos. Pode-se dizer que no primeiro caso grande parte da fundamentação computacional refere-se à teoria algébrica e no segundo à teoria das redes.

Na teoria da computação clássica estas noções são equivalentes. Entretanto, sob o ponto de vista da funcionalidade dos sistemas envolvidos, estas noções não são equivalentes, ou como se referem na literatura específica, não são dinamicamente equivalentes.

A lógica linear introduzida por Girard tornou explícita as interpretações computacionais relacionadas com recursos, reconhecida como a *lógica ciente dos recursos*. Isto se justifica pois as fórmulas são interpretadas como recursos e as regras lógicas como regras para utilização destes recursos. A conscientização é pertinente porque as provas geradas pelo sistema não permitem o uso irrestrito dos recursos, mas apenas aqueles assumidos anteriormente como hipóteses.

Baseando-se nestas interpretações, a noção de concorrência surge na estruturação das regras e não sendo portanto creditada a uma particular regra ou a um específico operador. Isto nos leva a crer que a interpretação dos recursos pode ser concebida como na semântica coerente da lógica linear. Ou ainda, recursos podem ser modelados como objetos de um domínio denominado espaço coerente, e o fluxo de recursos fluindo através

dos operadores lineares, tornando compatíveis as interpretações computacionais e denotacionais. Isto foi bem observado por Costa e Moreira [COS 94] num trabalho que mostra algumas conexões entre sistemas lógicos e dinâmicos, num tratamento uniforme para os sequentes lógicos e os sistemas de transições das redes.

Neste sentido, a teoria da aproximação sugerida por Scott, provê uma fundamentação matemática para a formalização destas idéias, capaz de interpretar a informação que flui quando os processos são executados. A idéia básica de Scott é representar tipos de dados por certos conjuntos parcialmente ordenados, denominados domínios. Quando a computação está baseada em um algoritmo, então cada um dos conjuntos de dados de entrada e saída é visto como um domínio. O programa que executa a computação é representado por funções entre domínios.

Analisadas todas estas idéias, este trabalho formalmente introduz a estrutura ordenada da máquina geométrica como o espaço coerente de processos computacionais, indutivamente gerado, e definido a partir da escolha do conjunto de ações rotuladas por posições de um espaço geométrico (definindo os processos elementares) e do conjunto finito de construtores de processos. Pela completação, todos os processos, inclusive aqueles sem restrição quanto ao tempo de execução ou espaço de memória, são interpretados neste domínio.

Ainda é apresentada uma generalização desta estrutura de modelo de máquina, formalizada pelo espaço coerente de processos computacionais transfinitos, propiciando interpretação para sistemas de computação distribuídos. A idéia para construir esta generalização sobre estruturas computacionais possivelmente infinitas, fundamenta-se na análise da construção dos ordinais transfinitos como a generalização das seqüências de números naturais, na Teoria dos Conjuntos, introduzida por Cantor, e apresentada por Stoll em [STO 61]. Neste livro, Stoll introduz os sistemas unários como uma fundamentação algébrica para indução e recursão nos números naturais. Estes estudos foram seguidos por Escardo [ESC 95] que introduz os sistemas binários para formalizar a indução e recursão na reta real e Reiser [REI 99], que apresentou uma versão categórica para os sistemas unários, com abordagem para a μ -recursão.

A motivação para este trabalho pode ser creditada ao desejo de dar continuidade a estas idéias ou de contribuir no sentido de que se tornem mais claras e precisas.

O trabalho fundamenta-se numa idéia de simples compreensão por se tratar de um modelo para interpretar o comportamento de programas concorrentes e não-determinísticos baseada na idéia de processo.

Constitui-se numa pesquisa bem limitada, pois restringe sua interpretação sobre processos matriciais que operam de forma sincronizada e em diferentes dimensões, apresentando uma abordagem muito interessante e intuitiva, características inerentes da estrutura sobre a qual o modelo é definido, os espaços coerentes.

Por fim, é preciso ainda dizer que este trabalho não tem a pretensão de responder a todas as questões que podem surgir a partir das idéias apresentadas. Como todo trabalho científico, o modelo de máquina geométrica é construído no sentido de responder alguns questionamentos dentro da teoria da computação, e colaborar na proposta de novas questões de pesquisa.

1.2 Objetivo e Etapas de Desenvolvimento

Esta tese tem por objetivo a construção do modelo de **máquina geométrica** para interpretação dos processos computacionais que envolvam paralelismo e não-determinismo, operando sobre estruturas matriciais com memória indexada por um espaço geométrico e utilizando como estrutura ordenada os espaços coerentes.

As diversas etapas que foram sendo vencidas antes que este objetivo tivesse sido

efetivamente alcançado estão especificadas logo a seguir.

1. A construção da estrutura ordenada do modelo de máquina, fundamentada nos domínios qualitativos introduzidos por Girard - os espaços coerente, divididos nas seguintes subetapas:
 - (a) a definição dos estados, dos processos computacionais e dos construtores de processos interpretados no modelo;
 - (b) a definição do espaço coerente de processos elementares e a construção do primeiro nível da estrutura do modelo;
 - (c) a formalização e análise dos morfismos que constroem os diferentes níveis da estrutura deste modelo;
 - (d) a completção da estrutura ordenada do modelo de máquina;
 - (e) a interpretação dos construtores de processos como morfismos na categoria $\mathcal{CospLin}$; e
 - (f) a generalização da estrutura do modelo, capaz de prover interpretação para sistemas de computação distribuídos.
2. A introdução de um modelo semântico, pela definição do espaço coerente dos estados de computação e de uma função de avaliação capaz de modelar o comportamento de todos os processos interpretados na estrutura do modelo.
3. A introdução de uma linguagem baseada em interpretações definidas sobre os objetos e morfismos da estrutura ordenada da máquina geométrica.

Para cada item e subitem detalhando o objetivo deste trabalho é dedicado um capítulo neste texto. Na próxima seção, estão resumidos os principais aspectos referentes à estrutura do modelo proposto e que também justificam a consistência deste trabalho.

1.3 Estrutura Ordenada do Modelo

A estrutura do modelo é indutivamente construída a partir do espaço coerente de processos elementares e dos construtores (funções lineares) que representarão as regras de construção dos processos modelados nesta máquina. Sua correspondente completção provê uma estrutura ordenada, capaz de modelar os processos computacionais (incluindo processos infinitos e transfinitos) considerando relevantes as características de concorrência e não-determinismo definidas sobre estruturas de memória.

Salienta-se que a construção dos espaços coerentes dos estados e dos processos computacionais é compatível com a representação geométrica da memória, ou seja, representada operacionalmente por estruturas matriciais (finitas ou infinitas) como vetores, matrizes, blocos tri-dimensionais e outras multi-dimensionais.

1.3.1 Espaço Coerente \mathbb{D}_n dos Processos Computacionais

O espaço coerente de processos indutivamente construído provê a interpretação para a família de todos os processos finitos no tempo. A noção mais intuitiva nesta interpretação está formalizada na definição da relação de coerência (reflexiva e simétrica) estabelecida sobre as ações rotuladas por posições do espaço geométrico. Esta relação define o grafo que explicitamente formaliza o conflito de acesso à memória. Ou seja, sob o conjunto de pontos compatíveis de tais grafos a coerência estrita interpreta a condição para modelar o paralelismo espacial.

Por outro lado, na construção dual, justificada pela presença da negação involutiva no grafo complementar, é interpretada a incoerência como condição para o não-determinismo. Para os demais construtores, o produto sequencial e a soma determinística, considera-se a interpretação associada aos morfismos na categoria $\mathit{CospLin}$ dos espaços coerentes e funções lineares.

A estrutura ordenada que deverá modelar o conjunto de processos será construída a partir da definição do espaço coerente plano \mathbb{D}_0 dos processos elementares, seguindo a metodologia proposta por Scott na construção do reticulado de fluxogramas [SCO 76]. A definição do espaço coerente \mathbb{D}_0 é obtida a partir de uma teia discreta, onde cada ponto interpreta um processo elementar. Um processo é elementar quando o resultado de sua aplicação altera o valor de, no máximo, uma entre todas as variáveis que definem cada estado de computação. A partir de \mathbb{D}_0 a estrutura será construída em níveis, sendo que cada nível define um novo espaço coerente \mathbb{D}_n construído de tal forma que satisfaça as condições:

1. deverá reconstruir os objetos do nível anterior, agora associados por um conjunto de índices capaz de identificar os construtores que os geraram,
2. preservar os relacionamentos entre objetos reconstruídos, e ainda
3. permitir a união indexada com outros tokens, referentes à teia do próximo nível.

Salienta-se que a definição dos construtores e destrutores de processos baseia-se na definição das funções projeções e imersões, que também formalizam o relacionamento entre os diferentes níveis da estrutura do modelo de máquina que se está propondo.

Os processos com tempo de execução finito são interpretados por objetos nos subespaços coerentes $\pi_{n,\infty}[\mathbb{D}_n] \subseteq \mathbb{D}_\infty$ que resultam de imersões aplicadas sobre cada um destes níveis. Isto significa que tais objetos são construídos nos subníveis \mathbb{D}_n e inseridos no espaço coerente \mathbb{D}_∞ resultante da completação. E, os processos cuja execução não pode ser limitada no tempo também serão interpretados por objetos definidos como o limite inverso de diagramas construídos a partir de projeções e imersões.

Portanto, os processos são interpretados por conjuntos coerentes de tokens indexados por dois conjuntos de índices disjuntos, onde

1. o índice superior direito indica seu posicionamento no espaço geométrico;
2. o índice inferior direito identifica o construtor de processo que o gerou e que foi modelado pela relação de coerência que constrói sua teia.

Neste último item, cada aplicação do construtor sobre um processo acrescenta mais um símbolo a cada índice, em cada token do subconjunto coerente que o interpreta. Portanto, a interpretação de um processo por um conjunto coerente, coincide em muitos casos com um conjunto de infinitos tokens indexados por conjuntos infinitos de símbolos que possui uma representação finita. O símbolo terminal em cada índice de cada token explicita o tipo de processo, finito ou infinito, no sentido temporal.

1.3.2 Espaço Coerente \mathbb{D}_∞ dos Processos Computacionais.

A posterior completação estende esta interpretação para os processos infinitos no tempo, construídos por diferentes abordagens. Neste sentido, este modelo de máquina torna viável qualquer uma das seguintes possibilidades de interpretação:

1. O espaço coerente $\mathbb{D}_\infty^{\rightarrow}$ interpretando processos que iniciam em determinado instante de tempo mas não se pode determinar quando será seu término. Assim, a determinação de novos processos caracteriza a construção por prefixo.

2. O espaço coerente $\mathbb{D}_{\infty}^{\leftarrow}$ interpretando processos que terminam em determinado instante de tempo mas não se pode determinar quando começaram. Neste caso, a determinação de novos processos caracteriza a construção por sufixo.
3. O espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$ interpretando processos que não possuem início nem término especificado por um determinado instante de tempo, mas é sempre possível determinar uma enumeração do tempo, a partir de qualquer instante que se conhece. Neste caso, a determinação de novos processos caracteriza a construção por infixo.
4. O espaço coerente $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ interpretando processos, cujos instantes que identificam seu início e término, estão bem determinados no tempo. Entretanto, não se pode determinar uma enumeração dos instantes, a partir do instante inicial, que atinja o instante final. Portanto, a determinação de novos processos caracteriza a construção por prefixo e sufixo, simultaneamente.

Os dois primeiros espaços coerente são isomorfos e foram formalmente definidos no texto. Em analogia, é formalizado o isomorfismo entre os últimos dois espaços coerentes e as correspondentes definições.

Estes espaços coerentes são algumas das possíveis interpretações apresentadas neste texto que determinam a estrutura ordenada e indutiva de diferentes modelos de máquinas geométricas. Estas diferenças são obtidas pela definição das imersões, nos subníveis de cada estrutura ordenada, que interpretam os construtores de processos que identificam cada um destes modelos de máquina.

Neste sentido, para cada interpretação de um processo infinito é possível ainda definir um conjunto enumerável de interpretações de processos transfinitos, que modelam a execução destes processos nas diferentes máquinas de um sistema de computação distribuído.

Esta possibilidade de modelar computações distribuídas, a partir da definição de espaços coerentes de processos transfinitos, formaliza o modelo de máquina geométrica distribuída.

1.3.3 Espaço Coerente $\mathbb{D}_{2\infty}$ dos Processos Transfinitos

Seguindo a mesma metodologia de construção, a definição do espaço coerente dos processos transfinitos constitui-se num domínio onde é possível interpretar também processos distribuídos, caracterizando neste caso o paralelismo temporal.

Para tal, considera-se um conjunto enumerável de máquinas, cada qual com memória independente e indexada por um espaço geométrico, mas sincronizadas no tempo, ou seja, operando a partir do mesmo instante.

A forma com que esta construção é proposta neste trabalho é interessante, sobretudo porque mostra que a modelagem de sistemas de computação distribuídos pode ser obtida como uma generalização da modelagem de sistemas de computação convencionais. Isto representa uma forte e direta ligação entre estes sistemas, geralmente tratados de forma isolada na literatura. Além disso, pela metodologia utilizada, é intrínseco que estes sistemas nascerem de forma bem intuitiva, conforme sugerido por Scott, devam ser as construções fundamentadas dentro da teoria das aproximações, ou mais especificamente, na Teoria dos Domínios.

A generalização da estrutura, interpretando processos computacionais possivelmente infinitos, distribuídos em um conjunto enumerável de máquinas, fundamenta-se na análise da construção dos ordinais transfinitos como seqüências generalizadas de números naturais.

Todas estas considerações foram importantes na definição dos morfismos que interpretam os construtores de processos na estrutura da máquina geométrica. Em especial, a partir da definição e análise do operador que interpreta o produto seqüencial, obtém-se

uma abordagem algébrica para as interpretações relacionadas com as construções temporais, baseada nos sistemas algébricos n -ários. Considerando-se ainda as construções de processos por prefixos e sufixos, definem-se as funções deslocamento, para a direita e para a esquerda, e apresenta-se uma abordagem gráfica para geração dos índices associados aos tokens que constroem as correspondentes interpretações.

1.4 Interpretação Semântica do Modelo

A noção de estado de computação é formalizada pelo conjunto de variáveis distribuídas num espaço geométrico e assumindo valores num conjunto V . Assim, a estas variáveis são atribuídos valores e nomes, identificando suas posições neste espaço geométrico que também modela a memória. Neste sentido, indica-se por I o conjunto de índices que representam as posições (pontos) do espaço geométrico.

Ao considerar a interpretação de computações não-determinísticas, cada estado computacional é interpretado por um conjunto coerente de traços lineares, de funções definidas do espaço coerente \mathbb{I} dos indexadores de pontos do espaço geométrico para o espaço coerente \mathbb{V} dos valores computacionais. Como consequência, o domínio semântico $\mathbb{S} \multimap \mathbb{S}$, capaz de avaliar o comportamento dos processos interpretados em \mathbb{D}_∞ , consiste em transformações sobre conjuntos de traços lineares.

As funções lineares além de serem contínuas no sentido de Scott, também são estáveis e lineares, propriedades que asseguram a existência e unicidade de uma menor aproximação na imagem, neste caso, caracterizada por tokens da teia sobre a qual se constrói \mathbb{V} , para todos os objetos do domínio \mathbb{I} limitados superiormente. As funções lineares podem portanto serem identificadas por subconjuntos especiais de pares ordenados de tokens, cujas teias constroem os espaços coerentes sobre os quais estas funções estão definidas. Cada elemento do conjunto $\mathbb{I} \multimap \mathbb{V}$ de todos os traços das funções lineares de \mathbb{I} para \mathbb{V} interpreta um estado computacional numa abordagem determinística.

Portanto, a família $\wp(\mathbb{I} \multimap \mathbb{V})$ de todos os conjuntos de traços lineares, ordenados pela inclusão, constitui-se no espaço coerente \mathbb{S} dos estados de computação. Da mesma forma, o conjunto de todos os traços das funções lineares definidas sobre o espaço coerente \mathbb{S} dos estados de computação, ordenados pela inclusão, define o espaço coerente $\mathbb{S} \multimap \mathbb{S}$.

Cada conjunto coerente em \mathbb{D}_∞ , interpretando um processo, está associado a uma função linear, cujo traço é um objeto de $\mathbb{S} \multimap \mathbb{S}$. Assim, a interpretação do comportamento de processos é obtida a partir da análise dos traços lineares das funções em $\mathbb{S} \multimap \mathbb{S}$.

1.5 Linguagem do Modelo

Além da estruturação do modelo e da apresentação de uma semântica denotacional baseada na análise dos objetos e morfismos da categoria $\mathit{CospLin}$, propõe-se também a definição de uma linguagem capaz de expressar, de uma forma mais operacional, as interpretações definidas sobre o espaço coerente de processos \mathbb{D}_∞ .

A formalização da identificação dos processos em \mathcal{P} com os conjuntos coerentes em \mathbb{D}_∞ e a interpretação dos construtores de processos como morfismos, definidos sobre tal espaço coerente de processos, é alcançada pelo operador linear $\mathfrak{R} : \Omega \rightarrow \mathbb{D}_\infty$.

Tal operador é responsável pela especificação equacional que deverá descrever o conjunto de propriedades associadas aos processos interpretados no espaço coerente \mathbb{D}_∞ de processos e sobre o qual pode-se construir as soluções de equações (recursivas) e inequações. O conjunto de constantes, variáveis e símbolos funcionais que constroem o conjunto Ω dos termos que podem aparecer na especificação da linguagem que se está propondo, definem sua assinatura.

Os operadores que definem o produto seqüencial, soma determinística e produto paralelo são interpretados na estrutura e deverão ser definidos nesta linguagem. Além destes, também é definido o operador não-determinístico, cuja interpretação é garantida pelo operador de *negação-linear*, involutivo, que define a teia sobre a qual se constroem objetos que interpretarão processos mutuamente exclusivos.

As construções recursivas da linguagem podem ser definidas a partir de duas formas distintas de interpretação, determinadas pela estrutura conceitual e indutiva do modelo. A primeira, denominada recursão espacial, relativa ao modo como os estados e processos computacionais são concebidos e que se reflete na estrutura do modelo aqui proposto, garante interpretação para estados e processos infinitos (memória infinita). Apoiada nesta interpretação, a linguagem suporta um operador que executa o paralelismo espacial, intrínseco à relação de coerência que define a teia geradora dos objetos que interpretam processos concorrentes em \mathbb{D}_∞ . Tal operador representa uma sincronização de ações interpretadas por processos concorrentes. Na outra interpretação, denominada recursão temporal, tornam-se explícitos o relacionamento entre os níveis que compõem a estrutura ordenada e o custo computacional associado aos processos finitos interpretados pelos objetos nestes níveis, em termos da quantidade de processos elementares envolvidos na sua construção e posterior execução, caracterizando as unidades de tempo computacional. Neste caso, tem-se assegurado pela completção, que tais equações recursivas têm solução em \mathbb{D}_∞ .

1.6 Estruturação do Texto

No Capítulo 2 apresenta-se um resumo sobre modelos de máquina e alguns importantes trabalhos na semântica denotacional.

A seguir, no Capítulo 3, define-se o conjunto de processos e de estados sobre a qual a estrutura da máquina geométrica está formalizada. O espaço coerente de processos elementares e todos os subespaços e morfismos envolvidos na definição do primeiro nível da estrutura ordenada deste modelo são introduzidos no Capítulo 4. Também apresenta-se neste capítulo a metodologia de construção, baseada na proposta de Scott para definição do reticulados de fluxogramas. No Capítulo 5, generalizam-se as construções, objetos e morfismos dos últimos dois capítulos para interpretar processos cuja execução está limitada a um intervalo de comprimento máximo igual a 2^{n+1} *utc* (unidades de tempo computacional).

A completção da estrutura indutiva do modelo é apresentada no Capítulo 6, e provê interpretação para todos os processos em \mathcal{P} , seja pela reconstrução dos objetos já definidos nos subespaços coerentes finitamente gerados, seja pela construção de novos objetos, aqueles que interpretam processos cuja execução não pode ser limitada no tempo.

O modelo de máquina geométrica distribuída, formalmente estruturada pelo domínio $\mathbb{D}_{\infty 2}$, onde os processos são determinados pelos respectivos rótulos no conjunto I de posições do espaço geométrico que modela a memória de cada máquina e pelos índices indicando a posição de cada máquina no espaço \mathcal{E} , enumerável, é apresentado no Capítulo 7.

A interpretação dos construtores de processos é formalizada por funções lineares apresentadas no Capítulo 8. Além das definições com as correspondentes restrições e observações, verifica-se também que o modelo contempla interpretação para construtores recursivos e apresentam-se alguns exemplos. As funções de representação (com exemplificação em anexo) e de interpretação relacionadas com a linguagem e semântica dos processos interpretados pela estrutura ordenada são formalizadas nos Capítulos 9 e 10, respectivamente. E no Capítulo 11 estão as considerações finais, trabalhos relacionados e possíveis aplicações e encerra-se o texto com as referências bibliográficas que ajudaram na fundamentação e na construção das idéias que definem esta tese.

Finalmente, salienta-se que este texto é rico em exemplificações e representações

gráficas. Justifica-se esta forma de apresentação pelo esforço em tornar mais acessível a compreensão das principais construções matemáticas que formalizam os objetivos propostos no início de nosso trabalho de pesquisa.

2 Histórico e Fundamentação

O objetivo deste trabalho é a definição de um modelo de máquina, e sua correspondente interpretação semântica, onde a noção de computação seja compatível com a noção de aproximação introduzida por Scott, e capaz de modelar o comportamento dos processos concorrentes e não-determinísticos executados pelos computadores atuais.

Esta difícil tarefa começa com o estudo, análise e seleção de alguns dos muitos e distintos modelos de máquinas que tem sido propostos na literatura, e das diversas teoria desenvolvidas na direção da modelagem dos sistemas dinâmicos utilizando as ferramentas computacionais.

Em [SCO 67] Scott apresenta uma terminologia uniforme para os diferentes tipos de máquinas, incluindo àquelas que serão citadas nos próximos parágrafos. Este trabalho enfatiza a separação entre os conceitos de máquina e programa e torna explícita algumas das vantagens de se utilizar o conceito matemático de função em comparação ao de conjunto (ou t -upla de conjuntos), na definição formal de uma máquina.

Ainda, como uma aplicação desta formalização do conceito de máquina, através da composição de funções, são analisadas as possíveis soluções para as seguintes questões:

1. A composição de funções computáveis em uma máquina é também computável nesta máquina?
2. Se duas funções são computáveis em diferentes máquinas, é possível definir uma generalização onde a composição de ambas seja também computável?

As respostas a estas questões foram amplamente estudadas, mas uma das abordagens mais interessantes pode ser obtida considerando-se uma teoria de tipos mais geral, a Teoria dos Domínios, apresentada na segunda seção deste capítulo, cujo principal utilização dentro da Teoria da Computação é prover uma semântica denotacional para as linguagens de programação.

Na primeira seção tem-se uma breve descrição dos modelos de máquinas seqüenciais e paralelas analisadas, alguns dos quais exerceram certa influência no trabalho de pesquisa apresentado ao longo deste texto.

2.1 Modelos de Máquinas

É concensual que a definição de uma máquina deve responder de forma simples e não-ambígua as seguintes questões: Como inserir dados na memória? Como são desempenhadas as operações e testes? E ainda, como retirar os resultados obtidos quando uma instrução de término é alcançada?

A especificação formal de máquina proposta por Scott [SCO 67] aplica a notação funcional para responder estas questões. Supondo os conjuntos I dos dados de entrada, F dos operadores, P dos testes e O dos dados de saída, uma máquina \mathcal{M} é uma função definida sobre o conjunto $\{I\} \cup F \cup P \cup \{O\}$ sempre que existem os conjuntos:

- X , que especifica o tipo dos dados de entrada,
- M , que especifica o tipo dos estados que constituem a memória e
- Y , que indica o tipo dos dados de saída da máquina;

e as seguintes aplicações são satisfeitas

1. $\mathcal{M}(I) \equiv \mathcal{M}_I: X \rightarrow M$,

2. $\mathcal{M}(F) \equiv \mathcal{M}_F : M \rightarrow M$, para todo $F \in F$,
3. $\mathcal{M}(P) \equiv \mathcal{M}_P : M \rightarrow \{T, F\}$, para todo $P \in P$, e
4. $\mathcal{M}(O) \equiv \mathcal{M}_O : M \rightarrow Y$.

Os programas são definidos por um conjunto de instruções contendo exatamente uma instrução inicial, e associando às demais instruções um único rótulo, de forma que não existam duas instruções distintas com o mesmo rótulo. Deste modo, os programas são vistos como esquemas, onde os símbolos para operações e predicados não possuem qualquer significado.

A computação de um programa, em uma determinada máquina, nos dá o significado de seguir o fluxo dos dados neste programa. O significado das instruções do programa constitui-se na descrição ou histórico deste fluxo.

Dois modelos de máquinas tem obtido uma posição dominante dentro da teoria da computação:

- a máquina de Turing, proposta por Alan Turing em 1936 e suas extensões constituem-se nos modelos mais utilizados para a formalização de um algoritmo;
- a máquina RAM (Random Access Machine) introduzida por Cook e Reckhow [COO 73] modelando o idealizado computador de Von Neumann, onde os programas estão dentro da memória da máquina.

2.1.1 Máquina de Turing

A máquina de Turing padrão pode ser pensada como um dispositivo de escolha automático que pode ser aplicado sobre quantidades finitas de estados computacionais. Existem dois tipos de estados computacionais especiais, o inicial e os finais. A máquina trabalha passo a passo, cada passo representa a transição de um estado para outro. Salienta-se que o estado subsequente pode coincidir com o estado anterior. O processo começa no estado inicial e termina em um estado final.

Toda a informação alcançada na execução do algoritmo é armazenada na memória, que constitui-se numa fita unidimensional, limitada à esquerda e tão grande quanto se deseja à direita, particionada por células. Esta fita constitui-se também no dispositivo de entrada e de saída dos dados. Cada célula pode armazenar o símbolo branco que interpreta a célula vazia na fita, ou o símbolo que identifica o início da fita, ou ainda um símbolo de um alfabeto previamente escolhido, que pode ser o alfabeto de entrada ou um outro auxiliar.

A máquina de Turing possui uma unidade de controle, refletindo o estado corrente da máquina, que contém a cabeça da fita - uma multifunção *leitura / gravação* capaz de se deslocar, tanto para a direita quanto para a esquerda, sobre a fita numa única célula. Cada passo pode descrever a leitura ou a escrita de um símbolo sobre a fita, durante o qual a cabeça esta posicionada sobre uma única célula da fita.

O desempenho de cada passo envolve uma seqüência de ações. Suponha que a cabeça *leitura / gravação* está sobre alguma célula e o corrente estado da máquina de Turing é finito. Então, dependendo do estado da máquina e da letra na célula, alguma letra é escrita nesta célula e a máquina vai para o próximo estado. A letra escrita pode inclusive ser a mesma da célula anterior. O processo continua desta forma, seqüencialmente, até atingir seu término, caracterizado por um estado finito.

Um programa neste caso, é a enumeração de todas as possíveis etapas da máquina de Turing. Em geral, o conteúdo dos programas depende do alfabeto que esta sendo usado.

Extensões da máquina de Turing são correntes na literatura, podendo diferir uma das outras em pequenos detalhes que não alteram o poder computacional deste formalismo.

Isto quer dizer que as classes de funções computáveis a partir destas variações são as mesmas alcançadas a partir da máquina padrão.

Em [BOA 90] são explorados os aspectos mais fundamentais da complexidade em modelos de máquinas. Para tal são analisados e comparados diversos modelos seqüenciais e paralelos. Algumas possíveis extensões citadas em [BOA 90], são transcritas logo a seguir:

- **fitas múltiplas**, é uma versão da máquina de Turing que possui várias fitas, onde cada fita é conectada a unidade de controle por meio de um cabeça de *leitura / gravação*;
- **fita infinita de duas vias** ocorre quando a fita pode ser infinita em ambas as direções;
- **múltiplas cabeças**, na qual cada cabeça na unidade de controle trabalha e se desloca independentemente;
- **fita bidimensional**, onde cada fita constitui-se numa grade bidimensional infinita.

O desempenho de cada uma destas extensões da máquina de Turing, em última instância, permanece o mesmo, embora ainda seja possível generalizá-las (computações não-determinísticas) ou recombina-las (máquina de duas cabeças e fita infinita).

Entretanto, em todas estas possíveis variações, a funcionalidade da máquina de Turing é descrita por algum programa. Neste sentido, a análise dos algoritmos deve ser alcançada pela análise destes programas. Desta forma, a exploração das propriedades essenciais das estruturas algorítmicas está diretamente relacionada com a exploração das estruturas de programas que descrevem estes algoritmos. Mas a tarefa de transcrever os algoritmos usados na prática como programas para a máquina de Turing não é uma tarefa fácil.

Apesar de seu imenso valor teórico, como um modelo computacional universal, a máquina de Turing possui muito pouco em comum com os computadores reais. A memória seqüencial da máquina de Turing dificulta o acesso imediato à informação armazenada em alguma célula que compõe os estados na memória.

Ou seja, não é uma característica intuitiva o endereçamento das células na memória, viabilizando um acesso direto sobre as mesmas. Esta característica de memória, intrínseca das máquinas de registradores, e compatível com os computadores reais é denominada memória de acesso aleatório.

2.1.2 Máquinas de Registradores.

As máquinas de registradores (RAM) constituem um modelo computacional para análise de algoritmos concretos baseada na fundamental distinção entre os conceitos de programa e de máquina. Esta distinção também fundamenta o modelo de máquina geométrico que se esta propondo, conforme mostram os próximos capítulos.

Neste sentido, as máquinas de registradores podem ser reconhecidas como modelos para computadores reais, bastando para tal uma redução no conjunto de instruções, relacionando apenas aquelas instruções essenciais. Por outro lado, máquinas de registradores não possuem algumas das características básicas dos computadores reais, pois

1. possuem um conjunto infinito de palavras, e admitem palavras infinitas,
2. possuem uma memória de endereçamento infinita,

sendo estas importantes características compatíveis com as linguagens de alto-nível.

Entretanto, este problema é facilmente contornado, pois toda máquina RAM está equipada com um compilador capaz de restringir tais características, de acordo com a implementação dos algoritmos ou a descrição dos seus programas.

A máquina RAM básica consiste numa unidade de controle (onde um programa é armazenado), um ou mais registradores de acumulação, um registrador contador e uma coleção infinita de registradores na memória. O acumulador e os registradores na memória não são limitados em seu tamanho, e qualquer estado computacional é uma configuração armazenada nos registradores na memória.

As intruções da máquina RAM são divididas em instruções que influenciam o fluxo de controle, instruções de entrada e saída de dados, instruções de transporte de dados e instruções de desempenho aritmético. Neste sentido, os programas na máquina RAM consistem em seqüências de instruções rotuladas. O significado de cada instrução é auto-explicativo, indicando em cada caso, como a instrução selecionada é ajustada no programa.

O não-determinismo é obtido quando cada linha de código do programa lista, na realidade, duas ou mais instruções, a partir da qual a máquina é capaz de escolher uma e efetivar sua execução.

O modelo de máquina proposto neste trabalho, no que se refere à modelagem de processos seqüenciais, tem como base o modelo de máquina RAM, da mesma forma que o reticulado de fluxogramas proposto por Scott. Neste caso, os rótulos identificam os registradores localizados no espaço geométrico I , que também modelam a memória. As operações básicas identificadas por processos elementares, estão rotuladas por posições do espaço I , que podem ser executada por quaisquer registradores instantaneamente.

Salientam-se, a seguir, algumas máquinas de registradores.

- A máquina **NORMA** (Number theOretical Register MACHine) introduzida por Bird, [BIR 76], é uma máquina de registradores que lembra a arquitetura básica dos computadores atuais. Possui um conjunto infinito (possivelmente enumerável) de registradores naturais com memória e três instruções elementares sobre cada registrador: adição e subtração do valor natural um e o teste que identifica se o valor armazenado no registrador é o número natural $zero$. A máquina NORMA é outro formalismo universal, extremamente simples em sua concepção, mas capaz de simular o poder computacional da máquina de Turing.
- A máquina **PRAM** proposta em [BOV 94], é uma extensão do modelo de máquina RAM para computações paralelas, definido por uma seqüência possivelmente infinita de processadores capazes de operar numa memória global compartilhada. Esta memória global também possui um número possivelmente infinito de células, mas tempo de acesso constante.

Apesar de não ser considerada um modelo de máquina real, PRAM é um instrumento muito útil para se estudar a estrutura lógica de computações paralelas no contexto onde os custos computacionais são geralmente ignorados. Pode-se afirmar que a maioria dos algoritmos desenvolvidos para modelos paralelos de computação são freqüentemente baseados em algoritmos desenvolvidos para a máquina PRAM.

Existem dois tipos de variações para a máquina PRAM, relacionados com o números de instruções :

1. SIMD-PRAM que compartilha múltiplos dados mas acessados por uma instrução única. Neste modelo, o conjunto de instruções da máquina RAM é estendido, permitindo que diferentes processadores possam acessar diferentes células da memória, simultaneamente. Portanto, em qualquer instante de tempo, todos os processadores executam a mesma instrução, embora em operandos distintos.

2. MIMD-PRAM que também compartilha múltiplos dados mas cujo acesso aos mesmo pode ser por obtido por múltiplas instruções. Nesta especificação, a máquina RAM é estendida de forma análoga, mas permitindo que os processadores executem tarefas distintas, simultaneamente.

Embora a caracterização de instrução múltipla viabiliza a possibilidade de flexibilidade no desenvolvimento do algoritmo, na prática estas considerações não são relevantes. É fácil mostrar que a máquina MIMD-PRAM pode ser simulada pela máquina SIMD-PRAM.

A especificação dos conflitos de acesso à memória global podem ocorrer quando mais de um registrador tenta ler ou escrever sobre a mesma célula de memória ao mesmo tempo. Entretanto, para descrever os algoritmos para a máquina PRAM, aplica-se as restrições para a escrita, e permite-se a leitura simultânea por vários processadores, na mesma célula de memória.

- Em [SHE 63], os estudos e aplicações mostram o uso das máquinas de registradores como modelos para a teoria básica da recursão. Neste caso a máquina utiliza apenas um conjunto finito, prefixado de seus registradores durante as computações, que pode ser descrito como uma unidade de controle acrescida de um conjunto finito de contadores.
- Minsky [MIN 67] mostrou o fato surpreendente de que, **a máquina de dois registradores** pode simular todo poder computacional de uma máquina universal.

A idéia de von Neumann de abdicar da unidade de controle finito de um computador, e armazenar também os programas na memória viabilizou o acesso à posições na memória pelo endereçamento direto ou indireto, aumentando o poder de processamento dos computadores, expresso nas novas tecnologias e linguagens atualmente em uso. Entretanto, alguns problemas foram surgindo, e por conseqüência alguns estudos foram desenvolvidos no sentido de solucioná-los. Alguns dos efeitos desagradáveis é conviver com a possibilidade de uso de registradores de forma esparsa, tornando possível a existência de regiões da memória onde nada está acontecendo, indicando uma possível fragmentação da memória. Além deste, outro fato que deve merecer cuidado se refere ao custo para a inicialização dos programas, quando, por exemplo, o começo de uma computação exige que todos os registradores envolvidos estejam zerados.

Na teoria dos sistemas de computação seqüenciais diversos modelos matemáticos tem sido estudados. Além dos já citados e comentados, tem-se ainda como muito divulgados e de grande aplicações os sistemas de Post e os sistemas de Markov, cujas referências podem ser encontradas em [BOA 90]. O principal resultado desta teoria foi mostrar que todos estes formalismos são equivalentes em relação ao comportamento das funções de entrada e saída que modelam os sistemas computacionais.

Entretanto, uma grande e expressiva quantidade de sistemas computacionais reais não são de natureza seqüencial. Em todos os níveis, desde um pequeno chip até uma rede mundial, o comportamento dos processos computacionais são distribuídos, no sentido que podem ser desempenhados de formas distintas e independentes, orientados por tarefas agendadas e norteadas por objetivos comuns.

A ausência de uma interpretação única para um algoritmo é usualmente chamada incompletude de sua descrição. Dentro do contexto computacional, um algoritmo é construído com o objetivo de obter a solução para um problema, de acordo com uma precisão desejada, de tal forma que garanta uma velocidade de execução compatível com tal abordagem. Além disso, devem ser explorados os recursos disponíveis e consideradas as propriedades relacionadas com a estrutura dos dados associados. Mas tais considerações raramente são explicitadas no algoritmo, quando muito ficam implícitas e sujeitas a diferentes interpretações nas possíveis implementações.

Por exemplo, se uma notação matemática é usada para descrever um algoritmo, nem sempre a ordem de execução de operações individuais é precisamente explicitada.

Isto significa que a implementação do algoritmo pode ser efetivada sem considerar relevante a ordem de execução de algumas operações, supondo que estas sejam equivalentes sob o ponto de vista matemático. Entretanto, o mesmo não ocorre sob o ponto de vista computacional e resulta em algumas propriedades indesejáveis como, por exemplo, a instabilidade numérica.

Isto ocorre por negligenciar uma análise de propriedades como associatividade, comutatividade, e distributividade que apesar de serem satisfeitas por vários conjuntos numéricos, não podem ser verificadas quando da execução do programa.

A análise da precisão dos resultados obtidos e o controle dos erros de arredondamento são inevitáveis, quando da execução de processos computacionais. A maioria das linguagens não oferece pistas de como é a estrutura interna de suas operações, e muitas vezes, devido à diversidade dos métodos de intermediar tais erros, um mesmo programa pode gerar resultados distintos quando executado em diferentes computadores. Esta é uma das principais causas da dificuldade de escrever software numérico portátil.

Em certas arquiteturas, além da execução dos programas demonstrar pouca ou quase nenhuma dependência na ordem de execução de operações individuais, o tempo exigido para execução não se relaciona com a troca e nem com o compartilhamento dos diferentes tipos de dados. Estas características foram herdadas das arquiteturas dos computadores de von-Neumann, mas sem relevância para computações seqüenciais.

Muitos trabalhos tem sido direcionados na busca de minimizar estes problemas principalmente na modelagem de grandes sistemas, mas os resultados ainda não são surpreendentes. Destacam-se a análise e limitação dos erros, que não se constitui numa abordagem construtiva e a análise intervalar, associada a um custo alto em termos de tempo computacional.

O uso eficiente das modernas arquiteturas paralelas implica na execução simultânea das ramificações dos dados computacionais que trabalham de forma independente. Não basta determinar como extrair os dados da memória, como desempenhar algumas manipulações e operações entre estes dados e como armazenar os resultados novamente na memória. É preciso especificar, explicitamente, sobre quais operações estes dados tem impacto. A ausência desta informação resulta no conhecido problema da dificuldade de se utilizar programas seqüenciais em computadores paralelos.

2.1.3 Máquina de Grafos

Uma proposta alternativa para estas questões que se tornaram muito atuais com o surgimento das máquinas paralelas é apresentada em [VOE 92], e que se identifica com o modelo de máquina paralela proposto nos próximos capítulos. Neste caso, utilizando uma abordagem teórica restrita a multigrafos dirigidos e acíclicos, o autor define a Máquina de Grafos. Nesta máquina, as operações são representadas por nodos e os arcos nos grafos descrevem a transferência de dados entre as operações.

A introdução de um sistema de coordenadas possibilita a indexação dos nodos, dos arcos e a definição de uma agenda (relativa ao tempo e espaço) associada a cada grafo. A máquina de grafos torna explícito o número e o tipo de seus processadores e de sua rede de comunicação.

A notação de grafos é uma poderosa linguagem para formalização das propriedades de um algoritmo, que torna viável o uso eficiente dos computadores paralelos. Neste sentido, o grafo de um algoritmo é capaz de explicitar

- as operações que podem ser executadas simultaneamente;
- as operações que devem ser executadas seqüencialmente;
- a dependência entre operações com rótulos distintos; e ainda
- o deslocamento do fluxo de dados na execução das implementações do algoritmo.

Sem perda de generalização, cada dois nodos admite no máximo um arco. Em

particular, um grafo de ordem linear é determinado pela enumeração de seus nodos, de acordo com o sentido dos seus arcos.

Algumas observações são apresentadas, pois ajudam na compreensão do modelo de máquina de grafos e se identificam com o trabalho desenvolvido nos próximos capítulos.

- As agendas generalizadas ou estritas associadas aos grafos.

As agendas informam e evolução temporal quando da execução da implementação do algoritmo. Quando satisfazem restrições impostas pelas características do algoritmo, viabilizando por exemplo, uma execução válida de acordo com uma determinada ordenação das operações, são denominada estritas.

As agendas são indicadas por vetores que especificam as restrições, associadas às operações do algoritmo indexadas por coordenadas destes vetores, que se referem ao tempo ou intervalo de tempo decorrido na sua execução.

Portanto, as agendas permitem reconhecer a oportunidade de se executar operações em paralelo. Existe, inclusive, uma correspondência biunívoca entre o conjunto das possíveis formas de paralelizar um algoritmo e o conjunto de agendas.

As agendas na máquina de grafos se identificam com os níveis da estrutura da máquina geométrica interpretada pelo espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$, onde o tempo é finito. Neste caso, caracterizando o paralelismo espacial restrito no tempo, e memória compartilhada. Além disso, na máquina geométrica distribuída, interpretada pelo espaço coerente $\mathbb{D}_{\infty}^{\rightarrow 2}$, as agendas também refletem os níveis da construção que neste caso, modela o paralelismo temporal, restrito agora ao número de máquinas envolvidos na modelagem do sistema computacional.

O tempo de execução de um algoritmo é outra característica importante para análise de sua eficiência e eficácia, que não depende apenas do algoritmo mas também da estrutura e característica do computador.

A implementação de computadores seriais, com um único processador, tem ao longo do tempo criado critérios bem-definidos para comparar algoritmos quanto ao tempo de execução. Tais procedimentos consistem na contagem das operações desenvolvidas quando da execução de ambos algoritmos, a partir da definição de um critério de precisão. Esta prática reflete a característica mais importante da implementação serial de algoritmos razoavelmente bons e que independem das possíveis modificações no hardware. Por esta razão a contagem de operações tornou-se um dos critérios mais eficientes para análise da complexidade de um algoritmo.

Entretanto, esta tarefa não é simples quando se trata de máquinas paralelas. Para tal, seja uma máquina paralela abstrata, possuindo infinitos processadores, todos eles capazes de medir seu desempenho na execução de uma operação em unidades de tempo, e de forma independente. Assumindo-se ainda que as tarefas de comunicação entre processadores, o tráfego de memória e a transferência de dados de entrada/saída, não possuem custo computacional (unidades de tempo).

- Tempo de execução e a constuição do grafo de um algoritmo.

Pode-se então utilizar a máquina paralela abstrata para comparar o tempo de execução de um algoritmo em um computador paralelo. Considere que o objetivo principal seja uma análise das propriedades associadas ao algoritmo, e que tal objetivo será alcançado ao examinar as diferenças entre as várias implementações de um mesmo algoritmo. De acordo com [VOE 92], as vantagens obtidas com a máquina de grafos, neste caso, são eminentes e podem ser estendidas no sentido de se examinar as várias implementações de vários algoritmos.

A existência de dispositivos de armazenamento de dados de alta capacidade não significa velocidade de acesso suficientemente baixa. Em muitas situações práticas, o tempo de acesso é substancialmente maior que o tempo para executar uma operação aritmética. Esta são razões plausíveis para justificar que as amostragens dos vários tipos de memórias sejam levadas em consideração quando da análise do algoritmo, considerando que os problemas devam ser resolvidos em tempo viável.

Além da variação entre os modelos ou memórias existentes (seqüenciais, endereçadas, associativas) os sistemas de computadores podem incluir vários bancos de memórias de mesmo tipo, que se comunicam via rede ou canais de comunicação, que também possuem limitações. Neste casos, o acesso nem sempre é imediato e as operações básicas de *leitura/escrita* podem não ser instantaneamente executadas.

- A memória computacional associada aos grafos de algoritmos.

O tamanho da memória RAM requerida para execução da implementação de um algoritmo reflete as características do algoritmo ou restringe-se apenas à sua execução, nível por nível? Em termos práticos, isto significa questionar se a execução em blocos, a partir de uma estrutura algébrica, de um processo matricial associado a um algoritmo melhora o seu desempenho computacional em relação a sua execução linha por linha, ou coluna por coluna?

A máquina de grafo constitui-se numa poderosa ferramenta na busca de respostas para estas questões. Isto se torna viável com a possibilidade de partição de um grafo em subgrafos, que possuem a mesma estrutura (inclusive sob o ponto de vista algébrico). Desta forma, cada subgrafo é visto como um macronodo, que podem ser executados de forma independente, seqüencial ou concorrentemente.

Por conseqüência, tem-se uma partição da memória em blocos, definidos pelas coordenadas de hiperplanos, resultando, no caso tridimensional, em paralelepípedos limitados. De acordo com [VOE 92], se apenas as operações pertencentes a cada macronodo são armazenadas na memória, e os macronodos forem executados seqüencialmente, e as trocas de dados entre os macronodos sendo desempenhadas por uma memória externa, a implementação do algoritmo é substancialmente eficiente. Isto porque o tamanho de memória externa é proporcional à área da superfícies dos macronodos, enquanto que o número de operações aritméticas que serão executadas pela máquina RAM dentro de cada paralelepípedo será proporcional ao seu volume.

Sabe-se que o volume cresce substancialmente mais rápido que a superfície de área, em relação ao crescimento das dimensões de cada paralelepípedo. Entretanto, a partição tem por objetivo melhorar este nível de desempenho do algoritmo, e ainda existe a possibilidade de que cada macronodo seja executado em paralelo. Se, além disso, ainda não se atingiu o desempenho desejado, pode-se executar os macronodos em paralelo, utilizando uma memória distribuída, com intercomunicação novamente estabelecida via memória externa.

Em [VOE 92], todas estas considerações são relevantes para a construção do grafo que descreve todas as características do algoritmo, para que depois, sejam explicitadas em sua implementação.

Intuitivamente, é possível se definir um relacionamento entre os grafos e as representações gráficas associadas às interpretações dos processos no espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$. Da mesma forma, é possível associar os subespaços vetoriais nos quais estão inseridos os vetores que representam as agendas que determinam as restrições de espaço e tempo, com os subespaços coerentes \mathbb{D}_n .

Por fim pode-se imaginar uma modelagem mais abrangente que contemple uma família maior de algoritmos, baseado num relacionamento intrínseco entre as teias geradoras dos espaços coerentes, que se constituem em grafos reflexivos e não-dirigidos, e os

grafos dirigidos e acíclicos propostos [VOE 92]. Ou seja, todos os processos matriciais correspondentes a algoritmos modelados pela máquina de grafos podem ser interpretados por conjuntos coerentes na estrutura ordenada sobre a qual se define a máquina geométrica. Entretanto, pode-se questionar se outros algoritmos podem ser interpretados. Isto não será desenvolvido deste trabalho, mas um referencial para trabalhos futuros.

Na próxima seção faz-se um histórico da teoria dos domínios semânticos desde sua fundamentação e apresentação formal por Scott até o final do século passado, com referência a alguns trabalhos que foram relevantes para alcançar nossos objetivos, mostrando também a diversidade de aplicação e os resultados atingidos na definição dos modelos computacionais existentes.

2.2 Modelos Semânticos

Os primeiros estudos em *computabilidade efetiva* usavam como ferramentas as funções recursivas, o λ -cálculo e as diferentes versões das máquinas de Turing. Nas últimas duas décadas, a partir da construção dos computadores eletrônicos, verifica-se a evolução das linguagens de programação de alto-nível, nas quais os programas passam a ser escritos de uma forma muito conveniente, independente da arquitetura de uma máquina específica.

A partir daí, o desenvolvimento de tais linguagens leva ao estudo e à análise mais profunda das questões referentes à sintaxe e à semântica associadas a estas linguagens.

Neste sentido, a sintaxe analisa a estruturação de strings de símbolos pertencentes a determinados alfabetos com os quais se pode escrever programas, obedecendo a critérios formais de especificação. Por outro lado, a semântica busca o significado deste programa ou, ainda, uma interpretação para o comportamento destes programas quando executados por um computador.

Para alcançar tal interpretação, duas importantes situações devem ser satisfeitas por qualquer teoria semântica: a primeira se refere à capacidade de interpretar a diversidade de tipos definidos pela linguagem, e a segunda se relaciona com a análise das propriedades de computabilidade das funções presentes no programa. Por exemplo, quando uma transformação de dados de entrada em dados de saída é associada a uma determinada função, a análise semântica simultânea envolve um estudo de seu domínio, contra-domínio e propriedades como monotonicidade, continuidade, estabilidade, linearidade e outras mais, como a composição. Em termos matemáticos, pela interpretação semântica de cada programa sintaticamente válido tem-se uma descrição da função computada pelo programa.

Esta integração entre semântica e sintaxe nos permite verificar as propriedades de um programa através de uma comparação direta entre sua denotação e sua especificação, constituindo-se numa poderosa ferramenta para verificação e validação de programas.

A semântica é essencialmente uma teoria matemática aplicada às linguagens de programação com o objetivo de dar o significado pretendido de tudo que é construído desde as descrições até as implementações. Seu papel se estende tanto na busca do desenvolvimento de métodos de verificação de programas e na análise sistemática de programas que já existem como também no desenho de novas linguagens, mais simples e mais regulares.

O significado dos programas pode ser descrito em termos de seqüências de etapas de computações em um computador abstrato ou idealizado. Esta abordagem caracteriza a *semântica operacional* que apesar de ser útil na formulação de técnicas precisas de implementação, tem dificuldades em mostrar equivalências entre diferentes estilos de implementações de uma linguagem pois não possui a representação padrão independente com a qual tal linguagem pode ser relacionada. Veja [MIL 90].

Numa abordagem mais abstrata, o significado de um programa é alcançado ao interpretar as descrições implícitas da linguagem, o que caracteriza a *semântica axiomática*

ou semântica das asserções. As duas principais características de tais sistemas formais é a coerência (toda expressão derivada do sistema deve ser verdadeira na interpretação pretendida) e a completude (toda expressão verdadeira derivável pertence ao sistema formal). Nesta abordagem destacam-se os relevantes trabalhos de Floyd e Hoare [HOA 69, FLO 67].

Ao contrário da semântica operacional, cuja ênfase está nos estados intermediários de uma computação, na *semântica denotacional* a observação se concentra no comportamento de um programa baseada na análise dos dados de entrada/saída, ignorando tais estados intermediários.

Neste sentido, a denotação de um programa se caracteriza pela existência de uma função de avaliação (ou interpretação) mapeando toda expressão válida na linguagem para seu significado em um apropriado domínio de entidades matemáticas. Esta função de avaliação tem uma propriedade muito conveniente, a composição (o significado de uma expressão é a composição dos significados referentes a cada subexpressão).

Dedica-se a seguir uma especial atenção à semântica denotacional, com ênfase na Teoria dos Domínios, que se constitui na área de maior interesse deste trabalho.

2.3 Semântica Denotacional das Linguagens de Programação Determinísticas

Os primeiros trabalhos na direção de uma análise conceitual das linguagens de programação se devem a Strachey, ao introduzir funções semânticas que mapeavam a sintaxe abstrada dos operadores e um operador de ponto fixo para denotar *loops*. Entretanto, a escolha em aplicar λ -abstrações (no lugar da notação funcional) para especificar as denotações levou a um problema de fundamentação.

Dana Scott [SCO 70] em 1970, estabelece a teoria dos domínios semânticos que provê uma fundamentação adequada para as descrições semânticas introduzidas por Strachey.

Em uma abordagem conjunta, Scott e Strachey [SCO 71] apresentam a essência da Semântica Denotacional onde as funções semânticas são contínuas e sua existência é garantida por um teorema de ponto fixo.

Em [SCO 72a] Scott introduz uma teoria abstrata de aproximações finitas, interpretadas como computações parciais, mas que também contempla limites infinitos como computação totais, e que por estas razões não se limita apenas à teoria clássica da computação. A primeira definição de domínio, apresentada por Scott [SCO 70, SCO 72, SCO 82a], foi a de reticulado (ou semi-reticulado) contínuo (veja também [GIE 80]), e, inspirado em noções topológicas [SMY 90, SMY 91], conseguiu definir diversos conceitos de interesse computacional sobre reticulados, como, por exemplo, a noção de função contínua e elemento compacto ou finito.

Posteriormente, em [SCO 72] Scott mostra que *espaços topológicos* podem ser inseridos em reticulados contínuos sempre que forem homeomórficos ao seu próprio espaço de funções e em [SCO 82] o autor faz uso da estrutura de sistemas de informação como uma forma alternativa de aplicar a mesma teoria na interpretação semântica para linguagens de programação de alto nível. Outras contribuições fundamentais foram [SCO 73, SCO 76, SCO 82, SCO 82a, SCO 90, REY 77].

Algumas idéias relativas ao assunto foram introduzidas anteriormente por Daniel Lacombe [LAC 55] na teoria da recursão. Contribuições importantes à teoria dos domínios também foram introduzidas independentemente por Y. L. Ershov [ERS 72]. Textos introdutórios sobre esta teoria apareceram ainda com a sua utilização em semântica denotacional de linguagens de programação em [STY 77]. Entretanto, sua introdução como uma teoria matemática em um texto didático aparece com Plotkin [PLO 81].

Esta teoria está já estabelecida [STV 94, EDA 97, GIE 80, JOH 82, GUN 92, ABR 94] e é na atualidade o maior paradigma em semântica denotacional de linguagens de programação. Veja também [KAM 84, KAM 84a, PLO 77, LAW 87, LAW 97]. Da mesma forma a equivalência de descrições denotacionais envolvendo diferentes domínios pode ser evidenciada, veja [MII 74].

2.3.1 Domínios de Scott

Neste seção apresentam-se as principais características da estrutura ordenada proposta por Scott, capaz de interpretar a informação que flui pelos diagramas de fluxos que compõe um programa. Tal interpretação constitui-se na principal motivação para o presente trabalho.

A idéia básica de Scott é representar tipos de dados por certos conjuntos parcialmente ordenados, denominados domínios. Quando a computação está baseada em um algoritmo, então cada um dos conjuntos de dados de entrada e saída é visto como um domínio. O programa que executa a computação é representado por certas funções entre domínios, denominadas de funções contínuas.

Uma *função contínua* é aquela que preserva a ordem de informação (quanto maior a informação dos dados de entrada, maior será a informação dos dados de saída) e os limites de computações infinitas no domínio (a informação total disponível como saída de uma seqüência infinita de elementos de entrada é o somatório total de toda a informação obtida de cada elemento de entrada).

A tese de Scott afirma que toda função computável é contínua no sentido acima. Funções com o mesmo domínio de entrada e saída, com a ordem pontual, formam um domínio de funções ou o espaço funcional. Toda a função contínua em um domínio com um menor elemento possui um menor ponto fixo no sentido do teorema de Tarski. Isto implica que o significado de um programa recursivo pode ser capturado como o ponto fixo de uma função de ordem mais alta, que é definida, pela recursão correspondente, no domínio de todas as funções de um dado tipo.

Existe uma variedade de categorias de domínios de acordo com as propriedades adicionais consideradas, desde que satisfaçam os critérios estabelecidos em [JUN 89].

Ainda, para modelar computações infinitas, exige-se que um domínio seja completo no sentido de que cada seqüência crescente de aproximações seja representada por um elemento no domínio, ou seja, deve possuir um supremo. Estes requisitos são suficientes para a obtenção de pontos fixos de funções contínuas e também para a construção de espaços funcionais. Obtém-se assim os domínios denominados de ordens parciais completas ou cpo's.

Como bem observado em [DIM 98], uma computação é executada sobre objetos concretos e o resultado da computação também é dado por uma seqüência de elementos concretos. Então, para modelar computações, é necessário abstrair a noção de ser um elemento concreto.

Tal abstração, na teoria dos domínios, é denominada de *elemento finito ou compacto*. Assim, exige-se que cada elemento de um domínio seja representado por todas as suas aproximações finitas ou compactas, ou seja, que cada elemento do domínio seja o supremo do conjunto dirigido de suas aproximações compactas. Quando isto acontece, tem-se os chamados cpo's algébricos. Os elementos compactos ou finitos do domínio formam a base do domínio.

A classe dos cpo's algébricos parece possuir as propriedades de computabilidade desejadas. Entretanto, observa-se a não existência de uma propriedade importante para a ciência da computação e também para partes da teoria da computabilidade: esta classe não é fechada para a construção de espaço funcional. Então, uma subclasse de cpo's algébricos é freqüentemente considerada - a dos que são consistentemente completos, ou seja, quando

todo subconjunto consistente do domínio possui um supremo. Estas estruturas, ou seja, os cpo's algébricos consistentemente completos são finalmente denominadas de *Domínios de Scott*, ou simplesmente domínios algébricos. Os espaços coerentes, estrutura fundamental deste trabalho, constituem um tipo especial de domínio algébrico.

Os elementos básicos podem também ser considerados como um conjunto de proposições lógicas que caracterizam qualquer elemento do domínio. Isto foi primeiramente observado por D. Scott [SCO 82] para os Domínios de Scott, tendo sido posteriormente generalizado para outras classes de domínios. Um domínio algébrico possui uma apresentação simples em termos de um sistema de informação, uma estrutura lógica sobre os elementos básicos que fornece uma indicação de como construir os elementos do domínio como as teorias da lógica correspondente. A lógica relativa a um domínio algébrico é a das propriedades observáveis do processo computacional. Esta *lógica de observações* está intimamente ligada à topologia de Scott para domínios [SMY 83a, VIC 87]. Um aberto de Scott pode ser visto como uma proposição sobre um tipo de dados, ou uma propriedade deste tipo.

2.3.2 Contribuições Posteriores

Nos últimos anos, tem surgido uma nova direção para aplicação de domínios contínuos em espaços clássicos da matemática. Tais domínios são generalizações dos domínios algébricos, conservando suas básicas propriedades. Edalat [EDA 97] afirma que domínios contínuos constituem a abordagem natural para a matemática contínua, pois considera que as representações nestes domínios são evidentemente mais diretas e simples que as representações em domínios algébricos.

Os trabalhos fundamentais com domínios contínuos de intervalos, no sentido de obter números reais efetivamente computáveis e as principais aplicações se devem a Edalat [EDA 95, EDA 95A, EDA 95b, EDA 96, EDA 97, ESC 94, ESC 95, ESC 96, ESC 97, EDA 98, EDA 98a, EDA 98b, ESC 99, ESC 99a].

Um domínio algébrico cuja base é enumerável, denominado de domínio ω -algébrico, pode ser dado de modo efetivo, segundo certas condições [WEI 80, ESC 97], com o objetivo de tornar a teoria construtiva e de definir a noção de elemento computável e função computável.

Phoa [PHO 94] introduziu um exemplo de uma nova abordagem para a teoria dos domínios, na qual uma categoria de domínios é construída diretamente de um modelo particular de computação, em uma estrutura intuitiva e natural, utilizando as ferramentas da teoria das categorias e teoria de topos. Bonsangue, Breugel e Rutten [BON 95] estudaram os espaços ultramétricos generalizados, que constituem uma generalização de pré-ordens e espaços ultramétricos ordinários, mostrando como construir a completação, topologia, e domínios potência, combinando a visão topológica de Smyth [SMY 83a, SMY 90, SMY 91, SMY 87] e a visão categórica de Lawvere [LAW 73] em espaços (ultra)métricos generalizados.

Destacam-se os trabalhos de Keye Martin [KEY 99] e Michael Mislove [MIS 99]. Martin desenvolve sua tese de doutorado no sentido de obter uma fundamentação para a computação, trabalhando com noção de medida em domínios, estendo posteriormente tal noção para interpretação de espaços topológicos [KEY 99a, KEY 98].

2.3.3 Teoria dos Domínios e Matemática Intervalar

Nos últimos anos os domínios contínuos têm sido utilizados em trabalhos sobre computação exata de números reais, resultando em algoritmos eficientes, em aritmética computacional de precisão infinita, em sistemas de funções iterativas, teoria de medidas e integração, principalmente com aplicações em matemática e física, como geometria fractal

e física estatística [EDA 97].

Os trabalhos de Edalat [EDA 95, EDA 95A, EDA 95b, EDA 96, ESC 94, EDA 98, EDA 98a, EDA 98b, EDA 97], Escardó [ESC 95, ESC 96, ESC 97, ESC 99, ESC 99a], Potts et al [POT 96, POT 97, POT 97a, POT 97b] baseiam-se na construção de um domínio ω -contínuo IR de intervalos reais fechados e limitados, ordenados pela ordem reversa da inclusão. Este tipo de construção foi originalmente proposta por Scott [SCO 70], com a introdução de um conjunto parcialmente ordenado, com um elemento topo representado pelo intervalo vazio, como um tipo de dado para números reais. Veja também [GIA 93, GIA 96]. Dave Plume [PLU 98], sob a orientação de Martin Escardó e Alex Simpson, desenvolveu uma calculadora para computação de números reais exatos, implementando algoritmos para operações aritméticas, funções transcendentais, integração, etc. Várias aplicações se tornaram possíveis a partir desses trabalhos, principalmente em diversas áreas da matemática, física e engenharia.

Acióly [ACI 91, ACI 91a, DIM 91, DIM 91a, DIM 91b] definiu domínios contínuos de intervalos, com base nos trabalhos originais de Scott, obtendo cpo's intervalares contínuos consistentemente completos com base enumerável, com o objetivo de apresentar uma fundamentação computacional para a Matemática Intervalar. Na continuação do trabalho de Acióly, pode-se citar também Bedregal [BED 94], apresentando um estudo detalhado dos sistemas de informação contínuos.

Uma outra abordagem baseada em domínios algébricos, mais especificamente, Espaços Coerentes, foi apresentada, mais recentemente, por Dimuro [DIM 98]. Esta tese de doutorado consistiu no desenvolvimento de uma metodologia para a obtenção de representações construtivas de sistemas ordenados de segunda ordem, baseadas em estruturas de espaços coerentes, com aplicação fundamental na Computação Científica e Matemática Intervalar.

Dimuro [DIM 98, DIM 2000, DIM 99] obteve como resultado uma representação global para os objetos ditos infinitos relativamente ao conteúdo de informação, como números reais e intervalos reais - *o Espaço Coerente Bi-Estruturado de Intervalos*-, de tal forma que possam ser obtidos modelos semânticos adequados para os processos computacionais envolvendo tais objetos.

Esta representação construtiva é denominada de global, pois, é realizada em dois níveis distinguíveis, compreendendo não somente a construção interna dos objetos da estrutura de informação, do conjunto das aproximações destes objetos e de sua topologia de informação, mas também sua estrutura externa de aplicação, representando um conjunto de operações algébricas, uma relação de ordem de posição, uma família de relações, uma família de funções elementares, uma subestrutura de medidas e uma topologia induzida por ela. Veja também [REI 97].

2.4 Semântica Denotacional das Linguagens de Programação Não-Determinísticas

Nos meados dos anos 70 já havia sido desenvolvido um número suficiente de técnicas de especificação da semântica denotacional para quaisquer linguagens de programação convencional ou sequencial. Nos anos que se seguem, o crescente interesse por sistemas envolvendo processos concorrentes levou ao desenvolvimento das linguagens de programação com construções possivelmente não-determinísticas.

O primeiro resultado neste sentido foi proposto por Robin Milner [MIL 75, MIL 90], que pode ser caracterizado por introduzir uma técnica usando os chamados oráculos, mas sem uma abstração denotacional suficiente, onde escolhas não-determinísticas não eram comutativas.

2.4.1 Semântica Algébrica

A semântica algébrica se identifica com o estudo da iteração e recursão usando ou a álgebra universal ou a correspondente abordagem categórica relacionada com as teorias algébricas introduzidas por Lawvere [LAW 73]. Busca-se com esta abordagem semântica um tratamento uniforme para temas como as gramáticas livres de contexto, fluxogramas e domínios recursivamente definidos.

Destaca-se nesta abordagem uma importante classe de modelos matemáticos para processos concorrentes, denominada classe do cálculo algébrico. Neste caso, processos são entendidos como uma classe de termos congruentes, segundo uma relação de congruência, responsável pela interpretação da igualdade de processos. As referências apresentadas abaixo são sugeridas em [MIL 90, WAG 90].

As construções clássicas de somas e produtos que se constituem em operações básicas de construção de processos e sua correspondente iteração leva à construção de uma álgebra de processos no sentido da Teoria de Modelos. A estrutura algébrica garante a existência e unicidade das soluções das equações, incluindo as definições recursivas.

Existem muitas e distintas abordagens algébricas para o tratamento e interpretação de processos. Em alguns importantes sistemas concorrentes, embora tenham uma apresentação algébrica, a aplicação de métodos analíticos (justificada pela semântica relacionada com a teoria das redes) é mais acentuada que as técnicas algébricas, restritas no caso ao modo pelo qual os processos são construídos (a partir de outros menores) e a utilização dos operadores da álgebra somente na definição de construtores de processos.

A ênfase algébrica ficou realmente caracterizada por Milner [MIL 80] que introduziu uma álgebra para sistemas de comunicação denominado CCS (Calculus of Communicating Systems). Este trabalho pioneiro no corpo da Álgebra dos Processos, se constitui na base de outros sistemas algébricos. Veja também [MIL 90]. Neste sistema tanto os processos sequenciais como os paralelos são representados por operadores de uma única teoria algébrica. Apresenta também o uso de congruências ao invés de equivalências - bis-similaridades - relacionando-as sempre com o significado operacional dos processos. Veja também [BAE 90].

De modo independente, Hoare [HOA 85] apresenta CSP (Communicating Sequential Processes) uma linguagem de programação concorrente que concorda com CCS na maneira primitiva de interação entre processos, denominada de sincronização de envio e recebimento de dados (mensagens) através de agente específicos.

A semântica interpretada por álgebras iniciais são equivalentes à semântica denotacional, com a vantagem de explicitar a sintaxe abstrata como uma álgebra inicial, de tal forma que as provas indutivas na semântica podem ser substituídas pela inicialidade na álgebra. Tal semântica admite extensões contínuas e as correspondentes construções duais (Coálgebras Finais) que também modelam os processos computacionais. Veja [GOG 77, RUT 93].

Um número considerável de resultados no contexto da abordagem algébrica para a semântica das linguagens de programação, com ênfase no relacionamento entre indução, inicialidade e computabilidade, pode ser encontrada em [MES 74]. Caracterizações indutivas da inicialidade, incluindo generalizações dos axiomas de Peano, construções em álgebras computáveis e seu relacionamento com modelos iniciais de representação finita, são também estudados e analisados detalhadamente. Veja também [BEA 94].

Em [REI 98, REI 98a, REI 99] apresenta-se uma abordagem categórica para indução e recursão nos naturais considerando a categoria US dos sistemas unários, formalizada por Escardó [ESC 95] e introduzida por Stoll [STO 61]. A composição de transformações naturais definidas por funtores formaliza a conexão fundamental entre esta e a categoria Set . Neste sentido, a inicialidade garante as construções recursivamente definidas sobre seus objetos. Considerando os sistemas unários como estruturas algébricas e seus homo-

morfismos como preservando tais estruturas, a categoria dos sistemas unários pode ser vista como uma subcategoria das álgebras. Aponta-se também para uma conexão entre as estruturas concretas em \mathcal{US} e as estruturas ordenadas em \mathcal{Poset} no sentido de buscar uma interpretação computacional para os esquemas relacionados com a μ -recursão. Os resultados alcançados se baseiam na caracterização dos naturais parciais como um *cpo* de todos os subobjetos do sistema unário dos números naturais estendidos.

2.4.2 Estruturas de Eventos

Um outro modelo significativo para computação de processos concorrentes são as estruturas de eventos que surgiram naturalmente como um subproduto no trabalho pioneiro de Kahn e Plotkin na fundamentação da semântica denotacional.

O mérito de Whinskel [WHI 77, WHI 88] foi torná-la uma teoria substancial para a semântica das linguagens de programação paralelas assim como as linguagens de alto nível como CCS e CSP [WHI 82, SAS 96], estabelecendo uma importante conexão entre a Teoria das Redes e a Teoria dos Domínios, compartilhando idéias de ambas [NIE 81].

As estruturas de eventos podem ser entendidas como um conjunto parcialmente ordenado de eventos associados a uma relação simétrica de conflito. A relação de ordem modela a dependência causal de tal forma que dois eventos que não são comparáveis e nem estão em conflito podem ocorrer sem ordenar suas ocorrências. As configurações ou estados associados a estas estruturas naturalmente refletem a informação obtida a partir da ocorrência dos eventos, caracterizam-se como um domínio de Scott. Veja [?] para uma abordagem categórica do relacionamento entre as diversas categorias de estruturas de eventos.

2.4.3 Domínios-Potência

Dentro da Teoria dos Domínios, os domínios-potência se caracterizam como ordens parciais completas com as quais é possível construir uma semântica denotacional adequada para linguagens de programação que envolvam computações não-determinísticas. Desde que são definidos sobre conjuntos de conjuntos, seus elementos representam conjuntos de diferentes cursos que uma computação não-determinística pode assumir.

As formas mais simples de domínios-potência foram introduzidas por Eagli e Milner mas sua construções eram restritas aos domínios planos ou discretos. Existem diversas maneiras de ordenar os elementos de um domínio-potência; e ainda mais, diferentes ordens produzem diferentes domínios-potência.

Plotkin apresenta uma construção de domínio-potência para uma classe geral de domínios algébricos, ou seja domínios que possuem uma base de elementos compactos. Embora a construção do domínio-potência de Plotkin produza um domínio algébrico a partir de outro domínio algébrico, nem sempre tem-se assegurado que estes domínios algébricos sejam fechados para o espaço de funções ou a exponenciação, construções vitais para a semântica denotacional. Entretanto Plotkin foi capaz de mostrar uma restrição, os SFP-domínios, e obter uma categoria simultaneamente fechada para as construções em domínio-potência e espaço funcional. Veja [SMY 83a] para uma apresentação formal da prova de que SFP constitui-se na maior categoria cartesiana fechada para as construções em domínios. Veja também os DI-domínios (ou domínios estáveis) na seção 2.5.4.

Os domínios-potência [STV 94, PLO 81] de Hoare e de Smith são construções categóricas duais, e o de Plotkin, uma generalização dos dois primeiros, mas todos eles estão baseados em três visões diferentes que expressam o tipo de informação que deve ser levado em conta nos processos não-determinísticos. Embora conceitualmente simples, os domínios-potência não são estruturas simples de serem aplicadas. Em [WHI 85] apresenta-se uma conexão entre as asserções da lógica modal e os domínios-potência que

podem interpretar computações não-determinísticas.

2.4.4 Semântica Parcialmente Aditiva.

Outras abordagens são também interessantes e foram objeto de investigação quando da estruturação deste trabalho, como por exemplo, a semântica parcialmente aditiva, apresentada em [MAN 86]. A característica mais acentuada desta semântica é a representação de programas através de funções parciais e de multi-funções de tal forma que os construtores de programas constituem-se em operações que manipulam tais funções, com referência a um tipo especial de operação soma, que leva a construções categóricas como os monóides parcialmente aditivos.

2.5 Semântica dos Tipos nas Linguagens de Programação

A noção de "tipo" fundamenta-se em duas linhas de desenvolvimento. Uma delas utiliza tipos para contornar problemas de fundamentação da matemática e assim, ao considerar tipos distintos resolver os problemas relacionados com paradoxos que levam a sistemas inconsistentes. A outra linha de desenvolvimento, mais recente, analisa e aplica a teoria dos tipos nas linguagens de programação. As linguagens de programação de alto nível adotam uma classificação dos dados em tipos, de tal forma que possam servir para diferentes propósitos. Destacam-se alguns nos próximos parágrafos. Os principais aspectos abordados nesta seção podem ser encontrados em [GUN 94, REY 74].

Uma das primeiras razões que levaram ao uso de tipos em linguagens de programação (caso do Fortran) foi o ganho de eficiência principalmente no gerenciamento da memória e a diminuição dos testes de validade de operandos em expressões.

Uma outra razão está relacionada com a propriedade de disciplinar os programas, de forçar restrições na forma de construção e definição dos mesmos. Em programas construídos usando estruturas de tipos é possível se descobrir erros em tempo de compilação, evitando que estes erros ocorram quando o programa é executado sobre a estrutura de dados definitiva. Entretanto, nem todos os programas que possuem bons desempenhos podem ser enquadrados nesta estrutura, ou ainda, o custo em reconstruir seus códigos ainda se mantém alto.

Por outro lado, os programas desenvolvidos na linguagem de tipos podem suportar dados abstratos e sobretudo a modularidade, que leva à separação entre a estrutura de suporte e a estrutura de aplicação, ambas relacionadas à especificação de novos programas. Estas duas estruturas podem ser automaticamente analisadas para determinar parcialmente a corretude do programa antes mesmo de ser realizada a compilação.

Mas a principal motivação para utilização de tipos em linguagens de programação é que seu uso facilita a compreensão e o significado dos programas. Neste sentido, uma linguagem baseada em tipos se constitui numa poderosa ferramenta de classificação de programas. Isto é alcançado pela abstração dos objetos em relação à estrutura que os representa, ou seja, a análise do tipo associado a cada objeto independente da seqüência de bits com que serão compilados.

2.5.1 Teoria dos Tipos em Programação

Uma das importantes contribuições da teoria matemática dos tipos para as linguagens de programação foi o uso de *funções de ordem-superior* na descrição e denotação de programas. Como exemplos salientam-se as linguagens Scheme baseadas no λ -cálculo não-tipado IEE:91 e ML [Milner:90]. Ambas empregam funções de ordem-superior como uma ferramenta para escrever um código suscinto e claro, além de capturar boas abstrações.

Outro modo de obter vantagens com o uso de tais linguagens é combiná-las com referências e atribuições.

Outra característica importante foi a aplicação de *tipos recursivos* relacionados com operadores de ponto-fixo, que também permitem uma validação parcial dos programas antes de sua compilação, mas que requerem por conseguinte, a construção de um sistema de inferência de tipos bem mais geral.

Em linguagens de programação baseadas em tipos é muito comum referências à idéia de que um símbolo pode ser interpretado por muitos tipos. O *polimorfismo* surge a partir do uso de uma variável que especifica um tipo parametrizado ou uma indeterminação para uma expressão. Assim, as funções polimórficas se constituem em termos do tipo abstração, ou ainda, são funções que podem ser avaliadas por outras de mesmo tipo. Isto é alcançado pela introdução do tipo variável e pela quantificação sobre todos os tipos.

Com exceção do ML-polimórfico, o λ -cálculo polimórfico, descoberto independentemente por Girard [GIR 92] que trabalhava em problemas da teoria da prova e por Reynolds [REY 74] que estava interessado no projeto de linguagens de programação, é um dos mais conhecidos e expressivos sistemas que usam o polimorfismo de tipos. Veja também [REY 84].

Uma outra noção de polimorfismo em linguagens de programação é a noção de um *subtipo* que busca uma classificação dos dados de acordo com as coleções de atribuições e propriedades de diferentes classes de uma categoria. A noção de hierarquia de sistemas está associada à classificação de diferentes subtipos.

2.5.2 Tipos Simples como Conjuntos

A notação λ , explicitamente introduzida pelo lógico Alonzo Church, foi desenvolvida no sentido de formalizar uma teoria geral para funções computáveis que pode ser estendida para uma teoria mais geral na matemática através das noções lógicas associada a esta. Na ciência da computação, o λ -cálculo tipado simples é o exemplo mais básico do cálculo tipado com uso de funções de ordem-superior. Nele são descritas uma coleção de termos e tipos juntamente com sistemas de equações. Veja também [STA 85] como uma referência relacionada à análise e aplicação de propriedades das relações lógicas, uma das ferramentas básicas para raciocinar sobre tipos. Um estudo geral sobre as relações lógicas pode ser encontrado em [MIT 90].

2.5.3 Tipos Simples como Domínios

Considerando-se que a noção de avaliação nas linguagens de programação não se reduz a uma teoria equacional, a semântica operacional em termos das equações do λ -cálculo tipado simples se reduz em uma linguagem inexpressiva.

Isto ocorre porque as estruturas necessárias para modelar tipos devem levar em conta conceitos computacionais como divergência e recursividade. Baseada nestas condições, a teoria dos domínios relacionada com as semânticas de tipos constituem-se numa poderosa linguagem de programação, podendo ser considerada uma extensão para o λ -cálculo tipado simples. Isto é alcançado quando, a partir da semântica operacional, se consegue gerar uma teoria equacional relativa a uma pré-ordem que induz as desejadas equações.

Diversas categorias cartesianas fechadas de domínios algébricos, incluindo os chamados Domínios de Scott, tem sido empregadas em semânticas de computação. São utilizadas para obter um modelo não-trivial [SCO 72, SCO 73] do λ -cálculo não-tipado [BAR 84] baseado em um domínio não-trivial isomórfico ao seu espaço funcional.

Também foi introduzida em [PLO 77] uma semântica denotacional para PCF (Programming Language for Computable Functions), um λ -cálculo tipado com tipos básicos para números naturais e valores booleanos acrescido de constantes para operações básicas

sobre estes tipos. PCF pode ser considerada como modelo teórico para linguagens de programação funcional.

Na busca de um modelo totalmente abstrato para a linguagem PCF foi e continua sendo pesquisada. Os bc-domínios (bound complete domains) são ordens parciais completas onde todo subconjunto que tem uma cota superior tem a menor cota superior, de tal modo construído que o espaço de funções contínuas, com a ordem ponto a ponto, é também um bc-domínio. Tais estruturas ordenadas podem ser usadas como modelos para o tipo PCF, entretanto não é um modelo totalmente abstrato. Conforme é mostrado em [PLO 77], pode ocorrer que dois termos tenham o mesmo comportamento operacional mas falham em ser iguais no modelo.

2.5.4 dI-Domínios

Com o mesmo objetivo de fundamentação e abstração para PCF, Gerard Berry e Curien [BER 78], introduzem os dI-domínios, mas o esforço de encontrar o modelo totalmente abstrato para o tipo PCF foi em vão. Tais estruturas ordenadas são tipos especiais de domínios de Scott, os quais possuem uma natureza mais operacional. Constituem-se em bc-domínios com duas propriedades adicionais:

1. a distributividade do supremo - operação binária que determina a menor das cotas superiores entre dois objetos do domínio - em relação ao ínfimo - operação binária que determina a maior das cotas inferiores entre dois objetos.
2. a propriedade I que assegura que o conjunto de aproximações de qualquer elemento compacto do domínio é sempre finito.

As funções entre dI-domínios são funções estáveis segundo uma ordem muito intuitiva, que leva em conta a maneira na qual elas computam. As funções estáveis podem prover uma semântica adequada para os algoritmos seqüenciais.

Salienta-se a linguagem CDS (Concrete Data Structures) proposta em [BER 78] que possui uma elegante teoria matemática onde estruturas de dados concretas e algoritmos formam uma categoria cartesiana fechada cujos objetos estão intimamente relacionados com estruturas de eventos e redes de Petri, veja [WHI 88].

Ao introduzir uma primitiva capaz de modelar o "ou paralelo", Plotkin introduz uma extensão da linguagem PCF que pode ser totalmente modelada por bc-domínios. Entretanto para os dI-domínios a questão permanece ainda em aberto.

Os dI-domínios podem ser representados como estruturas de eventos [ZHA 91] que podem ser aplicados na modelagem de processos concorrentes e para relacionar domínios mais abstratos com estruturas mais concretas como as redes de Petri.

A mais larga categoria cartesiana fechada dos domínios estáveis foi estudada em [ZHA 96a]. Um estudo relacionando a categoria dos SFP-domínios e a categoria dos domínios estáveis é apresentada em [ZHA 91a].

2.5.5 Espaços Coerentes: Domínios de Girard

Com o objetivo de dar uma semântica denotacional para a Lógica Linear, Girard [GIR 86, GIR 87, GIR 89] introduziu o estudo de uma importante subcategoria dos dI-domínios - os *espaços coerentes*-, onde os objetos são conjuntos construídos segundo uma relação reflexiva e simétrica, denominada de relação de coerência, e a ordem de informação é a relação de inclusão entre conjuntos. Os espaços coerentes ou domínios de Girard foram usados para modelagem de sistemas \mathbf{F} [GIR 86].

Girard justifica seu trabalho, partindo de que existem várias discussões em torno de estudos realizados em semântica denotacional de cálculos formais. Segundo Girard [GIR 89], a idéia fundamental de semântica denotacional é permitir interpretar redução

de expressões (uma noção dinâmica) por igualdade de valores (uma noção estática). Explicando de outra forma, são modelados os invariantes do cálculo.

A semântica denotacional é feita com a utilização da Teoria dos Domínios de Scott [STY 77, SMY 77, PLO 81, STV 94]. A particularidade da idéia de Scott estava em que o tipo de espaço topológico que adotou permitiu compatibilizar a estrutura topológica com a estrutura de ordem dos objetos.

Girard observa, contudo, que esta concepção topológica não se adapta muito bem para a construção de espaços funcionais [GIR 89]: "De que modo converge uma seqüência de funções: puntualmente ou uniformemente?"

A mais comum, mas não universal, resposta para esta questão é utilizar a topologia de abertos-compactos, na qual uma função encontra-se em um dado conjunto aberto se, quando restringida a um conjunto compacto específico, seus valores encontram-se em um conjunto aberto específico. Esta topologia somente é bem comportada quando os espaços são localmente compactos (todo ponto possui uma base de vizinhanças compactas), e mesmo assim o espaço funcional computacional não necessita ser localmente compacto.

Para resolver estes problemas, Scott foi levado a impor restrições drásticas a seus espaços topológicos, que foram afastados do espírito geométrico tradicional da topologia. De fato, seus espaços são realmente apenas conjuntos parcialmente ordenados com supremos dirigidos [JUN 89], onde, segundo Girard, a topologia é uma característica casual.

Salienta-se, entretanto, que existe uma visão lógica da topologia, que foi estabelecida em um contexto científico computacional. Veja [ABR 87, SMY 83a, VIC 87].

Por outro lado, a teoria dos Espaços Coerentes [GIR 86, GIR 87, GIR 89, LAF 88, TRO 92, SEL 96, DIM 96, DIM 96a, DIM 98, DIM 2000, DIM 99] também chamados de domínios de Girard, considera admissíveis no conjunto de funções somente funções que, além de contínuas no sentido de Scott (preservam supremos de conjuntos dirigidos), também preservam conjunções (interseções, ínfimos) limitadas superiormente (*pullbacks*), propriedade denominada de estabilidade, introduzida originalmente por Berry [BER 78], com o objetivo de fornecer uma caracterização semântica para algoritmos seqüenciais.

Em [GIR 96] é apresentada uma semântica denotacional baseada em espaços de Banach, com a idéia de dar uma versão contínua de espaços coerentes.

Observe que a noção principal nos modelos da lógica linear é a de função linear, que são fechadas para elementos primos completos. Entretanto, existem muitas estruturas que, embora adequadas para a lógica linear, não determinam domínios algébricos primos, e, além disso, domínios de Scott em geral não são algébricos primos.

Zhang [ZHA 96] procurou generalizar a noção de função linear, introduzindo a noção (mais fraca) de função quasi-linear, dando origem à classe dos domínios algébricos quasi-primos. Em [ZHA 89, ZHA 96] apresenta-se um estudo da categoria monoidal simétrica destes domínios, cujos morfismos são as funções quasi-lineares, representando-os como sistemas de informação irreduzíveis.

Mostra-se também que considerando as funções contínuas segundo Scott, estes domínios formam uma categoria cartesiana fechada. Veja [HUT 95] para um estudo sobre a categoria PRIME dos domínios algébricos primos.

Neste trabalho, os espaços coerentes constituem a estrutura semântica fundamental e por conseqüência, tem-se a noção de que a linguagem baseada em "tipos" deverá interpretar as propriedades observáveis das estruturas computacionais modeladas pelo espaço coerente de processos.

Com base na compatibilidade binária de estruturas atômicas de informação, a noção mais intuitiva deste trabalho está na definição da relação de coerência, que define o grafo sobre o qual se constrói este domínio semântico. Sobre o conjunto de pontos compatíveis de tais grafos, a coerência estrita interpreta a condição implícita para modelar o paralelismo - a concorrência entre posições de memória. Na construção dual, justificada pela

presença da negação involutiva no grafo complementar, a incoerência interpreta a condição para o não-determinismo - o conflito de acesso à memória.

2.5.6 Isomorfismo de Curry-Howard

O paradigma seguido por Girard para aplicação da lógica linear na computação é o isomorfismo de Curry-Howard [GIR 89], no qual as proposições ou fórmulas lógicas são interpretadas por tipos (formulas-as-types), as provas como programas (proof-as-processes) e o processo de normalização das provas como as computações dos programas. Este paradigma constitui-se na fundamentação para todas pesquisas que se desenvolvem interligando a lógica intuicionística, a programação funcional e a teoria das categorias.

O relacionamento entre o λ -cálculo tipado e as provas em analogia à correspondência entre a redução de termos à forma normal e a normalização das provas, constituem-se em relevantes exemplos. Ou seja, constituem-se em uma aplicação formal da lógica intuicionística na busca de um nível de abstração adequado para especificação de linguagens baseadas na programação funcional. Neste sentido salienta-se o estudo de Abramsky [ABR 90] no sentido de prover uma interpretação computacional concreta da lógica linear a partir do isomorfismo citado.

No caso da lógica linear intuicionística tal estudo levou a um refinamento do λ -cálculo, possibilitando um controle mais refinado sobre a ordem de avaliação mantendo o conteúdo lógico dos programas através de suas provas e a normalização através das computações. Os procedimentos computacionais de tal semântica operacional são de natureza eminentemente paralela e produzem uma dicotomia entre a avaliação *elazy* e a avaliação *eager*.

Neste estudo foi introduzida a idéia de processos representando provas de sequentes. Entretanto, como Abramsky admite apenas sequentes de um único lado observa-se uma restrição na aplicação do isomorfismo de Curry-Howard, ou seja, a noção de provas como processos se reduz apenas à noção de fórmulas como tipos.

Isto é estabelecido baseado na correspondência de que cada prova é a prova de um sequente cujo componente é unitário. Nesta perspectiva sequentes com mais de um componente (no caso, multi-conjuntos de fórmulas) são associados a tipos de processos, pois representam as provas tendo como objetos o tipo definido por sequentes unitários. Uma fórmula constitui-se na interpretação de um tipo de porta de interface (entrada ou saída) para um processo e um multiconjunto de fórmulas é considerada como um tipo de processo, especificado pelos tipos de suas portas de interface.

Esta restrição foi muito bem observada por Costa e Moreira [COS 94] onde uma conexão entre lógica e sistemas dinâmicos é apresentada baseada na interpretação computacional do cálculo de sequentes de dois lados. Nesta abordagem, o modelo de processo é o modelo tradicional de processos sequenciais paralelos, derivado do paradigma de programação procedural. Neste modelo a distinção entre processos (sequentes) e tipos de recursos (fórmulas) sobre os quais estes operam é fortemente enfatizada.

No cálculo de sequentes de um lado, uma fórmula e sua negação são conectadas pelo operador identidade de transformação de processos. Neste sentido, não é atribuído um tipo específico à operação de negação. Ou seja, uma fórmula e sua negação são denotacionalmente do mesmo tipo. Entretanto representam tipos diferentes de portas de interface de processos, no caso entrada e saída. Portanto representam tipos distintos de processos mas interpretam o mesmo tipo de recurso. Isto pode ser compreendido considerando-se que o operador negação não indica um novo recurso (tipo diferente), mas diferentes direções dos recursos no processo.

Considerando-se a estrutura que está sendo proposta, onde cada espaço coerente e sua construção dual, resultante da aplicação do operador negação, não podem ser conectados pela transformação identidade na categoria $\mathcal{CospLin}$, ou melhor, até o momento

não foi observada tal situação, considera-se que a interpretação resultante seguirá, sem restrições, o paradigma associado ao isomorfismo de Curry-Howard.

Acrescenta-se por fim o estudo proposto em [ALB 97] onde um novo sistema de dedução natural para a lógica linear é introduzido relacionado com o trabalho de Troelstra [TRO 92]. Neste caso, mantém-se as regras de introdução mas permite-se somente uma regra de eliminação, definida pela troca de uma conclusão por uma hipótese na aplicação da negação linear. Este trabalho faz uma correspondência interessante com a semântica operacional apresentada em [ABR 90] mas ambos são aplicações da lógica linear na modelagem de processos computacionais.

3 Estados e Processos Computacionais no Modelo de Máquina Geométrica

Este capítulo é organizado, principalmente, com o objetivo de ressaltar as principais características do modelo computacional apresentado neste trabalho.

Inspirada na teoria dos sistemas dinâmicos, a idéia intuitiva deste trabalho é a modelagem de uma máquina por uma estrutura matemática, onde os conceitos básicos como memória e processadores são construídos a partir de um espaço geométrico - um modelo de **máquina geométrica**. Neste trabalho, estamos interessados apenas na disposição dos pontos deste espaço geométrico, sem analisar suas propriedades métricas. Mais precisamente, o conjunto S de estados e o conjunto \mathcal{P} de processos são, respectivamente, definidos pelos conjuntos de valores e de ações, ambos rotulados por elementos do conjunto I , representando posições num espaço geométrico.

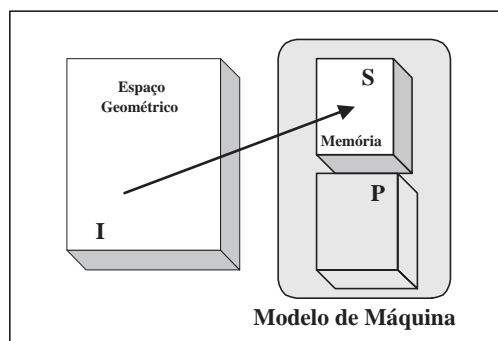


FIGURA 3.1 - Modelo MG com Memória Infinita.

Conforme Figura 3.1, os pontos do espaço geométrico podem ser entendidos como posições de memória e portanto a modelagem contempla interpretação para estados sem restrição quanto ao seu tamanho - um modelo para **memória infinita**. Baseado nisto, a noção de estado computacional é apresentada na primeira seção deste capítulo.

Por outro lado, a noção de processo computacional está relacionada com a noção de transformações entre estados de computação. Compatível com a idéia de que a memória não é necessariamente limitada, a máquina modela processadores sem restrição quanto ao seu tamanho - um modelo para **processadores infinitos**.

Quando, a partir de um estado inicial cada transformação alcança um estado final, o tempo de execução associado a tal transformação é indicado por um número finito de unidades de tempo computacional (*utc*). O modelo de máquina que se está propondo contempla interpretação tanto para estas transformações como para aquelas que não possuem restrição quanto ao seu tempo de execução - um modelo para **interpretações infinitas**.

A interpretação da computação executada por um processo infinito é representada por uma coleção de funções, sendo que cada função desta coleção interpreta o comportamento deste processo em um determinado tempo computacional e constitui-se numa aproximação ou representação parcial. Entretanto, o estado final só é alcançado em tempo infinito, formalmente definido como o limite de suas computações parciais, caracterizando a completção do modelo de máquina.

Um processo computacional elementar é visto como um conjunto (enumerável) de funções que, após execução, altera apenas uma posição de cada estado de computação. Todo processo que, após ou durante execução, altera mais de uma posição em cada es-

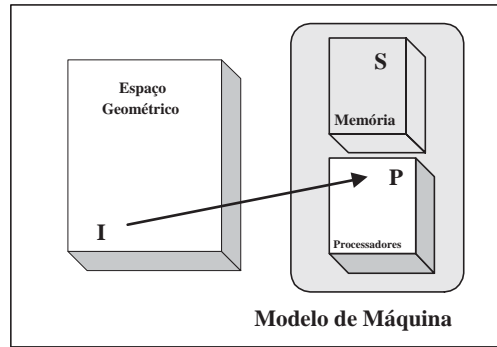


FIGURA 3.2 - Modelo MG com Processadores Infinitos.

tado computacional pode ser construído por sucessivas aplicações de construtores sobre processos elementares - um modelo de **construção indutiva**.

Baseada nas idéias apresentadas nos parágrafos anteriores é construída a estrutura do modelo considerando-se a noção fundamental de processo computacional elementar uni-dimensional. Entretanto, esta noção pode ser estendida ao se considerar processos computacionais elementares bi-dimensionais, como por exemplo uma estrutura matricial. Tem-se então um modelo de máquina para manipulação de **estruturas multi-dimensionais**.

3.1 Estados Computacionais no Modelo MG

No modelo MG (determinístico) a noção de estado de computação, formalmente apresentada logo a seguir, foi estruturada a partir de um conjunto de variáveis distribuídas num espaço geométrico e assumindo valores num conjunto V . Assim, a estas variáveis são atribuídos nomes e valores. O nome de cada variável indica a posição (rótulo em I) que ela se encontra no espaço geométrico. Cada nome é identificado por um subconjunto ordenado de rótulos e portanto a dimensão de cada estado computacional é determinada pelo conjunto I de rótulos.

Quando cada rótulo é indicado por um conjunto unitário tem-se um estado computacional uni-dimensional. Em particular, $I = \omega$ sempre que o conjunto de rótulos é enumerável. Se, além disso I é finito tem-se a estrutura vetorial.

Da mesma forma, quando o nome de cada estado é dado por um par ordenado de rótulos, tem-se um estado computacional bi-dimensional. Em particular, quando é possível modelar o espaço geométrico por um conjunto enumerável, indica-se $I = \{i_{nm}\}_{n,m \in \omega}$, caracterizando a estrutura matricial.

Esta noção de estado computacional é generalizada quando os elementos em I são subconjuntos ordenados e finitos de índices. Neste caso tem-se um estado computacional multi-dimensional, onde os nomes e os valores das variáveis são indicados por subconjuntos finitos de elementos do conjunto enumerável w , onde $I = \{i_X\}_{X \in \wp(\omega)}$.

No desenvolvimento deste trabalho são considerados os estados computacionais uni-dimensionais, mas salienta-se que o modelo proposto pode ser considerado na sua concepção mais genérica, conforme o sugerido acima.

3.1.1 Estados Computacionais Determinísticos

A próxima definição formalmente introduz um estado computacional determinístico. Em seguida, esta definição é estendida de tal forma que o modelo MG possa representar

processos e computações não-determinísticas.

Definição 3.1 *Considere I como um conjunto de rótulos e $V = Val \cup \{\#\}$, onde Val indica um conjunto de valores computacionais e $\#$ é um valor computacional distinguido. A família de todos os valores rotulados é definida pelo conjunto $S = [I \rightarrow V]$ de todas as funções (totais) de I em V , $S = \{\{v_i\}_{i \in I} \mid s : I \rightarrow V \text{ e } s(i) = v_i\}$. Neste caso, um elemento $s \in S$ é denominado um **estado computacional determinístico**.*

Salienta-se que, quando I é uni-dimensional, tem-se $s \in S$ é um estado computacional uni-dimensional e determinístico. Além disso, $null = \{s(i) = v_i = \#\}$ representa o estado computacional que não possui valores em Val , denominado estado computacional uni-dimensional, determinístico e indefinido.

Ao longo deste trabalho, considera-se $I = \omega$ e $S = [I \rightarrow V]$ é definida por

$$S = \{\{v_n\}_{n \in \omega} \mid s : \omega \rightarrow V \text{ e } s(n) = v_n\}.$$

Neste caso, um estado computação uni-dimensional, determinístico e enumerável é um conjunto infinito, enumerável, de valores em V , sendo indicado por

$$s = \{v_0, v_1, \dots, v_n, \dots\} = \{v_n\}_{n \in \omega} = \{s(n)\}_{n \in \omega} \in S.$$

3.1.2 Estados Computacionais

A interpretação e representação dos processos computacionais não-determinísticos no modelo MG é compatível com a idéia de conflito de acesso à posições comuns de memória que ocorre entre processos mutuamente exclusivos e será também concebida como uma transformação de estados, no caso não-determinísticos.

Um estado computacional é definido como um conjunto de estados computacionais determinísticos $s \in [S \rightarrow S]$.

Definição 3.2 *Considere o conjunto S de todos os estados computacionais determinístico conforme Definição 3.1 e seu correspondente conjunto das partes, $\wp(S)$. Um **estado computacional** (determinístico ou não) é definido pela expressão*

$$\mathbf{s} = \{s \mid s \in S = [I \rightarrow V]\} \in \wp(S),$$

significando que \mathbf{s} denota um conjunto de funções $s : I \rightarrow V$, onde o grafo de cada função s consiste num conjunto de valores rotulados pelas infinitas posições de memória. O conjunto de todos os estados computacionais é indicado pela expressão $\wp(S)$ ¹.

Pela última definição, quando \mathbf{s} é unitário tem-se um estado computacional determinístico, indicado por $\mathbf{s} \equiv s$. Além disso, toda função $f \in [S \rightarrow S]$ pode ser identificada por uma função $\mathbf{f} \in [\wp(S) \rightarrow \wp(S)]$.

3.2 Processos Computacionais no Modelo MG

Neste trabalho, apresenta-se um modelo de construção e representação de processos no qual a interpretação de um processo computacional enumerável e uni-dimensional \mathbf{p} baseia-se em questões fundamentais, assim resumidas:

1. A modelagem considera processos computacionais \mathbf{p} posicionados em um espaço geométrico que não possui restrição quanto ao seu tamanho. Além disso, tal posicionamento é determinante para a formalização das relações que satisfazem as condições de concorrência síncrona e de conflito de acesso à memória.

¹A equivalência na notação $S \equiv \wp(S)$ será aplicada neste texto sempre que não houver dúvida.

2. Cada processo computacional \mathbf{p} é construído a partir de processos elementares pela aplicação de construtores que definem sua execução num determinado tempo computacional e modelam a noção de aproximação em relação aos demais processos do conjunto \mathcal{P} de todos os processos, indutivamente definido.
3. O modelo caracteriza cada processo computacional \mathbf{p} como uma transformação de estados de computação. Neste sentido, o estado final obtido após aplicação do processo \mathbf{p} será definido levando-se em conta o estado inicial e as ações rotuladas por posições em I , as quais identificam cada um dos processos elementares envolvidos em sua construção.

Estas questões norteiam a construção da estrutura deste modelo caracterizada como um espaço coerente e definem a linguagem construída a partir deste domínio.

Visando facilitar a compreensão das próximas definições, primeiramente serão introduzidos os processos e os construtores determinísticos. Posteriormente estas definições são estendidas para contemplar modelagem para o não-determinismo.

Com o objetivo de formalizar a modelagem de processos como ações rotuladas por posições num espaço geométrico, define-se a seguir, o conjunto \mathcal{A} das ações computacionais. Por conseguinte, os elementos do subconjunto \mathcal{A}_i das ações computacionais rotuladas por $i \in I$ também serão definidas. Para tal, consideram-se:

1. $\tilde{S} = [I \rightsquigarrow V]$ como o conjunto de todas as funções parciais definidas de I para V , onde $S = [I \rightarrow V] \subseteq \tilde{S} = [I \rightsquigarrow V]$.
2. $\mathcal{A} = [\tilde{S} \rightarrow V]$ como o conjunto de (expressões) ações computacionais, onde cada elemento em \mathcal{A} é uma função que associa uma função parcial $\tilde{s} = \{v_i\}_{i \in X \subseteq I} \in \tilde{S}$ a um valor $v \in V$. Neste caso, $\mathcal{A} = \{\tilde{p} : \tilde{S} \rightarrow V \mid \tilde{p}(\tilde{s}) = \tilde{p}(\{v_i\}_{i \in X \subseteq I}) = v\}$;
3. $\mathcal{A}_I = \bigcup \{\mathcal{A}_i\}_{i \in I}$ como o conjunto de ações rotuladas por posições em I e $\mathcal{A}_i = [\tilde{S} \rightarrow \{v_i\}]$ como o subconjunto de ações computacionais rotuladas pela posição $i \in I$.

Com base nesta notação, quando $I = \omega$ cada rótulo $k \in \omega$ determina uma ação rotulada $\tilde{p}^{(k)} \in \mathcal{A}_k$, tal que $\tilde{p}^{(k)} : \tilde{S} \rightarrow \{v_k\}$, $\tilde{p}^{(k)}(\tilde{s}) = \tilde{p}^{(k)}(\{v_i\}_{i \in X \subseteq \omega}) = v_k$.

3.2.1 Processo Identidade-Default

O processo identidade-default determinístico, indicado por *skip*, constitui-se também num elemento do conjunto \mathcal{P} , neste caso uma função em $[S \rightarrow S]$, cuja ação associada é considerada não observável. A Figura 3.3 mostra sua representação gráfica.

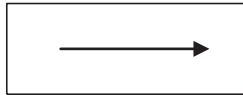


FIGURA 3.3 - Processo *skip*.

Definição 3.3 Seja $pr^{(n)} : S \rightarrow V$ a função projeção dada por

$$pr^{(n)}(s) = pr^{(n)}(\{s(i)\}_{i \in I}) = s(n), \quad \forall \{s(i)\}_{i \in I} \in S.$$

O processo indentidade-default *skip* : $S \rightarrow S$ é definido por $skip = \{pr^{(i)}\}_{i \in I}$. Em particular, se $I = \omega$ tem-se

$$\text{skip} = \{pr^{(n)}\}_{n \in \omega} = \{pr, pr^{(1)}, \dots, pr^{(n-1)}, pr^{(n)}, pr^{(n+1)}, \dots\} \in P.$$

Assim, $\text{skip}(s) = \{pr^{(n)}\}_{n \in \omega}(s) = \{pr^{(n)}(s)\}_{n \in \omega} = \{s(n)\}_{n \in \omega} = \{v_n\}_{n \in \omega} = s, \forall s \in S$.

Considerando-se as Definições 3.2 e 3.3 o processo identidade-default (determinístico ou não) é definido logo a seguir.

Definição 3.4 Considere $\mathbf{s} = \{s \mid s \in \mathbf{s} \text{ e } s : S \rightarrow S\}$. O processo identidade-default $\text{skip} : \wp(S) \rightarrow \wp(S)$ é definido pela expressão $\text{skip}(\mathbf{s}) = \{\text{skip}(s) : S \rightarrow S \mid s \in \mathbf{s}\}^2$.

3.2.2 Processo Computacional Elementar

Um processo elementar caracteriza-se por modificar o valor de apenas uma variável em cada estado computacional que resulta após sua execução. Isto pode ser observado na Figura 3.4. Todo processo que, após execução, altera mais de uma variável em cada estado computacional pode ser construído por sucessivas aplicações de construtores sobre processos elementares. Com ênfase nesta idéia é indutivamente construída a estrutura do modelo MG.

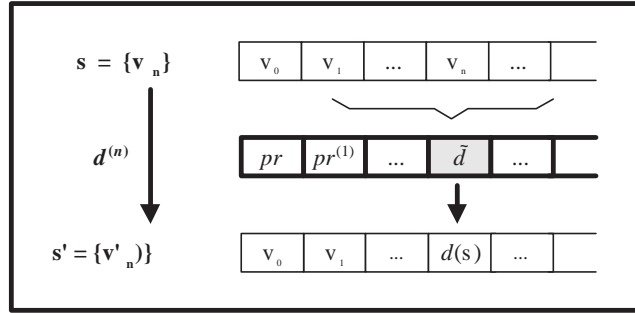


FIGURA 3.4 - Processo Elementar Enumerável Uni-Dimensional $d^{(n)}$.

Definição 3.5 Considere os conjuntos I e S , a função projeção $pr^{(k)} : S \rightarrow S$ apresentadas na Definição 3.3 e a função parcial $\tilde{d} : \tilde{S} \rightarrow V$. O conjunto $\mathcal{D} \subseteq [S \rightarrow S]$ possui como elementos as funções $d^{(n)} : S \rightarrow S$ definidas pela expressão

$$d^{(n)} = \{\tilde{t}^{(k)}\}_{k \in I} \quad \text{onde} \quad \tilde{p}^{(k)}(s) = \begin{cases} pr^{(k)}(s), & \text{se } k \neq n, \\ \tilde{d}(s), & \text{caso contrário.} \end{cases}$$

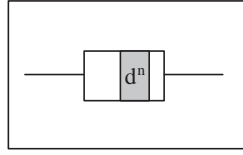
Cada função $d^{(n)} : S \rightarrow S \in \mathcal{D}$ é denominada um **processo computacional elementar determinístico**. Se $I = \omega$ tem-se

$$\mathcal{D} = \{\{pr, pr^{(1)}, \dots, pr^{(n-1)}, \tilde{d}, pr^{(n+1)}, \dots\} \mid pr^{(i)} \in \mathcal{A}_i, \tilde{d} \in \mathcal{A}, i, n \in \omega, n \neq i \text{ e } \tilde{d} \neq pr\}.$$

Neste caso, $d^{(n)} \equiv \{pr, pr^{(1)}, \dots, pr^{(n-1)}, \tilde{d}, pr^{(n+1)}, \dots\}$ é denominado processo elementar enumerável, determinístico e uni-dimensional $d^{(n)}$ e $\mathcal{D}_0 = \{\text{skip}\} \cup \mathcal{D} \subseteq [S \rightarrow S]$.

Em particular, considere $\# \in V$ conforme Definição 3.1. A função construída considerando-se a ação-nula rotulada pelo n -ésimo elemento de I define o processo elementar $null^{(n)} = \{\tilde{p}^{(k)}\}_{k \in I}$, onde

$$\tilde{p}^{(k)}(s) = \begin{cases} pr^{(k)}(s), & \text{se } n \neq k, \\ \#, & \text{caso contrário.} \end{cases}$$

FIGURA 3.5 - Processo Elementar $d^{(n)}$.

Salienta-se que, a execução do processo elementar $d^{(n)}$ resulta num estado computacional de saída igual ao de entrada, exceto na n -ésima posição, que recebe o valor da imagem pela função $\tilde{d} : \tilde{S} \rightarrow V$. A Figure 3.5 mostra a representação do processo elementar $d^{(n)}$.

O processo elementar (determinístico ou não) é definido logo a seguir, generalizando as idéias apresentadas no parágrafo anterior.

Definição 3.6 Considere $\mathbf{s} = \{s \mid s \in \mathbf{s} \text{ e } s : S \rightarrow S\}$ e $d^{(k)} : S \rightarrow S$ de acordo com as Definições 3.2 e 3.5. O **processo elementar** $\mathbf{d}^{(k)} : \wp(S) \rightarrow \wp(S)$ é definido pela expressão

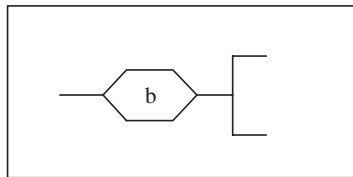
$$\mathbf{d}^{(k)}(\mathbf{s}) = \{d^{(k)}(s) \mid s \in \mathbf{s} \text{ e } d^{(k)} \in [S \rightarrow S]\}.$$

3.2.3 Testes Computacionais

Considere agora o conjunto $B = \{V, F\} \cup \{bool\}$ dos valores booleanos acrescido de um valor distingüido, indicado aqui por *bool*, interpretando o elemento botton na estrutura ordenada onde estão representados os valores booleanos extendidos.

Definição 3.7 Considere os conjuntos B dos valores Booleanos e S dos estados computacionais. Cada função $b : S \rightarrow B$ define um **teste computacional determinístico**. $\mathcal{B} \equiv [S \rightarrow B]$ indica o conjunto de todos os testes computacionais determinísticos.

Quando $I = \omega$ tem-se que S denota o conjunto de estados computacionais, uni-dimensionais e determinísticos e \mathcal{B} indica o conjunto de todos os testes computacionais enumeráveis, uni-dimensionais e determinísticos.

FIGURA 3.6 - Teste Computacional b .

Considerando-se as Definições 3.2 e 3.7 apresenta-se logo a seguir, um teste computacional (determinístico ou não).

Definição 3.8 Considere $\mathbf{s} = \{s \mid s \in \mathbf{s} \text{ e } s : S \rightarrow S\}$ e $b : S \rightarrow B$. O teste elementar $\mathbf{b} : \wp(S) \rightarrow B$ é definido pela expressão

²A equivalência na notação $skip \equiv \mathbf{skip}$ será aplicada neste texto sempre que não houver dúvida.

$$\mathbf{b}(\mathbf{s}) = \begin{cases} x, & \text{se } \bigcap_{s \in \mathbf{s}} \{b(s)\} = \{x\} \neq \emptyset, \\ \text{bool}, & \text{caso contrário.} \end{cases}$$

Neste trabalho, processos elementares serão indicados pelas letras $d^{(k)}, e^{(l)}, f^{(m)}, g^{(n)}, \dots$ sendo que d, e, f, g, \dots representam funções que correspondem as ações associadas a cada processo e k, l, m, n, \dots indicam os nomes das posições de memória que correspondem aos argumentos destas funções.

3.2.4 Função Posição $\Upsilon_{[S \rightarrow S]}$

Considerando que cada estado computacional $s \in S = [I \rightarrow V]$, conforme Definição 3.1, consiste um conjunto, possivelmente infinito, de valores rotulados por posições de um conjunto I e relacionados com os pontos de um espaço geométrico, defini-se a seguir a função posição $\Upsilon_{[S \rightarrow S]}$ que associa cada função em $[S \rightarrow S]$ a um subconjunto, possivelmente infinito, em $\wp(I)$. Intuitivamente, significa que $\Upsilon_{[S \rightarrow S]}(p)$ indica o conjunto de todas as posições de memória cujos valores são alterados quando o processo p é executado.

Definição 3.9 *Sejam as funções parciais $\tilde{p}^{(k)} \in \mathcal{A}_I$ indicando (expressões) ações computacionais rotuladas por posições em I . Para todo $p \in [S \rightarrow S]$, $p = \{\tilde{p}^{(k)}\}_{k \in I}$, tem-se*

$$\Upsilon_{[S \rightarrow S]} : [S \rightarrow S] \rightarrow \wp(I), \quad \Upsilon_{[S \rightarrow S]}(p) = \bigcup_{k \in I} \Upsilon_{\mathcal{A}_I}(\tilde{p}^{(k)}),$$

sempre que $\Upsilon_{\mathcal{A}_I} : \mathcal{A}_I \rightarrow \wp(I)$ é definida pela expressão

$$\Upsilon_{\mathcal{A}_I}(\tilde{p}^{(k)}) = \{k\}, \text{ se } p \neq pr, \quad \text{e} \quad \Upsilon_{\mathcal{A}_I}(\tilde{p}^{(k)}) = \emptyset, \text{ caso contrário.}$$

Pela última definição tem-se

1. $\Upsilon_{[S \rightarrow S]}(d^{(k)}) = \{k\}$ significando que a execução do processo elementar $d^{(k)}$ altera apenas o valor da k -ésima posição de memória;
2. $\Upsilon_{[S \rightarrow S]}(\text{skip}) = \emptyset$ significando que a execução do processo skip não altera os valores das posições de memória.

Na construção da função posição $\Upsilon_{\wp(S) \rightarrow \wp(S)}$ apresentada logo abaixo, foram consideradas as Definições 3.2 e 3.9 .

Definição 3.10 *Sejam $\mathbf{s} = \{s \mid s \in \mathbf{s} \text{ e } s : S \rightarrow S\}$ e $\wp(S)$ o conjunto de todos os estados computacionais. A função posição $\Upsilon_{\wp(S) \rightarrow \wp(S)} : [\wp(S) \rightarrow \wp(S)] \rightarrow \wp(I)$ é dada pela expressão $\Upsilon_{\wp(S) \rightarrow \wp(S)}(\mathbf{s}) = \bigcup_{s \in \mathbf{s}} \Upsilon_{S \rightarrow S}(s)$.*

3.3 Construtores de Processos

O modelo que se está propondo interpretará as seguintes operações básicas de construção de processos:

1. operador $Id_{\wp(S) \rightarrow \wp(S)}$,
2. operadores produto sequencial, produto paralelo e soma não-determinística,
3. operador soma determinística.

As próximas seções além de fornecerem as características básicas de cada um dos construtores de processos estudados neste trabalho, apresentam também suas representações e descrições formais. A simbologia utilizada para denotar estes operadores é sobrecarregada pois é idêntica, tanto para o caso restrito a computações determinísticas quanto para o caso mais genérico, envolvendo estados computacionais possivelmente não-determinísticos.

Os construtores de processos, como transformações de estados computacionais possivelmente não-determinísticos, serão formalizados logo a seguir. Para tal, considera-se a seguinte equivalência : $\mathbf{p} \in [\wp(S) \rightarrow \wp(S)]$ se, e somente se, existe $p \in [S \rightarrow S]$ tal que

$$\mathbf{p}(\mathbf{s}) = \bigcup_{s \in \mathbf{s}} \mathbf{p}(\{s\}) = \{p(s) \mid p \in [S \rightarrow S] \text{ e } s \in \mathbf{s}\}.$$

3.3.1 Construtor Identidade

Pode-se pensar, inicialmente, em uma transformação de processos particular, na qual o processo de entrada coincide com o processo de saída. Esta transformação é definida pela função $Id_{\wp(S) \rightarrow \wp(S)}$, apresentada a seguir

$$Id_{\wp(S) \rightarrow \wp(S)} : [\wp(S) \rightarrow \wp(S)] \rightarrow [\wp(S) \rightarrow \wp(S)], \quad \mathbf{p} \mapsto Id_{\wp(S) \rightarrow \wp(S)}(\mathbf{p}) = \mathbf{p},$$

ou ainda, $Id_{\wp(S) \rightarrow \wp(S)}(\mathbf{p})(\mathbf{s}) = \{s = Id_{S \rightarrow S}(s) \mid s \in \mathbf{s}\}$.

3.3.2 Produto Seqüencial

O produto seqüencial de processos é interpretado pela composição entre as operações que representam os processos envolvidos, obedecendo portanto a uma ordem de execução. Este construtor não é comutativo.

Suponha primeiramente o caso em que $p, q : S \rightarrow S$ representam processos determinísticos, sendo que p é finito. Então o processo $p \cdot q : S \rightarrow S$ primeiro executa p e, somente após o seu término, inicia a execução do processo q . Neste caso, o tempo de execução do processo resultante depende do tempo de execução dos processos p e q .

A Figura 3.7 representa o produto seqüencial entre os processos elementares determinísticos $d^{(k)}$ e $e^{(l)}$.

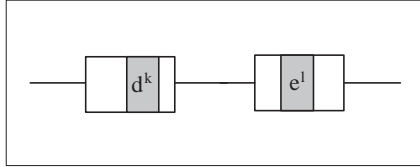


FIGURA 3.7 - Produto Seqüencial $d^{(k)} \cdot e^{(l)}$.

O produto seqüencial de processos, possivelmente não-determinísticos, é formalizado logo a seguir. Para tal, considera-se a seguinte equivalência $\mathbf{p} \in [\wp(S) \rightarrow \wp(S)]$ se, e somente se, existe $p \in [S \rightarrow S]$ tal que

$$\mathbf{p}(\mathbf{s}) = \{p(s) \mid p \in [S \rightarrow S] \text{ e } s \in \mathbf{s}\} = \bigcup_{s \in \mathbf{s}} \mathbf{p}(\{s\}).$$

Definição 3.11 Considere $\wp(S)$ como o conjunto de estados computacionais. O construtor produto seqüencial é definido pela expressão

$$\cdot : [\wp(S) \rightarrow \wp(S)]^2 \rightarrow [\wp(S) \rightarrow \wp(S)], \quad (\mathbf{p}, \mathbf{q}) \mapsto \mathbf{p} \cdot \mathbf{q}, \quad \text{onde}$$

$$(\mathbf{p} \cdot \mathbf{q})(\mathbf{s}) = \bigcup_{s \in \mathbf{s}} (\mathbf{q} \circ \mathbf{p})(\{s\}).$$

De acordo com a Definição 3.11 e sabendo que $(\mathbf{p} \cdot \mathbf{q})(\{s\}) = \{(p \circ q)(s)\}$, tem-se

$$(\mathbf{p} \cdot \mathbf{q})(\mathbf{s}) = \{(p \cdot q)(s) \mid p, q \in [S \rightarrow S] \text{ e } s \in \mathbf{s}\}.$$

3.3.3 Soma Determinística

A soma determinística entre processos está associada a uma escolha representada por um teste elementar, considerando-se a família de testes elementares \mathcal{B} . Formalmente,

Definição 3.12 Consider $\mathbf{p}, \mathbf{q} \in [\wp(S) \rightarrow \wp(S)]$ e $b \in \mathcal{B}$. O construtor soma determinística é indicado pela expressão

$$+_b : [\wp(S) \rightarrow B] \times [\wp(S) \rightarrow \wp(S)]^2 \rightarrow [\wp(S) \rightarrow \wp(S)], \quad (\mathbf{b}, \mathbf{p}, \mathbf{q}) \mapsto \mathbf{p} +_b \mathbf{q} \text{ onde}$$

$$(\mathbf{p} +_b \mathbf{q})(s) = \begin{cases} \mathbf{p}(s) = \bigcup_{s \in \mathbf{s}} \mathbf{p}(\{s\}), & \text{se } \mathbf{b} \text{ é satisfeito,} \\ \mathbf{q}(s) = \bigcup_{s \in \mathbf{s}} \mathbf{q}(\{s\}), & \text{se } \mathbf{b} \text{ não é satisfeito,} \\ \text{skip,} & \text{caso contrário.} \end{cases}$$

Pela Definição 3.12, a seguinte igualdade é válida: $(\mathbf{p} +_b \mathbf{q})(s) = \{(p +_b q)(s) \mid s \in \mathbf{s}\}$,

$$(p +_b q)(s) = \begin{cases} p(s), & \text{se } b \text{ é satisfeito,} \\ q(s), & \text{se } b \text{ não é satisfeito,} \\ \text{skip,} & \text{caso contrário.} \end{cases}$$

A Figura 3.8 mostra uma representação da soma determinística entre os processos elementares $d^{(k)}$ e $e^{(l)}$, relacionados com o teste $b \in \mathcal{B}$. Neste caso, $d^{(k)} +_b e^{(l)}$ é um processo determinístico que, ou executa $d^{(k)}$, ou executa $e^{(l)}$ (indicando ou exclusivo). Se a verdade relacionada pelo teste b é válida, então o processo $d^{(k)}$ será executado, caso contrário $e^{(l)}$ será executado.

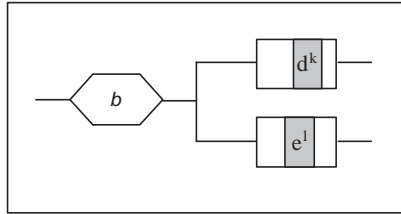


FIGURA 3.8 - Soma Determinística $d^{(k)} +_b e^{(l)}$.

Salienta-se ainda que, o tempo de execução da soma determinística $d^{(k)} +_b e^{(l)}$ será determinado ou pelo tempo de execução do processo $d^{(k)}$ ou pelo tempo do processo $e^{(l)}$, dependendo do valor booleano do teste b . Contudo, neste modelo, o teste b não tem tempo de execução associado, ou ainda, seu tempo de execução é zero. Este construtor não é comutativo.

3.3.4 Produto Paralelo

A construção de um novo processo pela aplicação do produto paralelo pressupõe a concorrência entre os processos. Nesta construção, o novo processo $\parallel -, d^{(k)}, e^{(l)}, - \parallel$ resulta da execução simultânea dos processos elementares concorrentes $d^{(k)}$ e $e^{(l)}$, veja Figura 3.9.

A concorrência está caracterizada pelos índices distintos ($k \neq l$) que definem cada um dos processos elementares $d^{(k)}$ e $e^{(l)}$. Dois processos são concorrentes sempre que a aplicação da função posição $\Upsilon_{[S \rightarrow S]}$, a cada um dos processos, resulta em imagens disjuntas. Ou ainda, a intersecção dos conjuntos que representam as posições de memória afetadas por cada um dos processos concorrentes corresponde ao conjunto vazio.

Formaliza-se a definição primeiramente para o caso determinístico.

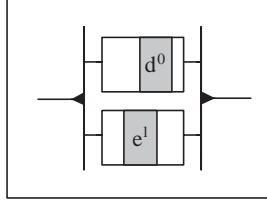


FIGURA 3.9 - Produto Paralelo $\| -, d^{(k)}, e^{(l)}, - \|$.

Definição 3.13 Sejam $p, q \in [S \rightarrow S]$ processos determinísticos e concorrentes, satisfazendo a condição $\Upsilon_{[S \rightarrow S]}(p) \cap \Upsilon_{[S \rightarrow S]}(q) = \emptyset$. O produto paralelo (restrito a estados determinísticos) é indicado pela expressão

$\| \| : [S \rightarrow S]^2 \rightarrow [S \rightarrow S]$, $(p, q) \mapsto \| -, p, q, - \|$, onde

$$\| -, p, q, - \| (s) = \{v_i\}_{i \in I}, v_i = \begin{cases} v'_i & \text{se } i \in \Upsilon_{[S \rightarrow S]}(p) \text{ e } p(s) = \{v'_i\}_{i \in I} \\ v''_i & \text{se } i \in \Upsilon_{[S \rightarrow S]}(q) \text{ e } q(s) = \{v''_i\}_{i \in I} \\ s(i) & \text{caso contrário} \end{cases}$$

Quando $p, q \in [S \rightarrow S]$, $\Upsilon_{[S \rightarrow S]}(p) \cap \Upsilon_{[S \rightarrow S]}(q) \neq \emptyset$, o produto paralelo determinístico está indefinido, ou ainda, $\| -, p, q, - \| = \text{skip}$.

No caso mais genérico,

Definição 3.14 Considere $\wp(S)$ indica o conjunto de estados computacionais, $\mathbf{p}, \mathbf{q} \in [\wp(S) \rightarrow \wp(S)]$ tais que $\Upsilon_{[\wp(S) \rightarrow \wp(S)]}(\mathbf{p}) \cap \Upsilon_{[\wp(S) \rightarrow \wp(S)]}(\mathbf{q}) = \emptyset$. Considerando-se a Definição 3.13, o construtor produto paralelo é definido pela expressão

$\| \| : [\wp(S) \rightarrow \wp(S)]^2 \rightarrow [\wp(S) \rightarrow \wp(S)]$, $(\mathbf{p}, \mathbf{q}) \mapsto \| -, \mathbf{p}, \mathbf{q}, - \|$, onde

$$\| -, \mathbf{p}, \mathbf{q}, - \| (\mathbf{s}) = \bigcup_{s \in \mathbf{s}} \| -, \mathbf{p}, \mathbf{q}, - \| (\{s\}) = \{ \| -, p, q, - \| (s) \mid s \in \mathbf{s}, p, q \in [S \rightarrow S] \}.$$

Se $\mathbf{p}, \mathbf{q} \in [\wp(S) \rightarrow \wp(S)]$ e $\Upsilon_{[\wp(S) \rightarrow \wp(S)]}(\mathbf{p}) \cap \Upsilon_{[\wp(S) \rightarrow \wp(S)]}(\mathbf{q}) \neq \emptyset$, o produto paralelo está indefinido, ou ainda, $\| -, \mathbf{p}, \mathbf{q}, - \| = \text{skip}$.

Considerando uma máquina com memória infinita e processadores com estrutura compatível, tem-se caracterizado o paralelismo real, aquele definido sobre processos concorrentes, executados simultaneamente - um modelo de máquina com **concorrência síncrona**.

3.3.5 Soma Não-Determinística

Dois processos são mutuamente exclusivos sempre que existir no mínimo uma posição de memória afetada por ambos processos. Neste sentido, a arbitrariedade da escolha constitui-se na principal característica da soma não-determinística de processos.

Formalmente,

Definição 3.15 Sejam $\mathbf{p}, \mathbf{q} \in [\wp(S) \rightarrow \wp(S)]$ e $\Upsilon_{[\wp(S) \rightarrow \wp(S)]}(\mathbf{p}) \cap \Upsilon_{[\wp(S) \rightarrow \wp(S)]}(\mathbf{q}) \neq \emptyset$. Tem-se então

$$| | : [\wp(S) \rightarrow \wp(S)]^2 \rightarrow [\wp(S) \rightarrow \wp(S)], (\mathbf{p}, \mathbf{q}) \mapsto | - | \mathbf{p} | \mathbf{q} | - |$$

indica o construtor de processos soma não-determinística, onde

$$| _ | \mathbf{p} | \mathbf{q} | _ | (s) = \mathbf{p}(s) \cup \mathbf{q}(s).$$

Caso contrário, sempre que $p, q \in [\wp(S) \rightarrow \wp(S)]$, $\Upsilon_{[\wp(S) \rightarrow \wp(S)]}(p) \cap \Upsilon_{[\wp(S) \rightarrow \wp(S)]}(q) = \emptyset$, a soma não-determinística está indefinida, ou seja, $| _ | \mathbf{p} | \mathbf{q} | _ | = \mathbf{skip}$.

Dois processos elementares são mutuamente exclusivos se estão definidos pelo mesmo índice k (rotulados pela mesma posição $k \in I$ do espaço geométrico). Assim, d^k e e^k são processos mutuamente exclusivos pois a execução de ambos muda o valor da variável associada à k -ésima posição de qualquer estado computacional. O tempo de execução associado ao processo resultante da escolha arbitrária $| _ | d^{(k)} | e^{(k)} | _ |$ depende do tempo de execução dos processos $d^{(k)}$ e $e^{(k)}$. A Figura 3.10 representa o processo $| _ | d^{(k)} | e^{(k)} | _ |$ resultante da aplicação do operador soma não-determinística sobre os processos elementares $d^{(k)}$ e $e^{(k)}$.

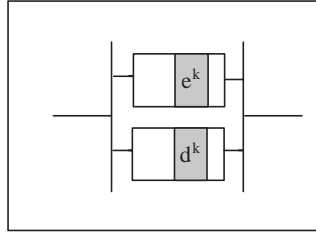


FIGURA 3.10 - Soma Não-Determinística $| _ | d^{(k)} | e^{(k)} | _ |$.

Encerra-se esta seção, ressaltando ainda que a composição infinita dos construtores gera novos processos que também são interpretados por objetos na estrutura do modelo - o espaço coerente de processos \mathbb{D}_∞ , cuja construção começa a ser apresentada no próximo capítulo.

Define-se logo a seguir um processo computacional e se dá uma idéia de como é modelada a noção de transformação de estados.

3.3.6 Construção Indutiva do Conjunto de Processos Computacionais

Definição 3.16 Considere a família $\wp(S)$ de todos os estados de computação. A família de todos os processos computacionais do modelo MG, indicada por \mathcal{P} , é indutivamente definida da seguinte forma

1. **skip** : $\wp(S) \rightarrow \wp(S) \in \mathcal{P}$, ou seja, a função identidade-default apresentada na Definição 3.3 é um processo computacional do modelo MG.
2. **d^(k)** : $\wp(S) \rightarrow \wp(S) \in \mathcal{P}$, ou seja, o processo elementar, formalizadamente apresentado na Definição 3.6, é também um processo computacional no modelo MG.

Nos próximos itens, considera-se $\mathbf{p}, \mathbf{q} : \wp(S) \rightarrow \wp(S)$ processos computacionais no modelo MG e $\mathbf{b} : \wp(S) \rightarrow \mathcal{B}$ um teste computacional no modelo MG.

3. A composição seqüencial é um programa no modelo MG, significando que $\mathbf{p} \cdot \mathbf{q} \in \mathcal{P}$ ou ainda, o produto seqüencial é um processo computacional no modelo MG.
4. A composição condicional, representando a soma determinística, é um programa no modelo MG. Logo $\mathbf{p} +_{\mathbf{b}} \mathbf{q} \in \mathcal{P}$.
5. A composição síncrona, representando o produto paralelo entre processos concorrentes, é um programa no modelo MG. Logo $\| _ | \mathbf{p}, \mathbf{q} | _ \| \in \mathcal{P}$.

6. A composição incondicional, representando a soma não-determinística entre processos conflitantes, é um programa no modelo MG. Logo $\lfloor _ \rfloor \mathbf{p} \mid \mathbf{q} \lfloor _ \rfloor \in \mathcal{P}$.

Quando I é uni-dimensional tem-se que $\mathbf{p} \in \mathcal{P}$ constitui-se num **processo computacional uni-dimensional**. Da mesma forma, se I é uni-dimensional e finito, $\mathbf{p} \in \mathcal{P}$ é denominado um **processo computacional uni-dimensional finito**.

Pela Definição 3.16, um processo é um conjunto subconjuntos de ações-expressões rotuladas por posições de um espaço geométrico. A transformação \mathbf{p} de um estado $\mathbf{s} \in \wp(S)$ é dada pelo conjunto $\mathbf{p}(\mathbf{s})$ de todos os subconjuntos de valores rotulados resultantes da aplicação de transformações $p : S \rightarrow S$ sobre cada estado computacional determinístico $s \in \mathbf{s}$.

Além disso, pela Definição 3.16, tem-se que a composição finita dos construtores de processos é também um processo,

$$\{\odot_{i=0}^n f_i \mid f_i \in \{Id_{\wp(S) \rightarrow \wp(S)}, \cdot, +, \parallel, \|\cdot\|\}\} \subseteq \mathcal{P}.$$

Observação 3.1 A completção da estrutura do modelo MG contempla interpretação para computações infinitas, incluindo a composição infinita dos construtores apresentados nesta seção. Neste sentido, a extensão do conjunto \mathcal{P} dos processos computacionais é induzida nos próximos capítulos pelo domínio \mathbb{D}_∞ , possuindo como elementos, além de todos os processos computacionais apresentados na Definição 3.16, também aqueles que possuem um tempo de execução infinito.

Observação 3.2 Finalmente, a última observação deste capítulo enfatiza que a aplicação de construtores distintos, sobre um mesmo estado computacional inicial, pode alcançar o mesmo estado final. Este é o caso dos processos construídos pela composição finita e constante sobre os construtores produto paralelo e produto seqüencial, representados logo a seguir, quando $I = \omega$:

1. $\mathbf{p} = \parallel _, d^{(0)}, d^{(1)}, \dots, d^{(n)}, _ \parallel e$
2. $\mathbf{q} = d^{(0)} \cdot d^{(1)} \cdot \dots \cdot d^{(n)}$.

Tais processos diferem no seu tempo de execução, 1 utc e n utc, respectivamente.

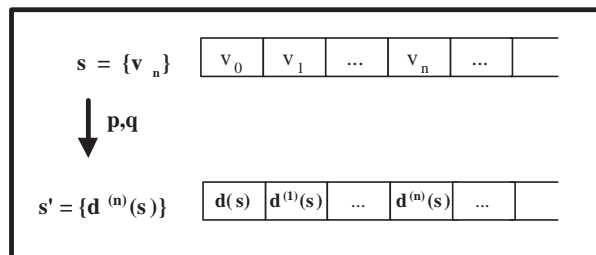


FIGURA 3.11 - Processos como Transformação de Estados Uni-Dimensionais.

4 Nível $\mathbb{D}_0 - \mathbb{D}_1$ da Construção do Espaço Coerente de Processos Gerados Indutivamente

A estrutura ordenada que modela a construção do conjunto \mathcal{P} de processos apresentado no capítulo 3 será indutivamente construída seguindo a metodologia sugerida por Scott [SCO 71].

Seguindo esta abordagem, o domínio semântico base do processo construtivo da estrutura deste modelo é o espaço coerente \mathbb{D}_0 , construído a partir do relacionamento baseado na coerência binária entre unidades atômicas de informações da teia \mathbb{D}_0 - grafo não-dirigido cujos tokens são os processos elementares rotulados por elementos $i_n \in I$.

Neste sentido, cada token da teia \mathbb{D}_0 é uma estrutura matemática formalizada na Definição 3.5 da seção 3.2 e denominada de processo computacional elementar. Portanto, o correspondente domínio \mathbb{D}_0 é denominado o espaço coerente dos processos elementares, onde cada subconjunto coerente, não-vazio, interpreta um processo elementar em $\mathcal{D} \subseteq \mathcal{P}$.

Tomando-se como base o espaço coerente \mathbb{D}_0 dos processos elementares, define-se o espaço coerente \mathbb{D}_1 , no qual são modelados todos os processos em \mathcal{P} executados em no máximo duas unidades de tempo computacional.

A metodologia de construção do primeiro nível $\mathbb{D}_0 - \mathbb{D}_1$ está esquematizada na Figura 4.1. Os espaços coerentes envolvidos nesta primeira etapa da construção serão definidos neste capítulo.

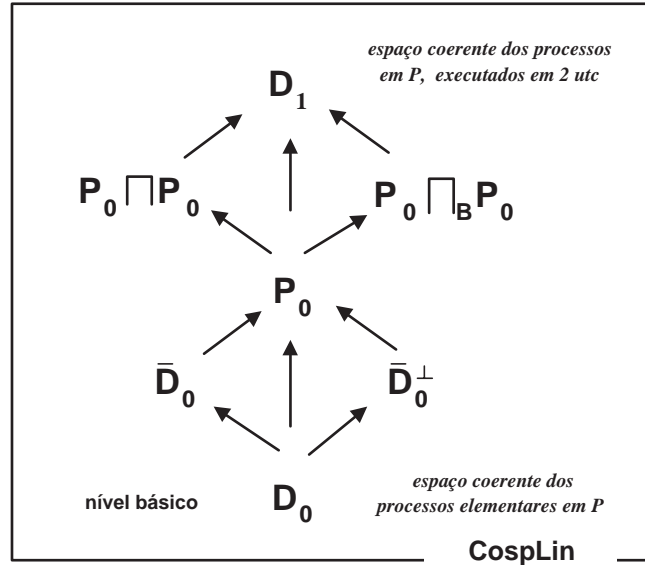


FIGURA 4.1 - Metodologia de Construção de $\mathbb{D}_0 - \mathbb{D}_1$.

4.1 Espaço Coerente \mathbb{D}_0 dos Processos Elementares.

Definição 4.1 *Seja $\mathbb{D}_0 \equiv (\mathcal{D}_0, \cong)$ a teia discreta definida pela coleção de todos os processos elementares em \mathcal{D}_0 considerando a igualdade como a relação de coerência associada. Neste caso,*

$$d^{(k)} \cong e^{(l)} \Leftrightarrow d = e \text{ e } k = l \Leftrightarrow d^{(k)} = e^{(l)}.$$

$\mathbb{D}_0 \equiv (Coh(\mathbf{D}_0), \subseteq)$ indica o **espaço coerente dos processos elementares**. Tem-se que $Coh(\mathbf{D}_0) = Coh(\mathcal{D}_0, \cong) = \{\emptyset\} \cup \{\{d^{(k)}\} \mid d^{(k)} \in \mathcal{D}_0\}$.

O processo identidade-default indicado por $skip \in \mathcal{D}_0$ é modelado pelo conjunto coerente vazio, o qual constitui-se no elemento minimal do domínio \mathbb{D}_0 , $\emptyset \subseteq x, \forall x \in \mathbb{D}_0$.

Cada conjunto coerente $\{d^{(k)}\} \in \mathbb{D}_0$, determinado pelo token $d^{(k)}$ na teia \mathbf{D}_0 , modela o processo elementar $d^{(k)} \in \mathcal{D}_0$.

Em particular tem-se que $\{null^{(k)}\}$ modela o processo $null^{(k)}$ apresentado na seção 3.2.2 do capítulo anterior.

Na Figura 4.2 pode-se observar alguns objetos do espaço coerente dos processos elementares.¹

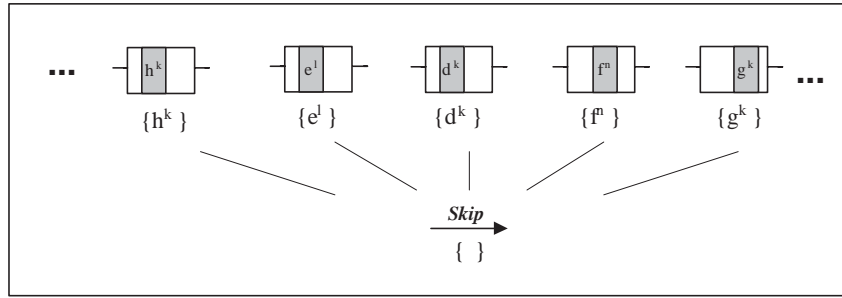


FIGURA 4.2 - O Espaço Coerente \mathbb{D}_0 dos Processos Elementares.

4.1.1 Função Posição em \mathbb{D}_0

A partir da definição da função posição que associa cada token da teia \mathbf{D}_0 a um subconjunto em $\wp(I)$ - o conjunto de todos os subconjuntos de indexadores de variáveis que definem os estados de computação, pode-se formalizar a noção de concorrência entre processos elementares.

Definição 4.2 Considere $\mathbb{D}_0 = (Coh(\mathbf{D}_0), \subseteq)$. A função posição $\Upsilon_{\mathbb{D}_0} : \mathbb{D}_0 \rightarrow \wp(I)$ é definida pela expressão

$$\Upsilon_{\mathbb{D}_0}(x) = \{k \mid d^{(k)} \in x\} = \bigcup v_{\mathbf{D}_0}(d^{(k)}), \quad \forall d^{(k)} \in x.$$

Neste caso, $v_{\mathbf{D}_0} : \mathbf{D}_0 \rightarrow \wp(I)$ é a função definida sobre os tokens da teia \mathbf{D}_0 dada por $v_{\mathbf{D}_0}(d^{(k)}) = \{k\}$. Em particular, $\Upsilon_{\mathbb{D}_0}(\emptyset) = \emptyset$.

Nas próximas seções serão consideradas construções definidas sobre o espaço coerente \mathbb{D}_0 dos processos elementares. Tais construções estão diretamente relacionadas com a família $Coh\mathbf{D}_0$ de subconjuntos coerentes de processos elementares. Adota-se então uma notação mais simplificada, $Coh\mathbf{D}_0 = \bar{\mathcal{D}}_0$. Neste caso, $d^{(k)} \in \mathbf{D}_0$ e $\{d^{(k)}\} = \bar{d}^{(k)} \in \bar{\mathcal{D}}_0$.

A unidade básica de informação (token) das teias que modelam as relações de concorrência e de conflito são os subconjuntos coerente do espaço coerente \mathbb{D}_0 . Neste caso, cada token da teia $\bar{\mathbf{D}}_0$ é um subconjunto unitário em $\wp(\mathcal{D}_0)$ e a correspondente união de tokens, possivelmente infinita, determina subconjuntos coerentes de processos elementares de \mathbb{D}_0 , definidos por posições distintas em I . Isto corresponde à formalização da idéia fundamental de um modelo de máquina com memória infinita capaz de executar um produto paralelo com infinitos fatores, em tempo finito (*1utc*).

¹O símbolo $\{ \}$ indica o conjunto vazio.

Neste sentido, primeiramente define-se o espaço coerente que interpreta os processos concorrentes. A partir de sua estrutura dual obtém-se a interpretação para o não-determinismo na soma de processos. Posteriormente o produto direto entre espaços coerentes formaliza a construção do espaço coerente que interpreta o produto seqüencial de processos. Finalmente, define-se o espaço coerente Booleano para a construção do espaço coerente que interpreta a soma determinística.

No primeiro nível da construção, todas as construções definidas nas próximas seções se restringem à modelagem de processos cuja execução se efetiva em no máximo *1utc*.

4.2 Espaço Coerente $\bar{\mathbb{D}}_0$ do PP de Processos Elementares

Nesta seção, o espaço coerente $\bar{\mathbb{D}}_0$ do produto paralelo de processos elementares é definido. A união de subconjuntos coerentes da teia $\bar{\mathbf{D}}_0$ interpreta o construtor produto paralelo de processos, restringindo sua aplicação sobre processos elementares.

Definição 4.3 A teia $\bar{\mathbf{D}}_0 \equiv (\bar{\mathcal{D}}_0, \approx)$ é construída considerando-se a família $\bar{\mathcal{D}}_0 \subseteq \wp(\mathcal{D}_0)$ e a relação de coerência dada por

$$\mathbf{a} \approx_{\bar{\mathbf{D}}_0} \mathbf{b} \quad \Leftrightarrow \quad \mathbf{a} = \mathbf{b} \quad \text{ou} \quad \Upsilon(\mathbf{a}) \cap \Upsilon(\mathbf{b}) = \emptyset,$$

sempre que $\mathbf{a}, \mathbf{b} \in \bar{\mathbf{D}}_0$. Assim, se $x \in \wp(\bar{\mathcal{D}}_0)$ (ou $x \in \wp(\wp(\mathcal{D}_0))$) e seus elementos são dois a dois coerentes pela relação $\approx_{\bar{\mathbf{D}}_0}$ então x é um subconjunto coerente de $\bar{\mathbf{D}}_0$. A coleção de tais subconjuntos coerentes, dada por

$$Coh\bar{\mathbf{D}}_0 = \{\emptyset\} \cup \{x \in \wp(\bar{\mathcal{D}}_0) \mid \bar{\mathbf{a}} \approx_{\bar{\mathbf{D}}_0} \bar{\mathbf{b}}, \forall \bar{\mathbf{a}}, \bar{\mathbf{b}} \in x\}$$

e ordenada pela inclusão, define o **espaço coerente do produto paralelo de processos elementares**, indicado por $\bar{\mathbb{D}}_0 \equiv (Coh\bar{\mathbf{D}}_0, \subseteq)$.

Em particular, quando $\mathbf{a}, \mathbf{b} \in \bar{\mathcal{D}}_0$, $\mathbf{a} = d^{(k)}$ e $\mathbf{b} = e^{(l)}$ tem-se

$$\overline{d^{(k)}} \approx_{\bar{\mathbf{D}}_0} \overline{e^{(l)}} \quad \Leftrightarrow \quad d^{(k)} = e^{(l)} \quad \text{ou} \quad l \neq k.$$

Alguns dos subconjuntos em $\bar{\mathbb{D}}_0$ estão representados na Figura 4.3.

Pela Definição 4.3 as seguintes interpretações podem ser feitas :

1. O processo identidade-default representado pela expressão *skip* e modelado pelo conjunto vazio, constitui-se no elemento minimal do domínio $\bar{\mathbb{D}}_0$, $\emptyset \subseteq x$, $\forall x \in \bar{\mathbb{D}}_0$.
2. O processo $\parallel skip \parallel \in \mathcal{D}$ é modelado pelo subconjunto coerente unitário $\{\emptyset\}$ que é coerente com todos os demais subconjuntos coerentes em $\bar{\mathbb{D}}_0$.
3. Cada token $\overline{d^{(k)}}$ da teia $\bar{\mathbf{D}}_0$ determina um subconjunto coerente unitário $\{\overline{d^{(k)}}\}$ que é modelado pelo processo computacional parcialmente definido em \mathcal{D} e representado pela expressão $\parallel -, d^{(k)}, - \parallel$.
4. De forma análoga, cada subconjunto coerente $\{\overline{d^{(k)}}, \overline{d^{(l)}}\} \in Coh(\bar{\mathbf{D}}_0)$ constitui-se num objeto parcial em $\bar{\mathbb{D}}_0$ e modela o processo resultante do produto paralelo entre os processos elementares $d^{(k)}$ e $d^{(l)}$, também indicado por uma expressão parcialmente definida em \mathcal{D} , no caso, $\parallel -, d^{(k)}, d^{(l)}, - \parallel$.

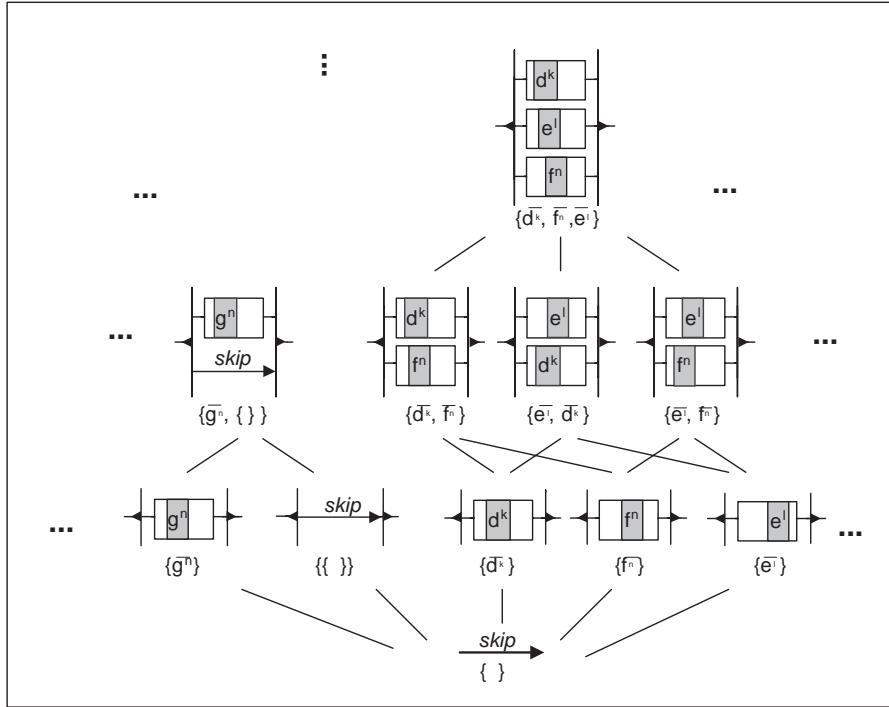


FIGURA 4.3 - O Espaço Coerente $\bar{\mathbb{D}}_0$ do PP de Processos Elementares.

4.2.1 Domínio Plano do PP de Processos Elementares

Quando o espaço coerente \mathbb{D}_0 de processos elementares se reduz ao espaço coerente trivial, o espaço coerente $\bar{\mathbb{D}}_0$ do produto paralelo de processos elementares é um domínio plano.

Neste caso, a coleção de todos os subconjuntos coerentes é dada pela união $Coh(\bar{\mathbb{D}}_0) = \{\emptyset, \{\emptyset\}\}$. Verifica-se também que não existem pares de processos elementares concorrentes em \mathcal{D}_0 , veja a Figura 4.4.

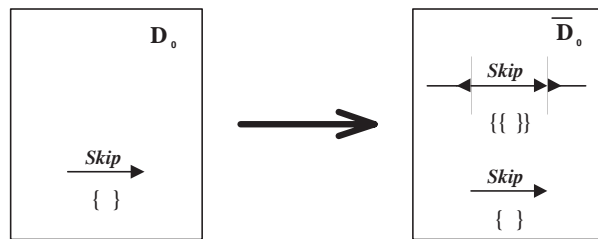


FIGURA 4.4 - O Espaço Coerente Plano $\bar{\mathbb{D}}_0$ do PP de Processos Elementares.

Considere que o conjunto de indexadores de posições é unitário, por exemplo $I = \{k\}$ e $\mathcal{A} = \mathcal{A}_k$. A relação de coerência que constrói a teia $\bar{\mathbb{D}}_0$, é dada por

$$\overline{d^{(k)}} \cong_{\bar{\mathbb{D}}_0} \overline{e^{(k)}} \Leftrightarrow d = e, \quad \text{sendo que} \quad \bar{\emptyset} \cong_{\bar{\mathbb{D}}_0} \overline{d^{(k)}}, \quad \forall \overline{d^{(k)}} \in \bar{\mathbb{D}}_0.$$

Neste caso, a coleção de todos os subconjuntos coerentes em $\bar{\mathbb{D}}_0$ é dada pela conjunto

$$\text{Coh}(\bar{\mathbb{D}}_0) = \{\emptyset\} \cup \{\{\overline{d^{(k)}}\}\} \cup \{\{\overline{d^{(k)}}, \emptyset\}\}, \quad \forall d \in \mathcal{A}.$$

Salienta-se ainda que tal domínio não é plano, $\{\overline{d^{(k)}}, \emptyset\} \in \bar{\mathbb{D}}_0$ é um objeto total e tem-se que $\emptyset \subseteq \{\overline{d^{(k)}}\} \subseteq \{\overline{d^{(k)}}, \emptyset\}$ assim como $\emptyset \subseteq \{\emptyset\} \subseteq \{\overline{d^{(k)}}, \emptyset\}$.

4.2.2 Função Posição em $\bar{\mathbb{D}}_0$

Definição 4.4 *Considere o espaço coerente $\bar{\mathbb{D}}_0 = (\text{Coh}(\bar{\mathbb{D}}_0), \subseteq)$. Sobre a teia $\bar{\mathbb{D}}_0$, a função posição $v_{\bar{\mathbb{D}}_0} : \bar{\mathbb{D}}_0 \rightarrow \wp(\omega)$ é dada por*

$$v_{\bar{\mathbb{D}}_0}(\mathbf{a}) = \{k\} \quad \text{se} \quad \mathbf{a} = \overline{d^{(k)}} \quad \text{e} \quad v_{\bar{\mathbb{D}}_0}(\emptyset) = \emptyset.$$

Então, define-se a função posição $\Upsilon_{\bar{\mathbb{D}}_0} : \bar{\mathbb{D}}_0 \rightarrow \wp(\omega)$ pela expressão $\Upsilon_{\bar{\mathbb{D}}_0}(x) = \bigcup_{a \in x} v_{\bar{\mathbb{D}}_0}(a)$. Em particular, $\Upsilon_{\bar{\mathbb{D}}_0}(\emptyset) = \emptyset$.

Este modelo contempla interpretação para infinitos processos concorrentes, executados num tempo finito, conforme pode ser verificado na próxima exemplificação. A completação deste modelo estende esta interpretação de forma a incluir processos concorrentes infinitos, mas que não possuem um tempo finito de execução.

Exemplificação 4.1

Considere o conjunto \mathcal{A} de ações e o conjunto $I = \{i_n\}_{n \in \omega}$ de rótulos. Define-se a seqüência

$$\{z_n\}_{n \in \omega} \subseteq \text{FinCoh}(\bar{\mathbb{D}}_0) \quad \text{tal que} \quad z_n = \bigcup_{i=0}^n \{\overline{d^{(i)}}\} = \{\overline{d^{(0)}}, \overline{d^{(1)}}, \dots, \overline{d^{(n)}}\}.$$

Neste caso, $z_n \subseteq z_{n+1}$. O supremo desta seqüência de subconjuntos dirigidos é indicado por $z = \bigcup_{n \in I} z_n$. Tem-se então que, $\forall n \in \omega$, $z_n \subseteq z$ e $\Upsilon(z) = I$.

O conjunto coerente $z \in \bar{\mathbb{D}}_0$ constitui-se num objeto total em $\bar{\mathbb{D}}_0$ e cada aproximação finita z_i de z é um elemento compacto de $\bar{\mathbb{D}}_0$.

Este subconjunto coerente permite interpretar o processo que executa simultaneamente, a ação $d \in \mathcal{A}$ em todas as posições $i \in I$ de memória, em no máximo *1utc*.

4.2.3 Relação de Coerência Estrita em $\bar{\mathbb{D}}_0$

Definição 4.5 *Considere a teia $\bar{\mathbb{D}}_0 \equiv (\bar{\mathbb{D}}_0, \approx)$. Os tokens \mathbf{a} e \mathbf{b} da teia $\bar{\mathbb{D}}_0$ são denominados **tokens estritamente coerentes** somente se, possuem imagens distintas pela função posição $\Upsilon_{\bar{\mathbb{D}}_0}$. Assim, sempre que $\mathbf{a}, \mathbf{b} \in \text{Coh}(\bar{\mathbb{D}}_0)$ tem-se*

$$\mathbf{a} \sim_{\bar{\mathbb{D}}_0} \mathbf{b} \Leftrightarrow (\mathbf{a} \approx_{\bar{\mathbb{D}}_0} \mathbf{b} \text{ e } \mathbf{a} \neq \mathbf{b}), \quad \Rightarrow \quad \Upsilon_{\bar{\mathbb{D}}_0}(\mathbf{a}) \cap \Upsilon_{\bar{\mathbb{D}}_0}(\mathbf{b}) = \emptyset.$$

Em particular, se $\mathbf{a} = \overline{d^{(k)}}$, $\mathbf{b} = \overline{e^{(l)}} \in \bar{\mathbb{D}}_0$ então $\overline{d^{(k)}} \sim \overline{e^{(l)}} \Rightarrow k \neq l$.

4.3 Espaço Coerente $\bar{\mathbb{D}}_0^\perp$ da SND de Processos Elementares

O espaço coerente $\bar{\mathbb{D}}_0^\perp$ é definido pela teia complementar $\bar{\mathbb{D}}_0^\perp \equiv (\bar{\mathbb{D}}_0, \approx_\perp)$. A relação de incoerência em $\bar{\mathbb{D}}_0$, tomada como a negação estrita da relação de coerência da Definição 4.5, é identificada como a coerência em $\bar{\mathbb{D}}_0^\perp$.

Definição 4.6 Considere a teia $\bar{\mathbb{D}}_0 \equiv (\bar{\mathbb{D}}_0, \approx_{\bar{\mathbb{D}}_0})$. A **incoerência** (*inconsistência, incompatibilidade*) entre os tokens da teia $\bar{\mathbb{D}}_0$ é dada por

$$\mathbf{a} \not\sim_{\bar{\mathbb{D}}_0} \mathbf{b} \Leftrightarrow \mathbf{a} = \mathbf{b} \text{ e } \Upsilon_{\bar{\mathbb{D}}_0}(\mathbf{a}) \cap \nu_{\bar{\mathbb{D}}_0}(\mathbf{b}) \neq \emptyset,$$

sempre que $\mathbf{a}, \mathbf{b} \in \bar{\mathbb{D}}_0$. Neste caso, \mathbf{a} e \mathbf{b} são tokens incoerentes da teia $\bar{\mathbb{D}}_0$.

Definição 4.7 Seja $\bar{\mathbb{D}}_0$ o espaço coerente do produto paralelo de processos elementares. O espaço coerente dual, denominado **espaço coerente da soma não-determinística de processos elementares** e indicado por

$$\bar{\mathbb{D}}_0^\perp \equiv (\text{Coh}(\bar{\mathbb{D}}_0^\perp), \subseteq) \equiv (\text{Coh}(\bar{\mathbb{D}}_0, \approx_{\bar{\mathbb{D}}_0^\perp}), \subseteq)$$

e representado na Figure 4.5, constitui-se na coleção de todos os subconjuntos coerentes da teia complementar $\bar{\mathbb{D}}_0^\perp$, ordenados pela relação de inclusão. Neste caso, a relação de coerência $\approx_{\bar{\mathbb{D}}_0^\perp}$ é dada por

$$\overline{d^{(k)}} \approx_{\bar{\mathbb{D}}_0^\perp} \overline{e^{(l)}} \Leftrightarrow \overline{d^{(k)}} = \overline{e^{(l)}} \text{ ou } \overline{d^{(k)}} \not\sim \overline{e^{(l)}}, \text{ sempre que } \overline{d^{(k)}}, \overline{e^{(l)}} \in \bar{\mathbb{D}}_0.$$

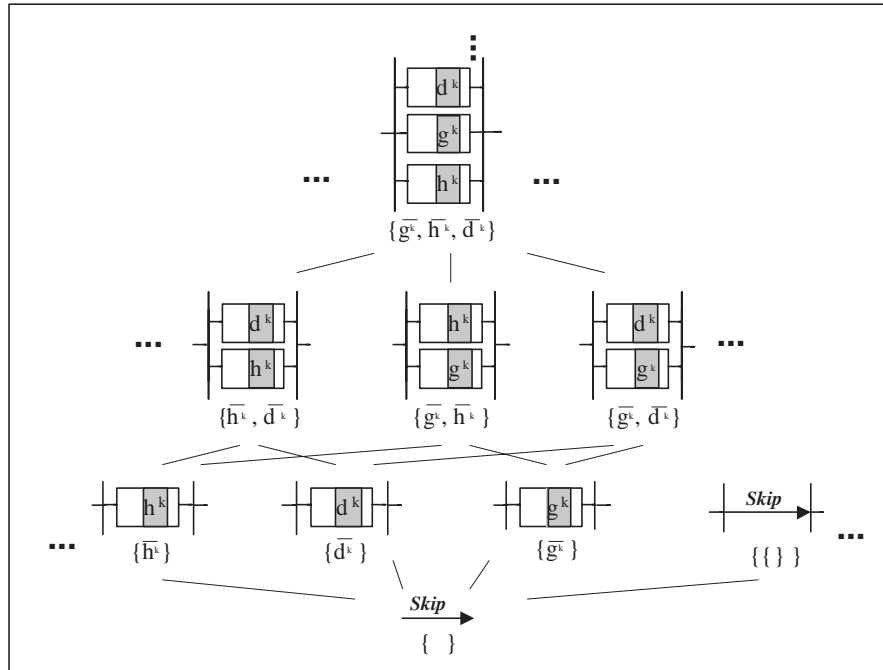


FIGURA 4.5 - O Espaço Coerente $\bar{\mathbb{D}}_0^\perp$ da SND de Processos Elementares.

Observa-se que, pela construção proposta, os tokens da teia $\bar{\mathbf{D}}_0^\perp$ interpretam processos elementares em conflito em \mathcal{D}_0 e a união de subconjuntos coerentes em $\bar{\mathbb{D}}_0^\perp$ interpreta a soma não-determinística de processos. Neste caso, a relação de conflito pode ser estendida sobre um número arbitrário de processos elementares.

A partir das representações gráficas e das correspondentes expressões indicando processos paralelos apresentadas na Figura 4.5, seguem-se as observações:

1. O processo identidade-default representado pela expressão $skip$ é interpretado pelo conjunto vazio - o elemento mínimo do domínio $\bar{\mathbb{D}}_0^\perp$.
2. O processo $| skip | \in \mathcal{D}$ é interpretado pelo subconjunto coerente unitário $\{\emptyset\}$ que não é coerente com nenhum outro subconjunto em $\bar{\mathbb{D}}_0^\perp$, exceto consigo mesmo.
3. Cada token $\overline{d^{(k)}}$ da teia $\bar{\mathbf{D}}_0$ determina um subconjunto coerente unitário $\{\overline{d^{(k)}}\}$ que interpreta o processo computacional representado pela expressão $| -, d^{(k)}, - |$ parcialmente definida em \mathcal{D} .
4. De forma análoga, cada subconjunto coerente $\{\overline{d^{(k)}}, \overline{e^{(k)}}\} \in Coh \bar{\mathbf{D}}_0^\perp$ constitui-se num objeto parcial em $\bar{\mathbb{D}}_0^\perp$ e interpreta o processo resultante da soma não-determinística entre os processos elementares $d^{(k)}$ e $e^{(k)}$, também indicado por uma expressão parcialmente definida em \mathcal{D} , no caso, $| -, d^{(k)} \mid e^{(k)}, - |$.

Observação 4.1

- $\{\emptyset\} \approx_{\bar{\mathbf{D}}_0} \{\overline{d^{(k)}}\}$ e $\{\emptyset\} \not\approx_{\bar{\mathbf{D}}_0^\perp} \{\overline{d^{(k)}}\}$, para todo $\{\overline{d^{(k)}}\}$.
- $\overline{d^{(k)}} \not\approx \overline{e^{(k)}} \Leftrightarrow \{\overline{d^{(k)}}, \overline{e^{(k)}}\} \not\subseteq Coh(\bar{\mathbf{D}}_0) \Leftrightarrow \{\overline{d^{(k)}}, \overline{e^{(k)}}\} \subseteq Coh \bar{\mathbf{D}}_0^\perp$.
- $\wp(\bar{\mathbf{D}}_0) \neq Coh \bar{\mathbf{D}}_0 \cup Coh \bar{\mathbf{D}}_0^\perp$ porque se $x = \{\overline{d^{(k)}}, \overline{d^{(l)}}, \overline{e^{(k)}}\}_{(k \neq l)}$ então $x \not\subseteq Coh \bar{\mathbf{D}}_0$ e $x \not\subseteq Coh \bar{\mathbf{D}}_0^\perp$.
- $\{\emptyset\} \cup \{\{\overline{d^{(k)}}\} \mid \overline{d^{(k)}} \in \bar{\mathbf{D}}_0\} = Coh(\bar{\mathbf{D}}_0) \cap Coh(\bar{\mathbf{D}}_0^\perp)$.
- $(\bar{\mathbb{D}}_0^\perp)^\perp = \bar{\mathbb{D}}_0$, baseada no funtor negação que é involutivo na categoria $CospLin$, [GIR 87].

4.3.1 Domínio Plano da SND de Processos Elementares

Seja $I = \omega$ o conjunto de rótulos de posições de um espaço geométrico e $\tilde{S} = [I \rightsquigarrow V]$ um conjunto unitário, por exemplo $\tilde{S} = \{d\}$. Sendo assim, $\mathcal{A} = \{d^{(n)}\}_{n \in \omega}$.

A teia $\bar{\mathbf{D}}_0$ é construída pela relação de coerência, dada por

$$\overline{d^{(k)}} \cong_{\bar{\mathbf{D}}_0^\perp} \overline{d^{(l)}} \Leftrightarrow k = l \quad \text{e} \quad \bar{\emptyset} \not\cong_{\bar{\mathbf{D}}_0^\perp} \overline{d^{(k)}}, \quad \forall \overline{d^{(k)}} \in \bar{\mathbf{D}}_0^\perp.$$

A coleção de todos os subconjuntos coerentes em $\bar{\mathbb{D}}_0^\perp$ é dada pelo conjunto

$$Coh \bar{\mathbf{D}}_0^\perp = \{\emptyset\} \cup \{\{\overline{d^{(k)}}\}\}_{k \in I}.$$

Nestas condições o espaço coerente $\bar{\mathbb{D}}_0^\perp$ é um domínio plano e cada conjunto coerente $\{\overline{d^{(k)}}\}$ é um objeto total. Assim, não existe par de processos elementares distintos em conflito em \mathcal{D}_0 . Veja Figura 4.6.

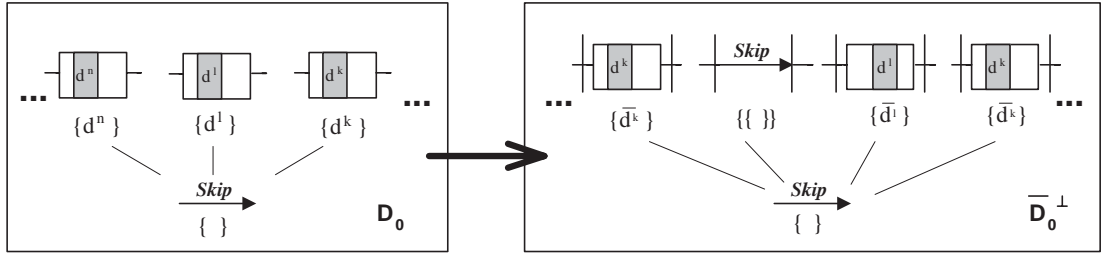


FIGURA 4.6 - O Domínio Plano $\bar{\mathbb{D}}_0^\perp$ da SND de Processos Elementares.

4.3.2 Função Posição em $\bar{\mathbb{D}}_0^\perp$

Definição 4.8 Considere $\bar{\mathbb{D}}_0^\perp = Coh(\bar{\mathbb{D}}_0^\perp, \subseteq)$. Seja $v_{\bar{\mathbb{D}}_0^\perp} : \bar{\mathbb{D}}_0^\perp \rightarrow \wp(\omega)$ dada por $v_{\bar{\mathbb{D}}_0^\perp}(\overline{d^{(k)}}) = \{k\}$ e $v_{\bar{\mathbb{D}}_0^\perp}(\{\emptyset\}) = \emptyset$. A função posição $\Upsilon_{\bar{\mathbb{D}}_0^\perp} : \bar{\mathbb{D}}_0^\perp \rightarrow \wp(\omega)$ é definida pela expressão

$$\Upsilon_{\bar{\mathbb{D}}_0^\perp}(x) = \bigcup_{a \in x} v_{\bar{\mathbb{D}}_0^\perp}(a).$$

Em particular, $\Upsilon_{\bar{\mathbb{D}}_0^\perp}(\emptyset) = \emptyset$.

Pelo modelo que se está definindo, o espaço coerente $\bar{\mathbb{D}}_0^\perp$ contempla interpretações para processos em conflito, finitos ou infinitos na interpretação espacial, desempenhados em *lutc*. A interpretação de tais processos como objetos em $\bar{\mathbb{D}}_0^\perp$ é alcançada pela aplicação da função posição $\Upsilon_{\bar{\mathbb{D}}_0^\perp}$. Veja a próxima exemplificação.

Exemplificação 4.2

Sejam \mathcal{A} um conjunto infinito e enumerável de ações e I um espaço geométrico onde $d_i \in \mathcal{A}$ e $i \in I$. Define-se seqüência

$$\{z_n\}_{n \in \omega} \subseteq FinCoh(\bar{\mathbb{D}}_0) \quad \text{tal que} \quad z_n = \bigcup_{i=0}^n \{\overline{d_i^{(k)}}\} = \{\overline{d_0^{(k)}}, \overline{d_1^{(k)}}, \dots, \overline{d_n^{(k)}}\}.$$

Neste caso, $z_n \subseteq z_{n+1}$. O supremo desta seqüência de subconjuntos dirigidos é indicado por $z = \bigcup_{n \in \omega} z_n$. Tem-se então que $\forall n \in \omega$, $z_n \subseteq z$ e $\Upsilon(z) = \{k\}$. Esse processo executa uma escolha arbitrária entre todas as possíveis ações $d_n \in \mathcal{A}$ na n -ésima posição de memória, em *lutc*, sendo interpretado pelo subconjunto coerente $z \in \bar{\mathbb{D}}_0^\perp$. Neste caso, z constitui-se num objeto total em $\bar{\mathbb{D}}_0^\perp$ e cada subconjunto finito z_n é uma aproximação finita de z , um elemento compacto de $\bar{\mathbb{D}}_0^\perp$.

4.4 Espaço Coerente \mathbb{P}_0

Nesta seção mostra-se a construção do espaço coerente \mathbb{P}_0 que resulta da soma direta dos espaços coerentes \mathbb{D}_0 , $\bar{\mathbb{D}}_0$ e $\bar{\mathbb{D}}_0^\perp$. Os objetos em \mathbb{P}_0 interpretam processos elementares e construções como o produto paralelo e a soma não-determinística aplicados sobre um número arbitrário de processos elementares.

Definição 4.9 A soma direta entre os espaços coerentes \mathbb{D}_0 , $\bar{\mathbb{D}}_0$ e $\bar{\mathbb{D}}_0^\perp$ define o espaço coerente indicado pela expressão

$$\mathbb{P}_0 \equiv \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp \equiv (Coh(\mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp), \subseteq).$$

Neste caso, a teia $\mathbf{P}_0 \equiv \mathbf{D}_0 \amalg \bar{\mathbf{D}}_0 \amalg \bar{\mathbf{D}}_0^\perp \equiv (\mathcal{D}_0 \dot{\cup} \bar{\mathcal{D}}_0 \dot{\cup} \bar{\mathcal{D}}_0, \approx_{\amalg})$ é definida pela união disjunta $\mathcal{D}_0 \dot{\cup} \bar{\mathcal{D}}_0 \dot{\cup} \bar{\mathcal{D}}_0 = \{0\} \times \mathcal{D}_0 \cup \{1\} \times \bar{\mathcal{D}}_0 \cup \{2\} \times \bar{\mathcal{D}}_0$ e pela relação de coerência \approx_{\amalg} dada por

$$(\alpha, a) \approx_{\amalg} (\beta, b) \Leftrightarrow \begin{cases} \alpha = \beta = 0 & \text{e } a \approx_{\mathbf{D}_0} b, \text{ ou} \\ \alpha = \beta = 1 & \text{e } a \approx_{\bar{\mathbf{D}}_0} b, \text{ ou} \\ \alpha = \beta = 2 & \text{e } a \approx_{\bar{\mathbf{D}}_0^\perp} b, \end{cases}$$

sempre que $a, b \in \mathcal{D}_0 \cup \bar{\mathcal{D}}_0$.

Cada token $(\alpha, a) \in \mathbf{D}_0 \amalg \bar{\mathbf{D}}_0 \amalg \bar{\mathbf{D}}_0^\perp$ será identificado por $a_\alpha \equiv (\alpha, a)$.

A Figura 4.7 mostra a construção de alguns objetos do espaço $\mathbb{P}_0 = \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp$.

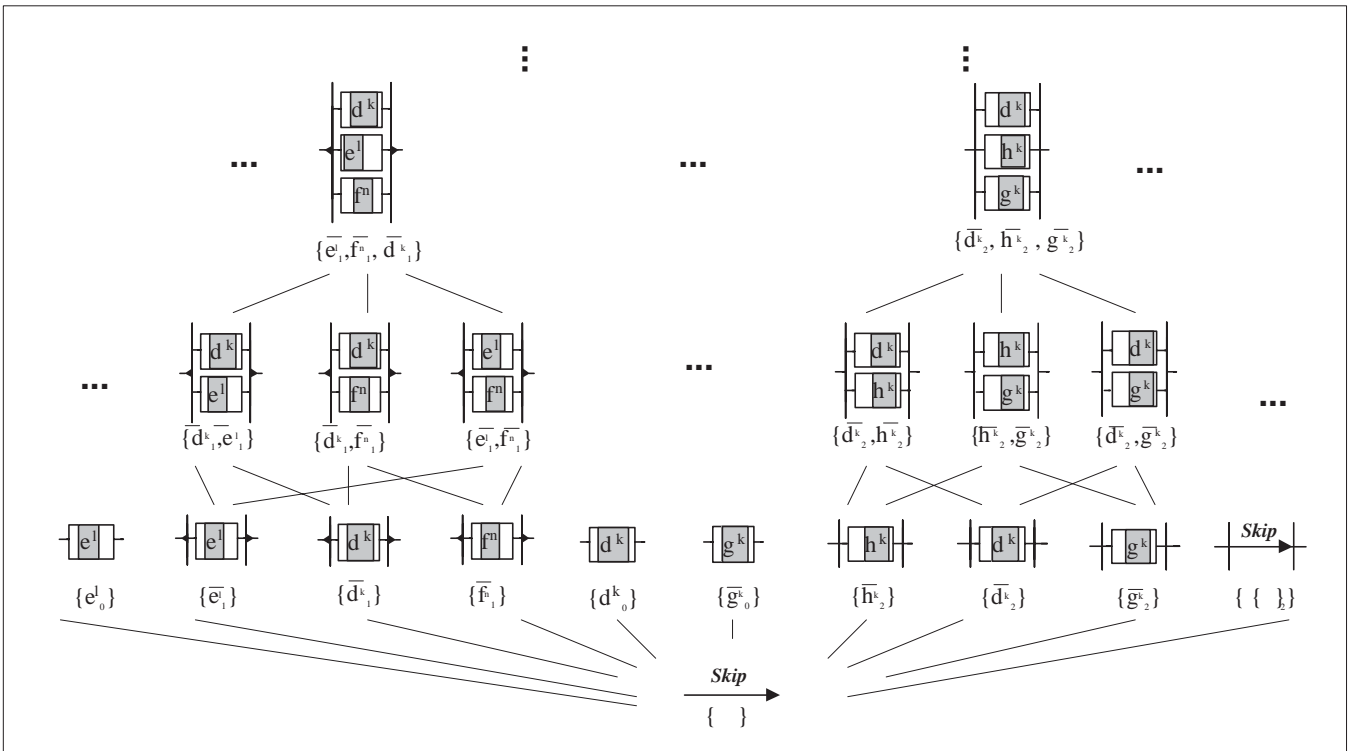


FIGURA 4.7 - O Espaço Coerente $\mathbb{P}_0 = \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp$ de Processos.

Observação 4.2

Considerando-se a indexação de tokens resultante da união disjunta em \mathbb{P}_0 , tem-se as seguintes interpretações, que estão representadas na Figura 4.7:

- $\mathbf{a}_\alpha \in \mathbf{P}_0$ e $\alpha = 0 \rightarrow \mathbf{a} = d^{(k)} \in \mathbf{D}_0$;
- $\mathbf{a}_\alpha \in \mathbf{P}_0$ e $\alpha = 1 \rightarrow \mathbf{a} = \overline{d^{(k)}} \in \bar{\mathbf{D}}_0$;
- $\mathbf{a}_\alpha \in \mathbf{P}_0$ e $\alpha = 2 \rightarrow \mathbf{a} = \overline{d^{(k)}} \in \bar{\mathbf{D}}_0^\perp$.

Observação 4.3

Baseada na associatividade da soma direta de um número finito de subespaços coerentes [GIR 87], a seguinte igualdade é satisfeita:

$$\mathbb{P}_0 \equiv \mathbb{D}_0 \coprod \bar{\mathbb{D}}_0 \coprod \bar{\mathbb{D}}_0^\perp = \mathbb{D}_0 \coprod (\bar{\mathbb{D}}_0 \coprod \bar{\mathbb{D}}_0^\perp) = (\mathbb{D}_0 \coprod \bar{\mathbb{D}}_0) \coprod \bar{\mathbb{D}}_0^\perp.$$

Observação 4.4

Considere agora a ordem de informação em \mathbb{P}_0 . A união indexada de pares de tokens coerentes na teia \mathbf{P}_0 gera os conjuntos coerentes em \mathbb{P}_0 de tal modo que índices distintos interpretam diferentes construtores de processos.

Nas exemplificações abaixo considera-se Ω - o conjunto de todas as expressões válidas na assinatura Σ , conforme Capítulo 9.

- índice 0 \Rightarrow construtor inclusão.

$$\begin{array}{ccc} \mathbb{P}_0 & \emptyset & \subseteq \{d_0^{(k)}\} \rightarrow \underline{\text{objeto total}} \\ & \downarrow & \downarrow \\ \Omega & skip & d^{(k)} \end{array}$$

- índice 1 \Rightarrow produto paralelo.

$$\begin{array}{ccccccc} & & & & & \underline{\text{objeto parcial}} & \\ & & & & & \nearrow & \\ \mathbb{P}_0 & \emptyset & \subseteq & \{\overline{d^{(k)}}_1\} & \subseteq & \{\overline{d^{(k)}}_1, \overline{e^{(l)}}_1\}_{(k \neq l)} & \subseteq & \{\overline{d^{(k)}}_1, \overline{e^{(l)}}_1, \overline{f^{(n)}}_1\}_{(k \neq l, l \neq n, n \neq k)} \cdots \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \Omega & skip & & \parallel -, d^{(k)}, - \parallel & & \parallel -, d^{(k)}, e^{(l)}, - \parallel & & \parallel -, d^{(k)}, e^{(l)}, f^{(n)}, - \parallel \cdots \end{array}$$

- índice 2 \Rightarrow soma não-determinística de processos.

$$\begin{array}{ccccccc} & & & & & \underline{\text{objeto parcial}} & \\ & & & & & \nearrow & \\ \mathbb{P}_0 & \emptyset & \subseteq & \{\overline{d^{(k)}}_2\} & \subseteq & \{\overline{d^{(k)}}_2, \overline{e^{(k)}}_2\}_{(d \neq e)} & \subseteq & \{\overline{d^{(k)}}_2, \overline{e^{(k)}}_2, \overline{f^{(k)}}_2\}_{(d \neq e, e \neq f, f \neq d)} \cdots \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \Omega & skip & & | -, d^{(k)}, - | & & | -, d^{(k)} \mid e^{(k)}, - | & & | -, d^{(k)} \mid e^{(k)} \mid f^{(k)}, - | \cdots \end{array}$$

Um objeto do domínio $\mathbb{D}_0 \coprod \bar{\mathbb{D}}_0 \coprod \bar{\mathbb{D}}_0^\perp$ é um conjunto coerente de tokens indexados por 0, 1 ou 2 que interpreta um processo elementar, um produto paralelo de processos elementares ou uma soma não-determinística de processos elementares, respectivamente. Neste sentido, \mathbb{P}_0 é uma extensão contínua, estável e linear para cada um dos domínios \mathbb{D}_0 , $\bar{\mathbb{D}}_0$ e $\bar{\mathbb{D}}_0^\perp$ e portanto todos os objetos construídos em qualquer um destes domínios podem ser reconstruídos em \mathbb{P}_0 .

As próximas observações se referem aos exemplos apresentados nos últimos itens.

- Os conjuntos coerentes $\{d^{(k)}\} \in \mathbb{D}_0$ e $\{d_0^{(k)}\} \in \mathbb{P}_0$ interpretam o mesmo processo $d^k \in \mathcal{D}$, em diferentes níveis da construção;
- Da mesma forma $\{\overline{d^{(k)}}, \overline{e^{(l)}}\}_{(k \neq l)} \in \bar{\mathbb{D}}_0$ e $\{\overline{d^{(k)}}_1, \overline{e^{(l)}}_1\}_{(k \neq l)} \in \mathbb{P}_0$ interpretam o produto paralelo $\parallel -, d^{(k)}, e^{(l)}, - \parallel$ em \mathcal{D} ;
- E ainda, se $\{\overline{d^{(k)}}, \overline{f^{(k)}}\}_{(d \neq f)} \in \bar{\mathbb{D}}_0^\perp$ interpreta a soma não-determinística $| -, d^{(k)} \mid f^{(k)}, - |$ em \mathcal{D} , então o mesmo é válido para $\{\overline{d^{(k)}}_2, \overline{f^{(k)}}_2\} \in \mathbb{P}_0$;
- Entretanto, em \mathbb{P}_0 não existem conjuntos coerentes cujos tokens possuem indexadores distintos. Isto é, sejam $\alpha, \beta \in \{0, 1, 2\}$ tais que $\{\mathbf{a}_\alpha\} \in \bar{\mathbb{D}}_0^\perp$, $\{\mathbf{b}_\beta\} \in \bar{\mathbb{D}}_0$ e $\alpha \neq \beta$. Tem-se que $\{\mathbf{a}_\alpha, \mathbf{b}_\beta\} \notin \mathbb{P}_0 = \mathbb{D}_0 \coprod \bar{\mathbb{D}}_0 \coprod \bar{\mathbb{D}}_0^\perp$.

Observação 4.5

De acordo com o exposto acima, cada índice em \mathbb{P}_0 representa um tipo de processo, com projeções distintas relativas ao espaço coerente \mathbb{D}_0 . Na Figura 4.8 apresenta-se a projeção $\Phi_0^{(0)} : \mathbb{P}_0 \rightarrow \mathbb{D}_0$, na qual os conjuntos coerentes que interpretam processos não-elementares são projetados para o conjunto coerente vazio, que interpreta o processo indefinido.

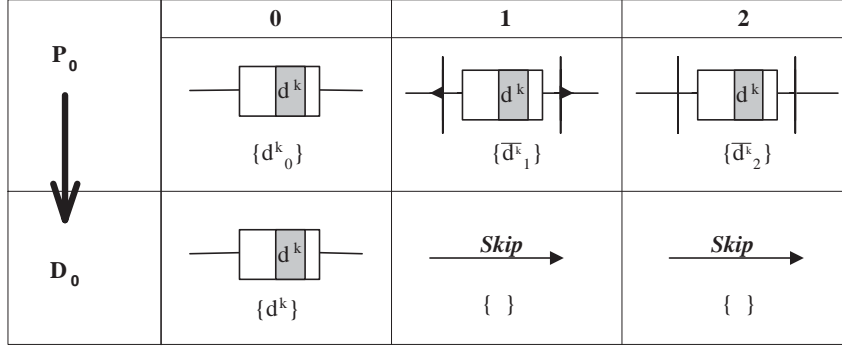


FIGURA 4.8 - A Função Projeção $\Phi_{\mathbb{D}_0}^0 : \mathbb{P}_0 \rightarrow \mathbb{D}_0$.

4.4.1 Imersões e Projeções do Espaço Poerente \mathbb{P}_0

Definição 4.10 *Considere os espaços coerentes \mathbb{P}_0 e \mathbb{D}_0 . As imersões $\phi_0^{(0)} : \mathbb{D}_0 \rightarrow \mathbb{P}_0$, $\phi_0^{(1)} : \bar{\mathbb{D}}_0 \rightarrow \mathbb{P}_0$ e $\phi_0^{(2)} : \bar{\mathbb{D}}_0^\perp \rightarrow \mathbb{P}_0$ são definidas pela expressão $\phi_0^{(\theta)}(x) = \{\theta\} \times x$, sempre que $\theta \in \{0, 1, 2\}$. Em particular, $\phi_0^{(\theta)}(\emptyset) = \emptyset$.*

Salienta-se que $\{\theta\} \times x \equiv \{\mathbf{a}_\theta \mid \mathbf{a} \in x\}$, onde \times indica o produto cartesiano de conjuntos. Neste caso, tem-se as implicações:

1. se $\theta = 0$ e $\mathbf{a}_\theta \in \mathbf{P}_0$, então $\mathbf{a} \in \mathbf{D}_0$;
2. se $\theta \in \{1, 2\}$ e $\mathbf{a}_\theta = \bar{a} \in \mathbf{P}_0$, então $a \in \mathbf{D}_0$ e $\mathbf{a} \in \text{Coh}(\mathbf{D}_0)$.

Proposição 4.1 *A imersão $\phi_0^{(0)} : \mathbb{D}_0 \rightarrow \mathbb{P}_0$ é monótona, contínua, estável e linear.*

Demonstração.

1. Sejam $x, x' \in \mathbb{D}_0$ e suponha $x \subseteq x'$. Então, tem-se que $\{\mathbf{a}_0 \mid \mathbf{a} \in x\} \subseteq \{\mathbf{a}_0 \mid \mathbf{a} \in x'\}$. Logo, $\phi(x)_0^{(0)} \subseteq \phi(x')_0^{(0)}$ e portanto $\phi_0^{(0)}$ é monotônica.
2. Suponha $X \subseteq \mathbb{D}_0$ dirigido com relação à inclusão. Então, pela monotonicidade da função $\phi_0^{(0)}$ acrescido do fato de que \mathbb{P}_0 é fechado para uniões dirigidas, tem-se que $\phi_0^{(0)}[X] \subseteq \mathbb{P}_0$ é também dirigido e $\bigcup \phi_0^{(0)}[X] \in \mathbb{P}_0$. Além disso,

$$\phi_0^{(0)}\left(\bigcup X\right) = \{(0, \alpha) \mid \alpha \in \bigcup X\} = \bigcup \{(0, \alpha) \mid \alpha \in x\}_{x \in X} = \bigcup \phi_0^{(0)}[X].$$

Conclue-se que $\phi_0^{(0)}$ é contínua.

3. Sejam $x, x' \in \mathbb{D}_0$. Sabendo que \mathbb{D}_0 e \mathbb{P}_0 são fechados para a interseção arbitrária, tem-se que $x \cap x' \in \mathbb{D}_0$ e $\phi_0^{(0)}(x) \cap \phi_0^{(0)}(x') \in \mathbb{P}_0$. Além disso,

$$\phi_0^{(0)}(x \cap x') = \{0\} \times x \cap x' = \{0\} \times x \cap \{0\} \times x' = \phi_0^{(0)}(x) \cap \phi_0^{(0)}(x').$$

Portanto $\phi_0^{(0)}$ é uma função estável.

4. Por fim, considere $X \subseteq \mathbb{D}_0$. Supondo que $\phi_0^{(0)}(\bigcup X) \in \mathbb{P}_0$ tem-se que

$$\phi_0^{(0)} \bigcup X = \{0\} \times \bigcup X = \bigcup (\{0\} \times x)_{x \in X} = \bigcup \{\phi_0^{(0)}(x) \mid x \in X\} = \bigcup \phi_0^{(0)}[X],$$

comprovando-se que $\phi_0^{(0)}$ satisfaz a linearidade.

Mostrou-se assim que a função $\phi_0^{(0)}$ é linear. \blacktriangle

Pela proposição acima $\phi_0^{(0)}$ é um morfismo na categoria $\mathcal{CospLin}$. O mesmo pode ser provado para as outras imersões relativas a \mathbb{P}_0 ,

$$\phi_0^{(1)} : \bar{\mathbb{D}}_0 \rightarrow \mathbb{P}_0, \phi_0^{(1)}(x) = \{1\} \times x \quad \text{e} \quad \phi_0^{(2)} : \bar{\mathbb{D}}_0^\perp \rightarrow \mathbb{P}_0, \phi_0^{(2)}(x) = \{2\} \times x.$$

Definição 4.11 Considere os espaços coerentes \mathbb{P}_0 , \mathbb{D}_0 , $\bar{\mathbb{D}}_0$ e $\bar{\mathbb{D}}_0^\perp$. As projeções $\Phi_0^{(0)} : \mathbb{P}_0 \rightarrow \mathbb{D}_0$, $\Phi_0^{(1)} : \bar{\mathbb{P}}_0 \rightarrow \bar{\mathbb{D}}_0$ e $\Phi_0^{(2)} : \bar{\mathbb{P}}_0 \rightarrow \bar{\mathbb{D}}_0^\perp$ são definidas pela expressão

$$\Phi_0^{(\alpha)}(x) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x\}$$

sempre que $\alpha \in \{0, 1, 2\}$. Em particular, $\Phi_0^{(\alpha)}(\emptyset) = \emptyset$ e $\phi_0^{(0)}$ é estrita.

Proposição 4.2 A função projeção $\Phi_0^{(0)}$ é monótona, contínua, estável e linear.

Demonstração.

1. Sejam $x, x' \in \mathbb{P}_0$. Suponha $x \subseteq x'$. Se $x = \emptyset$ tem-se que $\Phi_0^{(0)}(\emptyset) = \emptyset \subseteq \Phi_0^{(0)}(x'), \forall x' \in \mathbb{P}_0$. Caso contrário, se $x \neq \emptyset$ tem-se que $x = x' = \{\mathbf{a}_\alpha\}$ e $\Phi_0^{(0)}(x) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x\} \subseteq \{\mathbf{a}' \mid \mathbf{a}'_\alpha \in x'\} = \Phi_0^{(0)}(x')$. Logo, $\Phi_0^{(0)}$ é monotônica.
2. Suponha agora $X \subseteq \mathbb{P}_0$ dirigido com relação à inclusão. Considerando-se que \mathbb{P}_0 é fechado para uniões dirigidas, e que a função $\Phi_0^{(0)}$ é monótona, $\Phi_0^{(0)}[X] \subseteq \mathbb{D}_0$ é também dirigido e $\bigcup \Phi_0^{(0)}[X] \in \mathbb{D}_0$. Além disso,

$$\Phi_0^{(0)}(\bigcup X) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in \bigcup X\} = \bigcup \{\mathbf{a} \mid \mathbf{a}_\alpha \in x\}_{x \in X} = \bigcup \Phi_0^{(0)}[X].$$

Portanto $\Phi_0^{(0)}$ é contínua.

3. Sejam $x, x' \in \mathbb{P}_0$ tais que $x \cap x' \in \mathbb{P}_0$. Se $x \cap x' = \emptyset$, desde que Φ é estrita a prova da estabilidade é imediata. Suponha então $x \cap x' \neq \emptyset$. Tem-se que

$$\Phi_0^{(0)}(x \cap x') = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x \cap x'\} = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x\} \cap \{\mathbf{a} \mid \mathbf{a}_\alpha \in x'\}.$$

Logo, $\Phi_0^{(0)}(x \cap x') = \Phi_0^{(0)}(x) \cap \Phi_0^{(0)}(x')$ portanto $\Phi_0^{(0)}$ é uma função estável.

4. Considere agora $X \subseteq \mathbb{D}_0$. Supondo que $\Phi_0^{(0)}(\bigcup X) \in \mathbb{P}_0$ tem-se que

$$\Phi_0^{(0)}(\bigcup X) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in \bigcup X\} = \bigcup \{\mathbf{a} \mid \mathbf{a}_\alpha \in x\}_{x \in X} = \bigcup \{\Phi_0^{(0)}[X]\},$$

o que mostra que $\Phi_0^{(0)}$ satisfaz a linearidade. Portanto $\Phi_0^{(0)}$ é uma função linear. \blacktriangle

Pela proposição acima a função projeção $\Phi_0^{(0)}$ é um morfismo na categoria $\mathcal{CospLin}$. O mesmo pode ser provado para as outras projeções (estritas) relativas a \mathbb{P}_0 ,

$$\Phi_0^{(1)} : \mathbb{P}_0 \rightarrow \bar{\mathbb{D}}_0, \quad \Phi_0^{(1)}(x) = \{\mathbf{a} | \mathbf{a}_1 \in x\} \quad \text{e} \quad \Phi_0^{(2)} : \mathbb{P}_0 \rightarrow \bar{\mathbb{D}}_0^\perp, \quad \Phi_0^{(2)}(x) = \{\mathbf{a} | \mathbf{a}_2 \in x\}.$$

Veja Figura 4.8 onde estão representadas as projeções $\Phi_0^{(0)}$ de um subconjunto coerente unitário, no primeiro subnível $\mathbb{D}_0 - \mathbb{P}_0$.

Observação 4.6 1. Considerando-se que \mathbb{D}_0 é plano, $\text{FinCoh}(\mathbf{D}_0) = \text{Coh}(\mathbf{D}_0)$ e portanto pode-se definir imersões de \mathbb{D}_0 para $\bar{\mathbb{D}}_0$ como também de \mathbb{D}_0 para $\bar{\mathbb{D}}_0^\perp$. Entretanto esta igualdade não se verifica nos demais níveis da estrutura deste modelo computacional.

2. Os morfismos apresentados na Figura 4.9 resumem o relacionamento entre os espaços coerentes envolvidos na construção do primeiro subnível ($\mathbb{D}_0 - \mathbb{P}_0$) do primeiro nível ($\mathbb{D}_0 - \mathbb{D}_1$) da estrutura do modelo proposto neste trabalho.

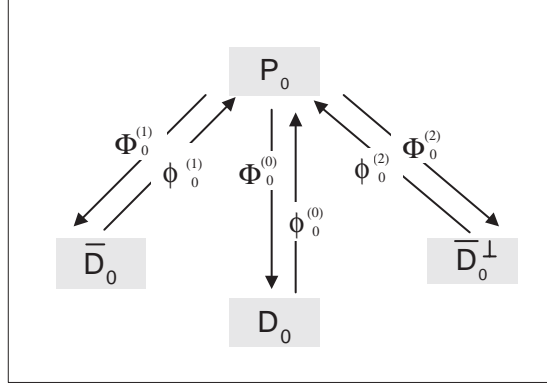


FIGURA 4.9 - As Projeções e Imersões em $\mathbb{D}_0 - \mathbb{P}_0$.

4.4.2 Função Posição em \mathbb{P}_0

Definição 4.12 Considere a teia \mathbf{P}_0 . Seja $v_{\mathbf{P}_0} : \mathbf{P}_0 \rightarrow \wp(\omega)$ dada por

$$v_{\mathbf{P}_0}(\mathbf{a}_\alpha) = \begin{cases} v_{\mathbf{D}_0}(\mathbf{a}) & \text{se } \alpha = 0, \\ v_{\bar{\mathbf{D}}_0}(\mathbf{a}) & \text{se } \alpha = 1, \\ v_{\bar{\mathbf{D}}_0^\perp}(\mathbf{a}) & \text{se } \alpha = 2, \\ \emptyset & \text{contrário;} \end{cases}$$

A função posição $\Upsilon_{\mathbb{P}_0} : \mathbb{P}_0 \rightarrow \wp(\omega)$ é definida por $\Upsilon_{\mathbb{P}_0}(x) = \bigcup_{\mathbf{a}_\alpha \in x} v_{\mathbf{P}_0}(\mathbf{a}_\alpha)$. Em particular, $\Upsilon_{\mathbb{P}_0}(\emptyset) = \emptyset$.

Nas próximas seções deste capítulo são definidos os espaços coerentes envolvidos na construção do segundo subnível do primeiro nível da estrutura do modelo proposto.

4.5 Espaço Coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$

Todos os subconjuntos coerentes resultantes do produto direto $\mathbb{P}_0 \amalg \mathbb{P}_0$ interpretam produtos seqüenciais de processos que são executados, no máximo, em *2 utc*.

Definição 4.13 *Considere o espaço coerente $\mathbb{P}_0 = \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp$. Seja*

$$\mathbf{P}_0 \amalg \mathbf{P}_0 \equiv (\mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp) \amalg (\mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp)$$

a teia definida pela união disjunta $(\mathbb{D}_0 \dot{\cup} \bar{\mathbb{D}}_0 \dot{\cup} \bar{\mathbb{D}}_0^\perp) \dot{\cup} (\mathbb{D}_0 \dot{\cup} \bar{\mathbb{D}}_0 \dot{\cup} \bar{\mathbb{D}}_0^\perp)$ e pela relação de coerência \approx_{\amalg} dada por

$$(\alpha, \mathbf{a}) \approx_{\amalg} (\beta, \mathbf{b}) \Leftrightarrow \begin{cases} \alpha = \beta \text{ e } \mathbf{a} \approx_{\amalg} \mathbf{b}, \text{ ou} \\ \alpha \neq \beta \end{cases}$$

sempre que $\mathbf{a}, \mathbf{b} \in \mathbb{D}_0 \cup \bar{\mathbb{D}}_0 \cup \bar{\mathbb{D}}_0^\perp$. A família de todos os subconjuntos coerentes de tal teia $\mathbf{P}_0 \amalg \mathbf{P}_0$, ordenada pela inclusão, define o espaço coerente

$$\mathbb{P}_0 \amalg \mathbb{P}_0 \equiv (\mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp) \amalg (\mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp)$$

denominado **espaço coerente do produto seqüencial de processos elementares** ou resultantes da aplicação do produto paralelo e da soma não-determinística sobre um número arbitrário de processos elementares.

Para cada token $(\alpha, \mathbf{a}) \in \mathbb{P}_0 \amalg \mathbb{P}_0$, considera-se a identificação $\mathbf{a}_\alpha \equiv (\alpha, \mathbf{a})$.

A Figura 4.10 mostra uma representação do espaço coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$ do produto seqüencial de processos executados, no máximo, em *2 utc*.

Exemplificação 4.3

Além de reconstruir todos os objetos que já pertenciam ao espaço coerente \mathbb{P}_0 , a indexação definida pela união disjunta juntamente com a relação de coerência \approx_{\amalg} dirige a construção de novos subconjuntos de tokens, dois a dois coerentes. Neste sentido, suponha os seguintes objetos em \mathbb{P}_0 :

1. $\{d_0^{(k)}\}, \{e_0^{(l)}\}, \{f_0^{(k)}\}$ representando os seguintes processos $d^{(k)}, e^{(l)}$ e $f^{(k)}$;
2. $\{\bar{d}_1^{(k)}\}, \{\bar{e}_1^{(l)}\}, \{\bar{f}_1^{(n)}\}$ e $\{\bar{d}_1^{(k)}, \bar{e}_1^{(l)}, \bar{f}_1^{(n)}\}_{(k \neq l, l \neq n, n \neq k)}$ cuja interpretação está respectivamente relacionada com os processos $\| -, d^{(k)}, - \|$, $\| -, e^{(l)}, - \|$, $\| -, f^{(n)}, - \|$ e $\| -, d^{(k)}, e^{(l)}, f^{(n)}, - \|$;
3. $\{\bar{d}_2^{(k)}\}, \{\bar{f}_2^{(k)}\}$ e $\{\bar{d}_2^{(k)}, \bar{f}_2^{(k)}\}_{(d \neq f)}$ representam somas não-determinísticas indicadas por $| -, d^{(k)}, - |$, $| -, f^{(k)}, - |$ e $| -, d^{(k)} \uparrow f^{(k)}, - |$, respectivamente.

Então novas interpretações são obtidas a partir destes subconjuntos coerentes em \mathbb{P}_0 considerando-se agora o espaço coerente resultante do produto direto $\mathbb{P}_0 \amalg \mathbb{P}_0$.

- O subconjunto coerente $x = \{d_{00}^{(k)}, e_{01}^{(l)}\}$ é um objeto total em $\mathbb{P}_0 \amalg \mathbb{P}_0$ que dá interpretação para o produto seqüencial entre os processos elementares $d^{(k)}$ e $e^{(l)}$, sendo indicado por $d^{(k)} \cdot e^{(l)}$. Salienta-se ainda que as expressões $d^{(k)} \cdot -$ e $- \cdot e^{(l)}$ são interpretadas pelos subconjuntos coerentes $x' = \{d_{00}^{(k)}\}$ e $x'' = \{e_{01}^{(l)}\}$. Tais subconjuntos constituem-se em aproximações distintas para o subconjunto x .

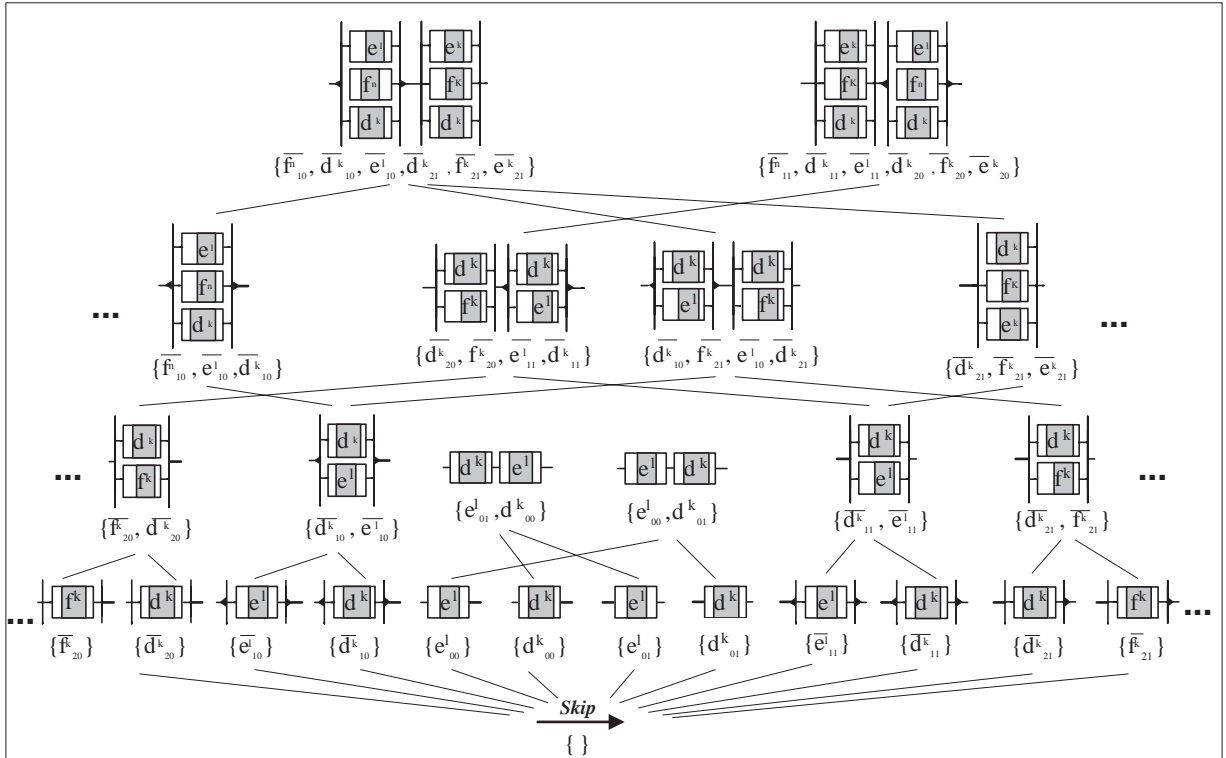


FIGURA 4.10 - O Espaço Coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$ do PS de Processos.

- Da mesma forma, o produto seqüencial $((\| _ , d^{(k)}, e^{(l)}, _ \|) \cdot _)$ que tem como primeiro fator o processo resultante do produto paralelo entre $d^{(k)}$ e $e^{(l)}$ é interpretado pelo subconjunto coerente $y = \{d^{(k)}_{10}, e^{(l)}_{10}\} \in \mathbb{P}_0 \amalg \mathbb{P}_0$. Neste exemplo, este subconjunto y é um objeto parcial do domínio $\mathbb{P}_0 \amalg \mathbb{P}_0$ pois dá uma informação parcial em relação ao processo seqüencial que está sendo interpretado. Ou seja, se conhece somente um dos fatores deste produto.
A união deste subconjunto coerente y com um outro conjunto coerente em $\mathbb{P}_0 \amalg \mathbb{P}_0$, cujos tokens são indexados por $\alpha_1 1, \alpha \in \{0, 1, 2\}$ resulta num subconjunto coerente que dará uma interpretação para um novo processo seqüencial, então com ambos fatores definidos, ou parcialmente definidos.
- Seja $z = \{\overline{d^{(k)}_{21}}, \overline{f^{(k)}_{21}}\}$ um objeto parcial em $\mathbb{P}_0 \amalg \mathbb{P}_0$ interpretando um produto seqüencial que tem como segundo fator a soma não-determinística $\| _ , d^{(k)} \mid f^{(k)}, _ \|$. Neste caso, pode-se fazer a união de z com outro objeto cujos tokens sejam indexados por $\alpha_1 0$. Tal união resulta em um novo objeto em $\mathbb{P}_0 \amalg \mathbb{P}_0$ que será aproximado pelos dois anteriores que lhe construíram, interpretando um novo produto seqüencial. Veja a Figura 4.11 que representa o conjunto coerente $y \cup z$.

A próxima exemplificação é uma generalização do último exemplo.

Exemplificação 4.4

Considere a família \mathcal{A}_k de todas as ações rotuladas por $i_k \in I$ e os subconjuntos coerentes em \mathbb{P}_0 definidos logo a seguir:

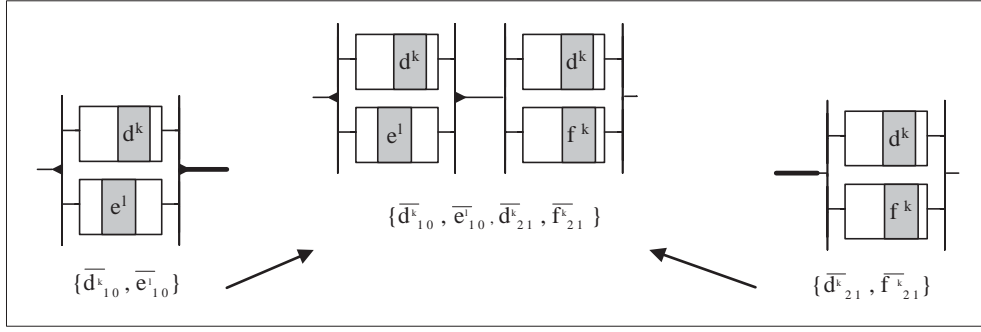


FIGURA 4.11 - Representação do PS com Interpretação em $\mathbb{P}_0 \amalg \mathbb{P}_0$.

- (1) $t' = \{\overline{d^k_{1\alpha}}\}_{k \in I}$ que interpreta o processo paralelo que executa simultaneamente a ação $d \in \mathcal{A}$ em todas as posições de memória, que correspondem aos pontos do espaço geométrico I ;
- (2) $t'' = \{\overline{d^k_{2\beta}}\}_{d \in \mathcal{A}}$, interpretando uma escolha não-determinística entre todas as ações $d : \tilde{S} \rightarrow V$ na mesma posição k .

1. Se $\alpha = 0$ and $\beta = 1$, o subconjunto coerente $t = t' \cup t'' = \{\overline{d^k_{10}}\}_{k \in I} \cup \{\overline{d^k_{21}}\}_{d \in \mathcal{A}}$ interpreta um produto seqüencial que executa primeiro o produto paralelo interpretado pelo subconjunto coerente t' e depois de sua finalização executa a soma não-determinística interpretada pelo subconjunto coerente t'' (indexação 21).
2. Entretanto, quando $\alpha = 1$ e $\beta = 0$ o referido subconjunto coerente interpreta o produto seqüencial na ordem inversa, ou seja, o produto paralelo interpretado por t' será executado após o término da execução da soma não-determinística interpretada por t'' .
3. Salienta-se, ainda, que cada conjunto coerente unitário $\{d^{(k)}\} \in \mathbb{D}_0$ define um conjunto coerente de tokens indexados $\{d^{(k)}_{00}, d^{(k)}_{01}\} \in \mathbb{P}_0 \amalg \mathbb{P}_0$ cuja interpretação corresponde ao produto seqüencial definido sobre o mesmo argumento, no caso, $d^{(k)} \cdot d^{(k)}$.

O conjunto coerente que interpreta o processo que executa o processo seqüencial $(\dots \underbrace{((d^{(k)} \cdot d^{(k)}) \cdot d^{(k)}) \cdot \dots \cdot d^{(k)}}_n)$ será construído em $\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}$ no nível $\mathbb{D}_{n-1} - \mathbb{D}_n$ da estrutura deste modelo.

Observação 4.7 Para todo token $\mathbf{a} \in \mathbb{P}_0$, considere $\{\mathbf{a}_{\alpha_1 \alpha_2}\} \in \mathbb{P}_0 \amalg \mathbb{P}_0$.

1. O índice α_2 informa a posição que o processo \mathbf{a} ocupa no produto seqüencial que está sendo interpretado, ou seja, ele indica se \mathbf{a} corresponde ao primeiro ou ao segundo fator deste produto.
2. O índice α_1 informa se \mathbf{a} interpreta um processo elementar ($\alpha_1 = 0$) ou um produto paralelo ($\alpha_1 = 1$) ou uma soma não-determinística ($\alpha_1 = 2$). Observe a Figura 4.12.

$\mathbb{P}_0 \amalg \mathbb{P}_0$	0	 $\{d^k_{00}\}$	 $\{\bar{d}^k_{10}\}$	 $\{\bar{d}^k_{20}\}$
	1	 $\{d^k_{10}\}$	 $\{\bar{d}^k_{11}\}$	 $\{\bar{d}^k_{21}\}$
\mathbb{D}_0		 $\{d^k\}$	$\xrightarrow{\text{Skip}}$ $\{\}$	$\xrightarrow{\text{Skip}}$ $\{\}$
		$\xrightarrow{\text{Skip}}$ $\{\}$	$\xrightarrow{\text{Skip}}$ $\{\}$	$\xrightarrow{\text{Skip}}$ $\{\}$

FIGURA 4.12 - Representações das Imersões e Projeções em $\mathbb{P}_0 \amalg \mathbb{P}_0$.

4.5.1 Projeções e Imersões do Espaço Coerente $\mathbb{P}_0 \amalg \mathbb{P}_0$

Definição 4.14 Considere \mathbb{P}_0 e $\mathbb{P}_0 \amalg \mathbb{P}_0$. As imersões $\psi_0^{(\alpha)} : \mathbb{P}_0 \rightarrow \mathbb{P}_0 \amalg \mathbb{P}_0$ são definidas pela expressão $\psi_0^{(\alpha)}(x) = \{\alpha\} \times x$, sempre que $\alpha \in \{0, 1\}$. Em particular, $\psi_0^{(\alpha)}(\emptyset) = \emptyset$, logo $\psi_0^{(\alpha)}$ é estrita.

Definição 4.15 Considere \mathbb{P}_0 e $\mathbb{P}_0 \amalg \mathbb{P}_0$. As projeções $\Psi_0^{(\alpha)} : \mathbb{P}_0 \amalg \mathbb{P}_0 \rightarrow \mathbb{P}_0$ são dadas pela expressão $\Psi_0^{(\alpha)}(x) = \{\mathbf{a} \mid (\alpha, \mathbf{a}) \in x\}$, sendo que $(\alpha, \mathbf{a}) \in |\mathbb{P}_0 \amalg \mathbb{P}_0|$, $\mathbf{a} \in |\mathbb{P}_0|$, e $\alpha \in \{0, 1\}$.

Proposição 4.3 As funções $\Psi_0^{(\alpha)} : \mathbb{P}_0 \amalg \mathbb{P}_0 \rightarrow \mathbb{P}_0$ e $\psi_0^{(\alpha)} : \mathbb{P}_0 \rightarrow \mathbb{P}_0 \amalg \mathbb{P}_0$, com $\alpha \in \{0, 1\}$, são monótonas contínuas estáveis e lineares.

Desde que o produto direto é o produto categórico na categoria CospLin , tem-se assegurado que as projeções $\Psi_0^{(\alpha)}$, com $\alpha \in \{0, 1\}$, são funções monótonas, contínuas, estáveis e lineares, portanto morfismos nesta categoria. As imersões $\psi_0^{(\alpha)}$ verificam estas mesmas propriedades. A demonstração será omitida, entretanto pode ser construída de forma análoga àquela da Proposição 4.5.

Veja a tabela da Figura 4.12, onde estão representadas as projeções de um conjunto coerente unitário em $\mathbb{P}_0 \amalg \mathbb{P}_0$.

4.5.2 Função Posição em $\mathbb{P}_0 \amalg \mathbb{P}_0$

Definição 4.16 Considere o espaço coerente $\mathbb{P}_0 \amalg \mathbb{P}_0 \equiv (\text{Coh}(\mathbb{P}_0 \amalg \mathbb{P}_0), \approx_{\amalg})$, a função $v_{\mathbb{P}_0} : \mathbb{P}_0 \rightarrow \wp(\omega)$ apresentada na Definição 4.12 e $v_{\mathbb{P}_0 \amalg \mathbb{P}_0} : \mathbb{P}_0 \amalg \mathbb{P}_0 \rightarrow \wp(\omega)$ a função dada por $v_{\mathbb{P}_0 \amalg \mathbb{P}_0}(\mathbf{a}_\alpha) = v_{\mathbb{P}_0}(\mathbf{a})$. A função posição $\Upsilon_{\mathbb{P}_0 \amalg \mathbb{P}_0} : \mathbb{P}_0 \amalg \mathbb{P}_0 \rightarrow \wp(\omega)$ é definida por $\Upsilon_{\mathbb{P}_0 \amalg \mathbb{P}_0}(x) = \bigcup_{\mathbf{a}_\alpha \in x} v_{\mathbb{P}_0 \amalg \mathbb{P}_0}(\mathbf{a}_\alpha)$. Em particular, $\Upsilon_{\mathbb{P}_0 \amalg \mathbb{P}_0}(\emptyset) = \emptyset$.

4.6 Espaço Coerente $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$

A construção de processos pela aplicação da soma determinística depende de uma escolha previamente determinada, associada a um teste. Define-se a seguir o espaço coerente plano \mathbb{B} que interpreta os testes do conjunto \mathcal{B} apresentado na Definição 3.7.

4.6.1 Espaço Coerente Plano de Testes Elementares

Definição 4.17 *Seja $\mathbf{B} \equiv (\mathcal{B}, \cong_{\mathcal{B}})$ a teia discreta definida pelo conjunto \mathcal{B} de testes booleanos e a relação de igualdade, $b \cong_{\mathcal{B}} b' \leftrightarrow b = b'$. A coleção de todos os conjuntos coerentes de testes da teia \mathbf{B} , ordenada pela inclusão, é denominado o **espaço coerente de testes**, indicado por $\mathbb{B} \equiv (\text{Coh}\mathbf{B}, \subseteq)$, de tal forma que $\text{Coh}\mathbf{B} = \{\emptyset\} \cup \{\{b\} | b \in \mathcal{B}\}$.*

Neste trabalho a construção do espaço coerente dos testes fundamenta-se na lógica binária, ao se considerar uma teia discreta cujos tokens são definidos na dualidade do verdadeiro e falso, no caso do cpo booleano. Entretanto, esta construção pode admitir outras abordagens lógicas, por exemplo a lógica fuzzy e a lógica intervalar.

Definição 4.18 *Seja $\bar{\mathcal{D}}_0$ a família de todos os subconjuntos coerentes finitos não-nulos de processos elementares em \mathbb{D}_0 . Considere também os espaços coerentes \mathbb{B} e $\mathbb{P}_0 \prod \mathbb{P}_0$. O **espaço coerente da soma determinística de processos** é definido pelo produto direto entre \mathbb{B} e $\mathbb{P}_0 \prod \mathbb{P}_0$, indicado por $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$. Sua correspondente teia é definida sobre o conjunto resultante da união disjunta*

$$\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0 \equiv ((\mathcal{B} \dot{\cup} \mathcal{P}_0 \dot{\cup} \mathcal{P}_0), \approx_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0})$$

onde $\mathcal{P}_0 = \mathcal{D}_0 \dot{\cup} \bar{\mathcal{D}}_0 \dot{\cup} \bar{\mathcal{D}}_0$. A correspondente relação de coerência $\approx_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0}$ é dada por:

$$(\alpha, \mathbf{a}) \approx_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0} (\beta, \mathbf{b}) \Leftrightarrow \begin{cases} \alpha = \beta = 2 \text{ e } \mathbf{a} \approx_{\mathbb{B}} \mathbf{b}, \text{ ou} \\ \alpha = \beta \neq 2 \text{ e } \mathbf{a} \approx_{\mathbb{P}_0} \mathbf{b}, \text{ ou} \\ \alpha \neq \beta. \end{cases}$$

Cada token $(\alpha, a) \in \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ será indicado por $a_{\alpha} \equiv (\alpha, a)$. A Figura 4.13 mostra um esquema de construção do espaço coerente $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$.

Observação 4.8 *No espaço coerente $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ a relação de coerência $\approx_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0}$ aplicada sobre tokens indexados pela união disjunta $\mathcal{B} \dot{\cup} \mathcal{P}_0 \dot{\cup} \mathcal{P}_0$ define a construção de novos objetos ao mesmo tempo que reconstrói os objetos dos espaços coerentes sobre os quais o produto direto é definido. Neste sentido, algumas observações se tornam importantes a fim de facilitar a compreensão desta etapa da estrutura proposta. Para tal, suponha os conjuntos coerentes em \mathbb{P}_0 referidos anteriormente na seção 9.4.*

1. O subconjunto coerente $\{\{b_2^{(n)}\}, \{d^{(k)00}\}, \{e_{01}^{(l)}\}\} \in \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ constitui-se num objeto total, identificado pela expressão $d^{(k)} +_b e^{(l)}$ e definido como a soma determinística entre os processos elementares $d^{(k)}$ e $e^{(l)}$, tendo como referência o teste $b^{(n)} \in \mathbb{B}$. Nesta interpretação se a verdade relativa ao teste b é satisfeita então o primeiro processo a ser executado será o processo elementar $d^{(k)}$. Caso contrário, a ordem de execução será invertida e $e^{(l)}$ será executado em primeiro lugar. Isto mostra desde já que a soma determinística não é comutativa. Veja Figura 4.14.

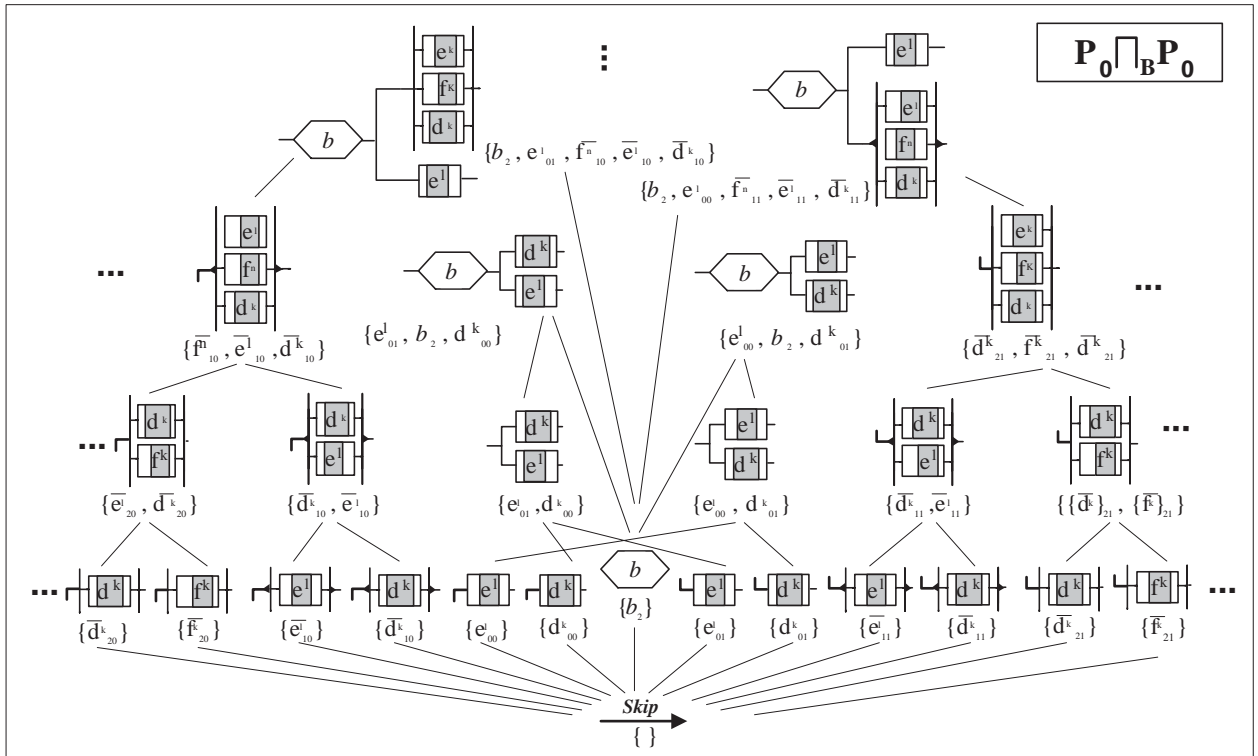


FIGURA 4.13 - O Espaço Coerente $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ da SD de Processos.

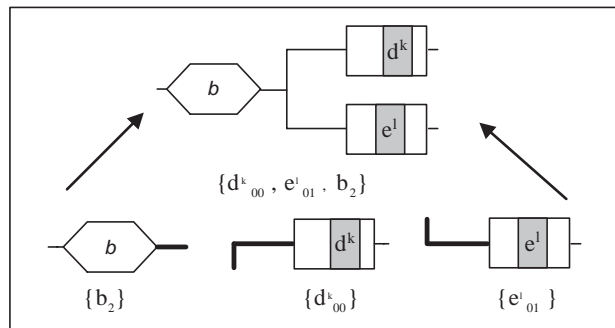


FIGURA 4.14 - A Construção da Soma Determinística $d^{(k)} +_b e^{(l)}$ em $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$.

2. Se $d \neq l$, o conjunto coerente $\{\overline{d^{(k)}}_{10}, \overline{d^{(l)}}_{10}\}$ é um objeto parcial em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$. Tal objeto interpreta a soma determinística que executa o produto paralelo identificado pela expressão $\| _, d^{(k)}, e^{(l)}, _ \|$ quando a verdade de algum teste for satisfeita. Neste caso não está definido ainda qual será o teste. Também não está definido o processo que deve ser executado caso esta possibilidade não se efetive. Veja a representação na Figura 4.15.

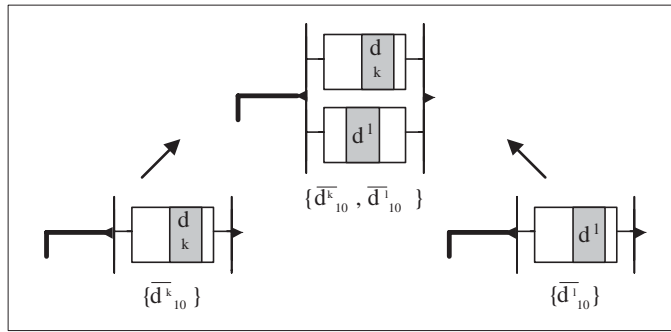


FIGURA 4.15 - A União Indexada de Tokens em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$.

3. Pode-se entretanto pensar na união de $\{\overline{d^{(k)}}_{10}, \overline{d^{(l)}}_{10}\}$ com o conjunto $\{e^{(l)}_{01}\}$. Tal união é também um objeto parcial em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$ interpretando a soma determinística que agora tem o processo que deve ser executado quando a falsidade de algum teste (por enquanto indefinido) for satisfeita.
4. Observe ainda que, satisfeita a desigualdade $k \neq l, l \neq n, n \neq k$, o subconjunto coerente $v = \{\overline{d^{(k)}}_{10}, \overline{d^{(l)}}_{10}, e^{(l)}_{01}, b_2^{(n)}\}$ em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$, identificado pela expressão $\| _, d^{(k)}, d^{(l)}, _ \| +_b e^{(l)}$ e representado na Figura 4.16, constitui-se numa aproximação para o conjunto coerente $\{\overline{d^{(k)}}_{10}, \overline{d^{(l)}}_{10}, \overline{d^{(n)}}_{10}, e^{(l)}_{01}, b_2^{(n)}\}$.

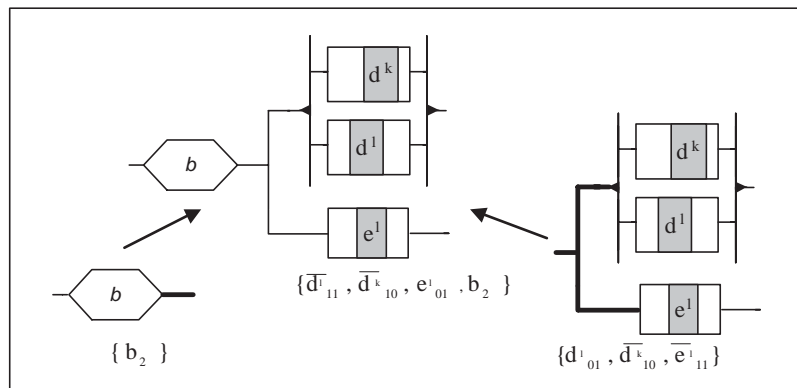


FIGURA 4.16 - Representação Gráfica da Construção do Processo v em $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$.

5. Para finalizar, suponha $\{\mathbf{a}_{\alpha_1\alpha_2}\} \in \mathbf{P}_0 \prod_{\mathbf{B}} \mathbf{P}_0$.

- (a) α_2 informa sobre a posição que tal processo ocupa na operação que ele interpreta, no caso uma soma determinística. Ou ainda, α_2 indica se o processo interpretado por \mathbf{a} pertence à primeira ou à segunda parcela desta soma.
- (b) α_1 informa se \mathbf{a} interpreta um processo elementar ($\alpha_1 = 0$) ou um produto paralelo ($\alpha_1 = 1$) ou uma soma não-determinística ($\alpha_1 = 2$). Além disso,
 - i. . se $\alpha_1 = 0$ então $\mathbf{a} \in \mathbf{D}_0$;
 - ii. . se $\alpha_1 \in \{1, 2\}$ então $\mathbf{a} = \bar{a} \in \text{Coh}(\mathbf{D}_0)$ e $a \in \mathbf{D}_0$;

4.6.2 Projeções e Imersões do $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$

Definição 4.19 Considere os espaços coerentes \mathbb{B} , \mathbb{P}_0 e $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$.

1. Se $\alpha \in \{0, 1\}$ então $\lambda_0^{(\alpha)} : \mathbb{P}_0 \rightarrow \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ é definida por $\lambda_0^{(\alpha)}(x) = \{\alpha\} \times x$;
2. Se $\alpha = 2$ então $\lambda_0^{(2)} : \mathbb{B} \rightarrow \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ é definida por $\lambda_0^{(2)}(x) = \{\alpha\} \times x$.

Em particular, $\lambda_0^{(\alpha)}(\emptyset) = \emptyset, \forall \alpha \in \{0, 1, 2\}$.

Definição 4.20 Considere os espaços coerentes \mathbb{B} , \mathbb{P}_0 e $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$. Definem-se as projeções

1. $\Lambda_0^{(\alpha)} : \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0 \rightarrow \mathbb{P}_0$ pela expressão $\Lambda_0^{(\alpha)}(x) = \{\mathbf{a} \mid (\alpha, \mathbf{a}) \in x\}$, sempre que $\alpha \in \{0, 1\}$. Neste caso, tem-se $(\alpha, \mathbf{a}) \in |\mathbf{P}_0 \prod_{\mathbb{B}} \mathbf{P}_0|$ e $\mathbf{a} \in |\mathbf{P}_0|$.
2. $\Lambda_0^{(\alpha)} : \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0 \rightarrow \mathbb{B}$ pela expressão $\Lambda_0^{(\alpha)}(x) = \{\mathbf{a} \mid (\alpha, \mathbf{a}) \in x\}$, sempre que $\alpha = 2$. Neste caso, $(\alpha, \mathbf{a}) \in |\mathbf{P}_0 \prod_{\mathbb{B}} \mathbf{P}_0|$ e $\mathbf{a} \in |\mathbf{B}|$.

Em particular, $\Lambda_0^{(\alpha)}(\emptyset) = \emptyset, \forall \alpha \in \{0, 1, 2\}$. Ou seja, as projeções $\Lambda_0^{(\alpha)}$ são estritas.

A Figura 4.17 apresenta as imersões de um processo elementar em $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ e a correspondente projeção $\Lambda_0^{(0)}$.

A próxima proposição assegura que tais funções são morfismos na categoria CospLin . A demonstração desta e das demais funções, projeções e imersões, apresentadas no final deste capítulo será omitida. Contudo, sua construção é análoga às demonstrações das Proposições 4.1 e 4.2.

Proposição 4.4 As funções $\Lambda_0^{(\alpha)} : \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0 \rightarrow \mathbb{P}_0$ e $\lambda_0^{(\alpha)} : \mathbb{P}_0 \rightarrow \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ apresentadas nas Definições 4.20 e 4.19 são morfismos em CospLin .

4.6.3 Função Posição em $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$

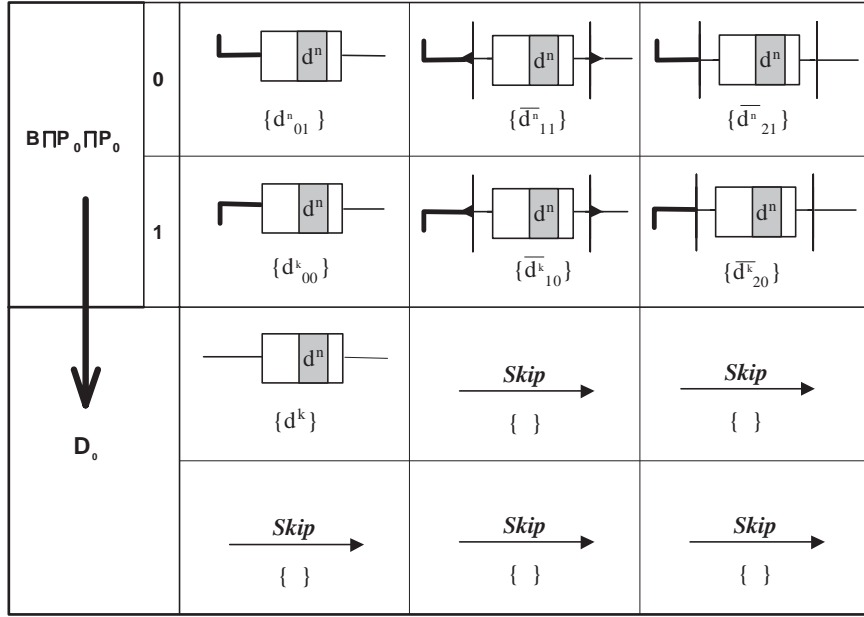
Definição 4.21 Considere os espaços coerentes \mathbb{B} , \mathbb{P}_0 , $\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$ e $\mathbb{B} \prod_{\mathbb{B}} \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0$. Considere também a função posição $\Upsilon_{\mathbb{P}_0}$ apresentada na Definição 4.12. Seja a função $v_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0} : \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0 \rightarrow \wp(\omega)$ dada pela expressão

$$v_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0}(\alpha, \mathbf{a}) = \begin{cases} \emptyset & \text{se } \alpha = 2, \\ v_{\mathbf{P}_0}(\mathbf{a}) & \text{caso contrário;} \end{cases}$$

A função posição $\Upsilon_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0} : \mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0 \rightarrow \wp(\omega)$ é definida da seguinte forma

$$\Upsilon_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0}(x) = \bigcup_{(\alpha, a) \in x} v_{\mathbf{P}_0 \prod_{\mathbb{B}} \mathbf{P}_0}(\alpha, a).$$

Em particular, $\Upsilon_{\mathbb{P}_0 \prod_{\mathbb{B}} \mathbb{P}_0}(\emptyset) = \emptyset$.

FIGURA 4.17 - Representações das Imersões sobre $\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0$.

4.7 Espaço Coerente \mathbb{D}_1

O espaço coerente \mathbb{D}_1 constitui-se na soma direta dos subespaços coerentes apresentados nas seções anteriores, sendo que sua definição formal é apresentada logo abaixo.

Definição 4.22 *Considere os espaços coerentes $\mathbb{P}_0 = \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp$, $\mathbb{P}_0 \amalg \mathbb{P}_0$ e $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$. A soma direta*

$$\mathbb{D}_1 = \mathbb{P}_0 \amalg (\mathbb{P}_0 \amalg \mathbb{P}_0) \amalg_{\mathbb{B}} (\mathbb{P}_0 \amalg \mathbb{P}_0)$$

define o espaço coerente $\mathbb{D}_1 \equiv (\text{Coh}(\mathbb{D}_1), \subseteq)$. Sua correspondente teia é definida sobre o conjunto resultante da união disjunta

$$|\mathbb{D}_1| \equiv (|\mathbb{D}_1|, \approx_{\mathbb{D}_1}) \equiv (\mathcal{P}_0 \dot{\cup} (\mathcal{P}_0 \dot{\cup} \mathcal{P}_0) \dot{\cup} (\mathcal{B} \dot{\cup} \mathcal{P}_0 \dot{\cup} \mathcal{P}_0), \approx_{\mathbb{D}_1})$$

onde $\mathcal{P}_0 = \mathcal{D}_0 \dot{\cup} \bar{\mathcal{D}}_0 \dot{\cup} \bar{\mathcal{D}}_0^\perp$. A correspondente relação de coerência $\approx_{\mathbb{D}_1}$ é dada por

$$(\alpha, \mathbf{a}) \approx_{\mathbb{D}_1} (\beta, \mathbf{b}) \Leftrightarrow \begin{cases} \alpha = \beta = 0 \text{ e } \mathbf{a} \approx_{\mathbb{P}_0} \mathbf{b}, \text{ ou} \\ \alpha = \beta = 1 \text{ e } \mathbf{a} \approx_{\mathbb{P}_0 \amalg \mathbb{P}_0} \mathbf{b}, \text{ ou} \\ \alpha = \beta = 2 \text{ e } \mathbf{a} \approx_{\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0} \mathbf{b} \end{cases}$$

sempre que $(\alpha, \mathbf{a}), (\beta, \mathbf{b}) \in |\mathbb{D}_1|$ e $\alpha, \beta \in \{0, 1, 2\}$.

Na Definição 4.22 também considera-se a identificação $(\alpha, \mathbf{a}) \equiv \mathbf{a}_\alpha$.

Ao contrário do que ocorre com o PP e a SND, os construtores de processos PS e SD são representados por operações externas do domínio \mathbb{P}_0 . Um subconjunto coerente em \mathbb{D}_1 interpreta no máximo uma destas operações. Em outras palavras, ainda não há interpretação para uma operação que executa um PS entre processos resultantes de outros produtos sequenciais ou de somas determinísticas. Tal interpretação será alcançada em outros níveis da construção.

Observação 4.9 Algumas considerações sobre os tokens de cada uma das teias geradoras dos espaços coerentes que definem \mathbb{D}_1 são importantes e serão apresentadas a seguir.

- $d^{(k)} \in \mathbf{D}_0 \rightarrow d_0^{(k)} \in \mathbf{P}_0 \rightarrow d_{00}^{(k)} \in \mathbf{D}_1$.
- $\overline{d^{(k)}} \in \overline{\mathbf{D}}_0 \rightarrow \overline{d^{(k)}}_1 \in \mathbf{P}_0 \rightarrow \overline{d^{(k)}}_{10} \in \mathbf{D}_1$.
- $\overline{d^{(k)}} \in \overline{\mathbf{D}}_0^\perp \rightarrow \overline{d^{(k)}}_2 \in \mathbf{P}_0 \rightarrow \overline{d^{(k)}}_{20} \in \mathbf{D}_1$.
- $d_0^{(k)} \in \mathbf{P}_0 \rightarrow d_{0\alpha_2}^{(k)} \in \mathbf{P}_0 \amalg \mathbf{P}_0 \rightarrow d_{0\alpha_2 1}^{(k)} \in \mathbf{D}_1$ sempre que $\alpha_2 \in \{0, 1\}$.
- $b \in \mathbf{B} \rightarrow b_2 \in \mathbf{P}_0 \amalg_{\mathbf{B}} \mathbf{P}_0 \rightarrow b_{22} \in \mathbf{D}_1$.
- $d_0^{(k)} \in \mathbf{P}_0 \rightarrow d_{0\alpha_2}^{(k)} \in \mathbf{B} \amalg \mathbf{P}_0 \amalg \mathbf{P}_0 \rightarrow d_{0\alpha_2 2}^{(k)} \in \mathbf{D}_1$ sempre que $\alpha_2 \in \{0, 1\}$.
- Suponha que $\alpha_1 \in \{1, 2\}$ e $\alpha_2 \in \{0, 1\}$, então:
 $\overline{d^{(k)}}_{\alpha_1} \in \mathbf{P}_0 \rightarrow \overline{d^{(k)}}_{\alpha_1 \alpha_2} \in \mathbf{B} \amalg \mathbf{P}_0 \amalg \mathbf{P}_0 \rightarrow \overline{d^{(k)}}_{\alpha_1 \alpha_2 2} \in \mathbf{D}_1$; e
 $\overline{d^{(k)}}_{\alpha_1} \in \mathbf{P}_0 \rightarrow \overline{d^{(k)}}_{\alpha_1 \alpha_2} \in \mathbf{P}_0 \amalg \mathbf{P}_0 \rightarrow \overline{d^{(k)}}_{\alpha_1 \alpha_2 1} \in \mathbf{D}_1$

A Figura 4.18 mostra as representações das imersões de um processo elementar em \mathbb{D}_1 e as correspondentes imagens pela projeção $\Pi_{1,0} : \mathbb{D}_1 \rightarrow \mathbb{D}_0$, definida na próxima seção.

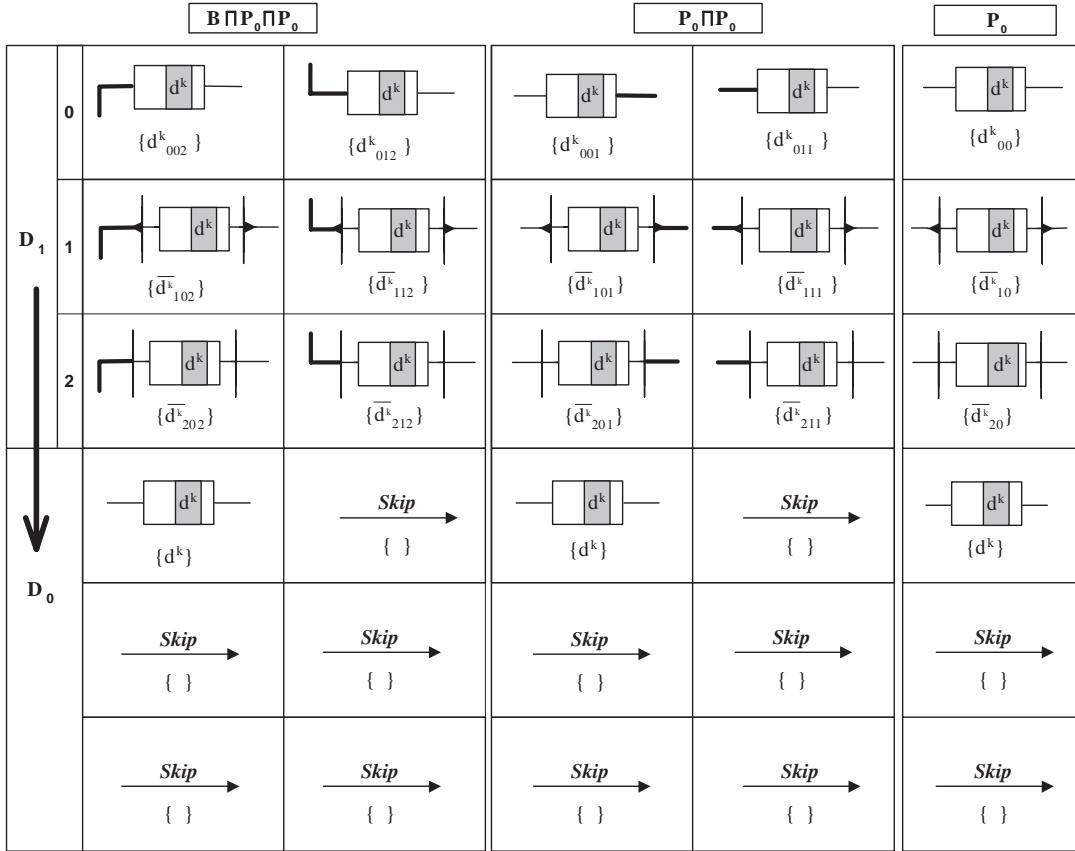


FIGURA 4.18 - Representações das Imersões sobre \mathbb{D}_1 .

Os morfismos apresentados na Figura 4.19 resumem o relacionamento entre os espaços coerentes envolvidos na construção dos subníveis $\mathbb{D}_0 - \mathbb{P}_0$ e $\mathbb{P}_0 - \mathbb{D}_1$, os quais compõem o primeiro nível $\mathbb{D}_0 - \mathbb{D}_1$ da estrutura do modelo de máquina proposto neste trabalho.

Comparando-se com a Figura 4.9, verifica-se o crescimento do processo construtivo da estrutura do modelo, através da presença de novas imersões e projeções. Tais funções serão objeto de estudo da próxima seção.

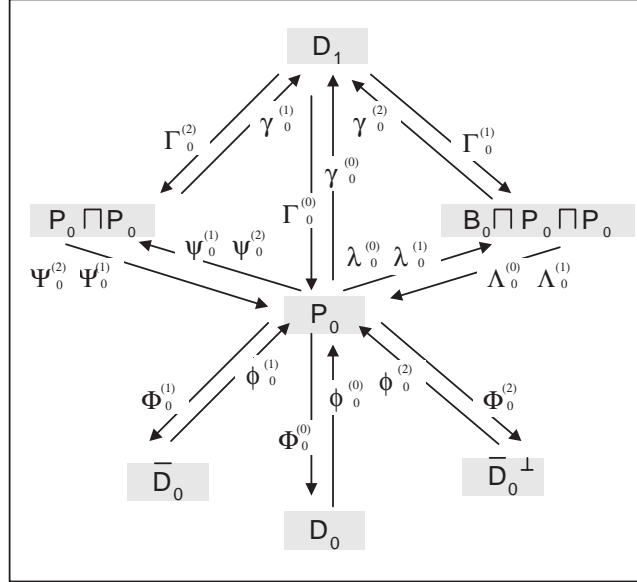


FIGURA 4.19 - Projeções e Imersões em $\mathbb{D}_0 - \mathbb{D}_1$.

A seguir, serão definidas e analisadas as projeções e imersões que formalizam o relacionamento entre os espaços coerentes que constroem o segundo subnível do primeiro nível da construção, no caso $\mathbb{P}_0 - \mathbb{D}_1$.

4.7.1 Imersões e Projeções do Subnível $\mathbb{P}_0 - \mathbb{D}_1$

Definição 4.23 Considere agora os espaços coerentes \mathbb{P}_0 , $\mathbb{P}_0 \amalg \mathbb{P}_0$, $\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0$ e \mathbb{D}_1 . As imersões

$$\gamma_0^{(0)} : \mathbb{P}_0 \rightarrow \mathbb{D}_1, \quad \gamma_0^{(1)} : \mathbb{P}_0 \amalg \mathbb{P}_0 \rightarrow \mathbb{D}_1 \quad \text{e} \quad \gamma_0^{(2)} : \mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0 \rightarrow \mathbb{D}_1$$

são definidas pela expressão $\gamma_0^{(\alpha)}(x) = \{\mathbf{a}_\alpha | \mathbf{a} \in x\}$, sempre que $\alpha \in \{0, 1, 2\}$. Em particular, $\gamma_0^\alpha(\emptyset) = \emptyset$.

Definição 4.24 Considere os espaços coerentes \mathbb{P}_0 , $\mathbb{P}_0 \amalg \mathbb{P}_0$, $\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0$ e \mathbb{D}_1 . De forma análoga define-se as projeções

$$\Gamma_0^{(0)} : \mathbb{D}_1 \rightarrow \mathbb{P}_0, \quad \Gamma_0^{(1)} : \mathbb{D}_1 \rightarrow \mathbb{P}_0 \amalg \mathbb{P}_0 \quad \text{e} \quad \Gamma_0^{(2)} : \mathbb{D}_1 \rightarrow \mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0$$

pela expressão $\Gamma_0^{(\alpha)}(x) = \{\mathbf{a} | \mathbf{a}_\alpha \in x\}$, sempre que $\alpha \in \{0, 1, 2\}$. Em particular, $\Gamma_0^{(\alpha)}(\emptyset) = \emptyset$. Veja Figura 4.19.

Proposição 4.5 Considere $\alpha \in \{0, 1, 2\}$. As funções estritas $\Gamma_0^{(\alpha)}$ e $\gamma_0^{(\alpha)}$, respectivamente apresentadas nas Definições 4.24 e 4.23, são morfismos na categoria CospLin .

4.7.2 Imersões e Projeções do Primeiro Nível $\mathbb{D}_0 - \mathbb{D}_1$

Conforme pode ser observado na Tabela 4.18 um conjunto coerente que interpreta um processo elementar em \mathbb{D}_0 pode ser imerso em \mathbb{D}_1 de três formas distintas: ou como um processo elementar (reconstruído), ou como um fator (da direita ou da esquerda) de um produto seqüencial ou ainda, como um termo de uma soma determinística (o primeiro ou o segundo). O mesmo pode ser alcançado para interpretações relacionadas ao produto paralelo em $\bar{\mathbb{D}}_0$ ou à soma não-determinística $\bar{\mathbb{D}}_0^\perp$.

A seguir são definidas as funções lineares que formalizam estas idéias.

Definição 4.25 *Considere os espaços coerentes \mathbb{D}_0 , $\bar{\mathbb{D}}_0$, $\bar{\mathbb{D}}_0^\perp$ e \mathbb{D}_1 como também os indexadores $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$ e $\beta \in \{0, 1, 2\}$. As imersões*

$$\pi_0^{(\alpha)} : \mathbb{D}_0 \rightarrow \mathbb{D}_1 \text{ se } \beta = 0, \quad \pi_0^{(\alpha)} : \bar{\mathbb{D}}_0 \rightarrow \mathbb{D}_1 \text{ se } \beta = 1 \quad e \quad \pi_0^{(\alpha)} : \bar{\mathbb{D}}_0^\perp \rightarrow \mathbb{D}_1 \text{ se } \beta = 2$$

são definidas pela expressão $\pi_0^{(\alpha)} = \{\mathbf{a}_\alpha \mid \mathbf{a} \in x\}$. Em particular, $\pi_0^{(\alpha)}(\emptyset) = \emptyset$.

Proposição 4.6 *Considere as imersões apresentadas na Definição 4.25. Para cada $\beta \in \{0, 1, 2\}$ as seguintes composições são satisfeitas:*

- $\pi_0^{(\beta 0)} = \gamma_0^{(0)} \circ \phi_0^{(\beta)}$,
- $\pi_0^{(\beta 01)} = \gamma_0^{(1)} \circ \psi_0^{(0)} \circ \phi_0^{(\beta)}$ e $\pi_0^{(\beta 11)} = \gamma_0^{(1)} \circ \psi_0^{(1)} \circ \phi_0^{(\beta)}$,
- $\pi_0^{(\beta 02)} = \gamma_0^{(2)} \circ \lambda_0^{(0)} \circ \phi_0^{(\beta)}$ e $\pi_0^{(\beta 12)} = \gamma_0^{(2)} \circ \lambda_0^{(1)} \circ \phi_0^{(\beta)}$,

onde as funções lineares $\gamma_0^{(\beta)}$, $\phi_0^{(\beta)}$, $\psi_0^{(0)}$, $\psi_0^{(1)}$, $\lambda_0^{(0)}$ e $\lambda_0^{(1)}$ foram apresentadas nas Definições 4.23, 4.1, 4.14 e 4.19, respectivamente.

Proposição 4.7 *As imersões apresentadas na Definição 4.25 são morfismos na categoria CospLin .*

As demonstrações para as proposições 4.7 e 4.6 são omitidas mas podem ser construídas considerando-se que a categoria CospLin é fechada para a composição de funções lineares.

Definição 4.26 *Considere os espaços coerentes \mathbb{D}_0 , $\bar{\mathbb{D}}_0$, $\bar{\mathbb{D}}_0^\perp$ e \mathbb{D}_1 como também os indexadores $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$ e $\beta \in \{0, 1, 2\}$. Definem-se as projeções estritas*

1. $\Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \mathbb{D}_0$, $\Pi_0^{(\alpha)}(x) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x \text{ e } \mathbf{a} \in \mathbf{D}_0\}$, se $\beta = 0$;
2. $\Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \bar{\mathbb{D}}_0$, $\Pi_0^{(\alpha)}(x) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x \text{ e } \mathbf{a} \in \bar{\mathbf{D}}_0\}$, se $\beta = 1$;
3. $\Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \bar{\mathbb{D}}_0^\perp$, $\Pi_0^{(\alpha)}(x) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x \text{ e } \mathbf{a} \in \bar{\mathbf{D}}_0\}$, se $\beta = 2$.

Consierando-se a Definição 4.26, sempre que $x \in (\gamma_1^{(0)} \circ \phi_0^{(1)})[\bar{\mathbb{D}}_0]$ ou $x \in (\gamma_1^{(0)} \circ \phi_0^{(2)})[\bar{\mathbb{D}}_0]$, tem-se $\Pi_0^{(00)}(x) = \emptyset$.

Proposição 4.8 *Considere as projeções apresentadas na Definição 4.26. Para cada $\beta \in \{0, 1, 2\}$ as seguintes igualdades são satisfeitas*

- $\Pi_0^{(\beta 0)} = \Phi_0^{(0)} \circ \Gamma_0^{(\beta)}$,
- $\Pi_0^{(\beta 01)} = \Phi_0^{(1)} \circ \Psi_0^{(0)} \circ \Gamma_0^{(\beta)}$ e $\Pi_0^{(\beta 11)} = \Phi_0^{(1)} \circ \Psi_0^{(1)} \circ \Gamma_1^{(\beta)}$,

- $\Pi_0^{(\beta 02)} = \Phi_0^{(2)} \circ \Lambda_0^{(0)} \circ \Gamma_0^{(\beta)}$ e $\Pi_0^{(\beta 12)} = \Phi_0^{(2)} \circ \Lambda_0^{(1)} \circ \Gamma_0^{(\beta)}$,

considerando-se que as funções lineares $\Gamma_0^{(\beta)}$, $\Phi_0^{(\beta)}$, $\Psi_0^{(0)}$, $\Psi_0^{(1)}$, $\Lambda_0^{(0)}$ e $\Lambda_0^{(1)}$ foram apresentadas nas Definições 4.23, 4.19, 4.15 e 4.2, respectivamente.

Proposição 4.9 As projeções $\Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \mathbb{D}_0$ apresentadas na Definição 4.26 são morfismos na categoria CospLin .

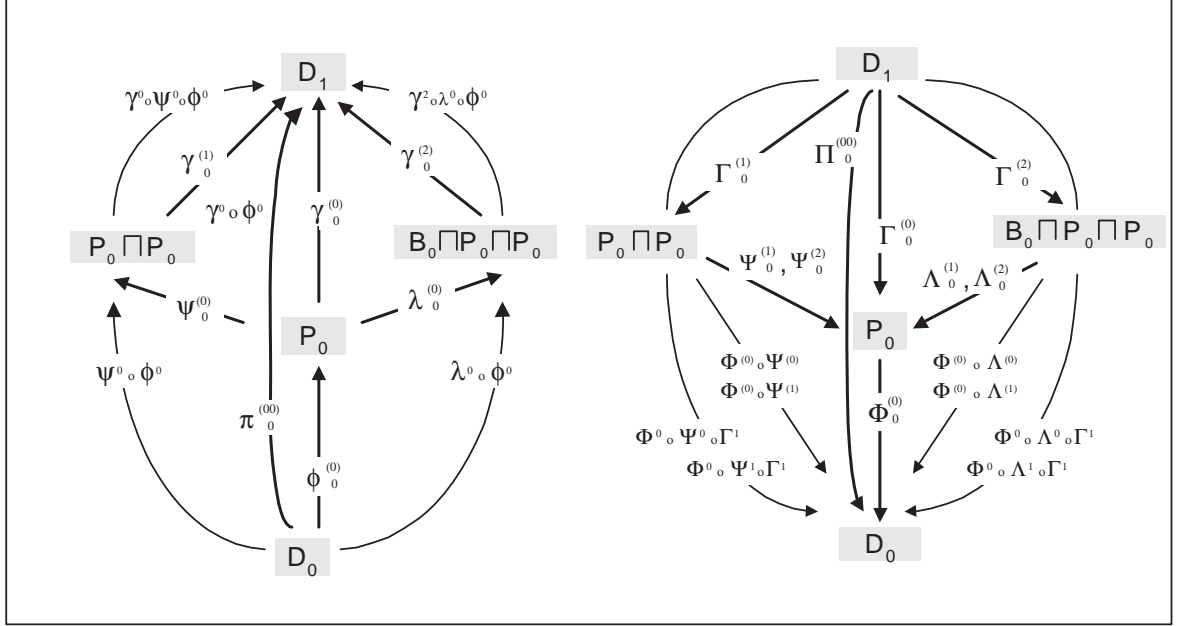


FIGURA 4.20 - Formalização do Relacionamento entre \mathbb{D}_0 e \mathbb{D}_1 .

Proposição 4.10 Considere os espaços coerentes \mathbb{D}_0 , $\bar{\mathbb{D}}_0$, $\bar{\mathbb{D}}_0^\perp$ e \mathbb{D}_1 como também os indexadores $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$ e $\beta \in \{0, 1, 2\}$. Tem-se

- $\langle \Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \mathbb{D}_0, \pi_0^{(\alpha)} : \mathbb{D}_0 \rightarrow \mathbb{D}_1 \rangle$, se $\beta = 0$,
- $\langle \Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \bar{\mathbb{D}}_0, \pi_0^{(\alpha)} : \bar{\mathbb{D}}_0 \rightarrow \mathbb{D}_1 \rangle$, se $\beta = 1$,
- $\langle \Pi_0^{(\alpha)} : \mathbb{D}_1 \rightarrow \bar{\mathbb{D}}_0^\perp, \pi_0^{(\alpha)} : \bar{\mathbb{D}}_0^\perp \rightarrow \mathbb{D}_1 \rangle$, se $\beta = 2$,

são pares-projeção para $(\mathbb{D}_1, \mathbb{D}_0)$, $(\mathbb{D}_1, \bar{\mathbb{D}}_0)$ e $(\mathbb{D}_1, \bar{\mathbb{D}}_0^\perp)$, respectivamente.

Pela proposição 4.10, verificam-se as condições

1. $\Pi_0^{(\alpha)} \circ \pi_0^{(\alpha)} = Id_{\mathbb{D}_0}$ e $\pi_0^{(\alpha)} \circ \Pi_0^{(00)} \subseteq Id_{\mathbb{D}_1}$ se $\beta = 0$,
2. $\Pi_0^{(\alpha)} \circ \pi_0^{(\alpha)} = Id_{\bar{\mathbb{D}}_0}$ e $\pi_0^{(\alpha)} \circ \Pi_0^{(00)} \subseteq Id_{\mathbb{D}_1}$ se $\beta = 1$,
3. $\Pi_0^{(\alpha)} \circ \pi_0^{(\alpha)} = Id_{\bar{\mathbb{D}}_0^\perp}$ e $\pi_0^{(\alpha)} \circ \Pi_0^{(00)} \subseteq Id_{\mathbb{D}_1}$ se $\beta = 2$.

Veja a Figura 4.20, onde estão representados estes morfismos e correspondentes composições.

De acordo com a Definição 4.26, se $x \in \mathbb{P}_0 \amalg \mathbb{P}_0$, a imagem $\Pi_0^{(001)}(x)$ consiste num subconjunto coerente em \mathbb{P}_0 que interpreta o primeiro termo do produto seqüencial interpretado por x . Intuitivamente, a transformação associada à projeção $\Pi_0^{(001)}$, para todo subconjunto coerente que interpreta um processo parcial, com um braço estendido para a direita, consiste na eliminação deste braço. Neste caso, a imagem modela um processo elementar.

A partir das restrições lineares das projeções apresentadas acima, é possível definir outras funções, conforme mostra a próxima definição.

Definição 4.27 *Considere os espaços coerentes \mathbb{D}_0 , \mathbb{P}_0 , $\mathbb{P}_0 \amalg \mathbb{P}_0$, $\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0$ e \mathbb{D}_1 que definem o primeiro nível da estrutura deste modelo computacional. Considere também as imersões $\gamma_0^{(\beta)}$, $\lambda_0^{(\beta)}$ e $\psi_0^{(\beta)}$ apresentadas na Definições 4.1, 4.14 e 4.19 e as projeções $\Pi_0^{(\alpha)}$ formalizadas na Definição 4.7, sempre que $\beta \in \{0, 1, 2\}$ e $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$.*

A função estrita $\Pi_0^+ : \mathbb{D}_1 \rightarrow \mathbb{D}_0$ é definida pela expressão

$$\Pi_0^+(x) = \begin{cases} \Pi_0^{(00)}(x) = \{\mathbf{a} \mid \mathbf{a}_{00} \in x\}, & \text{se } x \in \gamma_0^{(0)}[\mathbb{P}_0], \\ \Pi_0^{(001)}(x) = \{\mathbf{a} \mid \mathbf{a}_{001} \in x\}, & \text{se } (x) \in \gamma_n^{(1)}[\mathbb{P}_0 \amalg \mathbb{P}_0], \\ \Pi_0^{(002)}(x) = \{\mathbf{a} \mid \mathbf{a}_{002} \in x\}, & \text{se } (x) \in \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0], \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma $\Pi_0^- : \mathbb{D}_1 \rightarrow \mathbb{D}_0$ é definida pela expressão

$$\Pi_n^-(x) = \begin{cases} \Pi_0^{(00)}(x) = \{\mathbf{a} \mid \mathbf{a}_{00} \in x\}, & \text{se } x \in \gamma_n^{(0)}, \\ \Pi_0^{(011)}(x) = \{\mathbf{a} \mid \mathbf{a}_{011} \in x\}, & \text{se } (x) \in \gamma_0^{(1)}[\mathbb{P}_0 \amalg \mathbb{P}_0], \\ \Pi_0^{(112)}(x) = \{\mathbf{a} \mid \mathbf{a}_{012} \in x\}, & \text{se } (x) \in \gamma_0^{(2)}[\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0], \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Proposição 4.11 *As projeções Π_0^+ , $\Pi_0^- : \mathbb{D}_0 \rightarrow \mathbb{D}_1$ apresentadas na Definição 4.27 são morfismos na categoria CospLin .*

A demonstração da Proposição 4.11 pode ser construída considerando-se as restrições lineares das projeções $\Pi_0^{(\alpha)}$.

4.7.3 Função Posição em \mathbb{D}_1

A construção do próximo nível $\mathbb{D}_1 - \mathbb{D}_2$ da estrutura baseia-se na definição dos espaços coerentes $\bar{\mathbb{D}}_1$ e $\bar{\mathbb{D}}_1^\perp$, cujas teias são construídas segundo uma relação de coerência induzida pela função posição $\Upsilon_{\mathbb{D}_1}$ sobre os subconjuntos coerentes em \mathbb{D}_1 . Justifica-se assim a próxima definição.

Definição 4.28 *Considere o espaço coerente \mathbb{D}_1 e as funções $v_{\mathbb{P}_0}$, $v_{\mathbb{P}_0 \amalg \mathbb{P}_0}$ e $v_{\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0}$ apresentadas nas Definições 4.12, 4.16 e 4.21, respectivamente. A função posição $\Upsilon_{\mathbb{D}_1} : \mathbb{D}_1 \rightarrow \wp(\omega)$ é definida pela expressão*

$$\Upsilon_{\mathbb{D}_1}(x) = \bigcup_{\mathbf{a}_\alpha \in x} v_{\mathbb{D}_1}(\mathbf{a}_\alpha),$$

sempre que $v_{\mathbb{D}_1} : \mathbb{D}_1 \rightarrow \wp(\omega)$ é dada por

$$v_{\mathbf{D}_1}(\mathbf{a}_\alpha) = \begin{cases} v_{\mathbf{P}_0}(\mathbf{a}) & \text{se } \alpha = 0, \\ v_{\mathbf{P}_0 \amalg \mathbf{P}_0}(\mathbf{a}) & \text{se } \alpha = 1, \\ v_{\mathbf{B} \amalg \mathbf{P}_0 \amalg \mathbf{P}_0}(\mathbf{a}) & \text{se } \alpha = 2, \\ \emptyset & \text{caso contrário.} \end{cases}$$

Por fim, apresenta-se uma exemplificação para ilustrar como se constroem os objetos do próximo nível da estrutura deste modelo, considerando-se inclusive os objetos totais do nível anterior.

Exemplificação 4.5

Esta exemplificação envolve construções sobre os processos apresentados nos exemplos 4.1 e 4.2. Para tal, considere um conjunto enumerável \mathcal{A} de ações e o conjunto I de indexadores onde $d_j \in \mathcal{A}$ e $i \in I$.

1. Seja $z_j = \bigcup_{i \in I} \{d_j^{(i)}\} \times \{(10)\} = \overline{\{d_j^{(i)}\}_{10}}_{i \in I} \in Coh \mathbf{D}_1$ interpretando um processo que executa a ação $d_j \in \mathcal{A}$ em todas as posições $i \in I$, simultaneamente, no primeiro instante t_0 .

Neste caso, para cada subconjunto coerente z_j definido a partir de uma ação d_j a função posição é dada por $\Upsilon_{\mathbf{D}_1}(z_j) = I$. A variação de j determina novas ações em \mathcal{A} caracterizando-se assim a situação de conflito entre todos os subconjuntos coerentes $z_j \in \mathbb{D}_1$. Portanto, tem-se que os subconjuntos z_j são tokens coerentes da teia $\bar{\mathbf{D}}_2^\perp$ e $\Pi_0^{(00)}(z_j) = \emptyset$.

2. Define-se então $y = \bigcup_{j \in \omega} (z_j \times \{(20)\}) = \bigcup_{j \in \omega} (\overline{\{d_j^{(i)}\}_{10}}_{i \in I} \times \{(20)\}) \in \bar{\mathbb{D}}_2^\perp$ que interpreta uma escolha arbitrária entre todos os processos paralelos interpretados por z_j , consideradas neste caso, todas as posições $i \in I$ de memória. O processo y é executado no segundo instante T_1 .

Nos próximos casos, a ordem de execução é exatamente a oposta.

3. $y_i = \bigcup_{j \in \omega} \{d_j^{(i)}\} \times \{(20)\} = \overline{\{d_j^{(i)}\}_{20}}_{j \in \omega} \in \mathbb{D}_1$ que interpreta uma escolha arbitrária entre todos os processos d_j , executados na mesma posição $i \in I$ de memória. Considerando-se que $\Upsilon_{\mathbf{D}_1}(y_i) = \{i\}$ e sempre que $i \neq j$, $\Upsilon_{\mathbf{D}_1}(y_i) \cap \Upsilon_{\mathbf{D}_1}(y_j) = \emptyset$, o que caracteriza a concorrência entre os processos interpretados por estes subconjuntos coerentes.
4. Então, $z = \bigcup_{i \in I} (z_i \times \{(10)\}) = \bigcup_{i \in I} (\overline{\{d_j^{(i)}\}_{20}}_{j \in \omega} \times \{(10)\}) \in \mathbb{D}_1$ modela um processo que executa a soma não-determinística interpretado por z_i , simultaneamente, em todas as posições $i \in I$.

Nesta segunda parte da exemplificação, o processo z é construído de tal forma que, uma escolha não-determinística é efetivada na primeira unidade de tempo. Logo após, no instante t_1 será executada o correspondente produto paralelo.

Veja na Figura 4.21 a representação gráfica para os processos interpretados pelos subconjuntos coerentes y e z .

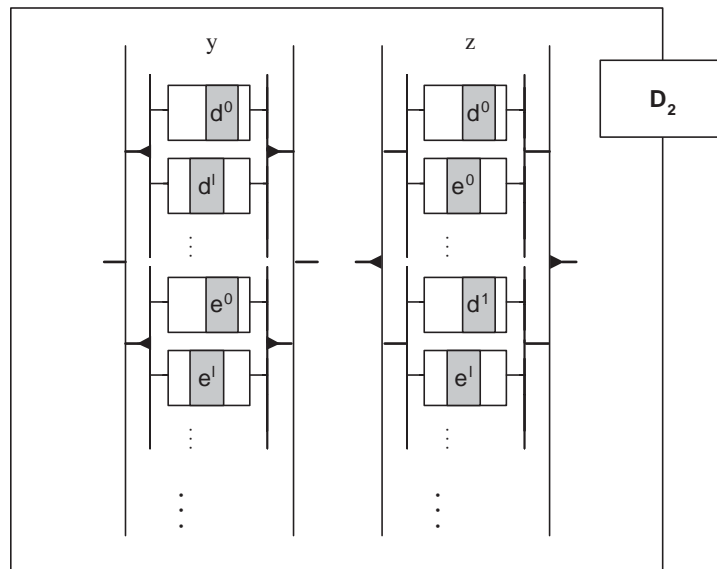


FIGURA 4.21 - Representação dos Processos Interpretados por $y, z \in \mathbb{D}_2$.

5 Nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ da Construção da Estrutura Ordenada do Modelo MG

Neste capítulo apresenta-se a construção do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ cujo desenvolvimento se efetiva de forma análoga ao do primeiro nível $\mathbb{D}_0 - \mathbb{D}_1$, conforme indicado na Figura 5.1.

Esta seção contém ainda exemplos e as correspondentes interpretações de conjuntos coerentes construídos em diferentes níveis, por todos os construtores modelados, de tal forma que facilitem sua compreensão.

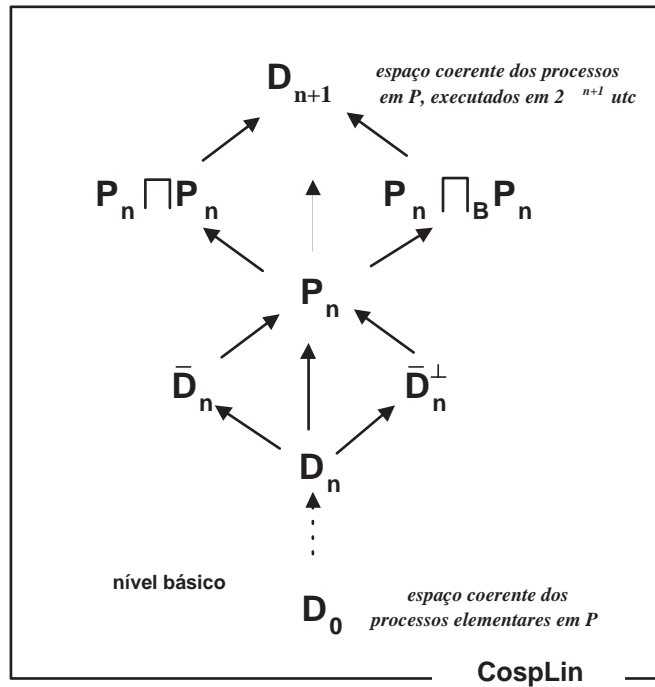


FIGURA 5.1 - Construção do Espaço Coerente de Processos \mathbb{D}_∞ .

5.1 Nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ da Construção

A base para a construção do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, da estrutura indutiva que caracteriza este modelo de máquina geométrica, será o espaço coerente \mathbb{D}_n , dos processos indutivamente gerados, cuja execução se fará, no máximo, em 2^n utc.

5.1.1 Espaço Coerente \mathbb{D}_n

O espaço coerente $\mathbb{D}_n \equiv (\text{Coh}(\mathbb{D}_n), \subseteq)$ é definido pela soma direta

$$\mathbb{D}_n \equiv \mathbb{P}_{n-1} \amalg (\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}) \amalg (\mathbb{B} \amalg \mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}).$$

Os objetos em \mathbb{D}_n interpretam processos que executam no máximo 2^n operações externas (produto seqüencial ou soma determinística) ou um número arbitrário de operações internas (soma não-determinística ou produto paralelo) interpretadas como objetos construídos

em \mathbb{P}_{n-1} . Isto quer dizer que conjuntos coerentes em \mathbb{D}_n e \mathbb{P}_{n-1} interpretam processos cuja execução se fará, no máximo, em 2^n e 2^{n-1} *utc*, respectivamente, considerando sempre que cada processo elementar no subespaço \mathbb{D}_0 está associado a um instante de tempo.

A relação de coerência que constrói a teia sobre a qual se define o espaço coerente $\bar{\mathbb{D}}_n$ é obtida a partir da definição da função posição $\Upsilon_{\mathbb{D}_n} : \mathbb{D}_n \rightarrow \wp(\omega)$ apresentada logo a seguir.

5.1.2 Função Posição em \mathbb{D}_n

Definição 5.1 *Considere os espaços coerentes \mathbb{D}_n e \mathbb{P}_{n-1} e também as funções $v_{\mathbb{D}_1}$, $v_{\mathbb{P}_0}$, $v_{\mathbb{P}_0 \amalg \mathbb{P}_0}$ e $v_{\mathbb{B} \amalg \mathbb{P}_0 \amalg \mathbb{P}_0}$ respectivamente apresentadas nas Definições 4.28, 4.12, 4.16 e 4.21. A função posição $\Upsilon_{\mathbb{D}_n} : \mathbb{D}_n \rightarrow \wp(I)$ é dada por*

$$\Upsilon_{\mathbb{D}_n}(x) = \bigcup_{\mathbf{a}_\alpha \in x} v_{\mathbb{D}_n}(\mathbf{a}_\alpha),$$

sempre que a função posição de tokens $v_{\mathbb{D}_n} : \mathbb{D}_n \rightarrow \wp(I)$ é definida pela expressão

$$v_{\mathbb{D}_n}(\mathbf{a}_\alpha) = \begin{cases} v_{\mathbb{P}_{n-1}}(\mathbf{a}) & \text{se } \alpha = 0, \\ v_{\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}}(\mathbf{a}) & \text{se } \alpha = 1, \\ v_{\mathbb{B} \amalg \mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}}(\mathbf{a}) & \text{se } \alpha = 2, \\ \emptyset & \text{caso contrário.} \end{cases}$$

Em particular, $\Upsilon_{\mathbb{D}_n}(\emptyset) = \emptyset$.

5.1.3 Espaço Coerente $\bar{\mathbb{D}}_n$

Definição 5.2 *Seja $\bar{\mathbb{D}}_n \equiv \text{Coh}(\mathbb{D}_n)$ a família de subconjuntos coerentes construídos sobre o espaço coerente $\mathbb{D}_n \equiv ((\text{Coh}(\mathbb{D}_n), \subseteq)$. Neste caso, $\bar{\mathbb{D}}_n \equiv (\bar{\mathbb{D}}_n, \approx_{\bar{\mathbb{D}}_n})$ denota a teia cuja relação de coerência é dada por*

$$x \approx_{\bar{\mathbb{D}}_n} y \Leftrightarrow \Upsilon_{\mathbb{D}_n}(x) \cap \Upsilon_{\mathbb{D}_n}(y) = \emptyset,$$

sempre que $x, y \in \bar{\mathbb{D}}_n$. O espaço coerente cujos objetos são subfamílias de subconjuntos coerentes em \mathbb{D}_n é indicado por

$$\bar{\mathbb{D}}_n \equiv (\text{Coh}(\bar{\mathbb{D}}_n), \subseteq) \equiv (\text{Coh}(\bar{\mathbb{D}}_n, \approx_{\bar{\mathbb{D}}_n}), \subseteq).$$

Um token da teia $\bar{\mathbb{D}}_n \equiv (\bar{\mathbb{D}}_n, \approx_{\bar{\mathbb{D}}_n})$ é um subconjunto coerente, possivelmente infinito, da família $\text{Coh}(\mathbb{D}_n)$. Assim sendo, um subconjunto coerente em $\bar{\mathbb{D}}_n$ é um elemento do conjunto potência $\wp(\text{Coh}(\mathbb{D}_n))$ cujos tokens satisfazem a relação de coerência $\approx_{\bar{\mathbb{D}}_n}$. Por esta relação, dois tokens são coerentes se a intersecção dos conjuntos de índices associado a cada token é disjunta.

No que se refere a relação de ordem, a qual modela a aproximação de objetos em \mathbb{D}_n , tem-se que cada parte atômica de informação $\mathbf{a} \in \mathbb{D}_n$ determina uma aproximação finita da forma $\{\mathbf{a}\} \in \text{Coh}(\mathbb{D}_n)$.

Se $\mathbf{a} \in x \in \text{Coh}(\mathbb{D}_n)$, as seguintes condições são satisfeitas:

1. $\bar{\mathbf{a}} \in \bar{\mathbb{D}}_n$ e $\{\bar{\mathbf{a}}\} \in \text{Coh}(\bar{\mathbb{D}}_n) = \text{Coh}(\text{Coh}(\mathbb{D}_n))$; e
2. $x \in \bar{\mathbb{D}}_n$ e $\{x\} \in \text{Coh}(\bar{\mathbb{D}}_n) = \text{Coh}(\text{Coh}(\mathbb{D}_n))$;

Contudo, verifica-se que $\{\bar{\mathbf{a}}\} \not\subseteq \{x\}$.

A coerência estrita entre tokens da teia $\bar{\mathbb{D}}_n$, construída considerando-se a relação $\sim_{\bar{\mathbb{D}}_n}$, é responsável pela interpretação dos processos concorrentes que agora não são mais necessariamente elementares, como no primeiro nível da construção proposta. Caracteriza-se então a concorrência de processos que são executados em no máximo 2^n *utc*.

Desta forma, o conjunto $\bar{\mathcal{D}}_n = Coh(\mathbf{D}_n)$ formaliza a idéia de que o modelo de máquina que estamos definindo tem memória infinita. Sobre a teia \mathbf{D}_n se constroem as interpretações para o produto paralelo e a soma não determinística de processos, possivelmente infinitos em relação ao conjunto \mathcal{A} de ações e ao espaço geométrico I que os definem. Entretanto, tais processos são limitados em relação ao tempo de execução.

O espaço coerente dual $\bar{\mathbb{D}}_n^\perp$ é definido a partir da relação de incoerência $\approx_{\bar{\mathbf{D}}_n}$, que corresponde a negação da coerência estrita em $\bar{\mathbf{D}}_n$. Sobre a teia complementar $\bar{\mathbf{D}}_n^\perp$ são construídos os conjuntos coerentes que interpretam processos em conflito, e que podem ser executados em no máximo 2^n *utc*.

5.1.4 Espaço Coerente $\bar{\mathbb{D}}_n^\perp$

Definição 5.3 *Considere a teia $\bar{\mathbf{D}}_n^\perp \equiv (Coh(\bar{\mathbf{D}}_n^\perp) \equiv (\mathcal{D}_n, \approx_{\bar{\mathbf{D}}_n^\perp})$ onde a relação de coerência é dada por*

$$x \approx_{\bar{\mathbf{D}}_n^\perp} y \Leftrightarrow x = y \text{ ou } x \approx_{\bar{\mathbf{D}}_n} y \Leftrightarrow x = y \text{ ou } \Upsilon_{\mathbf{D}_n}(x) \cap \Upsilon_{\mathbf{D}_n}(y) \neq \emptyset.$$

A família de todos os subconjuntos coerente da teia $\bar{\mathbf{D}}_n^\perp$, ordenada pela inclusão, define o espaço coerente da soma não-determinística de processos executados em no máximo 2^n *utc*, indicado por

$$\bar{\mathbb{D}}_n^\perp \equiv ((Coh(\mathcal{D}_n, \approx_{\bar{\mathbf{D}}_n^\perp}), \subseteq) \equiv (Coh(\bar{\mathbf{D}}_n^\perp), \subseteq).$$

Seja o espaço coerente resultante da soma direta $\mathbb{P}_n = \mathbb{D}_n \amalg \bar{\mathbb{D}}_n \amalg \bar{\mathbb{D}}_n^\perp$. Definem-se os espaços coerentes:

1. $\mathbb{P}_n \amalg \mathbb{P}_n$, cujos objetos interpretam processos seqüenciais finitos, que serão executados em 2^n *utc*;
2. $\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$, onde são modelados os processos que executam, no máximo, n escolhas ou testes, interpretados pelos objetos em \mathbb{B} .

Portanto, o espaço coerente $\mathbb{D}_{n+1} \equiv (Coh(\mathbf{D}_{n+1}), \subseteq)$ é dado pela soma direta

$$\mathbb{D}_{n+1} \equiv \mathbb{P}_n \amalg (\mathbb{P}_n \amalg \mathbb{P}_n) \amalg (\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n).$$

A formalização destas definições é análoga aquelas apresentadas em 4.4, 4.5 e 4.6.

As próximas seções tornam explícito, através de imersões e projeções, os relacionamentos entre os espaços coerentes que compõem o nível $\mathbb{D}_n - \mathbb{D}_{n+1}$. Cada imersão determina um tipo de interpretação nos diferentes níveis da estrutura. Por conseqüência, a formalização da extensão deste modelo pela definição do limite inverso, conforme proposto por Scott em [SCO 72], é apresentada de diferentes maneiras, no Capítulo 6. Para alcançar este objetivo, consideram-se as correspondentes projeções associadas a cada uma das imersões.

5.2 Imersões e Projeções do Nível $\mathbb{D}_n - \mathbb{D}_{n+1}$

5.2.1 Imersões $\pi_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$

Intuitivamente, cada aplicação da imersão $\pi_n^{(001)}$ estende um braço (2^n) para a direita. Desta forma, todo objeto na imagem $\pi_n^{(001)}[\mathbb{D}_n]$ interpreta o prefixo de um processo seqüencial cujo tempo máximo de execução será 2^{n+1} *utc*.

Portanto, as imersões $\pi_n^{(001)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, além de permitirem a transformação dos objetos do nível anterior, caracterizando-os como processos seqüenciais parciais, pela indexação $(\alpha 01)$ dos tokens dos subconjuntos em \mathbb{D}_n , também permite a eliminação de indeterminações. Estas indeterminações são eliminadas pela união com subconjuntos indexados por $(\beta 11)$ em \mathbb{D}_{n+1} , sempre que $\alpha, \beta \in \{0, 1, 2\}$.

Uma análise semelhante pode ser feita com relação à imersão $\pi_n^{(011)}$. No produto direto, a união de tokens indexados por $(\alpha 01)$ interpreta a parte do processo que será primeiramente executada (prefixo). Conforme mostra a Figura 5.2, a segunda união começa após o traço-conexão central e introduz a parte do processo que será executado posteriormente. Neste caso, esta parte do processo é modelada pelos objetos na imagem $\pi_n^{(011)}[\mathbb{D}_n]$, interpretando o sufixo de um processo seqüencial cujo tempo máximo de execução será 2^{n+1} *utc*.

Por fim, no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ também é possível uma análise dos demais construtores, que se diferenciam do produto seqüencial. Tem-se que as transformações associadas às imagens pelas imersões $\pi_n^{(\theta)}$

(1). indicam a possibilidade de definição de 2^n termos numa soma determinística após a execução de n testes, se $\theta \in \{\alpha 02, \beta 12\}$; e

(2). correspondem ao operador inclusão, sempre que $\theta = \alpha 0$

As imersões $\pi_n^{(\theta)}$ são formalmente apresentadas na próxima definição.

Definição 5.4 *Considere os espaços coerentes $\mathbb{D}_n, \bar{\mathbb{D}}_n, \bar{\mathbb{D}}_n^\perp$ e as somas diretas $\mathbb{P}_n = \mathbb{D}_n \amalg \bar{\mathbb{D}}_n \amalg \bar{\mathbb{D}}_n^\perp$ e $\mathbb{D}_{n+1} = \mathbb{P}_n \amalg (\mathbb{P}_n \amalg \mathbb{P}_n) \amalg (\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n)$. Considere também os indexadores $\beta \in \{0, 1, 2\}$ e $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$. A expressão $\pi_n^{(\alpha)}(x) = \{\mathbf{a}_\alpha \mid \mathbf{a} \in x\}$ define as imersões*

- $\pi_n^{(\alpha)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ se $\beta = 0$,
- $\pi_n^{(\alpha)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ se $\beta = 1$, e
- $\pi_n^{(\alpha)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ se $\beta = 2$.

Em particular, $\pi_n^{(\alpha)}(\emptyset) = \emptyset$.

No texto que se segue, utilizam-se as notações indicadas nos próximos itens.

- Na indexação dos tokens de subconjuntos coerentes em $\pi_n^{(\theta)}[\mathbb{D}_n]$, utiliza-se a notação

$$\underbrace{\theta. \theta. \dots \theta}_{n \text{ vezes}} \equiv \theta^n.$$

- A composição finita das imersões $\pi_{m,n}^{(\theta)} : \mathbb{D}_m \rightarrow \mathbb{D}_{n+1}$ será indicada por

$$\pi_{m,n}^{(\theta)} = \begin{cases} id_{\mathbb{D}_n}, & \text{se } m \geq n \\ \pi_n^{(\theta)} \circ \dots \circ \pi_{m+1}^{(\theta)} \circ \pi_m^{(\theta)}, & \text{se } m < n. \end{cases}$$

Para cada indexador $\theta \in \{00, 001, 002\}$, a tabela na Figura 5.2, que usa notação acima, mostra a representação dos processos interpretados pelos subconjuntos coerentes na imagem da imersão $\pi_n^{(001)}[\mathbb{D}_n]$, aplicada sobre um subconjunto coerente unitário que interpreta um processo elementar. Nestas exemplificações, tem-se

$$\{d_{(001)^n}^{(k)}\} = \pi_{n-1}^{(\theta)} \circ \pi_{n-2}^{(\theta)} \circ \dots \circ \pi_1^{(\theta)} \circ \pi_0^{(\theta)} (\{d^{(k)}\}) = \pi_{0,n}^{(001)} (\{d^{(k)}\}).$$

Verifica-se facilmente que, a imersão $\pi_n^{(001)}$ transforma o subconjunto coerente $\{d^{(k)}\}$, interpretando o processo elementar $d^{(k)}$, no subconjunto coerente $\{d_{(001)^{n+1}}^{(k)}\}$, interpretando o produto seqüencial, parcial e finito, indicado pela expressão

$$\underbrace{(((d^{(k)} \cdot_{-2^0}) \cdot_{-2^1}) \cdot \dots \cdot_{-2^n})}_{n+1 \text{ vezes}} \equiv d^{(k)} \cdot_{-2^n}.$$

O intervalo de tempo entre o início e o término da execução de tal processo parcialmente determinado será, no máximo, 2^{n+1} . Contudo, o primeiro processo a ser executado neste produto é conhecido.

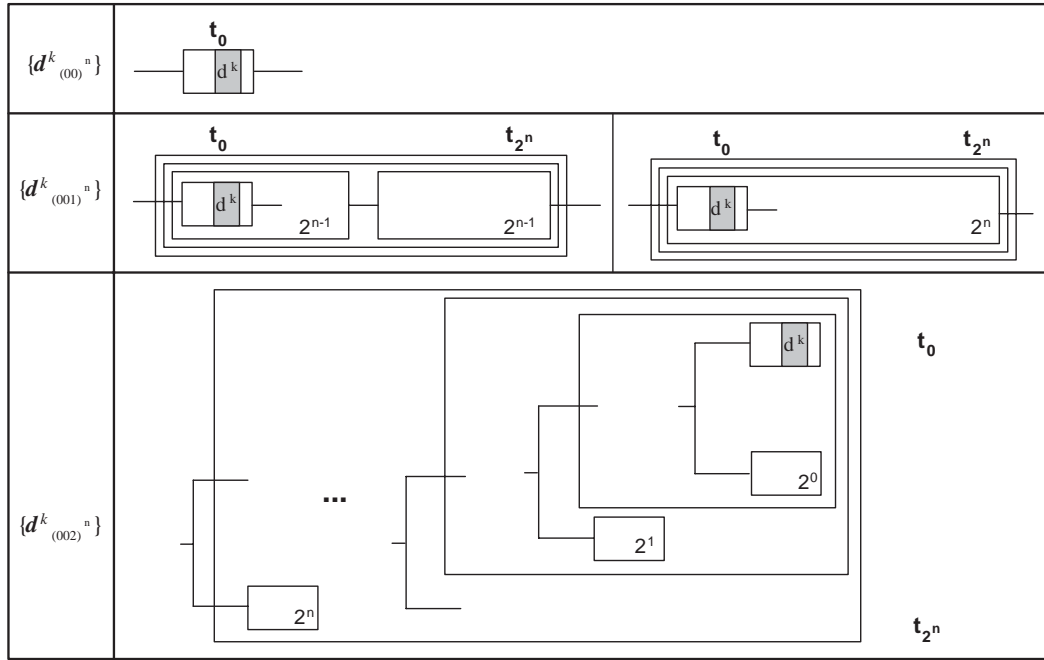


FIGURA 5.2 - Interpretação da Composição de Imersões em $\pi_n^\theta[\mathbb{D}_n] \subseteq \mathbb{D}_\infty$.

A seguir apresentam-se formalmente as imersões que modelam o relacionamento entre os espaços coerente que compõem o nível $\mathbb{D}_n - \mathbb{D}_{n+1}$.

Proposição 5.1 *Considere os domínios \mathbb{D}_n , $\bar{\mathbb{D}}_n$, $\bar{\mathbb{D}}_n^\perp$, \mathbb{P}_n , e \mathbb{D}_{n+1} , os indexadores $\theta \in \{0, 1\}$, $\beta \in \{0, 1, 2\}$ e $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$ e as imersões $\pi_n^{(\alpha)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ definidas acima. Considere também as funções lineares*

1. $\gamma_n^{(0)} : \mathbb{P}_n \rightarrow \mathbb{D}_{n+1}$, $\gamma_n^{(1)} : \mathbb{P}_n \amalg \mathbb{P}_n \rightarrow \mathbb{D}_{n+1}$ e $\gamma_n^{(2)} : \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n \rightarrow \mathbb{D}_{n+1}$, definidas pela expressão $\gamma_n^{(\beta)}(x) = \{\mathbf{a}_\alpha \mid a \in x\}$.
2. $\phi_n^{(0)} : \mathbb{D}_n \rightarrow \mathbb{P}_n$, $\phi_n^{(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{P}_n$, $\phi_n^{(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{P}_n$ e $\phi_n^{(\beta)}(x) = \{\mathbf{a}_\beta \mid a \in x\}$.
3. $\psi_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{P}_n \amalg \mathbb{P}_n$, dada por $\psi_n^{(\theta)}(x) = \{\mathbf{a}_\theta \mid a \in x\}$.
4. $\lambda_n^{(\theta)} : \mathbb{P}_n \rightarrow \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$, dada por $\lambda_n^{(\theta)}(x) = \{\mathbf{a}_\theta \mid a \in x\}$.

Para cada β , as seguintes igualdades são satisfeitas:

- $\pi_n^{(\beta 0)} = \gamma_n^{(0)} \circ \phi_n^{(\beta)}$;
- $\pi_n^{(\beta 01)} = \gamma_n^{(1)} \circ \psi_n^{(0)} \circ \phi_n^{(\beta)}$ e $\pi_n^{(\beta 11)} = \gamma_n^{(1)} \circ \psi_n^{(1)} \circ \phi_n^{(\beta)}$;
- $\pi_n^{(\beta 02)} = \gamma_n^{(2)} \circ \psi_n^{(0)} \circ \phi_n^{(\beta)}$ e $\pi_n^{(\beta 12)} = \gamma_n^{(2)} \circ \psi_n^{(1)} \circ \phi_n^{(\beta)}$.

A demonstração é imediata, considerando-se as funções apresentadas nos itens 1, 2, 3 e 4.

5.2.2 Projeções $\Pi_n^{(\theta)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$

A noção intuitiva relacionada com as projeções $\Pi_n^{(\theta)}$, formalizadas na Definição 5.5, também está fundamentada na análise da construção de um processo cuja execução está limitada a um intervalo de tempo. Conforme descrito em seções anteriores, este intervalo de tempo começa a ser definido logo no primeiro nível da construção indutiva das interpretações dos processos, junto com a escolha e a modelagem dos processos elementares - executados num instante de tempo. Portanto, os diferentes níveis da construção do espaço coerente de processos indutivamente gerados modelam a construção temporal do modelo de máquina geométrica.

Baseados nestas considerações, seja o subconjunto $x \in \mathbb{P}_n \amalg \mathbb{P}_n$, interpretando um produto seqüencial. A função $\Pi_n^{(001)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, aplicada sobre x , projeta como imagem o primeiro fator do produto seqüencial interpretado por x em $\mathbb{D}_n \amalg \mathbb{D}_n$. Ou ainda, $\Pi_n^{(001)}(x)$ interpreta o maior prefixo de x executado em 2^nutc .

Sendo assim, a projeção $\Pi_n^{(001)}$ retira o braço estendido para a direita pela imersão $\pi_n^{(001)}$, logo $\Pi_n^{(001)} \circ \pi_n^{(001)}(x) = x$. Os indexadores dos tokens que definem tal subconjunto-imagem são seqüências finitas de comprimento máximo 2^n .

O comportamento associado a outra função, $\Pi_n^{(011)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, é análogo considerando-se agora o segundo termo de cada objeto x do produto direto $\mathbb{P}_n \amalg \mathbb{P}_n$. Neste caso, a projeção $\Pi_n^{(011)}(x)$ interpreta o maior sufixo de x executado em 2^nutc . Assim, intuitivamente esta projeção retira o braço estendido para a esquerda pela imersão $\pi_n^{(011)}$, ou seja, $\Pi_n^{(011)} \circ \pi_n^{(011)}(x) = x$.

Em ambos os casos, os subconjuntos coerentes nas imagens $\Pi_n^{(001)}[\mathbb{D}_{n+1}]$ e $\Pi_n^{(011)}[\mathbb{D}_{n+1}]$ interpretam a redução ou a eliminação dos braços, pela esquerda ou pela direita, que caracterizam os processos seqüenciais. Entretanto, esta redução ou eliminação não ocorre simultaneamente.

Por outro lado, seja o subconjunto $x \in \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ interpretando uma soma determinística. A função $\Pi_n^{(002)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, aplicada sobre x , projeta como imagem o primeiro termo da soma interpretada por x . Intuitivamente, $\Pi_n^{(002)}(x)$ interpreta o processo que será executado quando for satisfeito o teste interpretado por $\Lambda_n(x) \in \Lambda_n[\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n]$.

Considerações semelhantes podem ser obtidas relacionando as interpretações na imagem $\Pi_n^{(012)}[\mathbb{D}_{n+1}]$ com escolhas determinísticas, quando os testes interpretados em $\Lambda_n(x) \in \Lambda_n[\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n]$ não são satisfeitos.

A definição formal destas projeções é apresentada logo a seguir.

Definição 5.5 *Considere os espaços coerentes \mathbb{D}_n , $\bar{\mathbb{D}}_n$, $\bar{\mathbb{D}}_n^\perp$, \mathbb{P}_n e \mathbb{D}_{n+1} e os indexadores $\beta \in \{0, 1, 2\}$ e $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$. As funções*

- $\Pi_n^{(\alpha)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ se $\beta = 0$,
- $\Pi_n^{(\alpha)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ se $\beta = 1$,

- $\Pi_n^{(\alpha)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ se $\beta = 2$,

são definidas pela expressão $\Pi_n^{(\alpha)}(x) = \{\mathbf{a} \mid \mathbf{a}_\alpha \in x\}$. Neste caso, $\Pi_n^{(\alpha)}(\emptyset) = \emptyset$.

As próximas proposições relacionam-se com as composições apresentadas nos itens 1, 2, 3 e 4 da proposição 5.1 e baseiam-se no fato de que *CospLin* é fechada para composição de funções lineares.

Proposição 5.2 *Considere os espaços coerentes $\mathbb{D}_n, \bar{\mathbb{D}}_n, \bar{\mathbb{D}}_n^\perp, \mathbb{P}_n$ e \mathbb{D}_{n+1} , os índices $\theta \in \{0, 1\}$, $\beta \in \{0, 1, 2\}$ e $\alpha \in \{\beta\} \times \{0, 1\} \times \{1, 2\} \cup \{\beta\} \times \{0\}$ e as projeções $\Pi_n^{(\alpha)}$ definidas acima. Além disso, considere as funções lineares*

1. $\Gamma_n^{(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{P}_n$, $\Gamma_n^{(1)} : \mathbb{D}_{n+1} \rightarrow \mathbb{P}_n \amalg \mathbb{P}_n$ e $\Gamma_n^{(2)} : \mathbb{D}_{n+1} \rightarrow \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_0$ definidas pela expressão $\Gamma_n^{(\beta)}(x) = \{\mathbf{a} \mid \mathbf{a}_\beta \in x\}$.
2. $\Phi_n^{(0)} : \mathbb{P}_n \rightarrow \mathbb{D}_n$, $\Phi_n^{(1)} : \mathbb{P}_n \rightarrow \bar{\mathbb{D}}_n$ e $\Phi_n^{(2)} : \mathbb{P}_n \rightarrow \bar{\mathbb{D}}_n^\perp$ onde $\Phi_n^{(\beta)}(x) = \{\mathbf{a} \mid \mathbf{a}_\beta \in x\}$.
3. $\Psi_n^{(\theta)} : \mathbb{P}_n \amalg \mathbb{P}_n \rightarrow \mathbb{P}_n$ dada por $\Psi_n^{(\theta)}(x) = \{\mathbf{a} \mid \mathbf{a}_\theta \in x\}$.
4. $\Lambda_n^{(\theta)} : \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n \rightarrow \mathbb{P}_n$, talque $\Lambda_n^{(\theta)}(x) = \{\mathbf{a} \mid \mathbf{a}_\theta \in x\}$.

As seguintes igualdades são satisfeitas:

- $\Pi_n^{(\beta 0)} = \Phi_n^{(\beta)} \circ \Gamma_n^{(0)}$,
- $\Pi_n^{(\beta 01)} = \Phi_n^{(\beta)} \circ \Psi_n^{(0)} \circ \Gamma_n^{(1)}$ e $\Pi_n^{(\beta 11)} = \Phi_n^{(\beta)} \circ \Psi_n^{(1)} \circ \Gamma_n^{(1)}$,
- $\Pi_n^{(\beta 02)} = \Phi_n^{(\beta)} \circ \Lambda_n^{(0)} \circ \Gamma_n^{(2)}$ e $\Pi_n^{(\beta 12)} = \Phi_n^{(\beta)} \circ \Lambda_n^{(1)} \circ \Gamma_n^{(2)}$.

Proposição 5.3 *As imersões e projeções apresentadas nas Definições 4.22 e 5.5 são morfismos na categoria *CospLin*.*

Proposição 5.4 *Para cada α explicitado na Definição 5.5, tem-se*

$$(\pi_n^{(\alpha)} \circ \Pi_n^{(\alpha)})[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1} \quad e \quad (\Pi_n^{(\alpha)} \circ \pi_n^{(\alpha)})[\mathbb{D}_n] = \mathbb{D}_n.$$

Para melhor compreensão das imersões e projeções definidas sobre cada um dos níveis da construção que se está propondo, se torna interessante, neste momento, a apresentação de alguns exemplos e suas correspondentes interpretações.

Dedica-se ainda, uma atenção especial para a construção gráfica destes exemplos, de forma que se tenha uma noção intuitiva de como a informação sobre os processos construídos está sendo armazenada.

5.2.3 Exemplificação da Aplicação das Funções $\pi_n^{(\theta)}$ e $\Pi_n^{(\theta)}$

Nesta exemplificação consideram-se diferentes processos seqüências definidos a partir de um mesmo conjunto de processos elementares, com diferentes interpretações em relação à construção dos processos nos diferentes níveis do modelo, mas preservando sempre a ordem de execução dos mesmos (ordem de construção).

Até a presente etapa de construção deste modelo, a interpretação se restringe a processos finitos em relação ao tempo de execução, por conseqüência, o mesmo se dá em relação às correspondentes aproximações.

Justifica-se, assim, a presença de retângulos vazios de diferentes tamanhos nas representações gráficas, interpretando processos parciais que podem sempre ser aproximados por processos finitos, de tamanho limitado.

Exemplificação 5.1

Este exemplo ilustra a construção dos objetos através de suas aproximações finitas. No domínio \mathbb{D}_2 , tem-se interpretação para todos os processos finitos (executados em no máximo 4 *utc*), inclusive as correspondentes aproximações, que podem conter “buracos” ou indeterminações de diferentes tamanhos (finitos), que identificam os diferentes níveis em que tais aproximações foram sendo definidas.

(1) Considere primeiramente, o subconjunto $x = \{d_{001.001}^{(k)}, e_{011.001}^{(l)}, f_{00.011}^{(n)}\} \in \mathbb{D}_2$ que interpreta o processo $(d^{(k)} \cdot e^{(l)}) \cdot (f^{(n)})$. A Figura 5.3 mostra a representação da construção do subconjunto coerente x a partir de seus tokens.

- Cada token do conjunto coerente x determina um conjunto coerente unitário que interpreta um processo. Neste sentido, as aproximações unitárias de x podem ser apresentadas como imagens pelas imersões. Ou ainda,

$$\begin{aligned} \{d_{001.001}^{(k)}\} &= \pi_1^{(001)} \circ \pi_0^{(001)} (\{d^{(k)}\}) = \pi_1^{(001)} (\{d_{(001)}^{(k)}\}), \\ \{e_{011.001}^{(l)}\} &= \pi_1^{(001)} \circ \pi_0^{(011)} (\{e^{(l)}\}) = \pi_1^{(001)} (\{e_{(011)}^{(l)}\}), \\ \{f_{00.011}^{(n)}\} &= \pi_1^{(011)} \circ \pi_0^{(00)} (\{f_{00.011}^{(n)}\}) = \pi_1^{(011)} (\{f_{00}^{(n)}\}), \end{aligned}$$

interpretando os respectivos processos parciais

$$(d^{(k)} \cdot -) \cdot {}_{-2} \quad , \quad (- \cdot e^{(l)}) \cdot {}_{-2} \quad \text{e} \quad {}_{-2} \cdot (f^{(n)}) \cdot {}_{-2}$$

- Da mesma forma as demais aproximações de x

$$\begin{aligned} \{d_{001.001}^{(k)}, e_{011.001}^{(l)}\} &= \pi_1^{(001)} (\{d_{001}^{(k)}, e_{011}^{(l)}\}), \\ \{d_{001.001}^{(k)}, f_{00.011}^{(n)}\} &= \{d_{001.001}^{(k)}\} \cup \{f_{00.011}^{(n)}\}, \\ \{e_{011.001}^{(l)}, f_{00.011}^{(n)}\} &= \{e_{011.001}^{(l)}\} \cup \{f_{00.011}^{(n)}\}, \end{aligned}$$

interpretam os processos respectivamente indicados por

$$(d^{(k)} \cdot e^{(l)}) \cdot {}_{-2} \quad , \quad (d^{(k)} \cdot -) \cdot (f^{(n)}) \quad \text{e} \quad (- \cdot e^{(l)}) \cdot (f^{(n)}) \cdot {}_{-2}$$

Salienta-se que os conjuntos coerentes $\{d_{001.001}^{(k)}, f_{00.011}^{(n)}\}$ e $\{e_{011.001}^{(l)}, f_{00.011}^{(n)}\}$ não são imagens destas imersões, são objetos construídos em \mathbb{D}_1 .

- Além disso, pelas projeções apresentadas nas Definições 5.5 e 5.10 tem-se que

$$\begin{aligned} \cdot x' &= \Pi_1^{(001)}(x) = \{d_{001}^{(k)}, e_{011}^{(l)}\} \text{ e } x'' = \Pi_1^{(011)}(x) = \{f_{00}^{(n)}\}; \\ \cdot \Pi_0^{(001)}(x') &= \{d^{(k)}\}, \Pi_0^{(001)}(x'') = \emptyset \text{ e } \Pi_0^{(00)}(x'') = \{f^{(n)}\}. \end{aligned}$$

(2) Sejam $n = 2$ e o conjunto coerente $s = \{d_{00.001}^{(k)}, e_{001.011}^{(l)}, f_{011.011}^{(n)}\} \in \mathbb{D}_2$ que interpreta o processo $(d^{(k)} \cdot (e^{(l)} \cdot f^{(n)}))$. Veja Figura 5.4.

- Considerando as projeções envolvendo \mathbb{D}_0 , \mathbb{D}_1 e \mathbb{D}_2 tem-se que

$$\begin{aligned} s' &= \Pi_2^{(001)}(s) = \Pi_{2,1}(s) = \{d_{00}^{(k)}\} \in \mathbb{D}_1 \quad \text{e} \\ s'' &= \Pi_1^{(00)}(s') = \Pi_{1,0}(s') = \{d^{(k)}\} \in \mathbb{D}_0. \end{aligned}$$

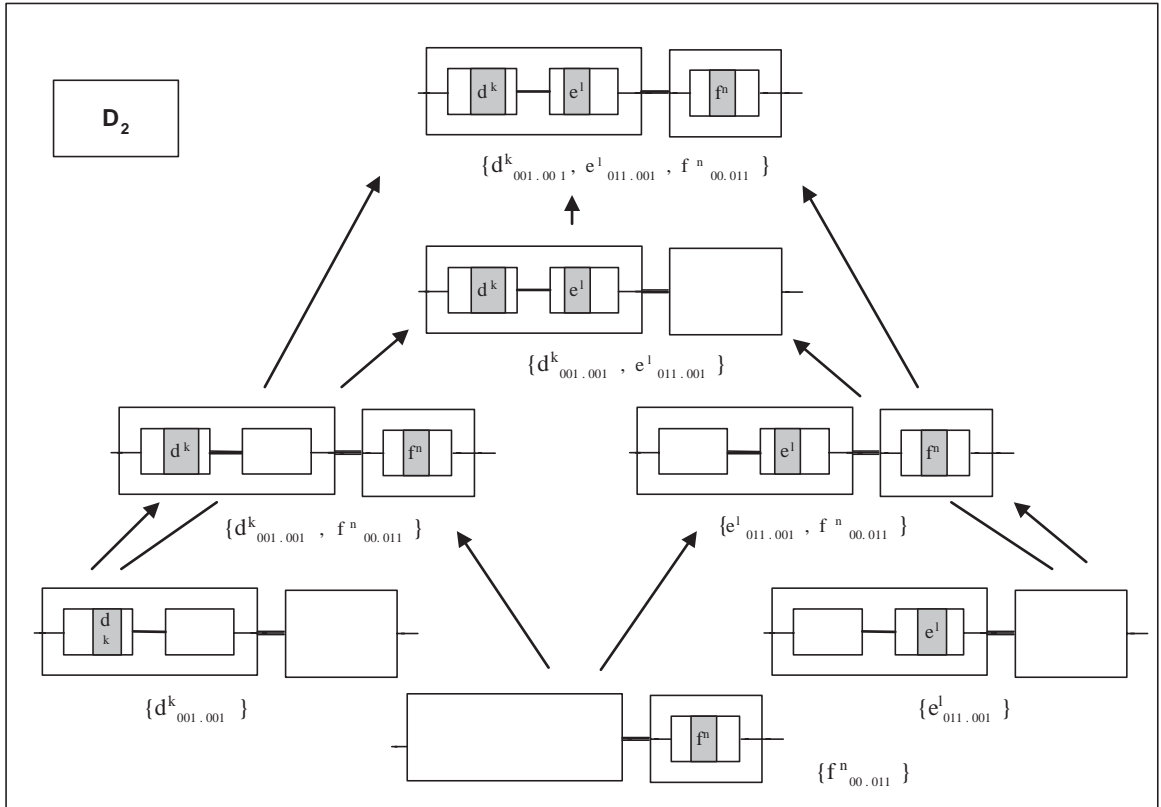


FIGURA 5.3 - Construção do Processo $(d^{(k)} \cdot e^{(l)}) \cdot (f^{(n)})$ em \mathbb{D}_2 .

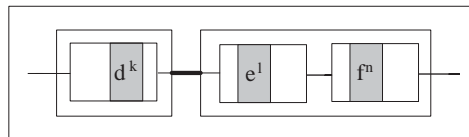


FIGURA 5.4 - Representação do Processo $(d^{(k)}) \cdot (e^{(l)}) \cdot f^{(n)}$.

- Pode-se observar ainda que algumas aproximações do conjunto coerente s podem ser expressas como imagens pelas imersões. Neste caso, tem-se as seguintes igualdades $\{d_{00.001}^{(k)}\} = \pi_1^{(001)} \circ \pi_0^{(00)}(\{d^{(k)}\})$, $\{e_{011.001}^{(l)}\} = \pi_1^{(001)} \circ \pi_0^{(011)}(\{e^{(l)}\})$ e $\{f_{00.011}^{(n)}\} = \pi_1^{(011)} \circ \pi_0^{(00)}(\{f^{(n)}\})$. Os conjuntos coerentes que resultam destas igualdades interpretam os processos parciais respectivamente indicados por

$$(d^{(k)} \cdot -) \cdot -_2, \quad (- \cdot e^{(l)}) \cdot -_2 \quad \text{e} \quad -_2 \cdot (f^{(n)}).$$

- Além disso, tem-se que $s''' = \{e_{001.011}^{(l)}, f_{011.011}^{(n)}\}$ é uma outra aproximação de s tal que $s''' = \pi_1^{(011)}(\{e_{001}^{(l)}, f_{011}^{(n)}\})$. Entretanto, o subconjunto coerente $\{d_{00.001}^{(k)}, e_{001.011}^{(l)}\}$ não pode ser expresso como uma imersão de \mathbb{D}_1 para \mathbb{D}_2 . Tal aproximação de s é construída em \mathbb{D}_2 .

Os conjuntos coerentes s e x , respectivamente representados nas Figuras 5.4 e 5.3 são objetos diferentes do mesmo domínio e, portanto, interpretam processos distintos. A semântica associada a estes objetos é que leva a uma igualdade que corresponderá à propriedade associativa, relativa ao produto seqüencial de processos. O mesmo pode ser observado com relação à Figura 5.5, no próximo exemplo.

(3) Seja $n = 3$ e $u = \{d_{00.001.001}^{(k)}, e_{00.011.001}^{(l)}, f_{00.00.011}^{(n)}\} \in \mathbb{D}_3$ o conjunto coerente que interpreta o processo $((d^{(k)}) \cdot (e^{(l)})) \cdot ((f^{(n)}))$. Na Figura 5.5, mostra-se a construção do espaço coerente u a partir de seus tokens, onde estão representadas as interpretações das próximas projeções. Verifica-se neste caso, as seguintes igualdades:

- $u' = \Pi_2^{(001)}(u) = \{d_{00.001}^{(k)}, e_{00.011}^{(l)}\} \in \mathbb{D}_2$ e $u'' = \Pi_1^{(001)}(u') = \{d_{00}^{(k)}\} \in \mathbb{D}_1$;
- $u''' = \Pi_0^{(00)}(u'') = \{d^{(k)}\} \in \mathbb{D}_0$.

Salienta-se entretanto que, $x \in \mathbb{D}_2$ não é uma aproximação para $u \in \mathbb{D}_3$. Isto ocorre porque a projeção de u , sobre \mathbb{D}_2 , não coincide com o conjunto coerente x . Pode-se dizer que x e u diferem em relação ao tempo em que as etapas que compõem cada um dos processos são executadas, embora sejam semanticamente equivalentes.

(4) Considere por fim, $v = \{d_{001.001.00}^{(k)}, e_{011.001.00}^{(l)}, f_{00.011.00}^{(n)}\} \in \mathbb{D}_3$ que interpreta o processo $((d^{(k)} \cdot e^{(l)}) \cdot (f^{(n)}))$, cuja construção está representada na Figura 5.6. Neste caso, v é a imersão de x em \mathbb{D}_3 .

- Tem-se que as projeções relacionadas com v são dadas por
 - $v' = \Pi_2^{(00)}(v) = \{d_{001.001}^{(k)}, e_{011.001}^{(l)}, f_{00.011}^{(n)}\} \in \mathbb{D}_2$,
 - $v'' = \Pi_1^{(001)}(v') = \{d_{001}^{(k)}, e_{011}^{(l)}\} \in \mathbb{D}_1$ e
 - $v''' = \Pi_0^{(011)}(v'') = \{d^{(k)}\} \in \mathbb{D}_0$ e $\Pi_0^{(001)}(v'') = \emptyset$.
- Nesta exemplificação os espaços coerentes u , v e x interpretam processos seqüenciais que, embora tenham a mesma interpretação semântica, possuem expressões distintas na linguagem porque são objetos distintos na estrutura do modelo.
- Considerando-se que u e v estão no mesmo nível no modelo, a árvore que descreve a construção de cada um destes objetos dentro do domínio \mathbb{D}_3 é diferente, sendo caracterizada por projeções distintas. Mostrou-se ainda que x é uma projeção de v pois $\Pi_{3,2}(v) = x$. Pela completção, x e v são ambas projeções de um mesmo objeto nos subníveis \mathbb{D}_2 e \mathbb{D}_3 , respectivamente.

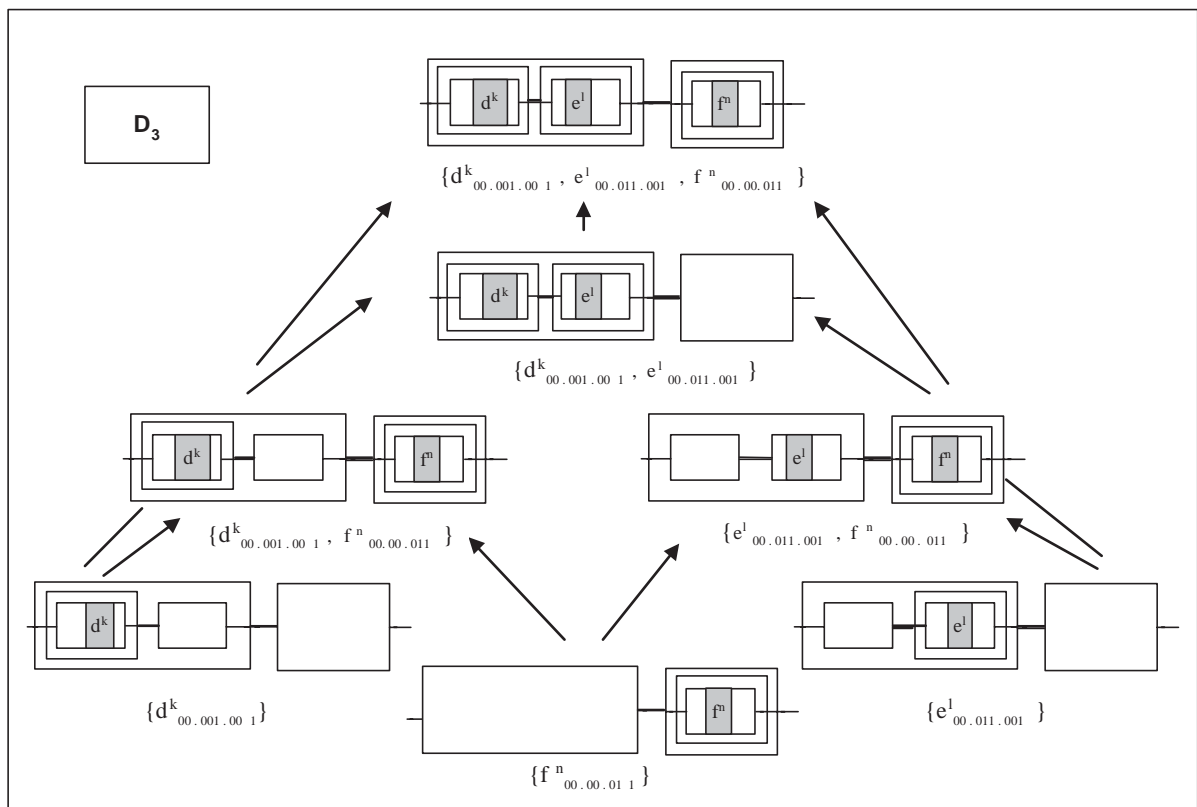


FIGURA 5.5 - Construção do Processo $((d^{(k)}) \cdot (e^{(l)})) \cdot ((f^{(n)}))$ em \mathbb{D}_3 .

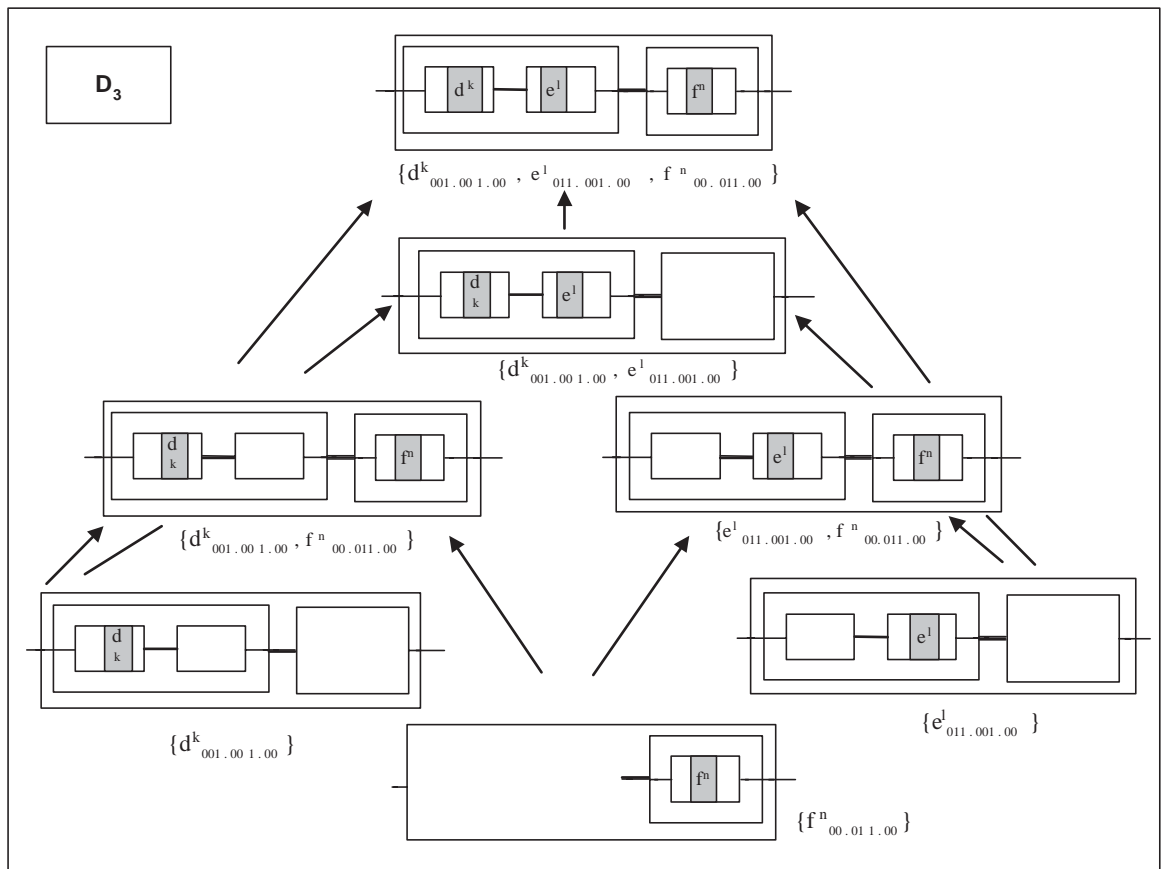


FIGURA 5.6 - Construção do Processo $((d^{(k)} \cdot e^{(l)}) \cdot (f^{(n)}))$ em \mathbb{D}_3 .

Por fim, deve-se salientar que as indeterminações de tamanho finito representadas por $_{-2^n}$ e interpretadas pelo conjunto vazio em \mathbb{D}_n podem ser substituídas por conjuntos coerentes construídos em níveis (ou subníveis) anteriores a \mathbb{D}_n . Neste sentido, $w = \{f_{00.011.00}^{(n)}\} \in \mathbb{D}_3$ é uma aproximação finita para todos os subconjuntos coerentes em \mathbb{D}_3 abaixo indicados:

$$\cdot \{f_{00.011.00}^{(n)}\}, \{d_{(001)^2.00}^{(k)}, f_{(00)^2.011}^{(n)}\}, \{e_{011.001.00}^{(l)}, f_{(00)^2.011}^{(n)}\}, \{d_{(001)^2.00}^{(k)}, e_{011.001.00}^{(l)}, f_{(00)^2.011}^{(n)}\}.$$

Exemplificação 5.2

Conjuntos coerentes que interpretam um produto paralelo ou uma soma de processos não-determinística são construídos pela união entre famílias de subconjuntos coerentes gerados no nível anterior da construção. Neste sentido, considere os subconjuntos coerentes $\{\overline{d_{001}^{(k)} d_{011.10}^{(k)}}, \overline{e_{001}^{(l)} e_{011.10}^{(l)}}\}$ e $\{\overline{d_{101.00}^{(k)}}, \overline{e_{101.00}^{(l)}}, \overline{d_{111.00}^{(k)}}, \overline{e_{111.00}^{(l)}}\}$ em \mathbb{D}_2 . Tais objetos interpretam processos parciais respectivamente indicados pelas expressões

$$\|_{-2}, d^{(k)} \cdot d^{(k)}, e^{(l)} \cdot e^{(l)},_{-2} \| \text{ e } (\|_{-}, d^{(k)}, e^{(l)},_{-} \| \cdot \|_{-}, d^{(k)}, e^{(l)},_{-} \|).$$

Seja $t = \{\overline{d_{001}^{(k)} d_{011.102}^{(k)}}, \overline{e_{001}^{(l)} e_{011.102}^{(l)}}, \overline{d_{101.012}^{(k)}}, \overline{e_{101.012}^{(l)}}, \overline{d_{111.012}^{(k)}}, \overline{e_{111.012}^{(l)}}, b_{22}\}$ o conjunto coerente em \mathbb{D}_3 que interpreta a soma determinística dada por

$$\|_{-2}, d^{(k)} \cdot d^{(k)}, e^{(l)} \cdot e^{(l)},_{-2} \| +_b (\|_{-}, d^{(k)}, e^{(l)},_{-} \| \cdot \|_{-}, d^{(k)}, e^{(l)},_{-} \|).$$

O processo parcial (finito) interpretado pelo subconjunto coerente t pode ser aproximado por outros processos, desde que indeterminações representadas pela expressão $_{-2^n}$ sejam substituídas por processos de igual ou menor tamanho.

Por outro lado, t também pode aproximar outros objetos, como por exemplo o espaço coerente $t' = t \times \{(011)\} \cup \{d_{00.00.00.001}^{(k)}\}$ em \mathbb{D}_4 que interpreta o processo P apresentado na Figura 5.7.

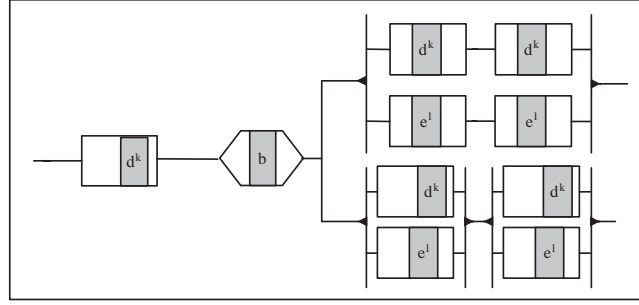
Supondo uma linguagem que transcreve as construções no domínio \mathbb{D}_4 tem-se que P é denotado pela expressão

$$(((d^{(k)}))) \cdot (\|_{-2}, d^{(k)} \cdot d^{(k)}, e^{(l)} \cdot e^{(l)},_{-2} \| +_b (\|_{-}, d^{(k)}, e^{(l)},_{-} \| \cdot \|_{-}, d^{(k)}, e^{(l)},_{-} \|)),$$

Neste caso, os processos que constroem o processo P serão interpretações dos subconjuntos coerentes obtidos por sucessivas aplicações das projeções, nos diferentes níveis da estrutura, conforme mostram os exemplos:

- $\Pi_{4,3}^{(001)}(t') = \{d_{00.00.00}^{(k)}\}$ e $\Pi_{4,3}^{(011)}(t') = t$;
- $\Pi_{4,2}^{(001)}(t') = \{d_{00.00}^{(k)}\}$ e $\Pi_{4,2}^{(011)}(t') = \emptyset$;
- $\Pi_{4,1}^{(001)}(t') = \{d_{00}^{(k)}\}$ e $\Pi_{4,1}^{(011)}(t') = \emptyset$;
- $\Pi_{4,0}^{(001)}(t') = \{d^{(k)}\}$ e $\Pi_{4,0}^{(011)}(t') = \emptyset$.

As próximas seções introduzem outros pares de funções imersão-projeção que também interpretam construtores de processos executados em tempo finito.

FIGURA 5.7 - Representação do Processo P .

5.3 Imersões π_n^{\rightarrow} , π_n^{\leftarrow} e Projeções Π_n^{\rightarrow} , Π_n^{\leftarrow} em $\mathbb{D}_n - \mathbb{D}_{n+1}$

Nesta seção são apresentadas as imersões $\pi_n^{\rightarrow}, \pi_n^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e as correspondentes projeções $\Pi_n^{\rightarrow}, \Pi_n^{\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, que interpretam construtores e destrutores de processos, respectivamente.

O limite para o diagrama definido a partir das projeções $\Pi_n^{\rightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ formaliza a completação da estrutura indutiva de um modelo de máquina geométrica capaz de interpretar, entre outros, os processos seqüenciais que são executados a partir de um tempo t_0 , mas que não possuem, necessariamente um tempo final.

Na construção dual, os limites para o diagrama definido a partir das projeções $\Pi_n^{\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ formalizam a completação da estrutura indutiva de um modelo de máquina geométrica capaz de interpretar, entre outros, os processos seqüenciais cuja execução é efetivada antes de um instante de tempo t_{-1} , mas que não possuem, necessariamente um tempo inicial.

Estas estruturas ordenadas, duais, são interpretadas pelos domínios $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$ apresentados no próximo capítulo. A notação é para lembrar a correspondência entre o conjunto $\mathcal{I}^{\rightarrow}$ dos indexadores dos tokens que definem as interpretações para processos seqüenciais em $\mathbb{D}_{\infty}^{\rightarrow}$ e o conjunto \mathbb{N} dos números naturais. Da mesma forma, o conjunto \mathcal{I}^{\leftarrow} dos indexadores dos tokens que definem as interpretações dos produtos seqüências em $\mathbb{D}_{\infty}^{\leftarrow}$ pode ser identificado pelo conjunto dos números inteiros-negativos. Neste caso, $\mathcal{I}^{\rightarrow} \equiv \mathbb{N} \equiv \mathcal{I}^{\leftarrow}$.

5.3.1 Imersões $\pi_n^{\leftarrow(\alpha)}$, $\pi_n^{\rightarrow(\alpha)}$

Apresentam-se aqui, as imersões $\pi_n^{\leftarrow(\alpha)}$, $\pi_n^{\rightarrow(\alpha)}$ referentes ao nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, cujas definições formais são obtidas pela análise de três casos. Considerando-se que o domínio \mathbb{D}_n é dado pela soma direta $\mathbb{P}_{n-1} \coprod (\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}) \coprod (\mathbb{B} \prod \mathbb{P}_{n-1} \prod \mathbb{P}_{n-1})$, justifica-se esta análise por casos na definição de cada construtor $\pi_n^{\leftarrow(\alpha)}$.

No subnível $\mathbb{D}_n - \mathbb{P}_n$, cada construtor interno está representado pelo indexador $\alpha \in \{0, 1, 2\}$, interpretando os operadores inclusão, produto paralelo ou soma não-determinística, respectivamente.

1. No subespaço $(\pi_n^{\rightarrow(0)} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}] \subseteq \pi_n^{(00)}[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1}$ são reconstruídas todas as interpretações de processos finitos, executados em 2^n *utc*. Do mesmo modo, em $(\pi_n^{\rightarrow(1)} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}] \subseteq \pi_n^{(10)}[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1}$ estão as interpretações para produtos paralelos entre os processos finitos, executados em 2^n *utc*. E por último, as somas não-determinísticas, executadas no máximo em 2^n *utc*, são interpretadas no subespaço $(\pi_n^{\rightarrow(2)} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}] \subseteq \pi_n^{(20)}[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1}$.

As considerações apresentadas no parágrafo anterior podem ser estendidas para as funções $\pi^{\leftarrow(\alpha)}$, pois a seguinte igualdade é sempre satisfeita:

$$(\pi_n^{\rightarrow(\alpha)} \circ \gamma_n^{(0)})[\mathbb{P}_n] = (\pi_n^{\leftarrow(\alpha)} \circ \gamma_n^{(0)})[\mathbb{P}_n].$$

2. Em $(\pi_n^{\rightarrow(\alpha)} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}] \subseteq \mathbb{D}_{n+1}$ estão as interpretações de prefixos de processos seqüenciais que podem ser executados em 2^{n+1} *utc*. Portanto,

$$(\pi_n^{\rightarrow(\alpha)} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}] \subset \pi_n^{(\alpha 01)}[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1}.$$

Intuitivamente, as imersões $\pi_n^{\rightarrow(\alpha)}$ e $\pi_n^{\leftarrow(\alpha)}$, quando aplicadas sobre subconjuntos que interpretam processos seqüenciais, constroem novos braços, para a direita e para a esquerda, respectivamente. Desta maneira, tais imersões sempre determinam objetos parciais em $\mathbb{P}_n \amalg \mathbb{P}_n$, onde a relação de coerência da teia $\mathbf{P}_n \amalg \mathbf{P}_n$ determina a possibilidade de união entre tais objetos para a construção de outras interpretações em \mathbb{D}_{n+1} .

Estes novos subconjuntos indexados nas imagens $(\pi_n^{\rightarrow(\alpha)} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}]$ e $(\pi_n^{\leftarrow(\alpha)} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}]$ interpretam os prefixos e sufixos dos novos processos, respectivamente.

3. No último caso, relacionado com as somas determinísticas, os novos subconjuntos nas imagens $(\pi_n^{\rightarrow(\alpha)} \circ \gamma_{n-1}^{(1)})[\mathbb{B} \amalg \mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}]$ e $(\pi_n^{\leftarrow(\alpha)} \circ \gamma_{n-1}^{(1)})[\mathbb{B} \amalg \mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}]$ representam a possibilidade de existência de um novo teste e todas as escolhas decorrentes do fato dele ser satisfeito. O número destas novas escolhas, para imersões no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ será, no máximo, igual a 2^n .

A Definição 5.6 formalmente introduz as imersões $\pi_n^{\rightarrow(\alpha)}$, $\pi_n^{\leftarrow(\beta)}$.

Definição 5.6 *Considere os espaços coerentes \mathbb{D}_n , \mathbb{P}_n , $\mathbb{P}_n \amalg \mathbb{P}_n$, $\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ e \mathbb{D}_{n+1} que definem o n -ésimo nível da estrutura deste modelo computacional. Considere também $\beta, \alpha \in \{0, 1, 2\}$ e as imersões $\gamma_n^{(\alpha)}$ e $\pi_n^{(\beta)}$, apresentadas na Proposição 5.1 e 5.4.*

As funções $\pi_n^{\rightarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\rightarrow(1)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\rightarrow(2)} : \mathbb{D}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas pela expressão

$$\pi_{n+1}^{\rightarrow(\alpha)}(x) = \begin{cases} \pi_{n+1}^{(\alpha 0)}(x) = \{\mathbf{a}_{\alpha 0} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(0)}[\mathbb{P}_n], \\ \pi_{n+1}^{(\alpha 01)}(x) = \{\mathbf{a}_{\alpha 01} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n], \\ \pi_{n+1}^{(\alpha 02)}(x) = \{\mathbf{a}_{\alpha 02} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(2)}[\mathbb{P}_n \amalg \mathbb{B} \amalg \mathbb{P}_n], \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Analogamente, as imersões $\pi_n^{\leftarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\leftarrow(1)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\leftarrow(2)} : \mathbb{D}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são formalizadas pela expressão

$$\pi_{n+1}^{\leftarrow(\beta)}(x) = \begin{cases} \pi_{n+1}^{(\beta 0)}(x) = \{\mathbf{a}_{\beta 0} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(0)}[\mathbb{P}_n], \\ \pi_{n+1}^{(\beta 11)}(x) = \{\mathbf{a}_{\beta 11} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n], \\ \pi_{n+1}^{(\beta 12)}(x) = \{\mathbf{a}_{\beta 12} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(2)}[\mathbb{P}_n \amalg \mathbb{B} \amalg \mathbb{P}_n], \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Quando $n = 0$, $\pi_0^{\rightarrow(\alpha)} = \pi_0^{\leftarrow(\alpha)} = \pi_0^{(\alpha 0)} = \{\mathbf{a}_{\alpha 0} \mid \mathbf{a} \in x \text{ e } \mathbf{a}_\alpha \in \mathbf{P}_0\}$.

Quando $\alpha = 0$, $\pi_n^{\rightarrow(0)} = \pi_n^{\leftarrow(0)}$ da mesma forma que $\pi_n^{\leftarrow(0)} = \pi_n^{\rightarrow(0)}$.

Em particular, $\pi_n^{\rightarrow(\alpha)}(\emptyset) = \pi_n^{\leftarrow(\alpha)}(\emptyset) = \emptyset$.

Proposição 5.5 As imersões $\pi_n^{\rightarrow(\alpha)}, \pi_n^{\leftarrow(\beta)}$ apresentadas na Definição 5.6 são morfismos na categoria $\text{Cosp}\mathcal{L}in$.

A demonstração da Proposição 5.5 pode ser construída considerando-se as restrições lineares das imersões $\pi_n^{(\alpha)}$.

A Figura 5.8 mostra algumas interpretações no subespaço $\pi_{n-1}^{\leftarrow}[\mathbb{D}_{n-1}] \subseteq \mathbb{D}_{n+1}$.

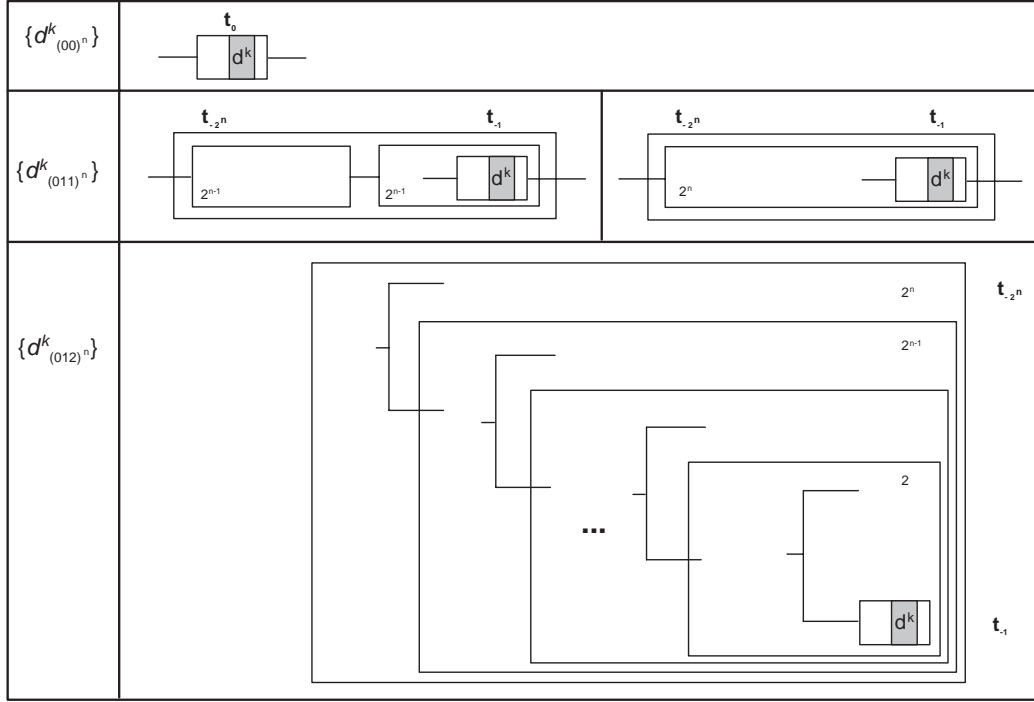


FIGURA 5.8 - Representação da Composição Finita da Imersão $\pi_{0,n-1}^{\leftarrow}[\mathbb{D}_0]$ em \mathbb{D}_n .

5.3.2 Projecções $\Pi_n^{\leftarrow(\alpha)}, \Pi_n^{\rightarrow(\alpha)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$

As próximas funções são extensões lineares das projecções $\Pi_0^{\leftrightarrow(\alpha)}$, apresentadas na Definição 5.7. Tais funções são definidas a partir das projecções $\Pi_n^{(\theta)}$, apresentadas na Definição 5.2.2.

Definição 5.7 Considere os espaços coerentes $\mathbb{D}_n, \mathbb{P}_n, \mathbb{P}_n \amalg \mathbb{P}_n, \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ e \mathbb{D}_{n+1} que definem o n -ésimo nível da estrutura deste modelo computacional. Considere também as imersões $\gamma_n^{(\beta)}, \lambda_n^{(\beta)}$ e $\psi_n^{(\beta)}$ apresentadas na Proposição 5.1 e as projecções $\Pi_n^{(\theta)}$ formalizadas na Definição 5.5, sempre que $\alpha \in \{0, 1, 2\}$.

As funções estritas $\Pi_n^{\rightarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n, \Pi_n^{\rightarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\rightarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela expressão

$$\Pi_n^{\rightarrow(\alpha)}(x) = \begin{cases} \Pi_n^{(\alpha 0)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\alpha 0} \in x\}, & \text{se } x \in \gamma_n^{(0)}[\mathbb{P}_n], \\ \Pi_n^{(\alpha 01)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\alpha 01} \in x\}, & \text{se } x \in \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n], \\ \Pi_n^{(\alpha 02)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\alpha 02} \in x\}, & \text{se } x \in \gamma_n^{(2)}[\mathbb{P}_n \amalg \mathbb{B} \amalg \mathbb{P}_n], \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, as projeções $\Pi_n^{\leftarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\leftarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\leftarrow(2)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n^\perp$ são definidas pela expressão

$$\Pi_n^{\leftarrow(\alpha)}(x) = \begin{cases} \Pi_n^{(\alpha 0)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\alpha 0} \in x\}, & \text{se } x \in \gamma_n^{(0)}, \\ \Pi_n^{(\alpha 11)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\alpha 11} \in x\}, & \text{se } x \in \gamma_n^{(1)}[\mathbb{P}_n \prod \mathbb{P}_n], \\ \Pi_n^{(\alpha 12)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\alpha 12} \in x\}, & \text{se } x \in \gamma_n^{(2)}[\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n], \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Quando $\alpha = 0$, $\Pi_n^{\rightarrow(0)} = \Pi_n^{\rightarrow}$ e $\Pi_n^{\leftarrow(0)} = \Pi_n^{\leftarrow}$.

Em particular, $\Pi_n^{\rightarrow(\alpha)}(\emptyset) = \Pi_n^{\leftarrow(\alpha)}(\emptyset) = \emptyset$.

Proposição 5.6 As projeções $\Pi_n^{\rightarrow(\alpha)}$, $\Pi_n^{\leftarrow(\alpha)}$ apresentadas na Definição 5.7 são morfismos na categoria CospLin .

A demonstração da Proposição 5.6 pode ser construída considerando-se as restrições lineares das projeções $\Pi_n^{(\alpha)}$.

Proposição 5.7 Considere os morfismos, na categoria CospLin , apresentados nas Definições 5.6 e 5.7. Tem-se as seguintes condições satisfeitas:

1. $\pi_n^{\rightarrow} \circ \Pi_n^{\rightarrow} \subseteq \mathbb{D}_{n+1}$ e $\Pi_n^{\rightarrow} \circ \pi_n^{\rightarrow} = \mathbb{D}_n$;
2. $\pi_n^{\rightarrow(1)} \circ \Pi_n^{\rightarrow(1)} \subseteq \mathbb{D}_{n+1}$ e $\Pi_n^{\rightarrow(1)} \circ \pi_n^{\rightarrow(1)} = \bar{\mathbb{D}}_n$;
3. $\pi_n^{\rightarrow(2)} \circ \Pi_n^{\rightarrow(2)} \subseteq \mathbb{D}_{n+1}$ e $\Pi_n^{\rightarrow(2)} \circ \pi_n^{\rightarrow(2)} = \mathbb{D}_n^\perp$.

As funções $\Pi_n^{\leftarrow(\alpha)}$ e $\pi_n^{\leftarrow(\alpha)}$ também satisfazem as condições apresentadas acima.

No texto que se segue, e em particular nos objetos representados na Figura 5.9, a composição finita das funções $f_n \in \{\Pi_n^{\rightarrow}, \Pi_n^{\leftarrow}, \pi_n^{\rightarrow}, \pi_n^{\leftarrow}\}$, no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, será indicada por $f_{n,m} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_m$ e definida pela expressão

$$f_{m,n} = \begin{cases} id_{\mathbb{D}_n}, & \text{se } m \leq n \\ f_m \circ f_{m-1} \circ \dots \circ f_{n+1} \circ f_n, & \text{se } m > n. \end{cases}$$

Conforme pode ser observado, nos próximos itens aplica-se a composição finita das projeções Π_n^{\rightarrow} e Π_n^{\leftarrow} , envolvendo interpretações construídas nos três primeiros níveis da estrutura e representadas na Figura 5.9.

- Seja x um objeto em \mathbb{D}_3 interpretando um processo seqüencial totalmente definido, dado pelo conjunto coerente

$$x = \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(k)}, d_{001.011.001}^{(k)}, d_{(011)2.001}^{(k)}, e_{(001)2.011}^{(l)}, e_{011.001.011}^{(l)}, e_{001.(011)2}^{(l)}, e_{(011)3}^{(l)}\}$$

cujas projeções são os subconjuntos coerentes

$$(1) \quad \Pi_2^{\rightarrow}(x) = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, d_{001.011}^{(k)}, d_{(011)2}^{(k)}\} \quad \text{e}$$

$$(2) \quad \Pi_2^{\leftarrow}(x) = \{e_{(001)2}^{(l)}, e_{011.001}^{(l)}, e_{001.011}^{(l)}, e_{(011)2}^{(l)}\}.$$

As correspondentes composições são dadas pelas igualdades

$$\bullet (\Pi_1^{\rightarrow} \circ \Pi_2^{\rightarrow})(x) = \Pi_{2,1}^{\rightarrow}(x) = \{d_{001}^{(k)}, d_{011}^{(k)}\};$$

$$\bullet (\Pi_1^{\leftarrow} \circ \Pi_2^{\leftarrow})(x) = \Pi_{2,1}^{\leftarrow}(x) = \{e_{001}^{(l)}, e_{011}^{(l)}\};$$

$$\bullet (\Pi_1^{\rightarrow} \circ \Pi_2^{\leftarrow})(x) = \{e_{001}^{(l)}, e_{011}^{(l)}\} \quad \text{e} \quad (\Pi_1^{\leftarrow} \circ \Pi_2^{\rightarrow})(x) = \{d_{001}^{(k)}, d_{011}^{(k)}\};$$

$$\bullet (\Pi_0^{\rightarrow} \circ \Pi_1^{\rightarrow} \circ \Pi_2^{\rightarrow})(x) = \Pi_{2,0}^{\rightarrow}(x) = \{d^{(k)}\};$$

$$\bullet (\Pi_0^{\leftarrow} \circ \Pi_1^{\leftarrow} \circ \Pi_2^{\leftarrow})(x) = \Pi_{2,0}^{\leftarrow}(x) = \{e^{(l)}\}.$$

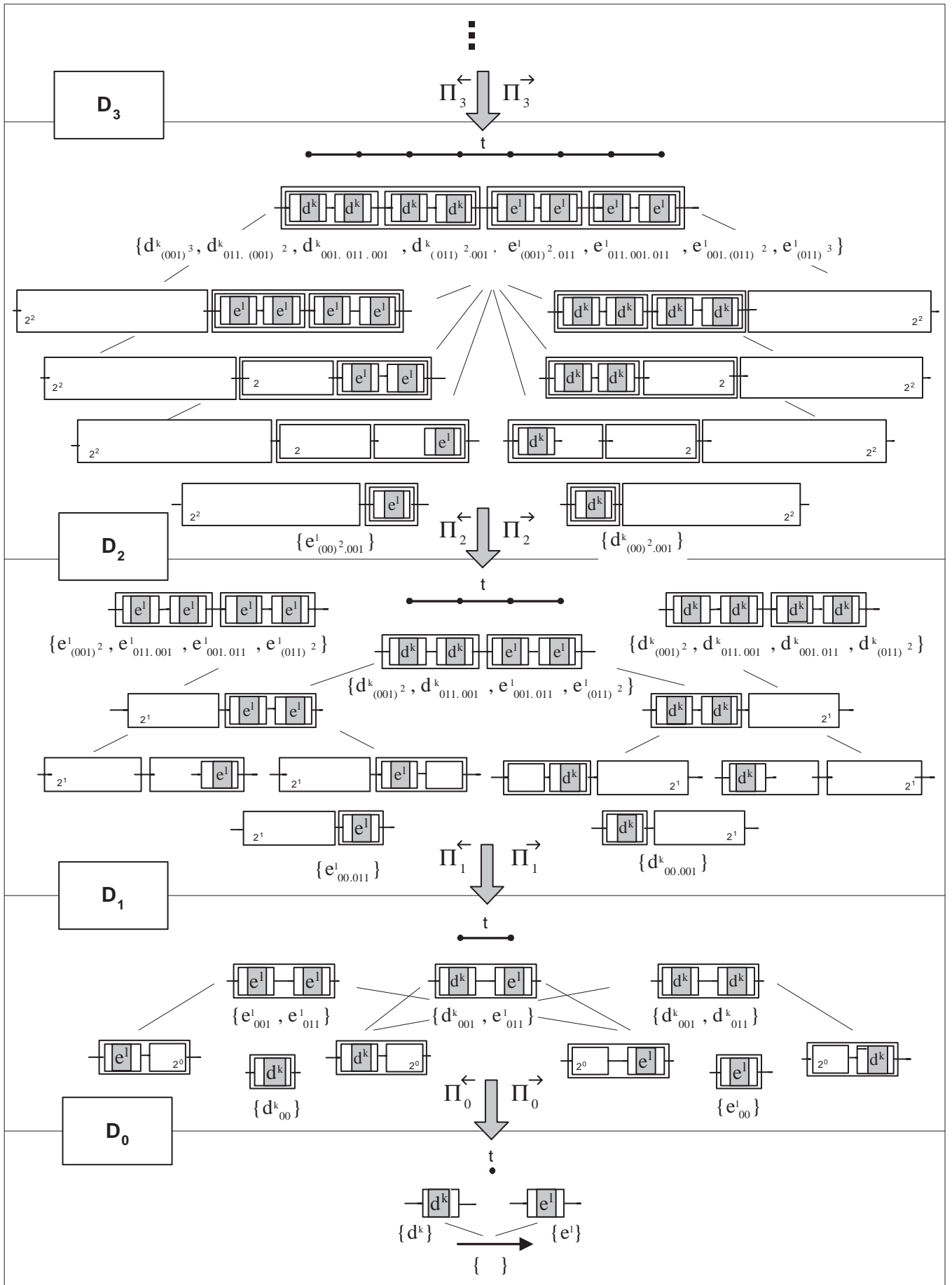


FIGURA 5.9 - Exemplicação das Projeções $\Pi_n^{\rightarrow}, \Pi_n^{\leftarrow}$ sobre \mathbb{D}_n .

- Considere agora $w = \{d_{(00)2.001}^{(k)}, d_{(011)3}^{(k)}, d_{001.(011)2}^{(k)}, d_{011.001.011}^{(k)}, d_{(001)2.011}^{(k)}\}$ e suas projeções $\Pi_2^{\rightarrow}(w) = \{d_{(00)2}^{(k)}\}$ e $\Pi_2^{\leftarrow}(w) = \{d_{(011)2}^{(k)}, d_{001.011}^{(k)}, d_{011.001}^{(k)}, d_{(001)2}^{(k)}\}$.

Algumas das possíveis composições são apresentadas logo a seguir

- $(\Pi_1^{\rightarrow} \circ \Pi_2^{\rightarrow})(w) = \Pi_{2,1}^{\rightarrow}(w) = \{d_{00}^{(k)}\} = (\Pi_1^{\leftarrow} \circ \Pi_2^{\rightarrow})(w)$;
- $(\Pi_1^{\rightarrow} \circ \Pi_2^{\leftarrow})(w) = \{d_{001}^{(k)}, d_{011}^{(k)}\} = \Pi_{2,1}^{\leftarrow}(w)$; e
- $\Pi_{2,0}^{\rightarrow}(x) = \{d^{(k)}\} = \Pi_{2,0}^{\leftarrow}(x)$.

5.4 Imersões e Projeções π_n^{\leftrightarrow} e Π_n^{\leftrightarrow}

Nesta seção são apresentadas as imersões $\pi_n^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e as correspondentes projeções $\Pi_n^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, que também interpretam construtores e destrutores de processos, respectivamente.

O limite para o diagrama definido a partir das projeções $\Pi_n^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ formalizam a completção da estrutura indutiva de um modelo de máquina geométrica capaz de interpretar, entre outros, os processos seqüenciais que não possuem início nem fim. Contudo, é sempre possível determinar os processos parciais que serão executados nos instantes t_{-1} e t_0 , ou em intervalos de tempo definidos a partir destes. Neste caso, na estrutura indutiva $\mathbb{D}_{\infty}^{\leftrightarrow}$ pode-se interpretar a construção de processos seqüenciais por infixação.

Com relação a seqüencialidade, pode-se pensar numa analogia entre o conjunto $\mathcal{I}^{\leftrightarrow}$ dos indexadores dos tokens das interpretações em $\mathbb{D}_{\infty}^{\leftrightarrow}$ e o conjunto dos números inteiros, ordenados na forma $\dots, -3, -2, -1, 0, 1, 2, 3, \dots$

5.4.1 Imersões $\pi_n^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$

As imersões π_n^{\leftrightarrow} mostram que a estrutura deste modelo de máquina geométrica é capaz de interpretar a aproximação de processos seqüências finitos que executam a construção por infixo. O subespaço $\pi_n^{\leftrightarrow}[\mathbb{D}_n]$ possui interpretação para processos parciais que não possuem prefixo e nem sufixo definidos, obtidos a partir de três casos.

1. Primeiramente, considere o caso em que $x \in (\phi_n^{(0)} \circ \gamma_n^{(0)})[\mathbb{D}_n]$. Tem-se então que, tanto x como sua imagem $\pi_n^{\leftrightarrow}(x) = \pi_n^{(00)}(x)$, interpretam o mesmo processo. Neste caso, a transformação associada a esta imersão coincide com o operador inclusão, o objeto é reconstruído em \mathbb{D}_{n+1} e seu conteúdo de informação preservado. O mesmo acontece quando $x \in (\phi_n^{(1)} \circ \gamma_n^{(0)})[\bar{\mathbb{D}}_n]$, ou ainda, quando $x \in (\phi_n^{(2)} \circ \gamma_n^{(0)})[\bar{\mathbb{D}}_n^{\perp}]$.

Nos próximos parágrafos são considerados os outros casos.

2. Intuitivamente, a imersão π_n^{\leftrightarrow} , quando aplicada sobre subconjuntos que interpretam processos seqüenciais, constrói braços opostos (para a direita e para a esquerda) de forma simultânea, possibilitando a união com novos subconjuntos e construindo outras interpretações. Estes novos subconjuntos, que não estão no subespaço $\pi_n^{\leftrightarrow}[\mathbb{D}_n]$, interpretarão os prefixos e sufixos dos novos processos, dependendo dos índices de seus tokens.

De acordo com o parágrafo anterior, a seqüencialidade é interpretada nos domínios \mathbb{D}_n , por uma ordem de construção temporal, explicitada nos objetos pela indexação dos seus tokens. Neste sentido, um novo objeto na imagem $\pi_n^{\leftrightarrow}[\mathbb{D}_{n-1} \prod \mathbb{D}_{n-1}] \subseteq \mathbb{D}_{n+1}$ interpreta um processo seqüencial parcial, sem prefixo nem sufixo definidos, com a informação adicional do tempo em que a parte determinada deste processo será executada.

3. No caso das somas determinísticas, os novos subconjuntos resultantes das imersões π_n^{\leftrightarrow} representam a possibilidade de existência de um novo teste e todas as escolhas decorrentes dele. O número destas novas escolhas, para imersões no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ será no máximo igual a 2^n . Assim, a indexação dos tokens dos subconjuntos em $\pi_n^{\leftrightarrow}[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n]$ indicam o número de escolhas possíveis associadas a cada interpretação.

Com o objetivo de simplificar a notação nas próximas seções, destes capítulo, as seguintes equivalências serão utilizadas

$$\gamma_n^{(0)} \equiv \gamma_n^{(0)}[\mathbb{P}_n], \quad \gamma_n^{(1)} \equiv \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n] \quad \text{e} \quad \gamma_n^{(2)} \equiv \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n].$$

As Definições 5.8 e 5.9 formalmente introduzem as imersões $\pi_n^{\leftarrow(\beta)}, \pi_n^{\rightarrow(\beta)}$, sempre que $\beta \in \{0, 1, 2\}$. A partir destas imersões, a função $\pi_n^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ é definida.

Definição 5.8 *Considere os espaços coerentes $\mathbb{D}_n, \mathbb{P}_n, \mathbb{P}_n \amalg \mathbb{P}_n, \mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ e \mathbb{D}_{n+1} que definem o n -ésimo nível da estrutura deste modelo computacional. Considere também as imersões $\gamma_n^{(\alpha)}, \lambda_n^{(\beta)}$ e $\psi_n^{(\beta)}$ apresentadas na Proposição 5.1 e as projeções $\Pi_n^{(\theta)}, \Gamma_n^{(\alpha)}, \lambda_n^{(\beta)}$ e $\Psi_n^{(\alpha)}$ formalizadas na Definição 5.5, sempre que $\beta, \alpha \in \{0, 1, 2\}$ e $\theta \in \{00, 001, 011, 002, 012\}$.*

As funções $\pi_n^{\leftarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}, \pi_n^{\leftarrow(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}, \pi_n^{\leftarrow(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas pela expressão

$$\pi_n^{\leftarrow(\alpha)}(x) = \begin{cases} \pi_{n+1}^{(\alpha 0)}(x) = \{\mathbf{a}_{\alpha 0} \mid \mathbf{a} \in x, \mathbf{a} \in \mathbb{D}_{n+1}\}, & \text{se } x \in \gamma_n^{(0)}, \\ (\pi_{n+1}^{(\alpha 01)} \circ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}))(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\pi_{n+1}^{(\alpha 02)} \circ (\gamma_n^{(2)} \circ \lambda_n^{(1)}) \circ (\Lambda_n^{(0)} \circ \Gamma_n^{(2)}))(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Analogamente, $\pi_n^{\rightarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}, \pi_n^{\rightarrow(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}, \pi_n^{\rightarrow(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são formalizadas pela expressão

$$\pi_n^{\rightarrow(\beta)}(x) = \begin{cases} \pi_{n+1}^{(\beta 0)}(x) = \{\mathbf{a}_{\beta 0} \mid \mathbf{a} \in x, \mathbf{a} \in \mathbb{D}_n\}, & \text{se } x \in \gamma_n^{(0)}, \\ (\pi_{n+1}^{(\beta 11)} \circ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}))(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\pi_{n+1}^{(\beta 12)} \circ (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}))(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Quando $n = 0$, $\pi_n^{\leftarrow(\beta)} = \pi_n^{\rightarrow(\beta)} = \pi_0^{(\beta 0)}$, definida em 5.4.

Em particular, $\pi_n^{\leftarrow(\beta)}(\emptyset) = \pi_n^{\rightarrow(\beta)}(\emptyset) = \emptyset, \forall n \in \omega$.

Definição 5.9 *Considere os espaços coerentes \mathbb{D}_n e \mathbb{D}_{n+1} e as funções estritas, lineares $\pi_n^{\leftarrow(\alpha)}, \pi_n^{\rightarrow(\beta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ apresentadas na Definição 5.8. As imersões $\pi_n^{\leftrightarrow(\alpha\beta)}$ no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ são definidas pela união de subconjuntos coerentes, explícita na próxima expressão*

$$\pi_n^{\leftrightarrow(\alpha)}(x) = \pi_n^{\leftarrow(\alpha)}(x) \cup \pi_n^{\rightarrow(\alpha)}(x).$$

Quando $\alpha = 0$, $\pi_n^{\leftrightarrow(\alpha)} = \pi_n^{\leftrightarrow}$.

Proposição 5.8 *As imersões estritas $\pi_n^{\leftarrow(\alpha)}, \pi_n^{\rightarrow(\beta)}$ e $\pi_n^{\leftrightarrow(\beta)}$ respectivamente apresentadas nas Definições 5.8, 5.9 são morfismos na categoria CospLin .*

A demonstração da proposição 5.11 será omitida. Entretanto, pode ser construída de forma análoga às provas das Proposições 5.9 e 5.10. Tais proposições se referem às projeções Π_n^{\leftrightarrow} , do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, na estrutura do modelo de máquina que estamos apresentando, e serão definidas logo a seguir.

No texto que se segue, adota-se a seguinte notação:

1. Na indexação dos tokens de subconjuntos em $\pi_n[\mathbb{D}_n]$, utiliza-se a notação

$$\cdot \underbrace{\theta . \theta . \dots . \theta}_{n \text{ vezes}} . \alpha \equiv \theta^n . \alpha \quad .$$

2. A composição finita das imersões $\pi_{m,n} : \mathbb{D}_m \rightarrow \mathbb{D}_{n+1}$ será indicada por

$$\cdot \pi_{m,n}^{\leftrightarrow} = \begin{cases} id_{\mathbb{D}_n}, & \text{se } m \geq n, \\ \pi_n^{\leftrightarrow} \circ \dots \circ \pi_{m+1}^{\leftrightarrow} \circ \pi_m^{\leftrightarrow}, & \text{se } m < n. \end{cases}$$

Em particular, $\pi_{0,n}^{\leftrightarrow} : \mathbb{D}_0 \rightarrow \mathbb{D}_{n+1}$ é indicada por $\pi_{0,n}^{\leftrightarrow} = \pi_n^{\leftrightarrow} \circ \dots \circ \pi_1^{\leftrightarrow} \circ \pi_0^{\leftrightarrow}$.

A tabela na Figura 5.10, construída de acordo com a notação sugerida logo a seguir, mostra a representação de alguns processos interpretados por subconjuntos coerentes na imagem da composição $\pi_{0,n}^{\leftrightarrow}[\mathbb{D}_0] \subseteq \mathbb{D}_{n+1}$.

Algumas das representações da Figura 5.10 se referem às seguintes imersões:

- $\pi_{0,n}^{\leftarrow}(\{d^{(k)}\}) = \{d_{(00)^{n+1}}^{(k)}\} = \pi_{0,n}^{\rightarrow}(\{d^{(k)}\}) = \pi_{0,n}^{\leftrightarrow}(\{d^{(k)}\})$;
- $\pi_{2,n}^{\leftarrow}(\{d_{00.001}^{(k)}\}) = \{d_{00.(011)^{n-1}.001}^{(k)}\}$ e $\pi_{2,n}^{\rightarrow}(\{d_{00.001}^{(k)}\}) = \emptyset$;
- $\pi_{2,n}^{\rightarrow}(\{d_{00.011}^{(k)}\}) = \{d_{00.(001)^{n-1}.011}^{(k)}\}$ e $\pi_{2,n}^{\leftarrow}(\{d_{00.011}^{(k)}\}) = \emptyset$.

Logo, tem-se $\pi_{2,n}^{\leftrightarrow}(\{d_{00.001}^{(k)}, d_{00.011}^{(k)}\}) = \{d_{00.(011)^{n-1}.001}^{(k)}, d_{00.(001)^{n-1}.011}^{(k)}\}$.

Considerando-se agora algumas projeções, como

- $\pi_{1,n}^{\leftarrow}(\{d_{011}^{(k)}\}) = \{d_{(001)^n.011}^{(k)}\}$ e $\pi_{1,n}^{\rightarrow}(\{d_{011}^{(k)}\}) = \emptyset$;
 - $\pi_{1,n}^{\leftarrow}(\{d_{001}^{(k)}\}) = \{d_{(011)^n.001}^{(k)}\}$ e $\pi_{1,n}^{\rightarrow}(\{d_{011}^{(k)}\}) = \emptyset$;
 - $\pi_{1,n}^{\leftrightarrow}(\{d_{001}^{(k)}\}) = \{d_{(011)^n.001}^{(k)}\}$ e $\pi_{1,n}^{\leftrightarrow}(\{d_{011}^{(k)}\}) = \{d_{(001)^n.011}^{(k)}\}$;
- obtém-se então $\pi_{1,n}^{\leftrightarrow}(\{d_{001}^{(k)}, d_{011}^{(k)}\}) = \{d_{(011)^n.001}^{(k)}, d_{(001)^n.011}^{(k)}\}$.

Seja $y = \{d_{002}^{(k)}, d_{012}^{(k)}\} \in \mathbb{D}_1$. Então, pela composição finita de imersões tem-se

- $\pi_{1,n}^{\leftrightarrow}(y) = \{d_{(012)^n.002}^{(k)}, d_{(002)^n.012}^{(k)}\}$ e
- $\{b_{22}, d_{(012)^n.002}^{(k)}, d_{(002)^n.012}^{(k)}\} = \{b_{22}\} \cup \pi_{1,n}(y) \in \mathbb{D}_{n+1}$.

5.4.2 Projeções $\Pi_n^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$

Assim como ocorre nas imersões $\pi_n^{(\alpha\beta)}$, as imagens determinadas pelas projeções Π_n^{\leftrightarrow} serão definidas como a união das imagens obtidas pelas funções lineares Π_n^{\leftarrow} e Π_n^{\rightarrow} . Isto está formalizado nas Definições 5.10 e 5.11.

A noção intuitiva relacionada com as projeções Π_n baseia-se também na análise da construção de um processo num intervalo finito de tempo.

Neste sentido, considere um subconjunto coerente $x \in \mathbb{D}_{n+1}$ que interpreta um produto sequencial executado em, no máximo 2^{n+1} *utc*. Neste caso, a projeção $\Pi_n^{\leftrightarrow}[\mathbb{P}_n \prod \mathbb{P}_n]$ interpreta a redução ou eliminação, simultâneamente, de ambos braços do produto sequencial interpretado pelo conjunto coerente x .

A seguir é formalizada a definição das projeções $\Pi_n^{\leftarrow}, \Pi_n^{\rightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$.

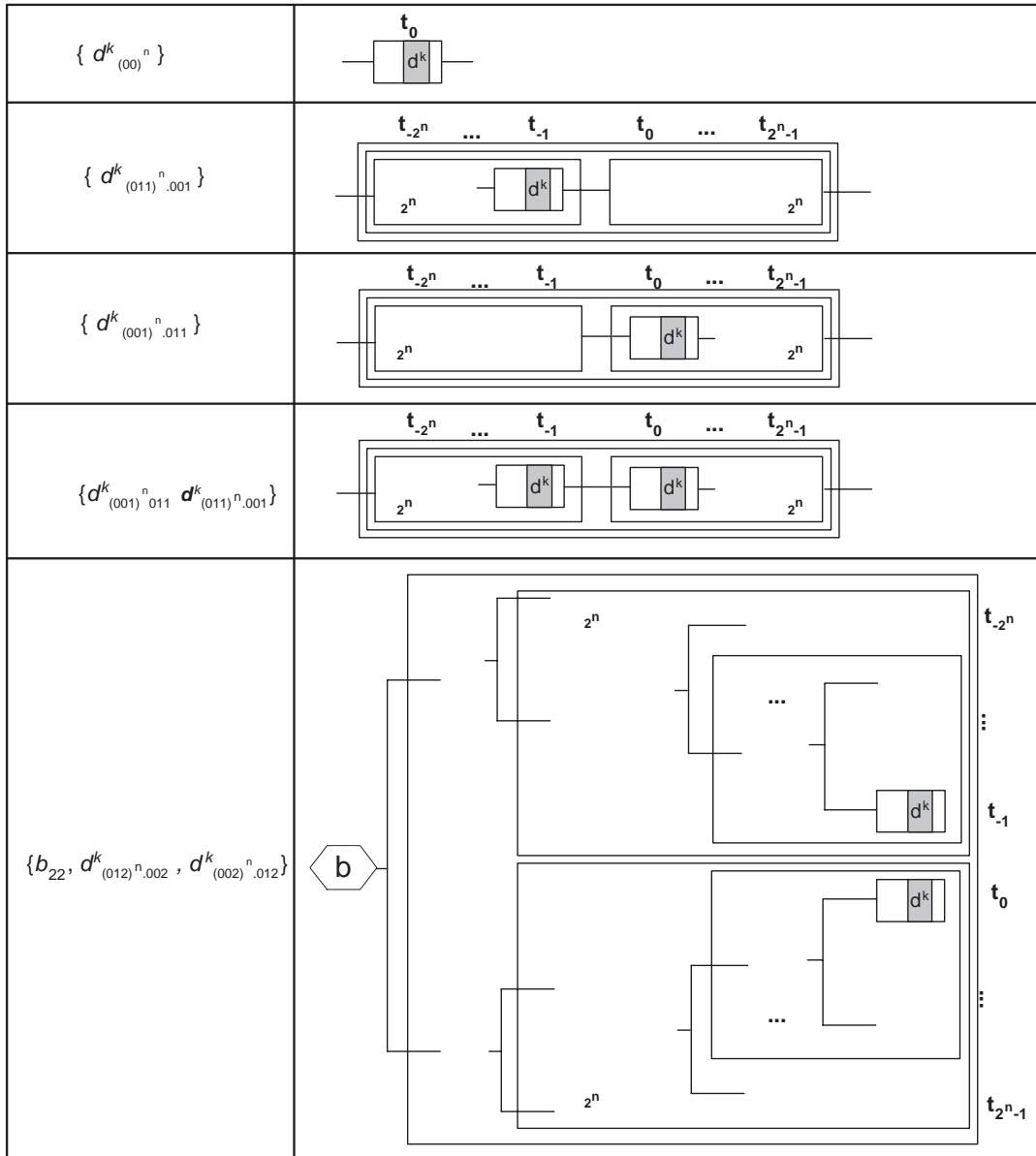


FIGURA 5.10 - Representação de Objetos em $\pi_{0,n}^{\leftrightarrow}[\mathbb{D}_n]$.

Definição 5.10 Considere os espaços coerentes \mathbb{D}_n , \mathbb{P}_n , $\mathbb{P}_n \amalg \mathbb{P}_n$, $\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ e \mathbb{D}_{n+1} que definem o n -ésimo nível da estrutura deste modelo computacional. Considere também as imersões $\gamma_n^{(\alpha)}$, $\lambda_n^{(\beta)}$ e $\psi_n^{(\beta)}$ apresentadas na Proposição 5.1 e as projeções $\Pi_n^{(\theta)}$, $\Gamma_n^{(\alpha)}$, $\lambda_n^{(\beta)}$ e $\Psi_n^{(\alpha)}$ formalizadas na Definição 5.5, sempre que $\beta \in \{0, 1\}$, $\alpha \in \{1, 2\}$ e $\theta \in \{00, 001, 011, 002, 012\}$.

As funções estritas $\Pi_n^{\leftarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\leftarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\leftarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela expressão

$$\Pi_n^{\leftarrow(\alpha)}(x) = \begin{cases} \Pi_n^{(\alpha 0)}(x) = \{\mathbf{a} \mid \mathbf{a}_{00} \in x, \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(\alpha 01)}(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(\alpha 02)}(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, $\Pi_n^{\rightarrow(\beta)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\rightarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\rightarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são dadas por

$$\Pi_n^{\rightarrow(\beta)}(x) = \begin{cases} \Pi_n^{(\beta 0)}(x) = \{\mathbf{a} \mid \mathbf{a}_{00} \in x, \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(\beta 11)}(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \lambda_n^{(1)}) \circ (\Lambda_n^{(0)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(\beta 12)}(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Em particular, quando $\alpha = \beta = 0$, $\Pi_n^{\leftarrow(\alpha)} = \Pi_n^{\leftarrow}$ e da mesma forma $\Pi_n^{\rightarrow(\beta)} = \Pi_n^{\rightarrow}$.

Proposição 5.9 As funções estritas $\Pi_n^{\leftarrow(\alpha)}$, $\Pi_n^{\rightarrow(\beta)}$ do nível $\mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas na Definição 5.10 são morfismos na categoria $\mathcal{CospLin}$.

Demonstração. Apresenta-se a seguir, a demonstração de que as funções $\Pi_n^{\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, na Definição 5.10, são funções lineares. Esta prova baseia-se nas restrições lineares definidas sobre as funções $\Pi_n^{(\alpha)}$, na Definição 5.1, sempre que $\beta \in \{0, 1, 2\}$ e $\alpha \in (\{\beta\} \times \{0, 1\} \times \{1, 2\}) \cup (\{\beta\} \times \{0\})$, podendo ser facilmente estendida para as demais projeções da Definição 5.10.

Primeiramente, observa-se que a função Π_n^{\leftarrow} está bem definida, considerando-se que o espaço coerente \mathbb{D}_{n+1} é definido por uma soma direta, no caso $\mathbb{D}_{n+1} = \mathbb{P}_n \amalg (\mathbb{P}_n \amalg \mathbb{P}_n) \amalg (\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n)$, de acordo com a Definição 4.22.

Agora, para mostrar que a função Π_n^{\leftarrow} é um morfismo em $\mathcal{CospLin}$, consideram-se a função linear $\Pi_n^{(00)}$ e os conjuntos

$$F_1 = \{\Pi_n^{(001)}, \Gamma_n^{(1)}, \Psi_n^{(0)}, \psi_n^{(0)}, \gamma_n^{(1)}\} \text{ e } F_2 = \{\Pi_n^{(002)}, \Gamma_n^{(2)}, \Lambda_n^{(1)}, \lambda_n^{(0)}, \gamma_n^{(2)}\}$$

de funções (projeção e imersão) lineares.

1. Sejam $x, x' \in \mathbb{D}_{n+1}$ tais que $x \subseteq x'$. Se $x = \emptyset$ tem-se $\Pi_n^{\leftarrow}(\emptyset) = \emptyset \subseteq \Pi_n^{\rightarrow}(x'), \forall x' \in \mathbb{D}_{n+1}$. Caso contrário, se $x \neq \emptyset$ e além disso tem-se

- (a) $x, x' \in \gamma_n^{(0)}[\mathbb{P}_n]$ e $\Pi_n^{(00)}$ é monótona, então $\Pi_n^{\leftarrow}(x) = \{\mathbf{a} \mid \mathbf{a}_{00} \in x\} \subseteq \{\mathbf{a}' \mid \mathbf{a}'_{00} \in x'\} = \Pi_n^{\leftarrow}(x')$.

- (b) $x, x' \in \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n]$, $n \neq 0$ e a monotonicidade das funções em F_1 , então $\Pi_n^{\leftarrow}(x') = (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)}(x') \subseteq (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)}(x) = \Pi_n^{\leftarrow}(x)$. Em particular, se $n = 0$ tem-se $\Pi_n^{\leftarrow}(x') = \Pi_n^{\leftarrow}(x) = \emptyset$.

- (c) $x, x' \in \gamma_n^{(2)}[\mathbb{B} \prod \mathbb{P}_n \prod \mathbb{P}_0]$, $n \neq 0$ e consideram-se as funções monótonas em F_2 , então $\Pi_n^{\leftarrow}(x) = (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)}(x) \subseteq (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)}(x') = \Pi_n^{\leftarrow}(x')$. Quando $n = 0$, tem-se $\Pi_n^{\leftarrow}(x') = \Pi_n^{\leftarrow}(x) = \emptyset$.
Logo, Π_n^{\leftarrow} é monotônica.

2. Suponha agora que $X \subseteq \mathbb{D}_{n+1}$ é dirigido com relação a inclusão. Considerando-se que \mathbb{D}_{n+1} é fechado para uniões dirigidas, e que a função Π_n^{\leftarrow} é monótona, $\Pi_n^{\leftarrow}[X] \subseteq \mathbb{D}_{n+1}$ é também dirigido e $\bigcup \Pi_n^{\leftarrow}[X] \in \mathbb{D}_{n+1}$.

- (a) Se $x \neq \emptyset$, $x, x' \in \gamma_n^{(0)}[\mathbb{P}_n]$ e sabendo que $\Pi_n^{(00)}$ é contínua, tem-se

$$\Pi_n^{\leftarrow}(\bigcup^\uparrow X) = \Pi_n^{(00)} = (\bigcup^\uparrow X) = \bigcup^\uparrow \{\Pi_n^{(00)}(x) | x \in X\} = \bigcup^\uparrow \{\Pi_n^{\leftarrow}(x) | x \in X\}$$

- (b) Se $x \neq \emptyset$, $x, x' \in \gamma_n^{(1)}[\mathbb{P}_n \prod \mathbb{P}_n]$ e considerando-se a continuidade das funções em F_1 , tem-se

$$\begin{aligned} \Pi_n^{\leftarrow}(\bigcup^\uparrow X) &= (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)}(\bigcup^\uparrow X) = \\ &= \bigcup^\uparrow \{(\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)}(x) | x \in X\} = \bigcup^\uparrow \{\Pi_n^{\leftarrow}(x) | x \in X\}. \end{aligned}$$

- (c) Se $x \neq \emptyset$, $x, x' \in \gamma_n^{(2)}[\mathbb{B} \prod \mathbb{P}_n \prod \mathbb{P}_n]$ e considerando-se a continuidade das funções em F_2 , tem-se

$$\begin{aligned} \Pi_n^{\leftarrow}(\bigcup^\uparrow X) &= (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)}(\bigcup^\uparrow X) = \\ &= \bigcup^\uparrow \{(\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)}(x) | x \in X\} = \bigcup^\uparrow \{\Pi_n^{\leftarrow}(x) | x \in X\}. \end{aligned}$$

Logo, $\Pi_n^{\leftarrow}(\bigcup^\uparrow X) = \bigcup^\uparrow \{\Pi_n^{\leftarrow}(x) | x \in X\}$, ou ainda, Π_n^{\leftarrow} é contínua.

3. Sejam $x, x' \in \mathbb{D}_{n+1}$ tais que $x \cap x' \in \mathbb{D}_{n+1}$. Se $x \cap x' = \emptyset$, desde que Π_n^{\leftarrow} é estrita sua prova é imediata. Caso contrário, se $x \cap x' \neq \emptyset$ e sabendo que

- (a) $x, x' \in \gamma_n^{(0)}[\mathbb{P}_n]$ e $\Pi_n^{(00)}$ é estável, tem-se

$$\Pi_n^{\leftarrow}(x \cap x') = \Pi_n^{(00)}(x \cap x') = \Pi_n^{(00)}(x) \cap \Pi_n^{(00)}(x') = \Pi_n^{\leftarrow}(x) \cap \Pi_n^{\leftarrow}(x').$$

- (b) $x, x' \in \gamma_n^{(1)}[\mathbb{P}_n \prod \mathbb{P}_n]$ e considerando estáveis as funções em F_1 , tem-se

$$\begin{aligned} \Pi_n^{\leftarrow}(x \cap x') &= ((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)})(x \cap x') = \\ &= ((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)})(x) \cap ((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)})(x'). \end{aligned}$$

- (c) $x, x' \in \gamma_n^{(2)}[\mathbb{P}_n \prod \mathbb{P}_n]$ e considerando estáveis as funções em F_2 , tem-se

$$\begin{aligned} \Pi_n^{\leftarrow}(x \cap x') &= ((\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)})(x \cap x') = \\ &= ((\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)})(x) \cap ((\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)})(x'). \end{aligned}$$

Em todos os casos, a igualdade $\Pi_n^{\leftarrow}(x \cap x') = \Pi_n^{\leftarrow}(x) \cap \Pi_n^{\leftarrow}(x')$ é satisfeita. Portanto Π_n^{\leftarrow} é uma função estável.

4. Considere agora $X \subseteq \mathbb{D}_{n+1}$. Sabendo que $\Pi_n^{\leftarrow}(\bigcup X) \in \mathbb{D}_n$, se $X = \emptyset$ tem-se $\Pi_n^{\leftarrow}(\bigcup \emptyset) = \emptyset = \bigcup \emptyset$. Caso contrário, isto é se $X \neq \emptyset$,

- (a) $X \subseteq \gamma_n^{(0)}[\mathbb{P}_n]$ e considerando a função linear $\Pi_n^{(00)}$,

$$\Pi_n^{\leftarrow}(\bigcup X) = \Pi_n^{(00)}(\bigcup X) = \bigcup \{\Pi_n^{(00)}(x) | x \in X\} = \bigcup \{\Pi_n^{\leftarrow}(x) | x \in X\}.$$

- (b) $X \subseteq \gamma_n^{(1)}[\mathbb{P}_n \prod \mathbb{P}_n]$ e as funções lineares em F_1 , então

$$\begin{aligned} \Pi_n^{\leftarrow}(\bigcup X) &= ((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)})(\bigcup X) = \\ &= \bigcup \{((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(001)})(x) | x \in X\} = \bigcup \{\Pi_n^{\leftarrow}(x) | x \in X\}. \end{aligned}$$

(c) $X \in \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_0]$, $n \neq 0$ e as funções lineares em F_2 , então

$$\begin{aligned} \Pi_n^{\leftarrow}(\bigcup X) &= (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)}(\bigcup X) = \\ &= \bigcup \{(\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(002)}(x) \mid x \in X\} = \bigcup \{\Pi_n^{\leftarrow}(x) \mid x \in X\}. \end{aligned}$$

Logo, Π_n^{\leftarrow} é linear.

Conclue-se então que $\Pi_n^{\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D} \in \text{MorCospLin}$. \blacktriangle

Definição 5.11 Considere os espaços coerentes \mathbb{D}_n e \mathbb{D}_{n+1} e as funções estritas, lineares $\Pi_n^{\leftarrow(\alpha)}$, $\Pi_n^{\rightarrow(\beta)}$ do nível $\mathbb{D}_{n+1} - \mathbb{D}_n$ apresentadas na Definição 5.10. As projeções $\Pi_n^{\leftrightarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\leftrightarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\leftrightarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela união dos subconjuntos coerentes dados pela expressão

$$\Pi_n^{\leftrightarrow(\alpha)}(x) = \Pi_n^{\leftarrow(\alpha)}(x) \cup \Pi_n^{\rightarrow(\alpha)}(x).$$

Quando $\alpha = 0$, $\Pi_n^{\leftrightarrow(\alpha)} = \Pi_n^{\leftrightarrow}$.

Proposição 5.10 As projeções $\Pi_n^{\leftrightarrow(\alpha)}$ do nível $\mathbb{D}_{n+1} - \mathbb{D}_n$, apresentadas na Definição 5.11, são morfismos na categoria CospLin .

Demonstração. Fixado $n \in \omega$, considere o domínio $\mathbb{D}_{n+1} = \mathbb{P}_n \amalg (\mathbb{P}_n \amalg \mathbb{P}_n) \amalg (\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n)$. Apresenta-se a seguir um esquema da prova baseado na linearidade das funções Π_n^{\leftarrow} , $\Pi_n^{\rightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, apresentadas na Definição 5.10.

1. Sejam $x, x' \in \mathbb{D}_{n+1}$ tais que $x \subseteq x'$.
Se $x = \emptyset$ tem-se que $\Pi_n^{\leftarrow}(\emptyset) = \emptyset \subseteq \Pi_n^{\leftarrow}(x')$, $\forall x' \in \mathbb{D}_{n+1}$.
Se $x \neq \emptyset$, pela monotonicidade das funções Π_n^{\leftarrow} , Π_n^{\rightarrow} tem-se

$$\Pi_n^{\leftarrow}(x') = \Pi_n^{\leftarrow}(x') \cup \Pi_n^{\rightarrow}(x') \subseteq \Pi_n^{\leftarrow}(x) \cup \Pi_n^{\rightarrow}(x) = \Pi_n^{\leftarrow}(x).$$

Logo, Π_n^{\leftarrow} é monotônica.

2. Suponha agora que $X \subseteq \mathbb{D}_{n+1}$ é dirigido com relação à inclusão. Considerando-se que \mathbb{D}_{n+1} é fechado para uniões dirigidas, e que a função Π_n é monótona, $\Pi_n[X] \subseteq \mathbb{D}_{n+1}$ é também dirigido e $\bigcup \Pi_n[X] \in \mathbb{D}_{n+1}$. Seja $X \neq \emptyset$ e considerando-se a continuidade das funções Π_n^{\leftarrow} e Π_n^{\rightarrow} tem-se

$$\Pi_n^{\leftarrow}(\bigcup^\uparrow X) = (\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(\bigcup^\uparrow X) = \bigcup^\uparrow \{(\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(x) \mid x \in X\}.$$

Tem-se então a igualdade $\Pi_n^{\leftarrow}(\bigcup^\uparrow X) = \bigcup^\uparrow \{\Pi_n^{\leftarrow}(x) \mid x \in X\}$. Portanto, Π_n é contínua.

3. Sejam $x, x' \in \mathbb{D}_{n+1}$ tais que $x \cap x' \in \mathbb{D}_{n+1}$. Se $x \cap x' = \emptyset$, desde que Π_n^{\leftarrow} é estrita sua prova é imediata. Suponha então $x \cap x' \neq \emptyset$ e considere as funções estáveis Π_n^{\leftarrow} e Π_n^{\rightarrow} . Tem-se

$$\begin{aligned} \Pi_n^{\leftarrow}(x \cap x') &= (\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(x \cap x') = (\Pi_n^{\leftarrow}(x) \cap \Pi_n^{\leftarrow}(x')) \cup (\Pi_n^{\rightarrow}(x) \cap \Pi_n^{\rightarrow}(x')) \\ &= (\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(x) \cap (\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(x') = \Pi_n^{\leftarrow}(x) \cap \Pi_n^{\leftarrow}(x') \end{aligned}$$

Logo, $\Pi_n^{\leftarrow}(x \cap x') = \Pi_n^{\leftarrow}(x) \cap \Pi_n^{\leftarrow}(x')$, provando que Π_n^{\leftarrow} é uma função estável.

4. Considere agora $X \subseteq \mathbb{D}_{n+1}$. Sabendo que $\Pi_n^{\leftarrow}(\bigcup X) \in \mathbb{D}_n$, tem-se que

$$\Pi_n^{\leftarrow}(\bigcup X) = (\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(\bigcup X) = \bigcup \{(\Pi_n^{\leftarrow} \cup \Pi_n^{\rightarrow})(x) \mid x \in \bigcup X\}$$

Assim sendo, tem-se $\Pi_n^{\leftrightarrow}(\cup X) = \cup\{\Pi_n^{\leftrightarrow}(x) \mid x \in \cup X\}$ o que mostra que Π_n^{\leftrightarrow} satisfaz a linearidade.

Portanto, pelo que foi mostrado nos itens anteriores desta demonstração, Π_n^{\leftrightarrow} é uma função linear, ou seja, um morfismo na categoria $Cosp\mathcal{L}in$. \blacktriangle

Proposição 5.11 *Considerando-se as imersões $\pi_n^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e as projeções $\Pi_n^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, apresentadas nas Definições 5.8, tem-se que $(\mathbb{D}_n, \mathbb{D}_{n+1})$ é um par projeção na categoria $Cosp\mathcal{L}in$.*

Pela proposição 5.11, valem as seguintes condições $\Pi_n^{\leftrightarrow} \circ \pi_n^{\leftrightarrow} = id_{\mathbb{D}_n}$ e $\pi_n^{\leftrightarrow} \circ \Pi_n^{\leftrightarrow} \subseteq id_{\mathbb{D}_{n+1}}$. Assim também, pode-se afirmar que $(\overline{\mathbb{D}}_n, \mathbb{D}_{n+1})$ e $(\overline{\mathbb{D}}_n^{\perp}, \mathbb{D}_{n+1})$ são pares-projeção na categoria $Cosp\mathcal{L}in$.

As observações referentes às projeções Π_n^{\leftrightarrow} feitas nos próximos parágrafos, assim como os exemplos, são obtidos a partir da análise entre os processos representados nas Figuras 5.9 e 5.11, onde são considerados os três primeiros níveis da estrutura.

Exemplificação 5.3

- Seja z o subconjunto coerente maximal, representado na Figura 5.11, dado por $z = \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(k)}, d_{001.011.001}^{(k)}, d_{(011)2.001}^{(k)}, d_{(001)2.011}^{(k)}, d_{011.001.011}^{(k)}, d_{001.(011)2}^{(k)}, d_{(011)3}^{(k)}\}$.

Tem-se $\Pi_2^{\leftarrow}(z) = \{d_{001.001}^{(k)}, d_{011.001}^{(k)}\}$ e $\Pi_2^{\rightarrow}(z) = \{d_{001.011}^{(k)}, d_{011.011}^{(k)}\}$, logo

- $(\Pi_2^{\leftarrow}(z) = \Pi_2^{\leftarrow} \cup \Pi_2^{\rightarrow})(x) = \{d_{001.001}^{(k)}, d_{011.001}^{(k)}, d_{001.011}^{(k)}, d_{011.011}^{(k)}\}$;
- $(\Pi_1^{\leftrightarrow} \circ \Pi_2^{\leftarrow})(z) = \{d_{001}^{(k)}, d_{011}^{(k)}\}$ e $(\Pi_0^{\leftrightarrow} \circ \Pi_1^{\leftrightarrow} \circ \Pi_2^{\rightarrow})(z) = \emptyset$.

Especialmente neste caso, tem-se

- $\Pi_2^{\leftarrow}(z) = \Pi_2^{\rightarrow}(z) = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, d_{001.011}^{(k)}, d_{(011)2}^{(k)}\} = \Pi_2^{\leftrightarrow}(x)$.

- Seja x o subconjunto coerente maximal em \mathbb{D}_3 , na Figura 5.9, dado por $x = \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(k)}, d_{001.011.001}^{(k)}, d_{(011)2.001}^{(k)}, e_{(001)2.011}^{(l)}, e_{011.001.011}^{(l)}, e_{001.(011)2}^{(l)}, e_{(011)3}^{(l)}\}$.

Considerando-se $\Pi_2^{\leftarrow}(z) = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}\}$ e $\overline{\Pi}_2(z) = \{e_{001.011}^{(l)}, e_{(011)2}^{(l)}\}$, tem-se a desigualdade $\Pi_2^{\leftarrow}(z) \neq \Pi_2^{\rightarrow}(z)$. Além disso,

- $\Pi_2^{\leftarrow}(x) = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, e_{001.011}^{(l)}, e_{(011)2}^{(l)}\}$;
- $(\Pi_1^{\leftrightarrow} \circ \Pi_2^{\leftarrow})(x) = \{d_{001}^{(k)}, e_{011}^{(l)}\}$ e $(\Pi_0^{\leftrightarrow} \circ \Pi_1^{\leftrightarrow} \circ \Pi_2^{\rightarrow})(x) = \emptyset$.

Aplicando as projeções definidas em seções anteriores, obtém-se:

- $\Pi_2^{(001)}(x) = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, d_{001.011}^{(k)}, d_{(011)2}^{(k)}\} = \Pi_2^{\rightarrow}(x)$;
- $\Pi_2^{(011)}(x) = \{e_{(001)2}^{(l)}, e_{011.001}^{(l)}, e_{001.011}^{(l)}, e_{(011)2}^{(l)}\} = \Pi_2^{\leftarrow}(x)$.
- $(\Pi_1^{\leftrightarrow} \circ \Pi_2^{\leftarrow})(x) = \{d_{001}^{(k)}, d_{011}^{(k)}\}$ e $(\Pi_1^{\leftarrow} \circ \Pi_2^{\leftarrow})(x) = \{e_{001}^{(l)}, e_{011}^{(l)}\}$.

Por fim, salienta-se que $x \notin \pi_3^{\rightarrow}[\mathbb{P}_2 \amalg \mathbb{P}_2]$ e $z \notin \pi_3^{\leftarrow}[\mathbb{P}_2 \amalg \mathbb{P}_2]$. Ambos são construídos em \mathbb{D}_3 .

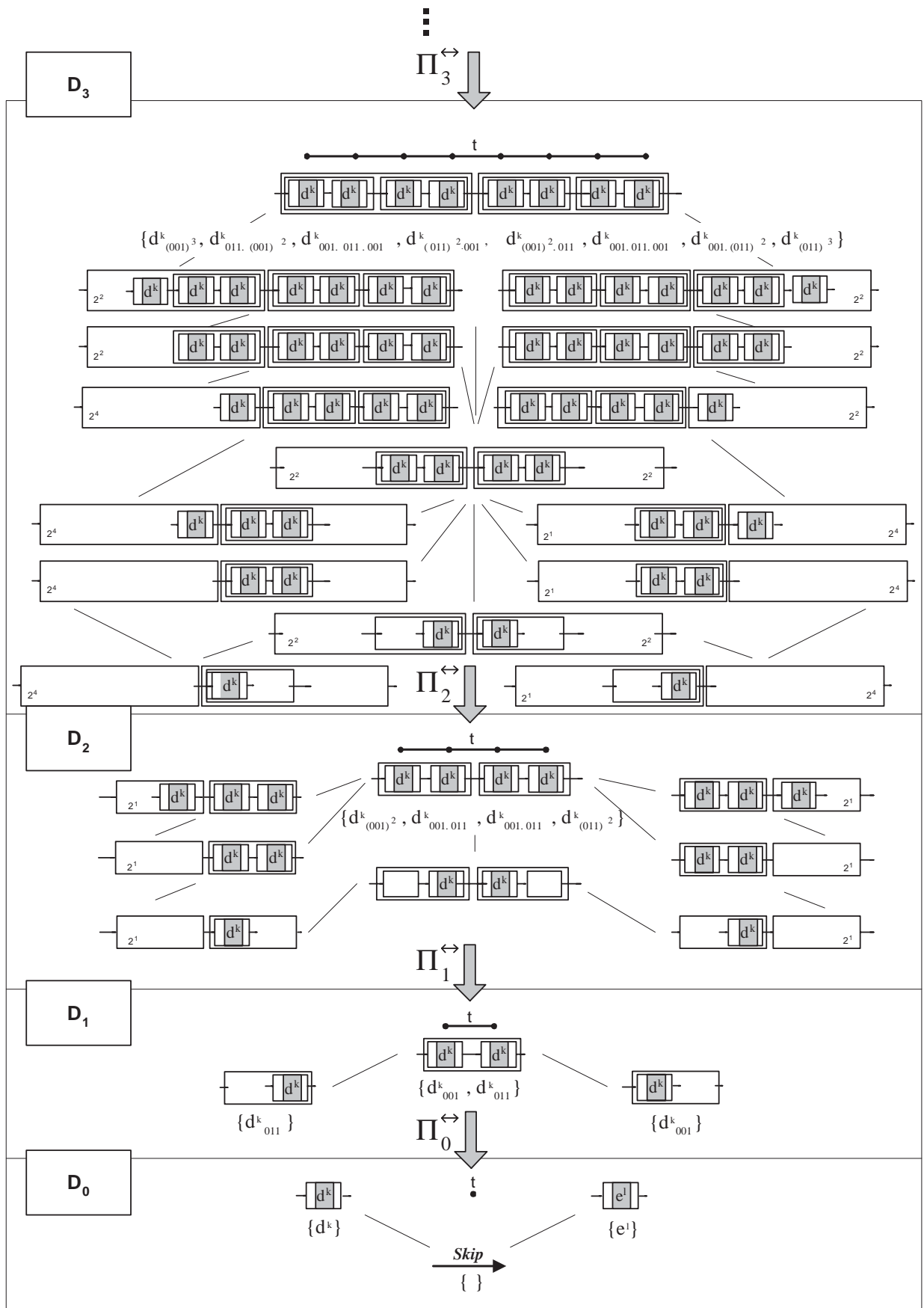


FIGURA 5.11 - Aplicações das Projeções $\Pi_{n,0}^{\leftrightarrow}$ sobre Objetos em \mathbb{D}_n .

5.5 Imersões e Projeções $\pi_n^{\rightarrow\leftarrow}$ e $\Pi_n^{\rightarrow\leftarrow}$

Nesta seção são apresentadas as imersões $\pi_n^{\rightarrow\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e as correspondentes projeções $\Pi_n^{\rightarrow\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, que também interpretam construtores de processos.

Neste caso, o limite para o diagrama definido a partir das projeções $\Pi_n^{\rightarrow\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ formaliza a completação da estrutura indutiva de um modelo de máquina geométrica computacional capaz de interpretar, entre outros, os processos seqüenciais que são executados dentro de um intervalo de tempo, possuindo portanto um tempo inicial e final de execução, mas que admitem um seqüência infinita de instantes de tempo computacional entre tais extremos.

Pode-se então verificar que, nesta estrutura, estão as interpretações para processos infinitos no sentido temporal, pré e pós fixados, simultâneamente.

Pode-se pensar numa analogia entre o conjunto $\mathcal{I}^{\rightarrow\leftarrow}$ dos indexadores dos tokens que interpretam tais produtos e o conjunto dos números inteiros ordenados na forma $0, 1, 2, \dots, \dots, -3, -2, -1$.

5.5.1 Imersões $\pi_n^{\rightarrow\leftarrow}$.

A definição das imersões $\pi_n^{\rightarrow\leftarrow}$ referentes ao nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, formalizadas em 5.12, baseia-se na análise dos três próximos casos.

1. Quando $x \in \gamma_{n-1}^{(0)}[\mathbb{P}_{n-1}]$, verificam-se as igualdades $\pi_n^{\rightarrow\leftarrow}(x) = \pi_n^{\leftarrow}(x) = \pi_n^{(00)}(x) \in \mathbb{P}_n \subseteq \mathbb{D}_{n+1}$, assegurando que no subespaço $(\pi_n^{\rightarrow\leftarrow} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}]$ são reconstruídas todas as interpretações de processos finitos, executados em 2^n *utc*.
2. Quando $x \in \gamma_{n-1}^{(1)}[\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}]$, tem-se $\pi_n^{\rightarrow\leftarrow}(x) \subseteq \mathbb{P}_n \amalg \mathbb{P}_n \subseteq \mathbb{D}_{n+1}$. Entretanto, $\pi_n^{\rightarrow\leftarrow}(x) \not\subseteq \pi_n^{(0\beta 1)}(x)$ e $\pi_n^{(0\beta 1)}(x) \not\subseteq \pi_n^{\rightarrow\leftarrow}(x)$.

No correspondente subespaço $\pi_n^{\rightarrow\leftarrow}[\mathbb{P}_n \amalg \mathbb{P}_n]$ estão as interpretações para o conjunto de processos seqüenciais, executados em 2^{n+1} *utc*, que possuem, simultâneamente, prefixos e sufixos definidos. Mas, neste caso, o tempo associado aos prefixos e sufixos representados por estas interpretações é, no máximo, igual a 2^{n-1} *utc*.

Isto significa, intuitivamente, que as imersões $\pi_n^{\rightarrow\leftarrow}$, quando aplicadas sobre subconjuntos que interpretam processos seqüenciais, estendem o comprimento dos braços que já haviam sido construídos por interpretações construídas em níveis anteriores. Mas neste caso, novos braços não serão construídos.

Portanto, tais imersões sempre determinam objetos parciais no subespaço $\mathbb{P}_n \amalg \mathbb{P}_n$ que interpretam processos pré e pós fixados, sendo que cada prefixo e sufixo deve ser executados em 2^{n-1} *utc*.

3. Considerando-se o caso das somas determinísticas, os novos subconjuntos na imagem $\pi_n^{\rightarrow\leftarrow}[\mathbb{P}_n \amalg_{\mathbb{B}} \mathbb{P}_n]$ representam a possibilidade de existência de novos testes e todas as escolhas decorrentes do fato deles serem ou não, satisfeitos. Mas, duas condições são conhecidas: o processo que será executado se o primeiro teste for satisfeito e o processo que será executado caso o último teste não for satisfeito.

Aplicações das imersões $\pi_{n,\infty}^{\rightarrow\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow\leftarrow}$, formalizadas na Definição 5.13, podem ser visualizadas na Figura 5.12.

Cada imersão $\pi_n^{\rightarrow\leftarrow}$ é definida pela união das imagens obtidas pelas imersões π_n^{\leftarrow} e π_n^{\rightarrow} , apresentadas logo abaixo.

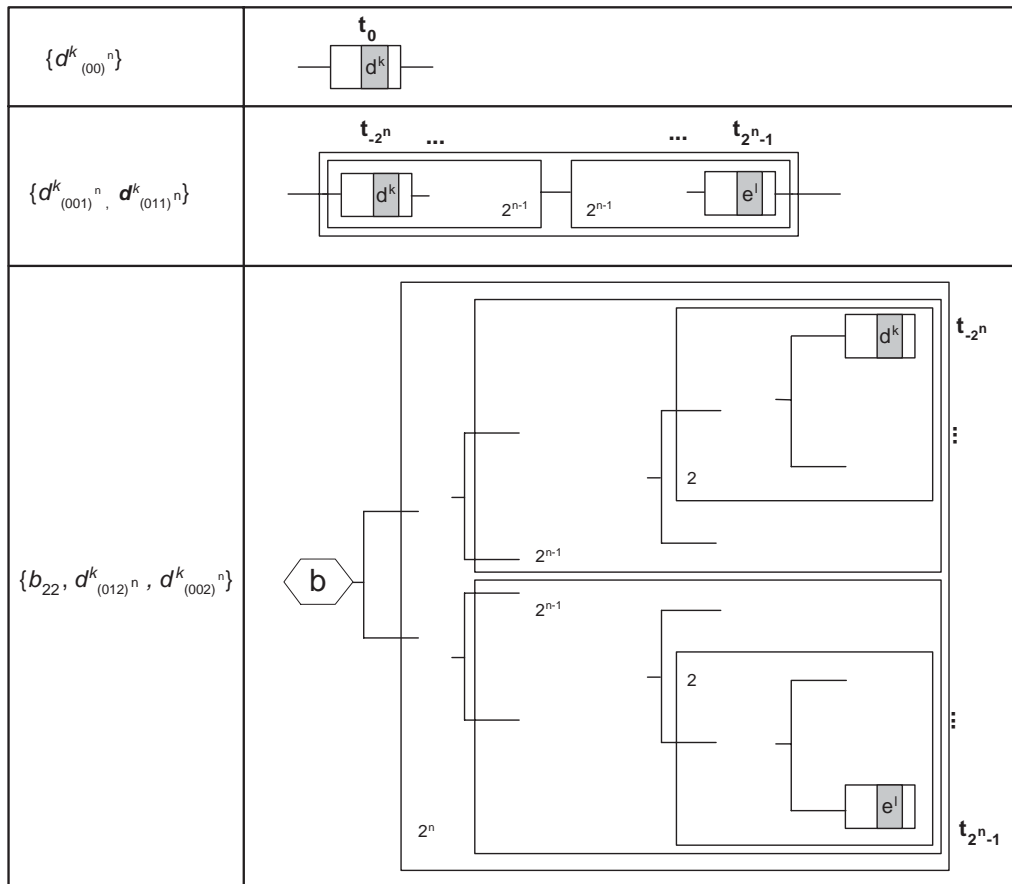


FIGURA 5.12 - Representação de Objetos em $\pi_{0,n}^{\rightarrow\leftarrow}[\mathbb{D}_n]$.

Definição 5.12 Considere os espaços coerentes \mathbb{D}_n , \mathbb{P}_n , $\mathbb{P}_n \amalg \mathbb{P}_n$, $\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ e \mathbb{D}_{n+1} que definem o n -ésimo nível da estrutura deste modelo computacional. Considere também $\beta, \alpha \in \{0, 1, 2\}$ e as imersões $\gamma_n^{(\alpha)}$ e $\pi_n^{(\beta)}$, apresentadas na Proposição 5.1 e 5.4.

As funções $\pi_n^{\leftrightarrow(\alpha)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\leftarrow(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\leftarrow(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas pela expressão

$$\pi_{n+1}^{\leftrightarrow(\alpha)}(x) = \begin{cases} \pi_{n+1}^{(\alpha 0)}(x) = \{\mathbf{a}_{\alpha 0} \mid \mathbf{a} \in x, \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x \in \gamma_n^{(0)}, \\ (\pi_{n+1}^{(\alpha 01)} \circ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}))(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\pi_{n+1}^{(\alpha 02)} \circ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)}))(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Analogamente, as imersões $\pi_n^{\leftrightarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\leftarrow(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\leftarrow(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são formalizadas pela expressão

$$\pi_{n+1}^{\leftarrow(\beta)}(x) = \begin{cases} \pi_{n+1}^{(\beta 0)}(x) = \{\mathbf{a}_{\beta 0} \mid \mathbf{a} \in x, \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x \in \gamma_n^{(0)}, \\ (\pi_{n+1}^{(\beta 11)} \circ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}))(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\pi_{n+1}^{(\beta 12)} \circ (\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)}))(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Considere os casos

1. se $n = 0$, valem então as igualdades $\pi_0^{\leftrightarrow(\alpha)} = \pi_0^{\leftarrow(\alpha)} = \pi_0^{(\alpha 0)}$.
2. se $\alpha = 0$, utiliza-se a notação $\pi_n^{\leftrightarrow(0)} \equiv \pi_n^{\leftrightarrow}$ e $\pi_n^{\leftarrow(0)} \equiv \pi_n^{\leftarrow}$, $\forall n \in \omega$.

Em particular, $\pi_n^{\leftrightarrow(\alpha)}(\emptyset) = \pi_n^{\leftarrow(\beta)}(\emptyset) = \emptyset$.

Considerações semelhantes às que foram apresentadas nas seções anteriores deste capítulo, referentes à composição finita de imersões e projeções, também se aplicam aos morfismos apresentados nesta seção.

Definição 5.13 Considere os espaços coerentes \mathbb{D}_n e \mathbb{D}_{n+1} e as funções estritas $\pi_n^{\leftrightarrow(\alpha)}$, $\pi_n^{\leftarrow(\beta)}$, no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ apresentadas na Definição 5.12. As imersões $\pi_n^{\rightarrow\leftarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\rightarrow\leftarrow(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\rightarrow\leftarrow(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas pela expressão

$$\pi_n^{\rightarrow\leftarrow(\alpha)}(x) = \pi_n^{\leftrightarrow(\alpha)}(x) \cup \pi_n^{\leftarrow(\alpha)}(x).$$

Quando $\alpha = 0$, utiliza-se a notação $\pi_n^{\rightarrow\leftarrow(0)} = \pi_n^{\rightarrow\leftarrow}$.

Em particular, $\pi_n^{\rightarrow\leftarrow(\alpha)}(\emptyset) = \emptyset$.

5.5.2 Projeções $\Pi_n^{\rightarrow\leftarrow}$

Definição 5.14 Considere os espaços coerentes \mathbb{D}_n , \mathbb{P}_n , $\mathbb{P}_n \amalg \mathbb{P}_n$, $\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n$ e \mathbb{D}_{n+1} que definem o n -ésimo nível da estrutura deste modelo computacional. Considere também as imersões $\pi_n^{(\theta)}$, $\lambda_n^{(\beta)}$ e $\psi_n^{(\beta)}$, apresentadas na Proposição 5.1, assim como as projeções $\Pi_n^{(\theta)}$, $\Gamma_n^{(\alpha)}$, $\lambda_n^{(\beta)}$ e $\Psi_n^{(\alpha)}$, formalizadas na Definição 5.5, sempre que $\beta \in \{0, 1\}$, $\alpha \in \{1, 2\}$ e $\theta \in \{00, 001, 011, 002, 012\}$.

As funções estritas $\Pi_n^{\leftrightarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\leftrightarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\leftrightarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela expressão

$$\Pi_n^{\leftrightarrow(\alpha)}(x) = \begin{cases} \Pi_n^{(\alpha 0)}(x), & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(\alpha 01)}(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \lambda_n^{(0)}) \circ (\Lambda_n^{(0)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(\alpha 02)}(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, $\Pi_n^{\leftarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\leftarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\leftarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela expressão

$$\Pi_n^{\leftarrow(\beta)}(x) = \begin{cases} \Pi_n^{(\beta 0)}(x), & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}) \circ \Pi_n^{(\beta 11)}(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \lambda_n^{(1)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)}) \circ \Pi_n^{(\beta 12)}(x), & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Definição 5.15 Considere os espaços coerentes \mathbb{D}_n e \mathbb{D}_{n+1} e as funções estritas $\Pi_n^{\rightarrow(\alpha)}$, $\Pi_n^{\leftarrow(\beta)}$ no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, apresentadas em 5.14. As projeções $\Pi_n^{\rightarrow\leftarrow(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\rightarrow\leftarrow(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\rightarrow\leftarrow(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela expressão

$$\Pi_n^{\rightarrow\leftarrow(\alpha)}(x) = \Pi_n^{\rightarrow(\alpha)}(x) \cup \Pi_n^{\leftarrow(\alpha)}(x).$$

Em particular, quando $\alpha = 0$, utiliza-se a notação $\Pi_n^{\rightarrow\leftarrow(0)} = \Pi_n^{\rightarrow\leftarrow}$.

Proposição 5.12 As projeções $\Pi_n^{\leftarrow(\alpha)}$, Π_n^{\leftarrow} , $\Pi_n^{\rightarrow\leftarrow}$ no nível $\mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, apresentadas nas Definições 5.14 e 5.15, assim como as imersões $\pi_n^{\rightarrow(\alpha)}$, $\pi_n^{\leftarrow(\beta)}$ e $\pi_n^{\rightarrow\leftarrow(\alpha)}$ definidas em 5.12 e 5.13, são morfismos na categoria $\mathit{CospLin}$.

A demonstração é análoga àquelas construídas em 5.9 e 5.10.

As representações de algumas interpretações em $\Pi_n^{\rightarrow\leftarrow}[\mathbb{D}_n]$ podem ser observadas na Figura 5.13, e sugere-se comparações com outras figuras apresentadas anteriormente, neste capítulo.

Neste sentido, considere a imersão $\pi_{m,n}^{\rightarrow\leftarrow} : \mathbb{D}_m \rightarrow \mathbb{D}_{n+1}$, dada pela composição finita

$$\pi_{m,n}^{\rightarrow\leftarrow} = \pi_m^{\rightarrow\leftarrow} \circ \pi_{m+1}^{\rightarrow\leftarrow} \circ \dots \circ \pi_{n-1}^{\rightarrow\leftarrow} \circ \pi_n^{\rightarrow\leftarrow}.$$

1. Seja $z = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, e_{001.011}^{(l)}, e_{(011)2}^{(l)}\}$ em \mathbb{D}_2 . Tem-se
 - $\pi_{2,n}^{\leftarrow}(z) = \{d_{(001)^{n+1}}^{(k)}, d_{011.(001)^n}^{(k)}\}$ e $\pi_{2,n}^{\leftarrow}(z) = \{e_{001.(011)^n}^{(l)}, e_{(011)^{n+1}}^{(l)}\}$, então
 - $\pi_{2,n}^{\rightarrow\leftarrow}(z) = \{d_{(001)^{n+1}}^{(k)}, d_{011.(001)^n}^{(k)}, e_{001.(011)^n}^{(l)}, e_{(011)^{n+1}}^{(l)}\}$.
2. Seja o subconjunto coerente $u \in \mathbb{D}_3$, dado por

$$u = \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(k)}, d_{001.011.001}^{(k)}, d_{(011)2.001}^{(k)}, e_{(001)2.011}^{(l)}, e_{011.001.011}^{(l)}, e_{001.(011)2}^{(l)}, e_{(011)3}^{(l)}\},$$
 Neste caso, $\pi_{3,n}^{\rightarrow\leftarrow}(u) = \pi_{3,n}^{\leftarrow}(u) \cup \pi_{3,n}^{\rightarrow}(u)$, onde

$$\begin{aligned} \bullet \pi_{3,n}^{\leftarrow}(u) &= \{d_{(001)^{n+1}}^{(k)}, d_{011.(001)^n}^{(k)}, d_{001.011.(001)^{n-1}}^{(k)}, d_{(011)2.(001)^{n-1}}^{(k)}\}. \\ \bullet \pi_{3,n}^{\rightarrow}(u) &= \{e_{(001)2.(011)^{n-1}}^{(l)}, e_{011.001.(011)^n}^{(l)}, e_{001.(011)^n}^{(l)}, e_{(011)^{n+1}}^{(l)}\}. \end{aligned}$$

Além disso, tem-se as projeções

$$\Pi_{3,2}^{\rightarrow\leftarrow}(u) = x \quad \text{e} \quad \Pi_{2,1}^{\rightarrow\leftarrow}(x) = \{d_{001}^{(k)}, e_{011}^{(l)}\}x.$$

A próxima proposição viabiliza a definição de um outro morfismo capaz de interpretar as construções de processos sequenciais por prefixo, sufixo e infixos, simultaneamente.

Proposição 5.13 Considere as imersões π_n^{\leftarrow} e $\pi_n^{\rightarrow\leftarrow}$. As seguintes igualdades entre subespaços-imagem são satisfeitas:

1. $(\pi_n^{\leftarrow(\beta)} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}] \cap (\pi_n^{\rightarrow\leftarrow(\beta)} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}] = \emptyset;$
2. $(\pi_n^{\leftarrow(\beta)} \circ \gamma_{n-1}^{(2)})[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}] \cap (\pi_n^{\rightarrow\leftarrow(\beta)} \circ \gamma_{n-1}^{(2)})[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}] = \emptyset;$
3. $(\pi_n^{\leftarrow(\beta)} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}] = (\pi_n^{\rightarrow\leftarrow(\beta)} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}].$

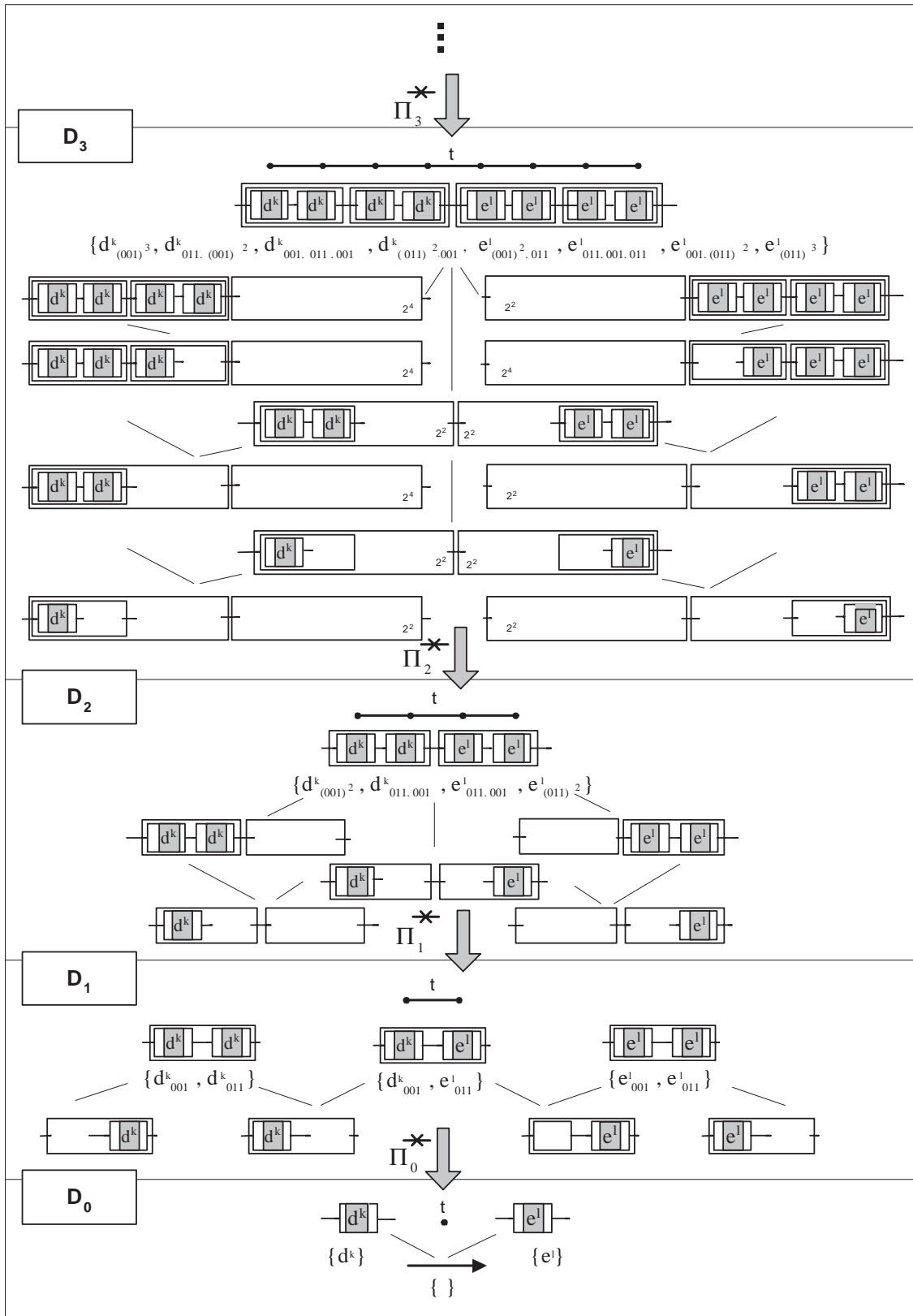


FIGURA 5.13 - Aplicação das Projeções Π_n^{\leftarrow} sobre Objetos em \mathbb{D}_n .

5.6 Imersões e Projeções π_n^{\rightleftarrows} e Π_n^{\rightleftarrows}

Apresenta-se agora as imersões $\pi_n^{\rightleftarrows} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e as correspondentes projeções $\Pi_n^{\rightleftarrows} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, que também interpretam construtores e destrutores de processos, respectivamente.

Neste caso, o limite para o diagrama definido a partir das projeções $\Pi_n^{\rightleftarrows} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ formaliza a completção da estrutura indutiva de um modelo de máquina geométrica computacional capaz de interpretar, entre outros, os processos seqüenciais infinitos, cujos fatores são também processos infinitos no sentido temporal, que admitem a construção simultâneamente por prefixo, sufixo e infixo.

Nos processos modelados em $\mathbb{D}_\infty^{\rightleftarrows}$ é sempre possível determinar o primeiro e o último processo a ser executado em um produto seqüencial infinito no sentido temporal. Além disso, pode-se determinar qual será o processo executado nos instantes t_{-n} e t_n ou qualquer intervalo finito obtido a partir destes, de forma isomorfa ao conjunto $0, 1, 2, \dots, \dots, -2, -1$. As imersões π_n^{\rightleftarrows} e as projeções Π_n^{\rightleftarrows} são definidas logo a seguir.

5.6.1 Imersões π_n^{\rightleftarrows}

Definição 5.16 *Considere as imersões $\pi_n^{\rightarrow(\alpha)}$, $\pi_n^{\leftarrow(\beta)}$ e $\pi_n^{\rightarrow\leftarrow}$ do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ e os indexadores $\alpha, \beta \in \{0, 1, 2\}$. As funções $\pi_n^{\rightleftarrows(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\rightleftarrows(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\rightleftarrows(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas pela expressão*

$$\pi_n^{\rightleftarrows(\beta)}(x) = \begin{cases} \pi_n^{\leftarrow(\beta)}(x) \cup \pi_n^{\rightarrow(\beta)}(x), & \text{se } \Pi_n^{\rightarrow}(x) \in \Gamma_n^{(1)}[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}], \\ \pi_n^{\rightarrow\leftarrow(\beta)}(x), & \text{caso contrário.} \end{cases}$$

onde as funções $\pi_n^{\rightleftarrows(0)}, \pi_n^{\leftarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\rightleftarrows(1)}, \pi_n^{\leftarrow(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\rightleftarrows(2)}, \pi_n^{\leftarrow(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são dadas pelas composições

$$\pi_n^{\rightleftarrows(\beta)} = \pi_n^{\rightarrow(\beta)} \circ \pi_{n-1}^{\leftarrow} \circ \Pi_n^{\rightarrow} \quad \text{e} \quad \pi_n^{\leftarrow(\alpha)} = \pi_n^{\leftarrow(\alpha)} \circ \pi_{n-1}^{\rightarrow\leftarrow} \circ \Pi_n^{\leftarrow}.$$

Quando $\beta = 0$, utiliza-se a notação $\pi_n^{\rightleftarrows(0)} = \pi_n^{\rightleftarrows}$.

Em particular, $\pi_n^{\rightleftarrows(\beta)}(\emptyset) = \emptyset$.

Os exemplos logo abaixo, facilitam a compreensão da Definição 5.16.

Exemplificação 5.4

Considere a função imersão $\pi_{m,n}^{\rightleftarrows} : \mathbb{D}_m \rightarrow \mathbb{D}_{n+1}$, dada pela composição finita

$$\pi_{m,n}^{\rightleftarrows} = \pi_m^{\rightleftarrows} \circ \pi_{m+1}^{\rightleftarrows} \circ \dots \circ \pi_{n-1}^{\rightleftarrows} \circ \pi_n^{\rightleftarrows}.$$

1. Seja $z = \{d_{(001)^2}^{(k)}, d_{011.001}^{(l)}, d_{001.011}^{(m)}, d_{(011)^2}^{(i)}\}$ em \mathbb{D}_2 . Tem-se
 - $\pi_{2,n}^{\rightleftarrows}(z) = \{d_{(001)^{n+1}}^{(k)}, d_{(011)^n.001}^{(l)}\}$ e $\pi_{2,n}^{\leftarrow}(z) = \{d_{(001)^n.011}^{(m)}, d_{(011)^{n+1}}^{(i)}\}$, então
 - $\pi_{2,n}^{\rightleftarrows}(z) = \{d_{(001)^{n+1}}^{(k)}, d_{(011)^n.001}^{(l)}, d_{(011)^n.001}^{(m)}, d_{(011)^{n+1}}^{(i)}\}$.
2. Seja $y = \{e_{(002)^2}^{(k)}, e_{012.002}^{(l)}, e_{002.012}^{(m)}, e_{(012)^2}^{(j)}\} \in \mathbb{D}_2$
 - $\pi_{2,n}^{\rightleftarrows}(y) = \{e_{(002)^{n+1}}^{(k)}, e_{(012)^n.002}^{(l)}\}$ e $\pi_{2,n}^{\leftarrow}(y) = \{e_{(002)^n.012}^{(m)}, e_{(012)^{n+1}}^{(j)}\}$, então
 - $\pi_{2,n}^{\rightleftarrows}(y) = \{e_{(002)^{n+1}}^{(k)}, e_{(012)^n.002}^{(l)}, e_{(012)^n.002}^{(m)}, e_{(012)^{n+1}}^{(j)}\}$.

3. Seja o subconjunto coerente $x \in \mathbb{D}_3$, dado por

$$x = \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(l)}, d_{001.011.001}^{(m)}, d_{(011)2.001}^{(j)}, e_{(001)2.011}^{(l)}, e_{011.001.011}^{(m)}, e_{001.(011)2}^{(m)}, e_{(011)3}^{(j)}\}.$$

Neste caso, $\pi_{3,n}^{\rightleftarrows}(x) = \pi_{3,n}^{\rightarrow}(x) \cup \pi_{3,n}^{\leftarrow}(x)$, sendo

- $\pi_{3,n}^{\rightarrow}(x) = \{d_{(001)^{n+1}}^{(k)}, d_{011.(001)^n}^{(l)}, d_{001.(011)^{n-1}.001}^{(m)}, d_{(011)^n.001}^{(j)}\}.$
- $\pi_{3,n}^{\leftarrow}(x) = \{e_{(001)^n.011}^{(k)}, e_{011.(001)^n.011}^{(l)}, e_{001.(011)^n}^{(m)}, e_{(011)^{n+1}}^{(j)}\}.$

Os subconjuntos $\pi_{2,n}^{\rightleftarrows}(z)$, $\pi_{2,n}^{\rightleftarrows}(y)$ e $\pi_{3,n}^{\rightleftarrows}(x)$ estão, respectivamente representados na Figura 5.14.

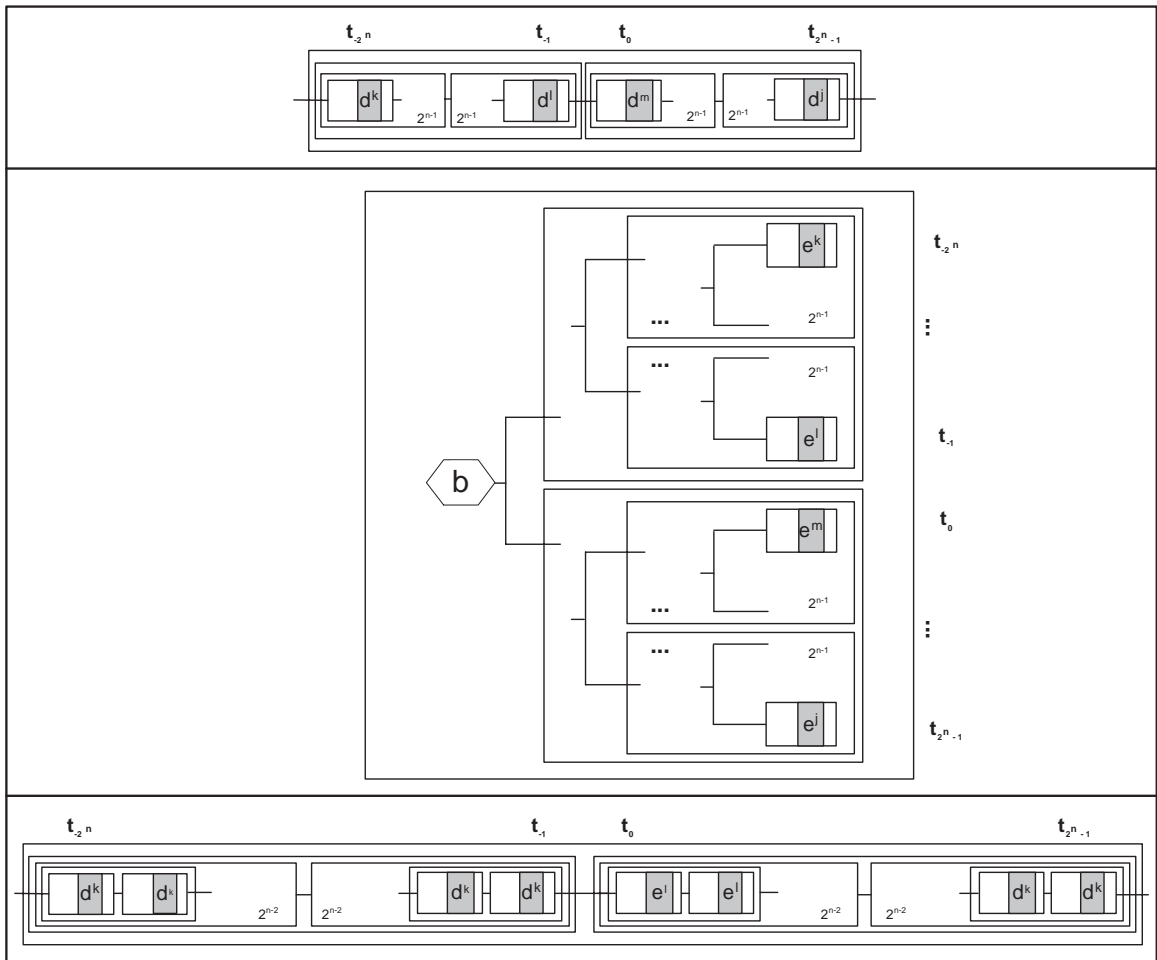


FIGURA 5.14 - Representação da Composição Finita das Imersões π_n^{\rightleftarrows} .

5.6.2 Projeções π_n^{\rightleftarrows}

Definição 5.17 Considere as imersões $\pi_n^{\rightarrow(\beta)}$, $\pi_n^{\leftarrow(\alpha)}$, as projeções $\Pi_n^{\rightarrow\leftarrow}$, Π_n^{\rightarrow} e Π_n^{\leftarrow} do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ e os indexadores $\alpha, \beta \in \{0, 1, 2\}$. As funções $\Pi_n^{\rightleftarrows(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\Pi_n^{\rightleftarrows(1)} : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\Pi_n^{\rightleftarrows(2)} : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas pela expressão

$$\Pi_n^{\rightleftharpoons(\beta)}(x) = \begin{cases} \Pi^{\rightleftharpoons(\beta)}(x) \cup \Pi_n^{\leftarrow(\beta)}(x), & \text{se } \Pi_n^{\rightarrow}(x) \notin \Gamma_n^{(0)}[\mathbb{P}_{n-1}], \\ \Pi^{\rightarrow(\beta)}(x), & \text{caso contrário.} \end{cases}$$

onde as funções $\Pi^{\rightleftharpoons(\beta)}, \Pi^{\leftarrow(\alpha)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ são dadas pelas respectivas composições

$$\Pi_n^{\rightleftharpoons(\beta)} = \pi_{n-1}^{\rightarrow(\beta)} \circ \Pi_{n-1}^{\rightarrow\leftarrow} \circ \Pi_n^{\rightarrow} \quad \text{e} \quad \Pi^{\leftarrow(\alpha)} = \pi_{n-1}^{\leftarrow(\alpha)} \circ \Pi_{n-1}^{\rightarrow\leftarrow} \circ \Pi_n^{\leftarrow}.$$

Quando $\beta = 0$, utiliza-se a notação $\Pi_n^{\rightleftharpoons(\beta)} = \Pi_n^{\rightleftharpoons}$.

Em particular, $\Pi_n^{\rightleftharpoons(\beta)}(\emptyset) = \emptyset$.

Proposição 5.14 *As projeções $\Pi_n^{\rightleftharpoons(\beta)}$ e as imersões $\pi_n^{\rightleftharpoons(\beta)}$ no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, apresentadas nas Definições 5.17 e 5.16 são morfismos na categoria CospLin .*

A demonstração é construída de forma análoga à demonstração da Proposição 5.9.

6 Completação do Espaço Coerente de Processos Gerados Indutivamente

A extensão do espaço coerente \mathbb{D}_n de processos finitamente gerados é formalizada com a construção e definição dos objetos e morfismos do espaço coerente \mathbb{D}_∞ e constitui-se no objeto de estudo deste capítulo.

Este novo espaço coerente, obtido pela completção, provê interpretação para todos os processos em \mathcal{P} , seja pela reconstrução dos objetos já definidos nos níveis \mathbb{D}_n , seja pela construção de novos objetos, aqueles que interpretam processos cuja execução não pode ser limitada no tempo.

A Figura 6.1 mostra o esquema completo da construção da estrutura sobre a qual se define a estrutura ordenada \mathbb{D}_∞ de uma máquina geométrica ¹, considerando-se a construção de processos por prefixação.

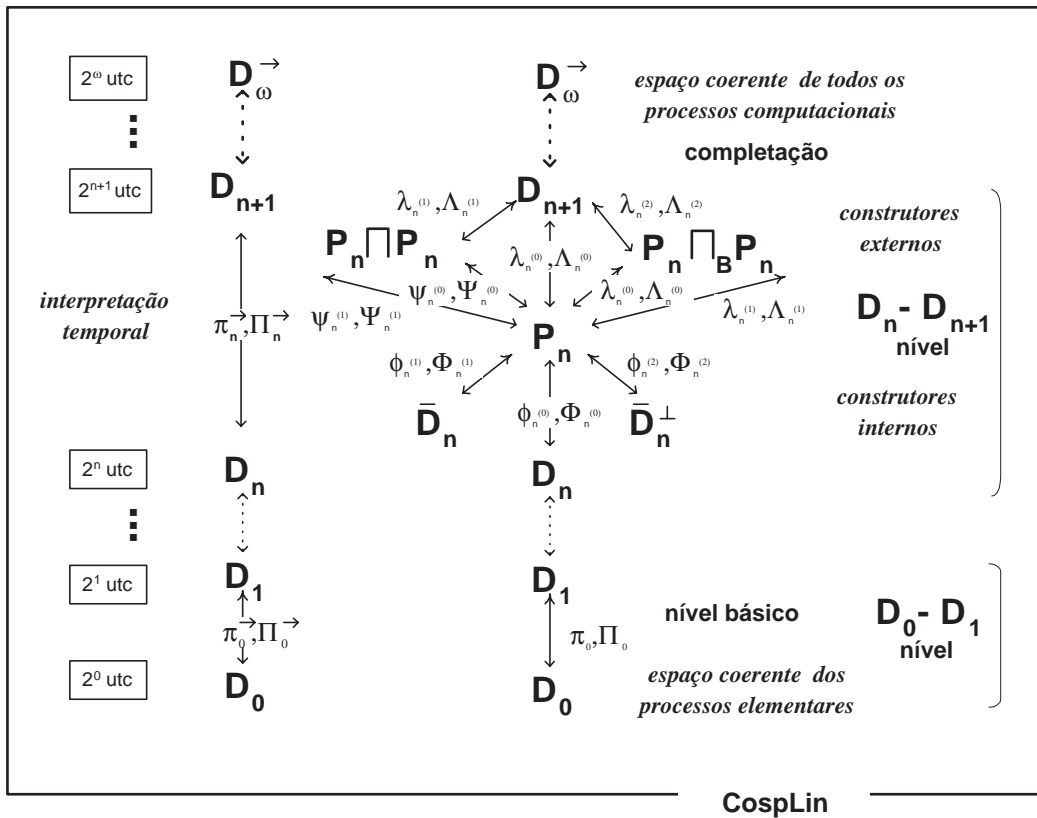


FIGURA 6.1 - Construção do Espaço Coerente de Processos \mathbb{D}_∞ .

Pela completção é possível interpretar cada processo em \mathcal{P} como um objeto do domínio \mathbb{D}_∞ , cuja estrutura ordenada e indutiva é suficientemente rica para explicitar aspectos importantes como :

¹Neste texto, utiliza-se a notação $\omega \equiv \infty$ nas representações gráficas.

- os **processos elementares** envolvidos na definição de cada processo, indicando as ações e posições que modelam as relações de concorrência e de conflito de acesso à memória (infinita);
- os **construtores** envolvidos na modelagem dos processos através da indexação associada a operadores lineares como a soma direta, o produto direto e operadores exponenciais, com manipulação de conjuntos-potência;
- a **complexidade de tempo** relacionada com o número máximo de operações associadas a cada execução (finita ou infinita) dos processos interpretados;
- a **quantidade de informação** que flui sobre cada processo caracterizada pela ordenação das interpretações;
- as **definições recursivas** associadas às interpretações dos processos modelados que estão fundamentadas na construção indutiva do modelo.

No que se refere a construção, todos os objetos em \mathbb{D}_∞ são definidos por suas aproximações - subconjuntos coerentes que interpretam processos parciais cuja execução se efetiva em tempo finito. Entretanto, isto não significa que tais objetos devam possuir representação finita no espaço geométrico ou no conjunto de ações.

Neste contexto, os novos objetos são obtidos como o limite inverso de seqüências de projeções definidas do espaço coerente \mathbb{D}_∞ sobre cada um dos subespaços \mathbb{D}_n . Mostra-se que estas projeções e as correspondentes imersões são morfismos na categoria *CospLin*.

A definição formal do espaço coerente \mathbb{D}_∞ de processos é apresentada a seguir e pode ser obtida de diferentes formas, dependendo do tipo de processo que se deseja interpretar.

De acordo com o que foi visto nos capítulos anteriores, busca-se uma interpretação para a memória e os processos computacionais posicionados em um espaço geométrico. Usufruindo desta construção geométrica e essencialmente posicional, todos os domínios que definem a estrutura indutiva deste modelo de máquina são capazes de interpretar as construções determinísticas, incluindo o paralelismo definido sobre estruturas matriciais que operam de forma independente e sincronizadas. Além destas, também as computações não-determinísticas, caracterizadas pelo conflito de acesso à memória, podem ser modeladas por esta estrutura construtiva.

No entanto, a construção temporal que leva à completção da estrutura indutiva deste modelo de máquina pode admitir diferentes abordagens, introduzindo diferentes domínios, sendo que alguns destes serão apresentados ao longo deste capítulo. As diferenças significativas entre tais domínios relacionam-se diretamente com às interpretações da seqüencialidade e da escolha determinística dos processos computacionais. Salienta-se ainda que cada domínio identifica uma enumeração distinta dos indexadores dos tokens de seus objetos.

Os domínios de interpretações abordados neste capítulo estão relacionados abaixo, seguidos de uma breve descrição.

1. D_∞^{\rightarrow} indica o espaço coerente onde os processos seqüenciais admitem a construção por prefixo e nas somas determinísticas se pode sempre determinar a primeira escolha mas não se conhece a última escolha;
2. D_∞^{\leftarrow} indica o espaço coerente onde os processos seqüenciais admitem a construção por sufixo e nas somas determinísticas se pode sempre determinar a última escolha, entretanto não é possível determinar a primeira escolha;
3. $D_\infty^{\leftrightarrow}$ indica o espaço coerente onde os processos seqüenciais admitem a construção por infixos e em somas determinísticas se pode sempre determinar as primeiras e as últimas escolhas;

4. $D_\infty^{\rightarrow\leftarrow}$ indica o espaço coerente onde os processos seqüenciais admitem a construção por prefixo e sufixo, simultâneamente;
5. $D_\infty^{\leftarrow\rightleftarrows}$ indica o espaço coerente onde os processos seqüenciais admitem a construção por prefixo, sufixo e infixos, simultâneamente.

6.1 Espaço Coerente \mathbb{D}_∞ dos Processos Computacionais

Esta seção introduz alguns conceitos importantes de forma genérica, os quais podem ser aplicados para qualquer uma das possíveis interpretações de processos que se deseja construir, utilizando este modelo de máquina.

Para tal, as imersões definidas no capítulo anterior serão, indistintamente, indicadas por $\pi_n : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e suas projeções por $\Pi_n : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$. Assim, $(\mathbb{D}_n, \mathbb{D}_{n+1})$ é um par-projeção onde $\pi_n \circ \Pi_n \subseteq \mathbb{D}_{n+1}$ e $\Pi_n \circ \pi_n = \mathbb{D}_n$.

A primeira definição apresentada nesta seção introduz o espaço coerente \mathbb{D}_∞ .

Definição 6.1 *O espaço coerente \mathbb{D}_∞ constitui-se no menor ponto fixo para a equação que define o espaço coerente dos processos finitos. Se $\mathbb{P}_\infty = \mathbb{D}_\infty \amalg \bar{\mathbb{D}}_\infty \amalg \bar{\mathbb{D}}_\infty^\perp$,*

$$\mathbb{D}_\infty = \mathbb{P}_\infty \amalg (\mathbb{P}_\infty \amalg \mathbb{P}_\infty) \amalg (\mathbb{B} \amalg \mathbb{P}_\infty \amalg \mathbb{P}_\infty).$$

As seções seguintes introduzem as extensões lineares $\pi_{n,\infty} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty$ para as imersões $\pi_n : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, e formalizam a noção de aproximação finita relacionada com a construção temporal. Todo subconjunto coerente em \mathbb{D}_n interpretando um processo cuja execução é finita – limitada por um intervalo ou instante de tempo, está associado a outro subconjunto em \mathbb{D}_∞ que interpreta a parte finita de outro processo, infinito no sentido temporal.

Desta forma, é possível se obter uma interpretação no domínio \mathbb{D}_∞ para todos processos em \mathcal{P} , incluindo produtos seqüenciais e somas determinísticas, executadas em tempo computacional infinito. Pode-se também interpretar um processo paralelo ou uma escolha não-determinística entre tais processos, desde que se possa garantir as respectivas condições de concorrência e conflito em \mathbb{D}_∞ .

As próximas definições formalizam estas idéias, enquanto as próximas seções especificam algumas das possíveis interpretações.

Definição 6.2 *Considere o espaço coerente \mathbb{D}_∞ e $x \in \mathbb{D}_\infty$ tal que $x = \bigcup_{n \in \omega}^\uparrow \{x_n\}$ onde $x_n \in \pi_{n,\infty}[\mathbb{D}_n]$. Considere também a função posição $\Upsilon_{\mathbb{D}_n}$ da Definição 5.9. A função $\Upsilon_{\mathbb{D}_\infty} : \mathbb{D}_\infty \rightarrow \wp(\omega)$ é definida por $\Upsilon_{\mathbb{D}_\infty}(x) = \bigcup \Upsilon_{\mathbb{D}_n}[\Pi_{\infty,n}(x)] = \bigcup \Upsilon_{\mathbb{D}_n}[\bigcup_{i \in \omega}^\uparrow \Pi_{\infty,n}(x_i)]$. Em particular, $\Upsilon_{\mathbb{D}_\infty}(\emptyset) = \emptyset$.*

Definição 6.3 *Seja $\bar{\mathbb{D}}_\infty \equiv \text{Coh}(\mathbb{D}_\infty)$ a família de subconjuntos coerentes construídos sobre o espaço coerente $\mathbb{D}_\infty \equiv ((\text{Coh}(\mathbb{D}_\infty), \subseteq)$.*

$\bar{\mathbb{D}}_\infty \equiv (\bar{\mathbb{D}}_\infty, \approx_{\bar{\mathbb{D}}_\infty})$ denota a teia cuja relação de coerência, que expressa a concorrência entre os processos interpretados pelos subconjunto coerentes x e y , é dada por

$$x \approx_{\bar{\mathbb{D}}_\infty} y \Leftrightarrow \Upsilon_{\mathbb{D}_\infty}(x) \cap \Upsilon_{\mathbb{D}_\infty}(y) = \emptyset \text{ ou } x = y, \quad \forall x, y \in \bar{\mathbb{D}}_\infty.$$

O espaço coerente cujos objetos são subfamílias de subconjuntos coerentes em \mathbb{D}_∞ construídos pela a relação $\approx_{\bar{\mathbb{D}}_\infty}$ é indicado por

$$\bar{\bar{\mathbb{D}}}_\infty \equiv (\text{Coh}\bar{\mathbb{D}}_\infty, \subseteq) \equiv (\text{Coh}(\bar{\mathbb{D}}_\infty, \approx_{\bar{\mathbb{D}}_\infty}), \subseteq).$$

A estrutura ordenada dual é construída sobre a teia complementar $\bar{\mathbb{D}}_\infty^\perp$, cuja relação de coerência, que expressa formalmente a relação de conflito entre os processos interpretados pelos conjuntos coerentes x e y , é definida por

$$x \approx_{\bar{\mathbb{D}}_\infty} y \Leftrightarrow \Upsilon_{\bar{\mathbb{D}}_\infty}(x) \cap \Upsilon_{\bar{\mathbb{D}}_\infty}(y) \neq \emptyset, \quad \forall x, y \in \bar{\mathbb{D}}_\infty.$$

O espaço coerente que interpreta a soma não-determinística de todos os processos (executados em tempo computacional finito ou infinito) é indicado por $\bar{\mathbb{D}}_\infty^\perp \equiv (\text{Coh}\bar{\mathbb{D}}_\infty^\perp, \subseteq) \equiv (\text{Coh}(\bar{\mathbb{D}}_\infty^\perp, \approx_{\bar{\mathbb{D}}_\infty}), \subseteq)$.

Neste momento, formaliza-se a noção de objeto compacto em \mathbb{D}_∞ e logo a seguir se mostra como são construídos os objetos totais (subconjuntos coerentes maximais em relação à inclusão).

6.1.1 Elementos Compactos do Domínio \mathbb{D}_∞

Um elemento $w \in \mathbb{D}_\infty$ é compacto se, e somente se, $w \subseteq \bigcup^\uparrow A \in \mathbb{D}_\infty$, onde A dirigido e existe $x \in \bigcup^\uparrow A$ tal que $w \subseteq x$.

Os objetos compactos em \mathbb{D}_∞ são construídos nos subespaços \mathbb{D}_n , ou seja, todo elemento compacto em \mathbb{D}_∞ interpreta um processo executado em um intervalo ou instante de tempo finito, logo $\mathbb{D}_\infty^c \subseteq \pi_{n,\infty}[\mathbb{D}_n]$.

A próxima proposição assegura que toda interpretação de um processo parcial cuja execução se efetiva em tempo finito, é um objeto compacto do domínio \mathbb{D}_∞ . A volta desta proposição nem sempre é verdadeira, o Exemplo 6.1, ao final deste capítulo, ilustra esta situação.

Proposição 6.1 *Considere a família \mathbb{D}_∞^c dos subconjuntos coerentes compactos do espaço coerente \mathbb{D}_∞ . Se $x \in \mathbb{D}_n^c$ então $\pi_{n,\infty}(x) \in \mathbb{D}_\infty^c$.*

6.1.2 Objetos Totais do Domínio \mathbb{D}_∞

Todo conjunto coerente $x \in \mathbb{D}_\infty$ é determinado como o limite de uma seqüência $\{x_n\}_{n \in \omega}$ de tal forma que $x_n \subseteq x$ e tem-se $x_n \in \pi_{n,\infty}[\mathbb{D}_n]$. Pela Proposição 6.1, cada $x_n \in \mathbb{D}_\infty^c$ constitui-se numa aproximação compacta para x . Além disso, conjuntos coerentes distintos correspondem a seqüências distintas, caracterizando a unicidade.

Considera-se agora a definição de objeto total:

Definição 6.4 *Um objeto $x \in \mathbb{D}_n$ é total se, e somente se, sempre que existe $y \in \mathbb{D}_n$ tal que $x \subseteq y$ tem-se que $x = y$.*

Pela definição anterior, um objeto total no espaço coerente \mathbb{D}_n é caracterizado por um conjunto maximal em relação à inclusão e interpreta um processo totalmente definido, executado em tempo finito. Os elementos (tokens) de um objeto total estão relacionados pela compatibilidade binária que caracteriza a teia \mathbf{D}_n do espaço coerente e não existe nenhum outro objeto deste espaço que se relacione com todos os tokens deste objeto que não esteja incluído nele.

O mesmo pode ser dito com relação aos objetos totais em \mathbb{D}_∞ , exceto que neste caso podem interpretar processos totalmente definidos, cujo tempo de execução pode ou não ser finito.

Salienta-se ainda que, tanto em \mathbb{D}_n como também em \mathbb{D}_∞ podem ocorrer objetos totais com um número finito de tokens. A Proposição 6.3 confirma esta idéia que se fundamenta no fato, já demonstrado em seções anteriores, de que as imersões são funções lineares, portanto monótonas e contínuas.

Nas próximas seções objetos maximais (sem limite de tempo de execução) são obtidos a partir da construção de diagramas definidos por composições infinitas de projeções. A construção dos objetos interpretados nestes diagramas leva em conta a união indexada relacionada com as possíveis projeções (direita e esquerda) quando se trata de processos seqüências ou (superior ou inferior) no caso de somas determinísticas, pois ambos construtores são interpretados por produtos diretos em $\mathcal{CospLin}$.

Encerra-se a primeira seção deste capítulo com a Tabela 6.1, que mostra um esquema reduzido contendo as categorias envolvidas na construção e completção do espaço coerente dos processos \mathbb{D}_∞ .

TABELA 6.1 - Categorias Relacionadas com a Construção do Modelo MG.

<i>Set</i>	<i>RGr</i>	<i>CospLin</i>
\mathcal{P}_∞	\mathbf{P}_∞	$\mathbb{P}_\infty = \mathbb{D}_\infty \amalg \bar{\mathbb{D}}_\infty \amalg \bar{\mathbb{D}}_\infty^\perp$
$\bar{\mathcal{D}}_\infty$	$\bar{\mathbf{D}}_\infty^\perp$	$\bar{\mathbb{D}}_\infty^\perp$
$\bar{\mathcal{D}}_\infty$	$\bar{\mathbf{D}}_\infty$	$\bar{\mathbb{D}}_\infty$
• \mathcal{D}_∞	\mathbf{D}_∞	$\mathbb{D}_\infty = \mathbb{P}_\infty \amalg (\mathbb{P}_\infty \amalg \mathbb{P}_\infty) \amalg (\mathbb{B} \amalg \mathbb{P}_\infty \amalg \mathbb{P}_\infty)$
\vdots	\vdots	\vdots
• \mathcal{D}_{n+1}	\mathbf{D}_{n+1}	$\mathbb{D}_{n+1} = \mathbb{P}_n \amalg (\mathbb{P}_n \amalg \mathbb{P}_n) \amalg (\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n)$
\mathcal{P}_n	\mathbf{P}_n	$\mathbb{P}_n = \mathbb{D}_n \amalg \bar{\mathbb{D}}_n \amalg \bar{\mathbb{D}}_n^\perp$
$\bar{\mathcal{D}}_n$	$\bar{\mathbf{D}}_n^\perp$	$\bar{\mathbb{D}}_n^\perp$
$\bar{\mathcal{D}}_n$	$\bar{\mathbf{D}}_n$	$\bar{\mathbb{D}}_n$
• \mathcal{D}_n	\mathbf{D}_n	$\mathbb{D}_n = \mathbb{P}_{n-1} \amalg (\mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1}) \amalg (\mathbb{B} \amalg \mathbb{P}_{n-1} \amalg \mathbb{P}_{n-1})$
\vdots	\vdots	\vdots
• \mathcal{D}_2	\mathbf{D}_2	$\mathbb{D}_2 = \mathbb{P}_1 \amalg (\mathbb{P}_1 \amalg \mathbb{P}_1) \amalg (\mathbb{B} \amalg \mathbb{P}_1 \amalg \mathbb{P}_1)$
\mathcal{P}_1	\mathbf{P}_1	$\mathbb{P}_1 = \mathbb{D}_1 \amalg \bar{\mathbb{D}}_1 \amalg \bar{\mathbb{D}}_1^\perp$
$\bar{\mathcal{D}}_1$	$\bar{\mathbf{D}}_1^\perp$	$\bar{\mathbb{D}}_1^\perp$
$\bar{\mathcal{D}}_1$	$\bar{\mathbf{D}}_1$	$\bar{\mathbb{D}}_1$
• \mathcal{D}_1	\mathbf{D}_1	$\mathbb{D}_1 = \mathbb{P}_0 \amalg (\mathbb{P}_0 \amalg \mathbb{P}_0) \amalg (\mathbb{P}_0 \amalg_{\mathbb{B}} \mathbb{P}_0)$
\mathcal{P}_0	\mathbf{P}_0	$\mathbb{P}_0 = \mathbb{D}_0 \amalg \bar{\mathbb{D}}_0 \amalg \bar{\mathbb{D}}_0^\perp$
$\bar{\mathcal{D}}_0$	$\bar{\mathbf{D}}_0^\perp$	$\bar{\mathbb{D}}_0^\perp$
$\bar{\mathcal{D}}_0$	$\bar{\mathbf{D}}_0$	$\bar{\mathbb{D}}_0$
• \mathcal{D}_0	\mathbf{D}_0	\mathbb{D}_0

$$\star \quad \bar{\mathcal{D}}_{n+1} = \text{Coh}(\mathbf{D}_n)$$

$$\star\star \quad \mathcal{D}_{n+1} = \mathcal{P}_n \dot{\cup} (\mathcal{P}_n \dot{\cup} \mathcal{P}_n) \dot{\cup} (\mathcal{B} \dot{\cup} \mathcal{P}_n \dot{\cup} \mathcal{P}_n)$$

$$\star\star\star \quad \mathcal{P}_n = \mathcal{D}_n \dot{\cup} \bar{\mathcal{D}}_n \dot{\cup} \bar{\mathcal{D}}_n$$

6.2 Espaço Coerente $\mathbb{D}_\infty^{(\theta)}$

Fixado o indexador $\theta \in \{00, 001, 011, 002, 012\}$, pelas Definições 6.1 e 6.3, tem-se

$$\mathbb{D}_\infty^{(\theta)} = \mathbb{P}_\infty^{(\theta)} \amalg (\mathbb{P}_\infty^{(\theta)} \amalg \mathbb{P}_\infty^{(\theta)}) \amalg (\mathbb{P}_\infty^{(\theta)} \amalg_{\mathbb{B}} \mathbb{P}_\infty^{(\theta)}),$$

sendo $\mathbb{P}_\infty^{(\theta)} = \mathbb{D}_\infty^{(\theta)} \amalg \overline{\mathbb{D}_\infty^{(\theta)}} \amalg \overline{\mathbb{D}_\infty^{(\theta)}}^\perp$.²

Embora restritos, a partir dos morfismos definidos sobre os domínios $\mathbb{D}_\infty^{(\theta)}$ são interpretados os demais construtores de processos. Justifica-se portanto, o estudo e análise das imersões e projeções $\pi_{n,\infty}^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{(\theta)}$ e $\Pi_{n,\infty}^{(\theta)} : \mathbb{D}_\infty^{(\theta)} \rightarrow \mathbb{D}_n$

No texto que se segue, utilizam-se as notações indicadas nos próximos itens.

1. Quando $\theta \in \{00, 001, 011, 002, 012\}$, na indexação dos tokens de subconjuntos coerentes em $\pi_{n,\infty}^{(\theta)}[\mathbb{D}_n]$ tem-se

$$: \theta \equiv \underbrace{\theta.\theta.\dots.\theta.\dots.}_{\text{infinitas vezes}}$$

2. Quando $\theta \in \{0\} \times (\{0\} \cup (\{0, 1\} \times \{1, 2\}))$, na composição finita das imersões $\pi_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ tem-se

$$\Pi_{m,n}^{(\theta)} = \begin{cases} id_{\mathbb{D}_n}, & \text{se } m \geq n \\ \pi_m^{(\theta)} \circ \pi_{n+1}^{(\theta)} \circ \dots \circ \pi_{n+1}^{(\theta)}, & \text{se } m < n. \end{cases}$$

$$\underbrace{\theta.\theta.\dots.\theta}_n \equiv \theta^n$$

3. A união dirigida relativa à inclusão \subseteq será denotada pela expressão $\bigcup_{n \in \omega}^\dagger$.

6.2.1 Imersões $\pi_{n,\infty}^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{(\theta)}$

Esta seção introduz as imersões $\pi_{n,\infty}^{(\theta)}$, que transformam objetos construídos nos subespaços \mathbb{D}_n em novos objetos em $\mathbb{D}_\infty^{(\theta)}$.

Embora as imersões $\pi_{n,\infty}^{(\theta)}$ sejam extensões das imersões já apresentadas na Definição 5.9, na seção 5.4.1, cada imersão identifica uma transformação distinta, conforme análise feita logo a seguir.

Neste sentido, um processo representado por um conjunto coerente x na imagem de uma imersão $\pi_{n,\infty}^{(\theta)}$ está posicionado no tempo em relação a um determinado instante t_0 , de tal forma que se pode determinar para qualquer parte finita, o intervalo de tempo (ou instante) em que esta aproximação será executada dentro deste processo.

Considerando-se que $\theta \in \{00, 001, 011, 002, 012\}$, estas imersões podem ocorrer de formas distintas, caracterizando interpretações que levam em conta, principalmente, o seu conteúdo de informação.

1. A primeira e mais simples função imersão interpreta um processo, cuja execução ocorre em tempo finito, por um subconjunto coerente de tokens indexados por $: 00 = 00.00. \dots$, indicando a composição infinita da função linear $\mathbf{F}_{(00)}$, que interpreta o operador inclusão. Neste caso, tem-se que $\pi_{n,\infty}^{(00)}(\mathbb{D}_n) \subseteq \mathbb{D}_\infty^{\rightrightarrows} \cap \mathbb{D}_\infty^{(00)}$.

Pode-se dizer que a indexação $: 00$ não altera os subconjuntos coerentes, apenas reescreve seus tokens indexando-os por uma expressão infinita. A ordenação destes objetos (parcial ou total) segundo o conteúdo de informação se mantém após

²Para simplificar a notação, $\mathbb{P}_\infty \amalg_{\mathbb{B}} \mathbb{P}_\infty \equiv \mathbb{P}_\infty \amalg_{\mathbb{B}} \mathbb{P}_\infty$.

imersão. Neste sentido, a análise da informação que flui pelos processos interpretados por tais objetos, estará disponível após um intervalo de tempo finito, que corresponde ao tempo computacional transcorrido entre o início e o término de sua execução.

As outras imersões modelam processos parciais resultantes da aplicação infinita dos morfismos que interpretam os construtores - produto seqüencial e soma determinística. Nestas situações, um conjunto coerente possui tokens indexados, onde cada índice é indicado por θ , sendo $\theta \in \{ : 001, : 011, : 002, : 012 \}$.

2. Os índices $: 001$ e $: 011$ identificam as imersões relacionadas com o primeiro e o segundo fator que compõe um processo resultante de um produto seqüencial. Portanto, tem-se as inclusões $\pi_{n,\infty}^{(001)}(\mathbb{D}_n) \subseteq \mathbb{P}_\infty^{(001)} \prod \mathbb{P}_\infty^{(001)}$ e $\pi_{n,\infty}^{(011)}(\mathbb{D}_n) \subseteq \mathbb{P}_\infty^{(011)} \prod \mathbb{P}_\infty^{(011)}$.

Por exemplo, as imersões $\pi_{n,\infty}^{(001)} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{(001)}$ transformam os objetos construídos em \mathbb{D}_n em novos objetos, os quais interpretam em $\mathbb{D}_\infty^{\rightarrow}$ processos seqüenciais parciais, que aproximam os processos infinitos no sentido temporal. Ou seja, no subespaço $\pi_{n,\infty}^{(001)}[\mathbb{D}_n] \subseteq \mathbb{D}_\infty^{(001)}$ pode-se interpretar produtos seqüenciais com prefixo finito, cujos infinitos fatores serão todos executados antes do instante t_0 .

Em outras palavras, pode-se olhar para tais imersões como morfismos que transformam processos finitos, que já foram construídos nos subespaços \mathbb{D}_n , em prefixos de novos processos seqüenciais parciais infinitos em $\mathbb{D}_\infty^{(001)}$. Destes processos parciais se conhece apenas um número finito de fatores. Neste caso, o primeiro fator que será executado é o processo interpretado por um subconjunto coerente em $\pi_{0,\infty}^{(001)}[\mathbf{D}_0]$, construído no nível \mathbf{D}_0 , no instante t_0 , e imerso em $\mathbb{D}_\infty^{(001)}$. A partir deste, a execução seqüencial dos demais fatores será feita obedecendo a ordem de construção.

Analogamente, as imersões $\pi_{n,\infty}^{(011)}$ interpretam transformações de processos finitos em sufixos de processos parciais em $\mathbb{D}_\infty^{(011)}$. Neste caso, destes processos seqüências parciais se conhece apenas um número finito de fatores, aqueles que correspondem aos últimos processos deste produto.

Agora ocorre o contrário, o último fator que será executado é o processo interpretado por um subconjunto coerente em $\pi_{0,\infty}^{(011)}[\mathbf{D}_0]$, construído no nível \mathbf{D}_0 , no instante t_{-1} , e imerso em $\mathbb{D}_\infty^{(011)}$. A partir deste, a execução seqüencial dos demais fatores será feita obedecendo a ordem inversa de construção.

Mais precisamente, a partir do último processo pode-se determinar o processo que será executado no instante de tempo anterior. Sendo assim, pela iteração da função imersão $\pi_{n,\infty}^{(011)}$ pode-se definir um processo seqüencial parcial, infinito no sentido temporal, do qual se conhece apenas uma parte finita, que será executada no intervalo de tempo $[t_{-n}, t_{-1}]$. Tal aproximação interpreta os (n) últimos processos do correspondente produto seqüencial infinito.

A construção (indutiva) destes processos define primeiramente o último processo do produto que é precedido pelo penúltimo, que por sua vez, é precedido pelo antepenúltimo, e assim continua, até atingir o processo executado no instante t_{-n} .

3. Analisando os objetos que interpretam as somas determinísticas, pode-se obter comentários semelhantes. Os indexadores $: 002$ e $: 012$, em uma soma determinística, identificam as imersões que admitem o primeiro ou o segundo termo na construção de uma soma.

Da mesma forma, tem-se as inclusões $\pi_{n,\infty}^{(002)}(\mathbb{D}_n) \subseteq \mathbb{B} \prod \mathbb{P}_\infty^{(002)} \prod \mathbb{P}_\infty^{(002)}$ e $\pi_{n,\infty}^{(012)}(\mathbb{D}_n) \subseteq \mathbb{B} \prod \mathbb{P}_\infty^{(012)} \prod \mathbb{P}_\infty^{(012)}$.

Salienta-se por fim que, na busca de interpretação para os processos infinitos no sentido do temporal estas imersões introduzem uma mudança em relação à informação que os novos subconjuntos coerentes passam a ter. Ou seja, pode ocorrer que objetos totais no sentido espacial em \mathbb{D}_n possam dar uma informação parcial sobre uma computação no sentido temporal em $\mathbb{D}_\infty^{(\theta)}$, como está ilustrado na Exemplicação 6.1.

As representações gráficas de processos interpretados pelas imersões $\pi_{0,\infty}^{(\theta)}$ aplicadas sobre um subconjunto coerente unitário, construído a partir de um token da teia \mathbf{D}_0 de processos elementares, podem ser observadas na Figura 6.2.

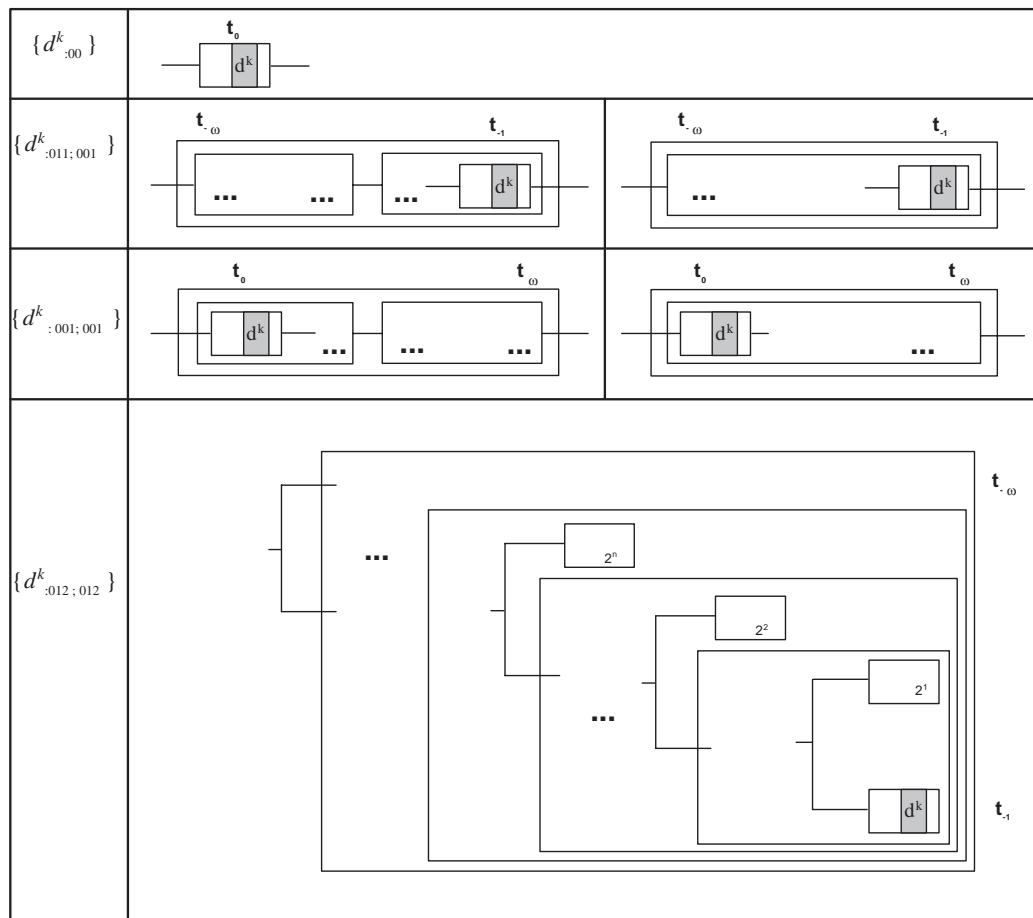


FIGURA 6.2 - Representação para Objetos em $\pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \in \mathbb{D}_\infty^{(\theta)}$.

Definição 6.5 Considere os espaços coerentes $\mathbb{D}_n, \mathbb{D}_\infty^{(\theta)}$. Considere também o indexador θ pertencente ao conjunto $\{00, 001, 011, 002, 012\}$. As imersões

$$\pi_{n,\infty}^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{(\theta)}$$

são definidas pela expressão $\pi_{n,\infty}^{(\theta)}(x) = \{\mathbf{a}.\theta \mid \mathbf{a} \in x\} = \{(\underbrace{\theta.\theta. \dots}_{\text{infinitas vezes}})\} \times x$. Salienta-

se que as imersões $\pi_{n,\infty}^{(\theta)}$ são estritas, $\pi_{n,\infty}^{(\theta)}[\emptyset] = \emptyset$.

Proposição 6.2 *Considere o indexador $\theta \in \{(00), (001), (011), (002), (012)\}$ e os espaços coerentes $\mathbb{D}_n, \mathbb{D}_\infty^{(\theta)}$. As imersões $\pi_{n,\infty}^{(\theta)}$ apresentadas na Definição 6.5 são monótonas, contínuas, estáveis e lineares.*

Demonstração. Fixado θ e por conseqüência \mathbb{D}_∞^θ na construção desta prova, indica-se $\pi_{n,\infty}^\theta$ simplesmente por $\pi_{n,\infty}$.

1. Sejam $x, x' \in \mathbb{D}_n$ e suponha $x \subseteq x'$. Tem-se que $\{\mathbf{a}.\theta \mid \mathbf{a} \in x\} \subseteq \{\mathbf{a}.\theta \mid \mathbf{a} \in x'\}$. Logo, $\pi_{n,\infty}(x) \subseteq \pi_{n,\infty}(x')$ e portanto $\pi_{n,\infty}$ é monotônica.
2. Suponha $X = \{x_i\}_{i \in \omega} \subseteq \mathbb{D}_n$ dirigido com relação à inclusão. Então pela monotonicidade da função $\pi_{n,\infty}$ acrescido do fato de que \mathbb{D}_∞ é fechado para uniões dirigidas, tem-se que $\pi_{n,\infty}[X] \subseteq \mathbb{D}_\infty$ é também dirigido e $\bigcup \pi_{n,\infty}[X] \in \mathbb{D}_\infty$. Então

$$\pi_{n,\infty}(\bigcup X) = \{\mathbf{a}.\theta \mid \mathbf{a} \in \bigcup X\} = \bigcup_i^\uparrow \{\mathbf{a}.\theta \mid \mathbf{a} \in x_i\} = \bigcup \pi_{n,\infty}[X].$$

Conclue-se que $\pi_{n,\infty}$ é contínua pois é monótona (1) e verifica-se a condição de continuidade.

3. Sejam $x, x' \in \mathbb{D}_n$ tais que $x \cup x' \in \mathbb{D}_n$. Sabendo que \mathbb{D}_n e \mathbb{D}_∞ são fechados para a interseção arbitrária, tem-se que $x \cap x' \in \mathbb{D}_n$ e $\pi_{n,\infty}(x) \cap \pi_{n,\infty}(x') \in \mathbb{D}_\infty$. Além disso,

$$\pi_{n,\infty}(x \cap x') = \{\mathbf{a}.\theta \mid \mathbf{a} \in x \cap x'\} = \{\mathbf{a}.\theta \mid \mathbf{a} \in x\} \cap \{\mathbf{a}.\theta \mid \mathbf{a} \in x'\}.$$

Logo $\pi_{n,\infty}(x \cap x') = \pi_{n,\infty}(x) \cap \pi_{n,\infty}(x')$ e portanto conclui-se que $\pi_{n,\infty}$ é estável pois é contínua (2) e verifica a condição de estabilidade.

4. Por fim, considere $X \subseteq \mathbb{D}_n$ tal que sempre que $x_0, x_1 \in X$ tem-se que $x_0 \cup x_1 \in \mathbb{D}_n$. Então

$$\pi_{n,\infty} \bigcup X = \{\mathbf{a}.\theta \mid \mathbf{a} \in \bigcup X\} = \bigcup \{\mathbf{a}.\theta \mid \mathbf{a} \in x \subseteq X\} = \bigcup \{\pi_{n,\infty}(x) \mid x \in X\}$$

o que mostra que $\pi_{n,\infty}$ satisfaz a linearidade.

Considerando os quatro itens, conclui-se que as funções $\pi_{n,\infty}^{(\theta)}$ são lineares, sempre que $\theta \in \{00, 001, 011, 002, 012\}$. \blacktriangle

Proposição 6.3 *Considere as famílias $\text{tot}(\mathbb{D}_n), \text{tot}(\mathbb{D}_\infty^{(00)})$ dos objetos totais em \mathbb{D}_n e $\mathbb{D}_\infty^{(00)}$, respectivamente. Sempre que $x \in \text{tot}(\mathbb{D}_n)$ tem-se que $\pi_{n,\infty}^{(00)}(x) \in \text{tot}(\mathbb{D}_\infty^{(00)})$.*

6.2.2 projeções $\Pi_{\infty,n}^{(\theta)}$

Esta seção introduz as projeções $\Pi_{\infty,n}^{(\theta)} : \mathbb{D}_\infty^{(\theta)} \rightarrow \mathbb{D}_n$ considerando-se as correspondentes imersões apresentadas na Definição 6.5.

Definição 6.6 *Considere o indexador $\theta \in \{00, 001, 011, 002, 012\}$, os espaços coerentes $\mathbb{D}_n, \mathbb{D}_\infty^{(\theta)}$ e $x = \bigcup_{i \in \omega}^\uparrow \{x_i\}$, onde cada $x_i \in \pi_{m,\infty}^{(\theta)}[\mathbb{D}_m]$ é uma aproximação finita de x . As projeções $\Pi_{\infty,n}^{(\theta)} : \mathbb{D}_\infty^{(\theta)} \rightarrow \mathbb{D}_n$ são definidas pela expressão*

$$\Pi_{\infty,n}^{(\theta)}(x) = \bigcup_{i \in \omega}^\uparrow (\Pi_{m,n}^{(\theta)} \circ \Pi_{\infty,m}^{(\theta)})(x_i),$$

sempre que $\Pi_{\infty,n}^{(\theta)}(x_i) = \{\mathbf{a} \mid \mathbf{a}.\theta \in x_i \text{ e } \mathbf{a} \in \mathbb{D}_n\}$. Em particular, $\Pi_{\infty,n}^{(\theta)}(\emptyset) = \emptyset$.

Proposição 6.4 *Considere o indexador $\theta \in \{(00), (001), (011), (002), (012)\}$ e os espaços coerentes $\mathbb{D}_n, \mathbb{D}_\infty^{(\theta)}$. As projeções apresentadas na Definição 6.6 são monótonas, contínuas, estáveis e lineares.*

Demonstração. Construída de forma análoga à apresentada na Proposição 6.2. ▲

A seguir são apresentadas algumas exemplificações da construção de objetos, com as respectivas representações gráficas das correspondentes interpretações em $\mathbb{D}_\infty^{\rightarrow}$ e $\mathbb{D}_\infty^{\leftarrow}$.

Exemplificação 6.1

Considere I enumerável e $d \in \mathcal{A}$. Define-se $z_n = \bigcup_{i=0}^n \{\overline{d^{(i)}}_{10}\} \in \mathbb{D}_1$ sendo $x_n = \pi_{1,\infty}(z_n) = \bigcup_{i=0}^n \{\overline{d^{(i)}}_{10:00}\} \in \mathbb{D}_\infty^{(00)}$ sua correspondente imersão. Então, tem-se

$$\bullet z_n = \{\overline{d^{(0)}}_{10}, \overline{d^{(1)}}_{10}, \dots, \overline{d^{(n)}}_{10}\} \Rightarrow x_n = \{\overline{d^{(0)}}_{10:00}, \overline{d^{(1)}}_{10:00}, \dots, \overline{d^{(n)}}_{10:00}\}$$

Neste caso, a seqüência de inclusões $\{z_n\}_{n \in \omega}$, $z_0 \subseteq z_1 \subseteq z_2 \subseteq \dots \subseteq z_n \subseteq \dots$ tem supremo, indicado por $z = \bigcup^\uparrow \{z_n\} \in \mathbb{D}_1$. Cada $z_n \in Coh[\mathbb{D}_1]$ é uma aproximação para $z \in tot(\mathbb{D}_1)$.

Analogamente, $x_n = \pi_{1,\infty}(z_n) = \{\mathbf{a}_{00} \mid \mathbf{a} \in z_n\} \in \mathbb{D}_\infty^{(00)}$ é também um objeto parcial e uma aproximação para $x = \bigcup^\uparrow \{x_n\}$. Cada subconjunto $x_n \in \mathbb{D}_\infty^{(00)}$ interpreta o processo que executa simultaneamente, no instante 0, a ação d nas n primeiras posições de memória.

Além disso, $\Upsilon_{\mathbb{D}_\infty^{(00)}}(x) = I$ e $x = \bigcup^\uparrow \pi_{1,\infty}^{(00)}(z_n) = \pi_{1,\infty}^{(00)}(\bigcup^\uparrow z_n) = \pi_{1,\infty}^{(00)}(z) \in \mathbb{D}_\infty^{(00)}$. Logo, $x \in \mathbb{D}_\infty^{\rightarrow}$ constitui-se num objeto total que modela o processo Q , o qual executa, simultaneamente, no instante 0, a ação d em todas as posições i de memória.

Agora, considere o subconjunto coerente $x'_n = \pi_{1,\infty}^{(001)}(z'_n) \in \mathbb{D}_\infty^{(001)}$ onde

$$\bullet z'_n = \bigcup_{i=0}^n \{\overline{d^{(i)}}_{101}, \overline{d^{(i)}}_{111}\} \in \mathbb{D}_1 \Rightarrow x'_n = \pi_{1,\infty}^{(001)}(z'_n) = \bigcup_{i=0}^n \{\overline{d^{(i)}}_{101:001}, \overline{d^{(i)}}_{111:001}\}.$$

Neste caso, x'_n é uma aproximação para $x' = \bigcup^\uparrow \{x'_n\} = \bigcup^\uparrow \pi_{1,\infty}^{(001)}(z'_n) \in \mathbb{D}_\infty$. Entretanto, $x' \in \mathbb{D}_\infty$ interpreta um processo parcial, indicado como o produto seqüencial Q' , possuindo infinitos fatores onde apenas os fatores executados nos instantes t_0 e t_1 são conhecidos. Os demais fatores de Q' ainda estão indefinidos.

Por fim, considerando-se o subconjunto coerente $x^{\mathbf{m}} \in \mathbb{D}_\infty^{(001)}$, dado pela união

$$x^{\mathbf{m}} = \bigcup_{n=0}^m \bigcup_{i \in I} \{\overline{d^{(i)}}_{(00)^n.111:001}\},$$

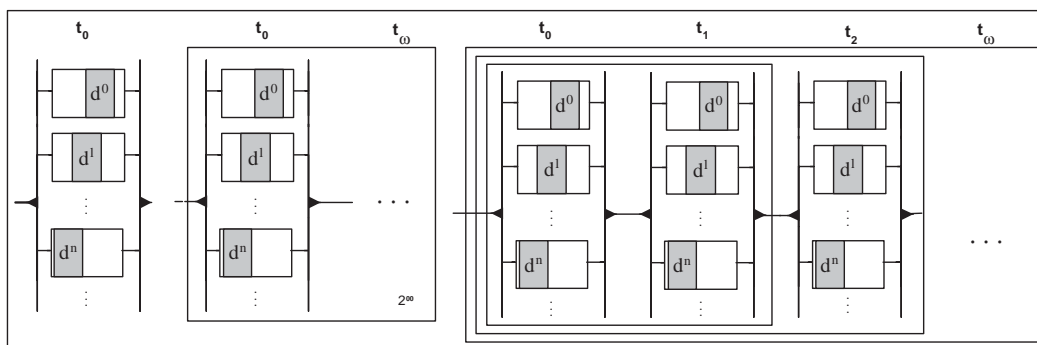
obtém-se $\mathbf{x} = \bigcup^\uparrow x^{\mathbf{m}} \in \mathbb{D}_\infty^{(00)}$, interpretando o produto seqüencial Q^∞ , cujos fatores (constantes) são executados em instantes de tempos distintos. Cada um destes fatores representa um processo que executa a ação d em todas as posições de memória, simultaneamente, em cada um destes instantes.

Encerramos este exemplo com a Figura 6.3 que apresenta uma representação dos processos Q , Q' e Q^∞ interpretados pelos subconjuntos coerentes x , x' e \mathbf{x} .

Exemplificação 6.2

- Seja $w_n \in \mathbb{D}_n$ tal que $\pi_{n,\infty}^\theta(w_n) \in \mathbb{D}_\infty^{(\theta)}$. Fixado $\theta = 00$ e considerando $\pi_{n,\infty}^{(00)} = \dots \circ \pi_{n+m}^{(00)} \circ \dots \circ \pi_{n+1}^{(00)} \circ \pi_n^{(00)}$, as seguintes implicações são verificadas:

$$\begin{aligned} \bullet w_0 = \{d^{(k)}\} &\rightarrow \pi_{0,\infty}^{(00)}(w_0) = \{d_{,00}^{(k)}\} \\ \bullet w_1 = \{d_{001}^{(k)}, d_{011}^{(k)}\} &\rightarrow \pi_{0,\infty}^{(00)}(w_1) = \{d_{001:00}^{(k)}, d_{011:00}^{(k)}\} \end{aligned}$$

FIGURA 6.3 - Representação para os Processos Q , Q' e Q^∞ .

$$\begin{aligned}
 \bullet w_2 &= \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, d_{00.011}^{(k)}\} \rightarrow \pi_{2,\infty}^{(00)}(w_2) = \{d_{(001)2:00}^{(k)}, d_{011.001:00}^{(k)}, d_{00.011:00}^{(k)}\} \\
 \bullet w_3 &= \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(k)}, d_{00.011.001}^{(k)}, d_{(00)2.011}^{(k)}\} \rightarrow \\
 &\pi_{3,\infty}^{(00)}(w_3) = \{d_{(001)3:00}^{(k)}, d_{011.(001)2:00}^{(k)}, d_{00.011.001:00}^{(k)}, d_{(00)2.011:00}^{(k)}\} \\
 &\vdots \\
 \bullet w_n &= \{d_{(001)n}^{(k)}, d_{011.(001)^{n-1}}^{(k)}, d_{00.011.(001)^{n-2}}, \dots, d_{(00)^{n-1}.011}^{(k)}\} \rightarrow \\
 &\pi_{n,\infty}^{(00)}(w_n) = w_n = \{d_{(001)n:00}^{(k)}, d_{011.(001)^{n-1}:00}^{(k)}, d_{00.011.(001)^{n-2}:00}, \dots, d_{(00)^{n-1}.011:00}^{(k)}\}
 \end{aligned}$$

- Se $:00 \equiv 00.00 \dots$ então $d_{:00}^{(k)} \neq d_{001:00}^{(k)} \neq d_{(001)2:00}^{(k)} \neq \dots \neq d_{(001)n:00}^{(k)}$.
- O processo $W00$, representado na Figura 6.4, é interpretado pelo conjunto coerente $\pi_{n,\infty}^{(00)}(w_n)$ dado por

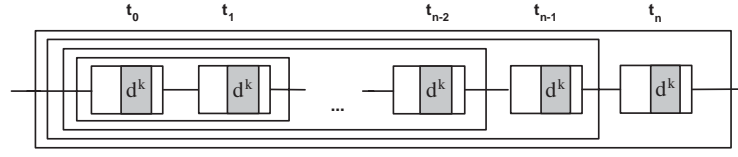
$$\underbrace{(((d^{(k)} \cdot d^{(k)}) \cdot d^{(k)}) \cdot \dots \cdot d^{(k)}) \cdot d^{(k)}}_{n+1 \text{ vezes}}$$

As imersões $\pi_{n,\infty}^{(00)}(w_n) \in \mathbb{D}_\infty^{(00)c}$ são objetos totais e interpretam processos cuja expressão é totalmente definida (sem indeterminação).

- Pode-se pensar em uma ordem de aproximação induzida pela inclusão, conforme é mostrado no Capítulo 9. Neste sentido, o processo interpretado pelo conjunto coerente $\pi_{n,\infty}^{(00)}(w_n)$ em $\mathbb{D}_\infty^{(00)}$ não é uma aproximação para o processo interpretado pelo conjunto coerente $\pi_{n+1,\infty}^{(00)}(w_{n+1})$, logo w_n também não aproxima w_{n+1} . Resumindo, $\pi_{n,\infty}^{(00)}(w_n) \not\subseteq \pi_{n+1,\infty}^{(00)}(w_{n+1}) \rightarrow w_n \not\subseteq w_{n+1}$.

Entretanto, é válida a igualdade $\Pi_{n+1,n}^{(001)}(w_{n+1}) = w_n$ e a inclusão $\pi_{n,\infty}^{(001)}(w_n) \subseteq \pi_{n+1,\infty}^{(001)}(w_{n+1})$, conforme pode ser observado na Exemplificação 6.4.

Exemplificação 6.3

FIGURA 6.4 - Representação do Processo $W00$.

- Seja $w'_n \in \mathbb{D}_n$ tal que $\pi_{n,\infty}^\theta(w'_n) \in \mathbb{D}_\infty$. Fixado $\theta = 00$ nas imersões $\pi_{n,\infty}^\theta$, as seguintes implicações são verificadas:
 - $w'_0 = \{d^{(k)}\} \rightarrow \pi_{0,\infty}^{(00)}(w'_0) = \{d_{:00}^{(k)}\}$
 - $w'_1 = \{d_{001}^{(k)}, d_{011}^{(k)}\} \rightarrow \pi_{1,\infty}^{(00)}(w'_1) = \{d_{001:00}^{(k)}, d_{011:00}^{(k)}\}$
 - $w'_2 = \{d_{00.001}^{(k)}, d_{001.011}^{(k)}, d_{(011)^2}^{(k)}\} \rightarrow \pi_{2,\infty}^{(00)}(w'_2) = \{d_{00.001:00}^{(k)}, d_{001.011:00}^{(k)}, d_{011.011:00}^{(k)}\}$
 - $w'_3 = \{d_{(00)^2.001}^{(k)}, d_{00.001.011}^{(k)}, d_{001.(011)^2}^{(k)}, d_{(011)^3}^{(k)}\} \rightarrow$
 $\pi_{3,\infty}^{(00)}(w'_3) = \{d_{(00)^2.001:00}^{(k)}, d_{00.001.011:00}^{(k)}, d_{001.(011)^2:00}^{(k)}, d_{(011)^3:00}^{(k)}\}$
 - \vdots
 - $w'_n = \{d_{(00)^{n-1}.001}^{(k)}, d_{(00)^{n-2}.001.011}^{(k)} \dots, d_{001.(011)^{n-1}}^{(k)}, d_{(011)^n}^{(k)}\} \rightarrow$
 $\pi_{n,\infty}^{(00)}(w'_n) = w'_n = \{d_{(00)^{n-1}.001:00}^{(k)}, d_{(00)^{n-2}.001.011:00}^{(k)} \dots, d_{001.(011)^{n-1}:00}^{(k)}, d_{(011)^n:00}^{(k)}\}$
- Se $:00 \equiv 00.00 \dots$ então $d_{:00}^{(k)} \neq d_{001:00}^{(k)} \neq d_{001.001:00}^{(k)} \neq \dots \neq d_{(001)^n:00}^{(k)}$.
- As imersões $\pi_{n,\infty}^{(00)}(w'_n) \in \mathbb{D}_\infty^{(00)c}$ também são objetos totais e interpretam processos cuja expressão é totalmente definida. O processo $W'00$, representado na Figura 6.4 e interpretado pelo conjunto coerente $\pi_{n,\infty}^{(00)}(w'_n)$ é indicado por

$$\underbrace{d^{(k)} \cdot (d^{(k)} \cdot \dots \cdot (d^{(k)} \cdot (d^{(k)} \cdot d^{(k)})))}_{n+1 \text{ vezes}}$$

- Verifica-se ainda a igualdade $\Pi_{n+1,n}^{(011)}(w'_{n+1}) = w_n$, embora as correspondentes inclusões não sejam válidas. No caso, $w'_n \not\subseteq w'_{n+1}$ e $\pi_{n,\infty}^{(00)}w'_n \not\subseteq \pi_{(n+1),\infty}^{(00)}w'_{n+1}$.
- Por último, considerando $w_0 = w'_0 = \{d^{(k)}\}$, salienta-se que
 - 1. $w'_{n+1} = \{d_{(00)^n}^{(k)}\} \times \{(001)\} \cup w'_n \times \{(011)\}$, e neste exemplo o novo processo w'_{n+1} é obtido ao se colocar $\{d_{(00)^n}^{(k)}\}$ como prefixo de w'_n ;
 - 2. $w_{n+1} = w_n \times \{(001)\} \cup \{d_{(00)^n}^{(k)}\} \times \{(011)\}$, e analogamente, na Exemplificação 6.2, tem-se w_{n+1} é obtido ao se colocar $\{d_{(00)^n}^{(k)}\}$ como sufixo de w'_n .

Exemplificação 6.4

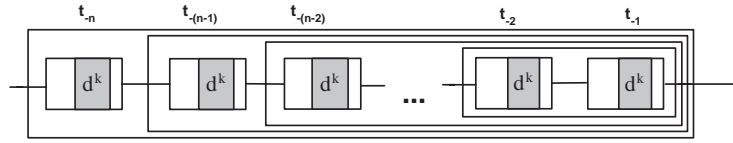


FIGURA 6.5 - Representação do Processo $W'00$.

- Fixando $\theta = 001$ nas próximas imersões $\pi_{n,\infty}^\theta$, tem-se as seguintes implicações:
 - $w_0 = \{d^{(k)}\} \rightarrow \pi_{0,\infty}^{(001)}(w_0) = \{d_{:001}^{(k)}\}$
 - $w_1 = \{d_{001}^{(k)}, d_{011}^{(k)}\} \rightarrow \pi_{1,\infty}^{(001)}(w_1) = \{d_{:001}^{(k)}, d_{011:001}^{(k)}\}$
 - $w_2 = \{d_{(001)2}^{(k)}, d_{011.001}^{(k)}, d_{00.011}^{(k)}\} \rightarrow \pi_{2,\infty}^{(001)}(w_2) = \{d_{:001}^{(k)}, d_{011:001}^{(k)}, d_{00.011:001}^{(k)}\}$
 - $w_3 = \{d_{(001)3}^{(k)}, d_{011.(001)2}^{(k)}, d_{00.011.001}^{(k)}, d_{(00)2.011}^{(k)}\} \rightarrow$
 $\pi_{3,\infty}^{(001)}(w_3) = \{d_{:001}^{(k)}, d_{011:001}^{(k)}, d_{00.011:001}^{(k)}, d_{(00)2.011:001}^{(k)}\}$
 - \vdots
 - $w_n = \{d_{(001)n}^{(k)}, d_{011.(001)^{n-1}}^{(k)}, d_{00.011.(001)^{n-2}}^{(k)}, \dots, d_{(00)^{n-1}.011}^{(k)}\} \rightarrow$
 $\pi_{n,\infty}^{(001)}(w_n) = \{d_{:001}^{(k)}, d_{011:001}^{(k)}, d_{00.011:001}^{(k)}, \dots, d_{(00)^{n-1}.011:001}^{(k)}\}$
- Se $:001 \equiv 001.001. \dots$ então $d_{:001}^{(k)} = d_{001:001}^{(k)} = d_{001.001:001}^{(k)} = \dots = d_{(001)^n:001}^{(k)}$ e portanto as seguintes inclusões são verificadas

$$\pi_{0,\infty}^{(001)}(w_0) \subseteq \pi_{1,\infty}^{(001)}(w_1) \subseteq \pi_{2,\infty}^{(001)}(w_2) \subseteq \dots \subseteq \pi_{n,\infty}^{(001)}(w_n) \in (\mathbb{D}_\infty^c).$$

- Verifica-se que cada um dos conjuntos coerentes $\pi_{n,\infty}^{(001)}(w_n)$ são objetos parciais em \mathbb{D}_∞ e interpretam processos indicados por expressões parcialmente definidas. O processo parcial $W001$ representado na Figura 6.6 é interpretado pelo conjunto coerente $\pi_{n,\infty}^{(001)}(w_n)$, sendo indicado pela expressão

$$\underbrace{d^{(k)} \cdot (d^{(k)} \cdot \dots \cdot (d^{(k)} \cdot (d^{(k)} \cdot \infty)))}_n$$

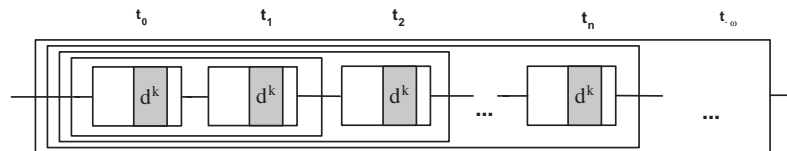


FIGURA 6.6 - Representação do Processo $W001$.

- Considerando uma ordem de aproximação induzida pela inclusão, pode-se dizer que o processo interpretado pelo conjunto coerente $\pi_{n,\infty}(w_n)$ é uma aproximação para o proceso interpretado pelo conjunto coerente $\pi_{n+1,\infty}(w_{n+1})$, pois tem-se a inclusão $w_n \subseteq w_{n+1}$ em $\mathbb{D}_\infty^{(001)}$.

- Além disso, tem-se $(\Pi_{\infty,n}^{(001)} \circ \pi_{n,\infty}^{(001)})(w_n) = w_n$.
- Por fim, consideram-se as seqüências
 - $\{w_n\}_{n \in \omega}$, cujos subconjuntos coerentes são tais que $w_n = \Pi_{n+1,n}^{(001)}(w_{n+1})$,
 - $\{\pi_{n,\infty}^{(001)}(w_n)\}_{n \in \omega}$ de imersões definidas de \mathbb{D}_n para $\mathbb{D}_\infty^{(001)}$.

Pela completção, tem-se assegurada a existência do supremo de cada uma destas seqüências ou subconjuntos dirigidos (pela inclusão) em $\mathbb{D}_\infty^{(001)}$. No caso,

$$\bigcup_{n \in \omega}^\uparrow \{w_n\} = \{d_{:001}^{(k)}, d_{011:001}^{(k)}, d_{00.011:001}^{(k)}, \dots, d_{(00)^{n-1}.011:001}^{(k)}, \dots\} = \bigcup_{n \in \omega}^\uparrow \{\pi_{n,\infty}^{(001)}(w_n)\}.$$

Exemplificação 6.5

- Fixando agora $\theta = 011$ nas imersões $\pi_{n,\infty}^\theta$, tem-se
 - $w'_0 = \{d^{(k)}\} \rightarrow \pi_{0,\infty}^{(011)}(w'_0) = \{d_{:011}^{(k)}\}$
 - $w'_1 = \{d_{001}^{(k)}, d_{011}^{(k)}\} \rightarrow \pi_{1,\infty}^{(011)}(w'_1) = \{d_{001:011}^{(k)}, d_{:011}^{(k)}\}$
 - $w'_2 = \{d_{00.001}^{(k)}, d_{001.011}^{(k)}, d_{011.011}^{(k)}\} \rightarrow \pi_{2,\infty}^{(011)}(w'_2) = \{d_{00.001:011}^{(k)}, d_{001.011:011}^{(k)}, d_{:001}^{(k)}\}$
 - $w'_3 = \{d_{00.00.001}^{(k)}, d_{00.001.011}^{(k)}, d_{001.011.011}^{(k)}, d_{011.011.011}^{(k)}\} \rightarrow$
 $\pi_{3,\infty}^{(001)}(w'_3) = \{d_{00.00.001:011}^{(k)}, d_{00.001:011}^{(k)}, d_{001:011}^{(k)}, d_{:011}^{(k)}\}$
 - \vdots
 - $w'_n = \{d_{(00)^{n-1}.001}^{(k)}, \dots, d_{001.011^{n-1}}^{(k)}, d_{(011)^n}^{(k)}\} \rightarrow$
 $\pi_{n,\infty}^{(001)}(w'_n) = \{d_{(001)^{n-1}:011}^{(k)}, \dots, d_{001:011}^{(k)}, d_{:011}^{(k)}\}$
- Se $:011 \equiv 011.011. \dots$ então $d_{:011}^{(k)} = d_{011:011}^{(k)} = d_{011.011:011}^{(k)} = \dots = d_{(011)^n:011}^{(k)}$ e portanto as seguintes inclusões são verificadas

$$\pi_{0,\infty}^{(011)}(w_0) \subseteq \pi_{1,\infty}^{(011)}(w_1) \subseteq \pi_{2,\infty}^{(011)}(w_2) \subseteq \dots \subseteq \pi_{n,\infty}^{(011)}(w_n) \in \mathbb{D}_\infty^{(001)\mathbf{c}}.$$

- O processo $W'011$ interpretado pelo conjunto coerente $\pi_{n,\infty}^{(011)}(w'_n)$ está representado na Figura 6.7 e é indicado pela expressão

$$\underbrace{(((\infty \cdot d^{(k)}) \cdot d^{(k)}) \cdot \dots \cdot d^{(k)}) \cdot d^{(k)}}_n$$

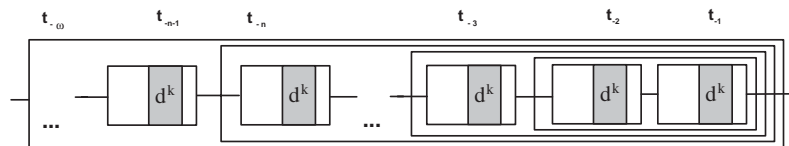


FIGURA 6.7 - Processo $W'011$.

- Verifica-se ainda que $w_n = \Pi_{n+1,n}^{(011)}(w_{n+1})$. Pela completação, existe o supremo para a seqüência $\{w_n\}_{n \in \omega}$ de subconjuntos coerentes $w_n \in \mathbb{D}_n$ e sua expressão é dada por

$$\bigcup_{n \in \omega}^\uparrow \{w_n\} = \{d_{:011}^{(k)}, d_{001:011}^{(k)}, d_{00.001:011}^{(k)}, \dots, d_{(00)^{n-1}.001:011}^{(k)}, \dots\} \in \mathbb{D}_\infty^{(011)}.$$

De acordo com sua construção indutiva, e correspondente completação, o modelo possui interpretação para composição infinita dos operadores que interpretam tanto produtos seqüenciais como somas determinísticas de processos computacionais.

Conforme pode ser observado nos exemplos, o operador produto seqüencial interpreta a operação de concatenação de strings (inclusive as infinitas) tanto à direita quanto à esquerda, caracterizada pelas indexações $:001$ e $:011$, respectivamente. As famílias de conjuntos coerentes resultantes da aplicação das imersões $\pi_{n,\infty}^{(001)}(w_n)$ e $\pi_{n,\infty}^{(011)}(w'_n)$ assim como $\pi_{n,\infty}^{(00)}(w_n)$ e $\pi_{n,\infty}^{(00)}(w'_n)$ são distintas e suas representações são também diferentes na linguagem, veja Capítulo 9. Entretanto, as interpretações semânticas coincidem, veja Capítulo 10. Considerando-se esta semântica associada, pode-se pensar em uma generalização da propriedade associativa, definida no caso, também para os objetos infinitos.

A seguir, representam-se os construtores (destrutores) de processos como imersões (projeções) definidas sobre os espaços coerentes $\mathbb{D}_\infty^\rightarrow$ e $\mathbb{D}_\infty^\leftarrow$. Nestes domínios, além da representação de todos os processos finitos, tem-se também a representação para processos seqüenciais pré e pós-fixados, respectivamente.

6.3 Espaços Coerentes $\mathbb{D}_\infty^\rightarrow$ e $\mathbb{D}_\infty^\leftarrow$

De acordo com as Definições 6.1 e 6.3, nesta seção são estudadas e analisadas as imersões e projeções definidas sobre espaços coerentes dados pelas equações

1. $\mathbb{D}_\infty^\rightarrow = \mathbb{P}_\infty^\rightarrow \amalg (\mathbb{P}_\infty^\rightarrow \amalg \mathbb{P}_\infty^\rightarrow) \amalg (\mathbb{P}_\infty^\rightarrow \amalg_{\mathbb{B}} \mathbb{P}_\infty^\rightarrow)$, onde $\mathbb{P}_\infty^\rightarrow = \mathbb{D}_\infty^\rightarrow \amalg \overline{\mathbb{D}_\infty^\rightarrow} \amalg \overline{\mathbb{D}_\infty^\rightarrow}^\perp$;
2. $\mathbb{D}_\infty^\leftarrow = \mathbb{P}_\infty^\leftarrow \amalg (\mathbb{P}_\infty^\leftarrow \amalg \mathbb{P}_\infty^\leftarrow) \amalg (\mathbb{P}_\infty^\leftarrow \amalg_{\mathbb{B}} \mathbb{P}_\infty^\leftarrow)$, onde $\mathbb{P}_\infty^\leftarrow = \mathbb{D}_\infty^\leftarrow \amalg \overline{\mathbb{D}_\infty^\leftarrow} \amalg \overline{\mathbb{D}_\infty^\leftarrow}^\perp$,

que contém um conjunto maior de interpretações construídas sobre os domínios $\mathbb{D}_\infty^{(\theta)}$.

6.3.1 Imersões $\pi_{n,\infty}^\rightarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\rightarrow$ e $\pi_{n,\infty}^\leftarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\leftarrow$

As imersões $\pi_{n,\infty}^\rightarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\rightarrow$ e $\pi_{n,\infty}^\leftarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\leftarrow$ são definidas nesta seção e se relacionam com os morfismos $\pi_n^\rightarrow, \pi_n^\leftarrow : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ na Definição 5.6.

Definição 6.7 *Considere os espaços coerentes $\mathbb{D}_n, \mathbb{D}_{n+1}, \mathbb{D}_\infty^\leftarrow$ e $\mathbb{D}_\infty^\rightarrow$ e as imersões $\pi_{n,\infty}^{(\theta)}$ apresentadas na Definição 6.5.*

Define-se a função imersão (estrita) $\pi_{n,\infty}^\rightarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\rightarrow$ pela expressão

$$\pi_{n+1,\infty}^\rightarrow(x) = \begin{cases} \pi_{n+1,\infty}^{(00)}(x), & \text{se } x \in \gamma_n^{(0)}[\mathbb{P}_n^\rightarrow], \\ \pi_{n+1,\infty}^{(001)}(x), & \text{se } x \in \gamma_n^{(1)}[\mathbb{P}_n^\rightarrow \amalg \mathbb{P}_n^\rightarrow], \\ \pi_{n+1,\infty}^{(002)}(x), & \text{se } x \in \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n^\rightarrow \amalg \mathbb{P}_n^\rightarrow], \\ \emptyset, & \text{caso contrário,} \end{cases}$$

da mesma forma que $\pi_{n,\infty}^\leftarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\leftarrow$ é dada pela expressão

$$\pi_{n+1,\infty}^\leftarrow(x) = \begin{cases} \pi_{n+1,\infty}^{(00)}(x) & \text{se } x \in \gamma_n^{(0)}[\mathbb{P}_n^\leftarrow], \\ \pi_{n+1,\infty}^{(011)}(x), & \text{se } x \in \gamma_n^{(1)}[\mathbb{P}_n^\leftarrow \amalg \mathbb{P}_n^\leftarrow], \\ \pi_{n+1,\infty}^{(012)}(x), & \text{se } x \in \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n^\leftarrow \amalg \mathbb{P}_n^\leftarrow], \\ \emptyset & \text{caso contrário,} \end{cases}$$

sabendo-se que, quando $x \in \mathbb{D}_0$, $\pi_{0,\infty}^{\rightarrow}(x) = \{\mathbf{a}_{:001} \mid \mathbf{a} \in x\}$ e $\pi_{0,\infty}^{\leftarrow}(x) = \{\mathbf{a}_{:011} \mid \mathbf{a} \in x\}$.

6.3.2 Projeções $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_n$ e $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_n$

Definição 6.8 Considere os espaços coerentes $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$ e as imersões $\gamma_n^{(\beta)}$, $\pi_{n,\infty}^{\leftarrow}$ e $\pi_{n,\infty}^{\rightarrow}$, respectivamente apresentadas na Proposição 5.1 e na Definição 6.7, sempre que $\beta \in \{0, 1, 2\}$. Considere também $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i \mid x_i \in \pi_{m,\infty}^{\rightarrow}[\mathbb{D}_m]\} \in \mathbb{D}_{\infty}^{\rightarrow}$ onde cada x_i é uma aproximação finita de x .

As projeções $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_n$ são definidas pela expressão $\Pi_{\infty,n}^{\rightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{\rightarrow}(x_i)$, sempre que

$$\Pi_{\infty,m}^{\rightarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x_i \in \pi_{m,\infty}^{(00)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\rightarrow}, \\ \{\mathbf{a} \mid \mathbf{a}_{:001} \in x \text{ e } \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x_i \in \pi_{m,\infty}^{(001)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\rightarrow} \prod \mathbb{P}_{\infty}^{\rightarrow}, \\ \{\mathbf{a} \mid \mathbf{a}_{:002} \in x \text{ e } \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x_i \in \pi_{m,\infty}^{(002)}[\mathbb{D}_m] \subseteq \mathbb{B} \prod \mathbb{P}_{\infty}^{\rightarrow} \prod \mathbb{P}_{\infty}^{\rightarrow}. \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Considere agora, $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i \mid x_i \in \pi_{m,\infty}^{\leftarrow}\} \in \mathbb{D}_{\infty}^{\leftarrow}$ onde cada x_i é uma aproximação finita de x . Da mesma forma, tem-se que as projeções $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_n$, são dadas pela expressão $\Pi_{\infty,n}^{\leftarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{m,n}^{\leftarrow} \circ \Pi_{\infty,m}^{\leftarrow}(x_i)$, sendo

$$\Pi_{\infty,n}^{\leftarrow}(x_i) = \begin{cases} (\{\mathbf{a} \mid \mathbf{a}_{:00} \in x \text{ e } \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x_i \in \pi_{m,\infty}^{(00)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\leftarrow}, \\ \{\mathbf{a} \mid \mathbf{a}_{:011} \in x \text{ e } \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x_i \in \pi_{m,\infty}^{(011)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\leftarrow} \prod \mathbb{P}_{\infty}^{\leftarrow}, \\ \{\mathbf{a} \mid \mathbf{a}_{:012} \in x \text{ e } \mathbf{a} \in \mathbf{D}_n\}, & \text{se } x_i \in \pi_{m,\infty}^{(012)}[\mathbb{D}_m] \subseteq \mathbb{B} \prod \mathbb{P}_{\infty}^{\leftarrow} \prod \mathbb{P}_{\infty}^{\leftarrow}. \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Em ambas definições, tem-se

1. $\Pi_{\infty,0}^{\rightarrow}(\{d_{:00}^{(k)}\}) = \Pi_{\infty,0}^{\rightarrow}(\{d_{:001}^{(k)}\}) = \Pi_{\infty,0}^{\rightarrow}(\{d_{:002}^{(k)}\}) = \{d^{(k)}\}$.
2. $\Pi_{\infty,0}^{\leftarrow}(\{d_{:00}^{(k)}\}) = \{d^{(k)}\}$ e $\Pi_{\infty,0}^{\leftarrow}(\{d_{:001}^{(k)}\}) = \Pi_{\infty,0}^{\leftarrow}(\{d_{:002}^{(k)}\}) = \emptyset$.
3. $\Pi_{\infty,0}^{\rightarrow}(\{d_{:011}^{(k)}\}) = \Pi_{\infty,0}^{\rightarrow}(\{d_{:012}^{(k)}\}) = \emptyset$.
4. $\Pi_{\infty,0}^{\leftarrow}(\{d_{:011}^{(k)}\}) = \Pi_{\infty,0}^{\leftarrow}(\{d_{:012}^{(k)}\}) = \{d^{(k)}\}$.

As definições apresentadas acima podem ser reescritas considerando-se as mesmas hipóteses. Neste caso, as projeções $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_n$ são dadas por $\Pi_{\infty,n}^{\rightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\rightarrow}(x_i)$, onde

$$\Pi_{\infty,n}^{\rightarrow}(x_i) = \begin{cases} (\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{(00)})(x_i), & \text{se } x_i \in \pi_{m,\infty}^{(00)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\rightarrow}, \\ (\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{(001)})(x_i), & \text{se } x_i \in \pi_{m,\infty}^{(001)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\rightarrow} \prod \mathbb{P}_{\infty}^{\rightarrow}, \\ (\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{(002)})(x_i), & \text{se } x_i \in \pi_{m,\infty}^{(002)}[\mathbb{D}_m] \subseteq \mathbb{B} \prod \mathbb{P}_{\infty}^{\rightarrow} \prod \mathbb{P}_{\infty}^{\rightarrow}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, tem-se $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_n$, podem ser expressas por $\Pi_{\infty,n}^{\leftarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\leftarrow}(x_i)$, sendo

$$\Pi_{\infty,n}^{\leftarrow}(x_i) = \begin{cases} (\Pi_{m,n}^{\leftarrow} \circ \Pi_{\infty,m}^{(00)})(x_i), & \text{se } x_i \in \pi_{m,\infty}^{(00)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\leftarrow}, \\ (\Pi_{m,n}^{\leftarrow} \circ \Pi_{\infty,m}^{(011)})(x_i), & \text{se } x_i \in \pi_{m,\infty}^{(011)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}^{\leftarrow} \prod \mathbb{P}_{\infty}^{\leftarrow}, \\ (\Pi_{m,n}^{\leftarrow} \circ \Pi_{\infty,m}^{(012)})(x_i), & \text{se } x_i \in \pi_{m,\infty}^{(012)}[\mathbb{D}_m] \subseteq \mathbb{B} \prod \mathbb{P}_{\infty}^{\leftarrow} \prod \mathbb{P}_{\infty}^{\leftarrow}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Proposição 6.5 *As projeções $\Pi_{\infty,n}^{\rightarrow}, \Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty} \rightarrow \mathbb{D}_n$ e as imersões $\pi_{\infty,n}^{\rightarrow}, \pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty} \rightarrow \mathbb{D}_n$, respectivamente apresentadas nas Definições 6.7 e 6.8, são morfismos na categoria CospLin .*

Demonstração. Restringe-se esta demonstração às projeções. Neste sentido, deve-se mostrar que as projeções $\Pi_{\infty,n}^{\rightarrow}$ são funções monótonas, contínuas, estáveis e lineares. As demonstrações para as demais funções são construções análogas.

1. Prova-se primeiramente que $\Pi_{\infty,n}^{\rightarrow}$ é monótona. Para tal, suponha $\{x_i\}_{i \in \omega}, \{x'_i\}_{i \in \omega}$ famílias dirigidas de subconjuntos coerentes e finitos tais que

$$x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\} \subseteq \bigcup_{i \in \omega}^{\uparrow} \{x'_i\} = x'.$$

Supondo que a composição finita $\Pi_{m,n}^{\rightarrow}$ é uma função monótona, e considerando o fecho inferior de $x' \in \mathbb{D}_{\infty}$, assegurando que para todo x'_i existe x_i e $x_i \subseteq x'_i$, uma das seguintes condições é sempre satisfeita

- (a) se $x_i, x'_i \in \pi_{m,\infty}^{(00)}[\mathbb{D}_m] \subseteq \mathbb{P}_{\infty}$, então

$$\Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:00 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}] \subseteq \Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:00 \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}];$$
- (b) se $x_i, x'_i \in \pi_{m,\infty}^{(001)}[\mathbb{D}_m] \subseteq [\mathbb{P}_{\infty} \prod \mathbb{P}_{\infty}]$,

$$\Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:001 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}] \subseteq \Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:001 \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}];$$
- (c) se $x_i, x'_i \in \pi_{m,\infty}^{(002)}[\mathbb{D}_m] \subseteq [\mathbb{B} \prod \mathbb{P}_{\infty} \prod \mathbb{P}_{\infty}]$,

$$\Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:002 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}] \subseteq \Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:002 \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_n\}];$$
- (d) caso contrário, é imediato que $\emptyset \subseteq \emptyset$.

Portanto, $\Pi_{\infty,n}^{\rightarrow}(x) \subseteq \Pi_{\infty,n}^{\rightarrow}(x')$.

2. A seguir, verifica-se a continuidade de $\Pi_{\infty,n}^{\rightarrow}$.

Seja $\mathcal{X} \neq \emptyset$ uma família dirigida, $\mathcal{X} \subseteq \mathbb{D}_{\infty}$. Para quaisquer $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\}$, $x' = \bigcup_{i \in \omega}^{\uparrow} \{x'_i\}$ em \mathcal{X} , tem-se que $x \cup x' \in \mathcal{X}$. Pela monotonicidade demonstrada no item anterior, $\Pi_{\infty,n}^{\rightarrow}(x) \subseteq \Pi_{\infty,n}^{\rightarrow}(x \cup x')$ e $\Pi_{\infty,n}^{\rightarrow}(x') \subseteq \Pi_{\infty,n}^{\rightarrow}(x \cup x')$. Portanto a família $\{\Pi_{\infty,n}^{\rightarrow}(\mathcal{X})\} \subseteq \mathbb{D}_n$ é dirigida. Considerando-se que \mathbb{D}_n é fechado para uniões dirigidas tem-se que $\bigcup^{\uparrow} \{\Pi_{\infty,n}^{\rightarrow}(\mathcal{X})\}_{i \in \omega} \in \mathbb{D}_n$.

Supondo que a composição finita $\Pi_{m,n}^{\rightarrow}$ é uma função contínua, nos próximos casos, aplica-se a definição da função $\Pi_{\infty,n}^{\rightarrow}$, tomando-se $\bigcup \mathcal{X}$ como argumento.

- (a) Se $\forall x_i \in \bigcup \mathcal{X}$, $x_i \in \pi_{m,\infty}^{(00)}[\mathbb{P}_m] \subseteq \mathbb{P}_n$, $\bigcup \mathcal{X} \in \mathbb{P}_{\infty}$ logo

$$(\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,n}^{\rightarrow})(\bigcup \mathcal{X}) = \bigcup \Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:00 \in x_i \subseteq \mathcal{X} \text{ e } \mathbf{a} \in \mathbf{D}_m\}];$$
- (b) Se $\forall x_i \in \bigcup \mathcal{X}$, $x_i \in \pi_{m,\infty}^{(001)}[\mathbb{P}_m] \subseteq \mathbb{P}_{\infty} \prod \mathbb{P}_{\infty}$, $\bigcup \mathcal{X} \in \mathbb{P}_{\infty} \prod \mathbb{P}_{\infty}$, logo

$$(\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,n}^{\rightarrow})(\bigcup \mathcal{X}) = \bigcup \Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:001 \in x_i \subseteq \mathcal{X} \text{ e } \mathbf{a} \in \mathbf{D}_m\}];$$
- (c) Se $\forall x_i \in \bigcup \mathcal{X}$, $x_i \in \pi_{m,\infty}^{(002)}[\mathbb{P}_m] \in [\mathbb{B} \prod \mathbb{P}_{\infty} \prod \mathbb{P}_{\infty}]$, $\bigcup \mathcal{X} \in \mathbb{B} \prod \mathbb{P}_{\infty} \prod \mathbb{P}_{\infty}$,

$$(\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,n}^{\rightarrow})(\bigcup \mathcal{X}) = \bigcup \Pi_{m,n}^{\rightarrow}[\{\mathbf{a} \mid \mathbf{a}:002 \in x_i \subseteq \mathcal{X} \text{ e } \mathbf{a} \in \mathbf{D}_m\}];$$
- (d) Caso contrário, é imediato que $\emptyset = (\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,n}^{\rightarrow})(\emptyset)$.

Em todos os casos, $\Pi_{\infty,n}^{\rightarrow}(\bigcup \mathcal{X}) = \bigcup \{\Pi_{\infty,n}^{\rightarrow}(x) \mid x \in \mathcal{X}\}$. Portanto $\Pi_{\infty,n}^{\rightarrow}$ é monótona e verifica a continuidade, comprovando que $\Pi_{\infty,n}^{\rightarrow}$ é contínua.

3. Mostra-se agora que $\Pi_{\infty,n}^{\rightarrow}$ verifica a propriedade da estabilidade. Suponha $\Pi_{m,n}$ estável e considere $x, x' \in \mathbb{D}_{\infty}$ tais que $x \cup x' \in \mathbb{D}_{\infty}$. Neste caso, seja $x'' = x \cap x' = \bigcup^{\uparrow} \{x''_i\}_{i \in \omega}$, onde $x''_i = x_i \cap x'_i, \forall i \in \omega$, considerando-se também que x_i, x'_i são aproximações finitas de x, x' , respectivamente.

(a) Se $x_i, x'_i \in \pi_{m,\infty}^{(00)}[\mathbb{P}_m], x''_i \in \pi_{m,\infty}^{(00)}[\mathbb{P}_m]$, logo

$$\Pi_{\infty,m}^{\rightarrow}(x''_i) = \{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_m\} \cap \{\mathbf{a} \mid \mathbf{a}_{:00} \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_m\};$$

(b) Se $x_i, x'_i \in \pi_{m,\infty}^{(001)}[\mathbb{P}_m], x''_i \in \pi_{m,\infty}^{(001)}[\mathbb{P}_m]$, logo

$$\Pi_{\infty,m}^{\rightarrow}(x''_i) = \{\mathbf{a} \mid \mathbf{a}_{:001} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_m\} \cap \{\mathbf{a} \mid \mathbf{a}_{:001} \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_m\};$$

(c) Se $x_i, x'_i \in \pi_{m,\infty}^{(002)}[\mathbb{P}_m], x''_i \in \pi_{m,\infty}^{(002)}[\mathbb{P}_m]$, logo

$$\Pi_{\infty,m}^{\rightarrow}(x''_i) = \{\mathbf{a} \mid \mathbf{a}_{:002} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_m\} \cap \{\mathbf{a} \mid \mathbf{a}_{:002} \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_m\};$$

(d) Caso contrário, $\Pi_{\infty,n}^{\rightarrow}(x \cap x') = \emptyset = \Pi_{\infty,n}^{\rightarrow}(x) = \Pi_{\infty,n}^{\rightarrow}(x')$.

Mostrou-se assim que, em todos os casos,

$$(\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{\rightarrow})(x''_i) = (\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,n}^{\rightarrow})(x_i) \cap (\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{\rightarrow})(x'_i),$$

implicando portanto, na seguinte igualdade

$$(\Pi_{m,n}^{\rightarrow} \circ \Pi_{\infty,m}^{\rightarrow})(\bigcup^{\uparrow} \{x''_i\}_{i \in \omega}) = \Pi_{m,n}^{\rightarrow}[\Pi_{\infty,n}^{\rightarrow}(\bigcup^{\uparrow} \{x_i\}_{i \in \omega}) \cap \Pi_{\infty,m}^{\rightarrow}(\bigcup^{\uparrow} \{x'_i\}_{i \in \omega})].$$

Portanto, $\Pi_{\infty,n}^{\rightarrow}(x'') = \Pi_{\infty,n}^{\rightarrow}(x \cap x') = \Pi_{\infty,n}^{\rightarrow}(x) \cap \Pi_{\infty,n}^{\rightarrow}(x')$. Assim, $\Pi_{\infty,n}^{\rightarrow}$ é contínua e satisfaz a propriedade da estabilidade, logo é uma função estável.

4. Para concluir, deve-se mostrar que $\Pi_{\infty,n}^{\rightarrow}$ satisfaz a propriedade da linearidade. Seja $\mathcal{X} \subseteq \mathbb{D}_{\infty}$ tal que para quaisquer $x, x' \in \mathcal{X}$, $x \cup x' \in \mathbb{D}_{\infty}$. Tomando-se $x \cup x' = \bigcup_{i \in \omega}^{\uparrow} \{x_i \cup x'_i\}$ como argumento para aplicação de $\Pi_{\infty,n}^{\rightarrow}$, tem-se

$$\begin{aligned} \Pi_{\infty,n}^{\rightarrow}(x \cup x') &= \{\mathbf{a} \mid \mathbf{a}_{:\theta} \in \bigcup_{i \in \omega}^{\uparrow} \{x_i \cup x'_i\}\} \\ &= \{\mathbf{a} \mid \mathbf{a}_{:\theta} \in \bigcup_{i \in \omega}^{\uparrow} \{x_i\}\} \cup \{\mathbf{a} \mid \mathbf{a}_{:\theta} \in \bigcup_{i \in \omega}^{\uparrow} \{x'_i\}\} \\ &= \Pi_{\infty,n}^{\rightarrow}(x) \cup \Pi_{\infty,n}^{\rightarrow}(x') \end{aligned}$$

sempre que $\theta \in \{00, 001, 002\}$. Acrescentando-se a esta última igualdade, a propriedade provada no item 2 e considerando-se ainda o fato de $\Pi_{\infty,n}^{\rightarrow}(\emptyset) = \emptyset$, conclui-se que $\Pi_{\infty,n}^{\rightarrow}(\bigcup \mathcal{X}) = \bigcup \{\Pi_{\infty,n}^{\rightarrow}(x) \mid x \in \mathcal{X}\}$. Portanto $\Pi_{\infty,n}^{\rightarrow}$ é linear. \blacktriangle

Proposição 6.6 *Considere as projeções $\Pi_{\infty,n}^{\rightarrow}, \Pi_{\infty,n}^{\leftarrow}$ e as imersões $\pi_{\infty,n}^{\rightarrow}, \pi_{\infty,n}^{\leftarrow}$, respectivamente apresentadas nas Definições 6.7 e 6.8. As seguintes condições são satisfeitas:*

1. $\Pi_{\infty,n}^{\rightarrow} \circ \pi_{\infty,n}^{\rightarrow} = \mathbb{D}_n$ and $\pi_{\infty,n}^{\rightarrow} \circ \Pi_{\infty,n}^{\rightarrow} \subseteq \mathbb{D}_{\infty}^{\rightarrow}$, e
2. $\Pi_{\infty,n}^{\leftarrow} \circ \pi_{\infty,n}^{\leftarrow} = \mathbb{D}_n$ and $\pi_{\infty,n}^{\leftarrow} \circ \Pi_{\infty,n}^{\leftarrow} \subseteq \mathbb{D}_{\infty}^{\leftarrow}$,

Portanto, $(\mathbb{D}_{\infty}^{\rightarrow}, \mathbb{D}_n)$ e $(\mathbb{D}_{\infty}^{\leftarrow}, \mathbb{D}_n)$ são pares-projeção na categoria CospLin .

6.3.3 Cone Limite $(\mathbb{D}_\infty^\rightarrow, \Pi_{\infty,n}^\rightarrow : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_n)$

As próximas definições levam em conta as imersões $\pi_{n,\infty}^\rightarrow : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^\rightarrow$ e as projeções $\Pi_{\infty,n}^\rightarrow : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_n$ apresentadas nas Definições 6.7 e 6.8.

Definição 6.9 *O diagrama \mathcal{K}^\rightarrow , representado por*

$$\mathbb{D}_0 \xleftarrow{\Pi_1^\rightarrow} \mathbb{D}_1 \xleftarrow{\quad} \dots \xleftarrow{\quad} \mathbb{D}_n \xleftarrow{\Pi_{n+1}^\rightarrow} \mathbb{D}_{n+1} \xleftarrow{\quad} \dots$$

é definido pela família de subespaços coerentes $\{\mathbb{D}_n\}_{n \in \omega}$ e pelas projeções $\Pi_n^\rightarrow : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas na Definição 5.7, de tal forma que para todo $w_{n+1} \in \mathbb{D}_{n+1}$ existe $w_n \in \mathbb{D}_n$ e $\Pi_n^\rightarrow(w_{n+1}) = w_n$.

Neste sentido, o diagrama \mathcal{K}^\rightarrow determina seqüências $\{w_n\}_{n \in \omega}$ de subconjuntos coerentes $w_n \in \mathbb{D}_n$ tal que $\Pi_n^\rightarrow(w_{n+1}) = w_n$.

Como conseqüência, cada seqüência $\{w_n\}_{n \in \omega}$ também determina outra seqüência $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_\infty^\rightarrow$ de aproximações finitas do conjunto coerente x , ou seja $x = \bigcup_{n \in \omega}^\uparrow x_n$, onde, para cada subconjunto coerente x_n existe $w_n \in \mathbb{D}_n$ tal que $\Pi_{\infty,n}^\rightarrow(x_n) = w_n$. Mais detalhadamente, pela Definição 6.8, $w_n = \Pi_n^\rightarrow(x_n) \Leftrightarrow \pi_{n,\infty}^\rightarrow(w_n) \subseteq x_n \subseteq x$, garantindo que uma das seguintes condições, é sempre satisfeita

1. $\pi_{n,\infty}^\rightarrow(w_n) = \{\mathbf{a}:00 \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$, ou
2. $\pi_{n,\infty}^\rightarrow(w_n) = \{\mathbf{a}:001 \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$, ou
3. $\pi_{n,\infty}^\rightarrow(x_n) = \{\mathbf{a}:002 \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$.

Portanto, a partir da seqüência $\{w_n\}_{n \in \omega}$, define-se a seqüência $\{x_n\}_{n \in \omega}$, cujos subconjuntos coerentes determinam um novo objeto $x \in \mathbb{D}_\infty^\rightarrow$, dado por

$$x = \bigcup_{n \in \omega}^\uparrow \{x_n\} \Leftrightarrow x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \subseteq x \text{ e } \Pi_{\infty,n}^\rightarrow(x_n) = w_n.$$

A seguir, define-se o limite inverso para o diagrama \mathcal{K}^\rightarrow que assegura interpretação para todos os processos, incluindo os processos seqüenciais infinitos no sentido temporal, mas que são executados a partir de um instante t_0 . As exemplificações 6.1 e 6.4, apresentadas no final deste capítulo, ilustram este caso.

Proposição 6.7 *Considere a família de projeções $\Pi_{\infty,n}^\rightarrow : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_n$, apresentada na Definição 6.8. O par $\mathbb{D}^\rightarrow \equiv (\mathbb{D}_\infty^\rightarrow, \Pi_{\infty,n}^\rightarrow : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama \mathcal{K}^\rightarrow apresentado na Definição 6.9.*

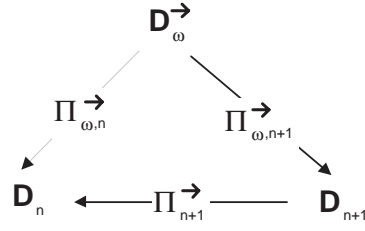
Demonstração. Considere os espaços coerentes \mathbb{D}_n e $\mathbb{D}_\infty^\rightarrow$ e as funções lineares $\Pi_{\infty,n}^\rightarrow : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_n$ e $\Pi_n^\rightarrow : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas nas Definições 6.8 e 5.7, respectivamente.

1. Mostra-se primeiramente que $\Pi_{\infty,n}^\rightarrow = \Pi_{n+1,n}^\rightarrow \circ \Pi_{\infty,n+1}^\rightarrow$, ou seja, satisfaz a comutatividade do diagrama

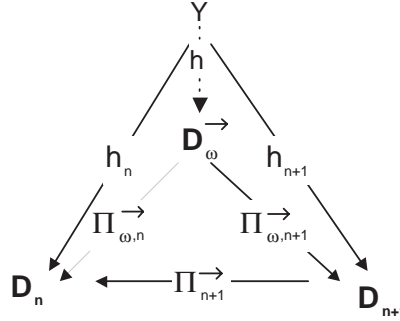
De fato, seja $x = \bigcup_{i \in \omega}^\uparrow \{x_i\} \in \mathbb{D}_\infty^\rightarrow$. Tem-se que

$$\begin{aligned} \Pi_{n+1,n}^\rightarrow \circ \Pi_{\infty,n+1}^\rightarrow(x) &= (\Pi_{n+1,n}^\rightarrow \circ \Pi_{\infty,n+1}^\rightarrow)(\bigcup_{i \in \omega}^\uparrow \{x_i\}) \\ &= \bigcup_{i \in \omega}^\uparrow \{(\Pi_{n+1,n}^\rightarrow \circ \Pi_{\infty,n+1}^\rightarrow)(x_i)\} \\ &= \bigcup_{i \in \omega}^\uparrow \{\Pi_{n+1,n}^\rightarrow(w_{n+1})\} = \bigcup_{i \in \omega}^\uparrow \{w_n\} \\ &= \bigcup_{i \in \omega}^\uparrow \{\Pi_{\infty,n}^\rightarrow(x_i)\} = \Pi_{\infty,n}^\rightarrow(\bigcup_{i \in \omega}^\uparrow \{x_i\}) \end{aligned}$$

Logo $\Pi_{n+1,n}^\rightarrow \circ \Pi_{\infty,n+1}^\rightarrow(x) = \Pi_{\infty,n}^\rightarrow(x)$, o que completa esta parte da demonstração.



2. Para finalizar, deve-se mostrar que o cone $(\mathbb{D}_\infty, \Pi_{\infty,n}^\rightarrow : \mathbb{D}_\infty \rightarrow \mathbb{D}_n)$ é um limite para os demais cones em relação ao diagrama \mathcal{K}^\rightarrow . Para tal, considere um outro cone $(\mathbb{Y}, f_n : \mathbb{Y} \rightarrow \mathbb{D}_n)$ para o diagrama \mathcal{K}^\rightarrow . Seja $h : \mathbb{Y} \rightarrow \mathbb{D}_\infty$ definida pela expressão $h(y) = \bigcup_{i \in \omega}^\uparrow \{\mathbf{a} \cdot \theta \mid \mathbf{a} \in f_n(y)\}$. É fácil verificar que $h \in Mor_{\mathcal{C}ospLin}$ e que $\Pi_{\infty,n}^\rightarrow \circ h = f_n$, ou seja, h verifica a comutatividade do diagrama



Mostra-se a seguir que h é única. Suponha $g : \mathbb{Y} \rightarrow \mathbb{D}_\infty$ que também verifica a comutatividade do último diagrama, ou seja, $\Pi_{\infty,n}^\rightarrow \circ g = f_n$, mas tal que existe $y \in \mathbb{Y}$ e $g(y) \neq f(y)$. Neste caso, tem-se que $(\Pi_{\infty,n}^\rightarrow \circ g)(y) \neq (\Pi_{\infty,n}^\rightarrow \circ f)(y)$ Supondo $g(y) = \bigcup_{i \in \omega}^\uparrow \{x_i\}$ e $h(y) = \bigcup_{i \in \omega}^\uparrow \{x'_i\}$, tem-se que

$$\Pi_{\infty,n}^\rightarrow(\bigcup_{i \in \omega}^\uparrow \{x_i\}) = \Pi_{\infty,n}^\rightarrow(\bigcup_{i \in \omega}^\uparrow \{x'_i\})$$

Aplicando a definição de $\Pi_{\infty,n}^\rightarrow$, resulta $\bigcup_{i \in \omega}^\uparrow \{\Pi_{\infty,n}^\rightarrow(x_i)\} = \bigcup_{i \in \omega}^\uparrow \{\Pi_{\infty,n}^\rightarrow(x'_i)\}$. Considerando a função $\pi_{n,\infty}^\rightarrow$ apresentada na Definição 6.7 e a Proposição 6.5, tem-se que $\pi_{n,\infty}^\rightarrow(\bigcup_{i \in \omega}^\uparrow \{\Pi_{\infty,n}^\rightarrow(x_i)\}) = \pi_{n,\infty}^\rightarrow(\bigcup_{i \in \omega}^\uparrow \{\Pi_{\infty,n}^\rightarrow(x'_i)\})$ Portanto, $\bigcup_{i \in \omega}^\uparrow \{x_i\} = \bigcup_{i \in \omega}^\uparrow \{x'_i\} \Leftrightarrow g(y) = f(y), \forall y \in \mathbb{Y}$.

Demonstrou-se assim que o par $(\mathbb{D}_\infty, \Pi_{\infty,n} : \mathbb{D}_\infty \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama \mathcal{K}^\rightarrow . \blacktriangle

6.3.4 Cone Limite $(\mathbb{D}_\infty^\leftarrow, \Pi_{\infty,n}^\leftarrow : \mathbb{D}_\infty^\leftarrow \rightarrow \mathbb{D}_n)$

A seguir, define-se o limite inverso para o diagrama \mathcal{K}^\leftarrow que assegura interpretação para todos os processos, incluindo os processos seqüenciais infinitos no sentido temporal, mas que são executados antes de um instante t_0 . Tais processos seqüenciais infinitos não possuem um começo mas certamente terminam no instante t_{-1} . Este é o caso da exemplificação 6.5, já apresentada neste capítulo.

Definição 6.10 O diagrama \mathcal{K}^{\leftarrow} , representado por

$$\mathbb{D}_0 \xleftarrow{\Pi_0^{\leftarrow}} \mathbb{D}_1 \longleftarrow \dots \longleftarrow \mathbb{D}_n \xleftarrow{\Pi_n^{\leftarrow}} \mathbb{D}_{n+1} \longleftarrow \dots$$

é definido pela família de subespaços coerentes $\{\mathbb{D}_n\}_{n \in \omega}$ e pelas projeções $\Pi_n^{\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas na Definição 5.7, de tal forma que para todo $w_{n+1} \in \mathbb{D}_{n+1}$ existe $w_n \in \mathbb{D}_n$ e $\Pi_n^{\leftarrow}(w_{n+1}) = w_n$.

Neste sentido, o diagrama \mathcal{K}^{\leftarrow} determina seqüências $\{w_n\}_{n \in \omega}$ de subconjuntos coerentes $w_n \in \mathbb{D}_n$ tal que $\Pi_{\leftarrow n}(w_{n+1}) = w_n$.

Como conseqüência, cada seqüência $\{w_n\}_{n \in \omega}$ também determina outra seqüência $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_\infty$ de aproximações finitas do conjunto coerente x , ou seja $x = \bigcup_{n \in \omega}^\uparrow x_n$, onde, para cada subconjunto coerente x_n existe $w_n \in \mathbb{D}_n$ tal que $\Pi_{n,\infty}^{\leftarrow}(x_n) = w_n$. Mais detalhadamente, pela Definição 6.8,

$$w_n = \Pi_n^{\leftarrow}(x_n) \Leftrightarrow \pi_{n,\infty}^{\leftarrow}(w_n) \subseteq x_n \subseteq x,$$

garantindo que uma das seguintes condições é sempre satisfeita

1. $\pi_{n,\infty}^{\leftarrow(00)}(w_n) = \{\mathbf{a}_{:00} \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$, ou
2. $\pi_{n,\infty}^{\leftarrow(011)}(w_n) = \{\mathbf{a}_{:011} \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$, ou
3. $\pi_{n,\infty}^{\leftarrow(012)}(x_n) = \{\mathbf{a}_{:012} \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$.

Portanto, a partir da seqüência $\{w_n\}_{n \in \omega}$, define-se a seqüência $\{x_n\}_{n \in \omega}$, cujos subconjuntos coerentes determinam um novo objeto $x \in \mathbb{D}_\infty$, dado por

$$x = \bigcup_{n \in \omega}^\uparrow \{x_n\} \Leftrightarrow x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \subseteq x \text{ e } \Pi_{\infty,n}^{\leftarrow}(x_n) = w_n.$$

A seguir, define-se o limite inverso para o diagrama \mathcal{K}^{\leftarrow} .

Proposição 6.8 Considere a família de projeções $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty \rightarrow \mathbb{D}_n$, apresentada na Definição 6.8. O par $\mathbb{D}^{\leftarrow} \equiv (\mathbb{D}_\infty, \Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama \mathcal{K}^{\leftarrow} apresentado na Definição 6.10.

Proposição 6.9 Considere os cones $\mathbb{D}^{\rightarrow} \equiv (\mathbb{D}_\infty, \Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_\infty \rightarrow \mathbb{D}_n)$ e $\mathbb{D}^{\leftarrow} \equiv (\mathbb{D}_\infty, \Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty \rightarrow \mathbb{D}_n)$, apresentados nas Seções 6.3.3 e 6.3.4. A função linear $\mathbf{H}_{\mathbb{D}_\infty, \mathbb{D}_\infty}^{\rightarrow, \leftarrow} : \mathbb{D}_\infty \rightarrow \mathbb{D}_\infty$ definida pela expressão

$$\mathbf{H}_{\mathbb{D}_\infty, \mathbb{D}_\infty}^{\rightarrow, \leftarrow}(x) = \begin{cases} x, & \text{se } x \in \mathbb{P}_\infty^{\rightarrow}, \\ \{\mathbf{a}_{:011} \mid \mathbf{a}_{:001} \in x\}, & \text{se } x \in \mathbb{P}_\infty^{\rightarrow} \prod \mathbb{P}_\infty^{\rightarrow}, \\ \{\mathbf{a}_{:012} \mid \mathbf{a}_{:002} \in x\}, & \text{se } x \in \mathbb{B} \prod \mathbb{P}_\infty^{\rightarrow} \prod \mathbb{P}_\infty^{\rightarrow}. \end{cases}$$

é um isomorfismo em CospLin .

6.4 Espaço Coerente $\mathbb{D}_\infty^{\leftrightarrow}$

De acordo com as Definições 6.1 e 6.3, considera-se aqui o espaço coerente

$$\mathbb{D}_\infty^{\leftrightarrow} = \mathbb{P}_\infty^{\leftrightarrow} \amalg (\mathbb{P}_\infty^{\leftrightarrow} \prod \mathbb{P}_\infty^{\leftrightarrow}) \amalg (\mathbb{P}_\infty^{\leftrightarrow} \prod_{\mathbb{B}} \mathbb{P}_\infty^{\leftrightarrow}) \quad , \quad \text{onde } \mathbb{P}_\infty^{\leftrightarrow} = \mathbb{D}_\infty^{\leftrightarrow} \amalg \overline{\mathbb{D}_\infty^{\leftrightarrow}} \amalg \overline{\mathbb{D}_\infty^{\leftrightarrow}}^\perp.$$

Os objetos em $\mathbb{D}_\infty^{\leftrightarrow}$ interpretam além de todos os processos finitos em \mathcal{P} , também aqueles processos resultantes da composição infinita dos operadores produto seqüencial e soma determinística, para os quais não se conhece o instante inicial e nem o instante final de sua execução. Entretanto, considerando a construção indutiva de $\mathbb{D}_\infty^{\leftrightarrow}$, é possível determinar os processos parciais que serão executados nos instantes t_{-1} e t_0 . Assim também, todos os outros processos executados em instantes ou intervalos de tempo que precedem t_{-1} ou sucedem t_0 também podem ser determinados.

6.4.1 Imersões $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$

As imersões $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ interpretam a construção por infixo de processos, e serão definidas pela união das imagens referentes às funções $\pi_{n,\infty}^{\leftarrow}$ e $\pi_{n,\infty}^{\rightarrow}$, conforme pode ser verificado logo a seguir.

Para tal, considera-se a notação

$$\gamma_n^{(0)} \equiv \gamma_n^{(n)}[\mathbb{P}_n], \quad \gamma_n^{(1)} \equiv \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n], \quad \gamma_n^{(2)} \equiv \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n].$$

Definição 6.11 *Seja o indexador $\theta \in \{00, 001, 002, 011, 012\}$. Considere os espaços coerentes \mathbb{D}_n e $\mathbb{D}_\infty^{\leftrightarrow}$ apresentadas na Definição 5.4 e 5.1 como também as funções $\pi_{n,\infty}^{(\theta)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_\infty$ na Definição 6.5 e as imersões e projeções em 5.2, 5.1.*

Define-se a função imersão $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ pela expressão

$$\pi_{n,\infty}^{\leftarrow}(x) = \begin{cases} \pi_{n,\infty}^{(00)}(x), & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)})(x) \times \{(: 011; 001)\}, & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)})(x) \times \{(: 012; 002)\}, & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma $\pi_{n,\infty}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ é definida pela expressão

$$\pi_{n,\infty}^{\rightarrow}(x) = \begin{cases} \pi_{n,\infty}^{(00)}(x) & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)})(x) \times \{(: 001; 011)\}, & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)})(x) \times \{(: 002; 012)\}, & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset & \text{caso contrário.} \end{cases}$$

Se $n = 0$ então $\pi_{0,\infty}^{\rightarrow} = \pi_{0,\infty}^{\leftarrow} = \pi_{0,\infty}^{(00)} = \{\mathbf{a}:00 \mid \mathbf{a} \in x\}$.

Considerando-se a Definição 6.11, seguem-se as igualdades.

1. se $x \in \gamma_n^{(0)}[\mathbb{P}_n]$, $\pi_{n,\infty}^{\leftarrow}(x) = \pi_{n,\infty}^{\rightarrow}(x) = \{\mathbf{a}:00 \mid \mathbf{a} \in x, \mathbf{a} \in \mathbf{D}_{n+1}\}$;
2. se $x \in \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n] \equiv \gamma_n^{(1)}$, então

$$\pi_{n+1,\infty}^{\leftarrow}(x) = (\pi_{n+1,\infty}^{(011)} \circ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}))(x) \times \{(001)\}$$
 e

$$\pi_{n+1,\infty}^{\rightarrow}(x) = (\pi_{n+1,\infty}^{(001)} \circ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}))(x) \times \{(011)\};$$
3. se $x \in \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n] \equiv \gamma_n^{(2)}$, tem-se

$$\pi_{n+1,\infty}^{\leftarrow}(x) = (\pi_{n+1,\infty}^{(012)} \circ (\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)}))(x) \times \{(002)\}$$
 e

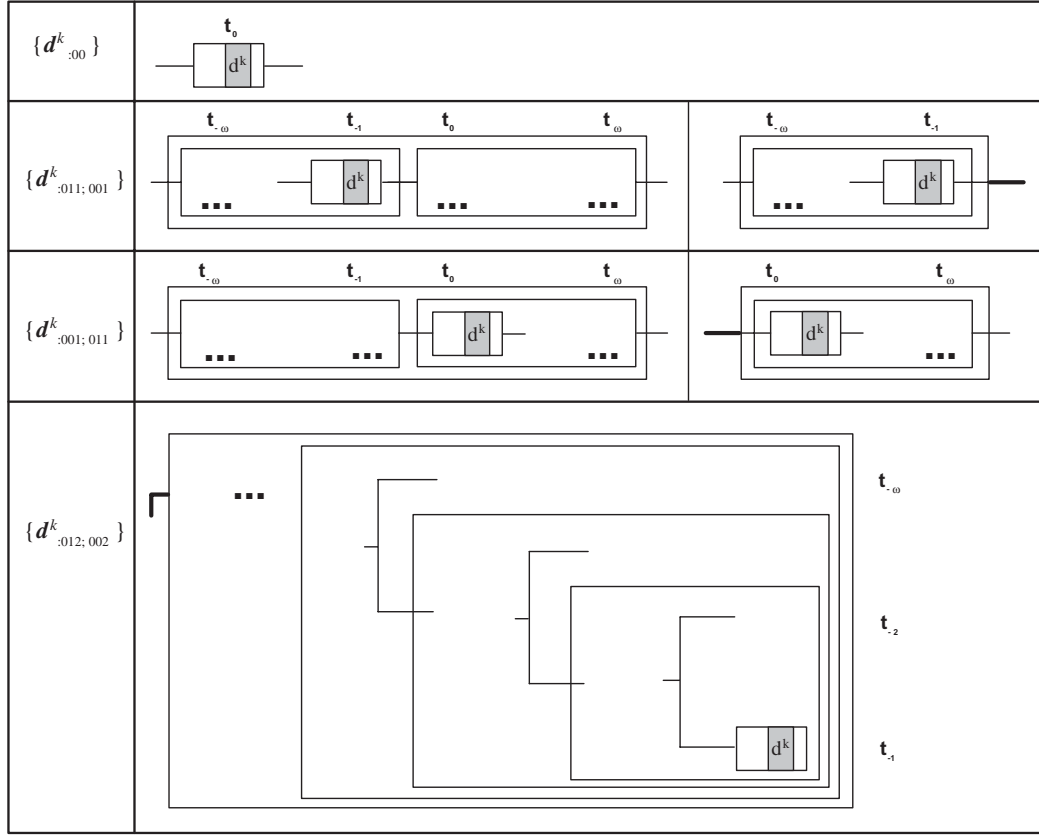
$$\pi_{n+1,\infty}^{\rightarrow}(x) = (\pi_{n+1,\infty}^{(002)} \circ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)}))(x) \times \{(002)\}.$$

Definição 6.12 *Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_\infty^{\leftrightarrow}$ e as imersões $\pi_{n+1,\infty}^{\leftarrow}$, $\pi_{n,\infty}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ apresentadas na Definição 6.11. A função estrita $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ é definida pela expressão*

$$\pi^{\leftrightarrow}(x) = \pi_{n,\infty}^{\leftarrow}(x) \cup \pi_{n,\infty}^{\rightarrow}(x).$$

Proposição 6.10 *Considere os espaços coerentes \mathbb{D}_n , \mathbb{D}_{n+1} e \mathbb{D}_∞ . As funções $\pi_{n,\infty}^{\leftarrow}$, $\pi_{n,\infty}^{\rightarrow}$, $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ apresentadas nas Definições 6.11 e 6.12 são morfismos na categoria CospLin .*

Aplicações das imersões $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$, apresentadas na Definição 6.12, sobre conjuntos coerentes unitários, podem ser visualizadas na Figura 6.8.

FIGURA 6.8 - Representações de Interpretações em $\pi_{n,\infty}^{\leftrightarrow}[\mathbb{D}_n]$

6.4.2 Projeções $\Pi_{\infty,n}^{\leftrightarrow}$

Os objetos na imagem das projeções $\Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n$ são definidos como a união dos subconjuntos coerentes em $\Pi_{\infty,n}^{\leftarrow}[\mathbb{D}_n]$ e $\Pi_{\infty,n}^{\rightarrow}[\mathbb{D}_n]$. Portanto, primeiramente apresentase a definição das funções $\Pi_{\infty,n}^{\leftarrow}$ e $\Pi_{\infty,n}^{\rightarrow}$ e as correspondentes proposições que asseguram sua linearidade.

Definição 6.13 *Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_n \amalg \mathbb{D}_n$ e $\mathbb{B} \amalg \mathbb{D}_n \amalg \mathbb{D}_n$ e as imersões $\gamma_n^{(\beta)}$ e $\pi_{n,\infty}$ respectivamente apresentados na Proposição 5.1 e na Definição 6.12, sendo $\beta \in \{0, 1, 2\}$. Considere também $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\} \in \mathbb{D}_{\infty}$ onde cada $x_i \in \mathbb{D}_{\infty}^c$ é uma aproximação finita de x .*

A função $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n$ é definida por $\Pi_{\infty,n}^{\leftarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\leftarrow}(x_i)$, sendo

$$\Pi_{\infty,n+1}^{\leftarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\leftarrow} \circ \gamma_n^{(0)}), \\ \{\mathbf{a} \mid \mathbf{a}_{:011;001} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(001)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\leftarrow} \circ \gamma_n^{(1)}), \\ \{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(002)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\leftarrow} \circ \gamma_n^{(2)}), \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n$, $\Pi_{\infty,n}^{\rightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\rightarrow}(x_i)$, e

$$\Pi_{\infty,n+1}^{\rightarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\rightarrow} \circ \gamma_n^{(0)}), \\ \{\mathbf{a} \mid \mathbf{a}_{:001;011} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(011)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\rightarrow} \circ \gamma_n^{(1)}), \\ \{\mathbf{a} \mid \mathbf{a}_{:002;012} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(012)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\rightarrow} \circ \gamma_n^{(2)}), \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Em ambas definições, se $n = 0$ tem-se

$$\Pi_{\infty,0}^{\leftarrow}(x_i) = \Pi_{\infty,0}^{\rightarrow}(x_i) = \begin{cases} \{d^{(k)}\} & \text{se } x = \{d_{:00}^{(k)}\}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Proposição 6.11 *As projeções estritas $\Pi_{\infty,n}^{\leftarrow}, \Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n$ apresentadas na Definição 6.13 são morfismos na categoria $\mathit{CospLin}$.*

Demonstração. Apresenta-se a demonstração de que $\Pi_{\infty,n}^{\leftarrow}$ são morfismos $\mathit{CospLin}$, ou seja, tais projeções $\Pi_{\infty,n}^{\leftarrow}$ são funções monótonas, contínuas, estáveis e lineares. O mesmo pode ser provado para as projeções $\Pi_{\infty,n}^{\rightarrow}$. O caso em que $n = 0$ é trivial. Considera-se então $n \in \omega$ mas $n \neq 0$.

1. Demonstra-se primeiramente que $\Pi_{\infty,n+1}^{\leftarrow}$ é monótona. Para tal, suponha $\{x_i\}_{i \in \omega}, \{x'_i\}_{i \in \omega}$ famílias dirigidas de subconjuntos coerentes e finitos tais que

$$x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\} \subseteq \bigcup_{i \in \omega}^{\uparrow} \{x'_i\} = x'.$$

Considerando-se o fecho inferior de $x' \in \mathbb{D}_{\infty}$, para todo x'_i existe x_i e $x_i \subseteq x'_i$, logo uma das seguintes condições é satisfeita

- (a) se $x_i, x'_i \in (\pi_{n+1,\infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, $\{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\} \subseteq \{\mathbf{a} \mid \mathbf{a}_{:00} \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}$;
- (b) se $x_i, x'_i \in (\pi_{n+1,\infty} \circ \gamma_n^{(1)})[\mathbb{P}_n \amalg \mathbb{P}_n]$, $\{\mathbf{a} \mid \mathbf{a}_{:001;011} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(001)\} \subseteq \{\mathbf{a} \mid \mathbf{a}_{:001;011} \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(001)\}$;
- (c) se $x_i, x'_i \in (\pi_{n+1,\infty} \circ \gamma_n^{(2)})[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n]$, $\{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(002)\} \subseteq \{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x'_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(002)\}$;
- (d) caso contrário, é imediato que $\emptyset \subseteq \emptyset$.

Logo, $\Pi_{\infty,n+1}^{\leftarrow}(x) \subseteq \Pi_{\infty,n+1}^{\leftarrow}(x')$.

2. A seguir, verifica-se a continuidade de $\Pi_{\infty,n+1}^{\leftarrow}$. Seja $\mathcal{X} \neq \emptyset$ uma família dirigida, $\mathcal{X} \subseteq \mathbb{D}_{\infty}$. Para quaisquer $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\}$, $x' = \bigcup_{i \in \omega}^{\uparrow} \{x'_i\}$ em \mathcal{X} , tem-se que $x \cup x' \in \mathcal{X}$. Pela monotonicidade comprovada no ítem anterior, $\Pi_{\infty,n+1}^{\leftarrow}(x) \subseteq \Pi_{\infty,n+1}^{\leftarrow}(x \cup x')$ e $\Pi_{\infty,n+1}^{\leftarrow}(x') \subseteq \Pi_{\infty,n+1}^{\leftarrow}(x \cup x')$. Portanto a família $\{\Pi_{\infty,n+1}^{\leftarrow}(\mathcal{X})\} \subseteq \mathbb{D}_{n+1}$ é dirigida. Considerando-se que \mathbb{D}_{n+1} é fechado para uniões dirigidas tem-se que $\bigcup^{\uparrow} \{\Pi_{\infty,n+1}^{\leftarrow}(\mathcal{X})\}_{i \in \omega} \in \mathbb{D}_{n+1}$.

Nos próximos casos, aplica-se a função $\Pi_{\infty,n+1}^{\leftarrow}$, tomando-se $\bigcup \mathcal{X}$ como argumento.

- (a) Se $\forall x \in \bigcup \mathcal{X}$, $x \in (\pi_{n+1,\infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, $\bigcup \mathcal{X} \in (\pi_{n+1,\infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, então $\Pi_{\infty,n+1}^{\leftarrow}(\bigcup \mathcal{X}) = \bigcup \{\{\mathbf{a} \mid \mathbf{a}_{:00} \in x \subseteq \mathcal{X} \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}\}$;
- (b) Se $\forall x \in \mathcal{X}$, $x \in (\pi_{n+1,\infty} \circ \gamma_n^{(1)})[\mathbb{P}_n \amalg \mathbb{P}_n]$, então $\Pi_{\infty,n+1}^{\leftarrow}(\bigcup \mathcal{X}) = \bigcup \{\{\mathbf{a} \mid \mathbf{a}_{:011;001} \in x \subseteq \mathcal{X} \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(001)\}\}$;
- (c) Se $\forall x \in \mathcal{X}$, $x \in (\delta_{n+1,\infty} \circ \gamma_n^{(2)})[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n]$, então $\Pi_{\infty,n+1}^{\leftarrow}(\bigcup \mathcal{X}) = \bigcup \{\{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x \subseteq \mathcal{X} \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(002)\}\}$;
- (d) Caso contrário, é imediato que $\emptyset = \Pi_{\infty,n+1}^{\leftarrow}(\emptyset)$.

Em todos os casos, $\Pi_{\infty,n+1}^{\leftarrow}(\bigcup \mathcal{X}) = \bigcup \{\Pi_{\infty,n+1}^{\leftarrow}(x) \mid x \in \mathcal{X}\}$. Portanto $\Pi_{\infty,n+1}^{\leftarrow}$ é monótona e satisfaz a continuidade, o que completa a demonstração de que esta função é contínua.

3. Mostra-se agora que $\Pi_{\infty, n+1}^{\leftarrow}$ satisfaz a propriedade da estabilidade. Suponha que $x, x' \in \mathbb{D}_{\infty}$ tais que $x \cup x' \in \mathbb{D}_{\infty}$. Neste caso, seja $x'' = x \cap x' = \bigcup^{\uparrow} \{x''_i\}_{i \in \omega}$ onde $x''_i = x_i \cap x'_i, \forall i \in \omega$.

- (a) Se $x, x' \in (\Pi_{n+1, \infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, $x'' = x \cap x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, logo $\Pi_{\infty, n+1}^{\leftarrow}(x \cap x') = \{\mathbf{a} \mid \mathbf{a}_{:00} \in (x \cap x')\} = \{\mathbf{a} \mid \mathbf{a}_{:00} \in x\} \cap \{\mathbf{a} \mid \mathbf{a}_{:00} \in x'\}$; sempre que $\mathbf{a} \in \mathbf{D}_{n+1}$.
- (b) Se $x, x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(1)})[\mathbb{P}_n]$, $x'' = x \cap x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(1)})[\mathbb{P}_n]$, então $\Pi_{\infty, n+1}^{\leftarrow}(x \cap x') = \{\mathbf{a} \mid \mathbf{a}_{:011;001} \in x\} \times \{(001)\} \cap \{\mathbf{a} \mid \mathbf{a}_{:011;001} \in x'\} \times \{(001)\}$;
- (c) Se $x, x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(2)})[\mathbb{P}_n]$, $x'' = x \cap x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(2)})[\mathbb{P}_n]$, então $\Pi_{\infty, n+1}^{\leftarrow}(x \cap x') = \{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x\} \times \{(002)\} \cap \{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x'\} \times \{(002)\}$;
- (d) Caso contrário, $\Pi_{\infty, n+1}^{\leftarrow}(x \cap x') = \emptyset = \Pi_{\infty, n+1}^{\leftarrow}(x) = \Pi_{\infty, n+1}^{\leftarrow}(x')$.

Mostrou-se que em todos os casos, $\Pi_{\infty, n}^{\leftarrow}(x \cap x') = \Pi_{\infty, n+1}^{\leftarrow}(x) \cap \Pi_{\infty, n+1}^{\leftarrow}(x')$. Assim, $\Pi_{\infty, n+1}^{\leftarrow}$ é contínua e verifica a propriedade da estabilidade, logo é uma função estável.

4. Para concluir, deve-se mostrar que $\Pi_{\infty, n+1}^{\leftarrow}$ satisfaz a propriedade da linearidade. Seja $\mathcal{X} \subseteq \mathbb{D}_{\infty}$ tal que, para quaisquer $x, x' \in \mathcal{X}$, $x \cup x' \in \mathbb{D}_{\infty}$. Tomando-se $x \cup x'$ como argumento para aplicação de $\Pi_{\infty, n+1}^{\leftarrow}$, tem-se os seguintes casos

- (a) Se $x, x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, $x'' = x \cup x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(0)})[\mathbb{P}_n]$, logo $\Pi_{\infty, n+1}^{\leftarrow}(x \cup x') = \{\mathbf{a} \mid \mathbf{a}_{:00} \in (x \cup x')\} = \{\mathbf{a} \mid \mathbf{a}_{:00} \in x\} \cup \{\mathbf{a} \mid \mathbf{a}_{:00} \in x'\}$, sempre que $\mathbf{a} \in \mathbf{D}_n$;
- (b) Se $x, x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(1)})[\mathbb{P}_n]$, $x'' = x \cup x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(1)})[\mathbb{P}_n]$, então $\Pi_{\infty, n+1}^{\leftarrow}(x \cup x') = \{\mathbf{a} \mid \mathbf{a}_{:011;001} \in x\} \times \{(001)\} \cup \{\mathbf{a} \mid \mathbf{a}_{:011;001} \in x'\} \times \{(001)\}$;
- (c) Se $x, x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(2)})[\mathbb{P}_n]$, $x'' = x \cup x' \in (\delta_{n+1, \infty} \circ \gamma_n^{(2)})[\mathbb{P}_n]$, então $\Pi_{\infty, n+1}^{\leftarrow}(x \cup x') = \{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x\} \times \{(002)\} \cup \{\mathbf{a} \mid \mathbf{a}_{:012;002} \in x'\} \times \{(002)\}$;
- (d) Caso contrário, $\Pi_{\infty, n+1}^{\leftarrow}(x \cup x') = \emptyset = \Pi_{\infty, n+1}^{\leftarrow}(x) = \Pi_{\infty, n+1}^{\leftarrow}(x')$.

Acrescentando-se a esta última igualdade, a propriedade demonstrada no item 2 e o fato de $\Pi_{\infty, n+1}^{\leftarrow}(\emptyset) = \emptyset$, tem-se que $\Pi_{\infty, n+1}^{\leftarrow}(\bigcup \mathcal{X}) = \bigcup \{\Pi_{\infty, n+1}^{\leftarrow}(x) \mid x \in \mathcal{X}\}$. Portanto $\Pi_{\infty, n+1}^{\leftarrow}$ é linear.▲

Definição 6.14 Considere os espaços coerentes \mathbb{D}_{∞} e \mathbb{D}_n e as projeções $\Pi_{n, \infty}^{\leftarrow}, \Pi_{n, \infty}^{\rightarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n$ apresentadas na Definição 6.13. Considere também $x \in \mathbb{D}_{\infty}$ como o supremo de um conjunto dirigido de aproximações finitas, $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\}$.

A função estrita $\Pi_{\infty, n}^{\leftrightarrow} : \mathbb{D}_{\infty} \rightarrow \mathbb{D}_n$ associa todo conjunto coerente x ao conjunto coerente $\Pi_{\infty, n}^{\leftrightarrow}(x) \in \mathbb{D}_n$ definido pela união

$$\Pi_{\infty, n}^{\leftrightarrow}(x) = (\Pi_{\infty, n}^{\leftarrow} \cup \Pi_{\infty, n}^{\rightarrow})(x),$$

Neste caso, pela Definição 6.13 tem-se

$$\Pi_{\infty, n}^{\leftrightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} ((\Pi_{\infty, n}^{\leftarrow} \cup \Pi_{\infty, n}^{\rightarrow})(x_i)) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty, n}^{\leftrightarrow}(x_i).$$

Em particular, $\Pi_{\infty, n}^{\leftrightarrow}(\emptyset) = \emptyset$.

Proposição 6.12 Considere os espaços coerentes \mathbb{D}_{∞} e \mathbb{D}_n , a seqüência $\{z_n\}_{n \in \omega} \subseteq (\mathbb{D}_n)$, com $z_n = \Pi_n(z_{n+1})$, apresentada na Definição 6.15 e as projeções $\Pi_{\infty, n}^{\leftrightarrow} : \mathbb{D}_{\infty} \rightarrow \mathbb{D}_n$ apresentadas na Definição 6.14. Para cada $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\}$ a seguinte condição é satisfeita

$$\Pi_{\infty,n}^{\leftrightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \{\Pi_{\infty,n}^{\leftrightarrow}(x_i)\} = z_n.$$

Demonstração. Considerando-se $\{x_i\}_{i \in \omega} \subseteq \pi_{i,\infty}^{\leftrightarrow}[\mathbb{D}_i]$ e os resultados da Proposição 6.1, é imediato que $z_n = \Pi_{\infty,n}^{\leftrightarrow}(x_n) \subseteq \bigcup_{i \in \omega}^{\uparrow} \{\Pi_{\infty,n}^{\leftrightarrow}(x_i)\}$. Para mostrar a volta, suponha primeiramente que $i \geq n$, então

$$\Pi_{\infty,n}^{\leftrightarrow}(x_i) = \Pi_{i,n}^{\leftrightarrow} \circ \Pi_{\infty,i}^{\leftrightarrow}(x_i) = \Pi_{i,n}^{\leftrightarrow}(z_i) = z_n. \quad (6.1)$$

Agora, se $i \leq n$ tem-se que

$$\Pi_{\infty,n}^{\leftrightarrow}(x_i) = \delta_{i,n} \circ \Pi_{\infty,i}^{\leftrightarrow}(x_i) = \delta_{i,n}(z_i) \subseteq z_n. \quad (6.2)$$

Logo, por (6.1) e (6.2) conclui-se que $\Pi_{\infty,n}^{\leftrightarrow}(x_i) \subseteq z_n$. Assim também $\bigcup_{i \in \omega}^{\uparrow} \{\Pi_{\infty,n}^{\leftrightarrow}(x_i)\} \subseteq z_n$. Portanto $\Pi_{\infty,n}^{\leftrightarrow}(x_n) = z_n = \bigcup_{i \in \omega}^{\uparrow} \{\Pi_{\infty,n}^{\leftrightarrow}(x_i)\}$, o que completa a prova. \blacktriangle

Como consequência da Proposição 6.12 segue o corolário.

Corolário 6.1 *Considere a seqüência $\{z_n\}_{n \in \omega}$ apresentada na Definição 6.15. Se $x = \bigcup_{i \in \omega}^{\uparrow} x_i \in \mathbb{D}_{\infty}$ então*

$$\Pi_{\infty,n}^{\leftrightarrow}(x) = \Pi_{\infty,n}^{\leftrightarrow}(x_n) = \Pi_{\infty,n}^{\leftrightarrow}(x_m) = z_n, \text{ sempre que } m \geq n.$$

Proposição 6.13 *As funções $\Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_{\infty} \rightarrow \mathbb{D}_n$, apresentadas na Definição 6.14, constituem-se em morfismos da categoria CospLin .*

Demonstração. Conseqüência da Proposição 6.11.

6.4.3 Cone Limite $(\mathbb{D}_{\infty}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_{\infty} \rightarrow \mathbb{D}_n)$.

Definição 6.15 *Considere as projeções $\Pi_n^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ e as imersões $\pi_{n+1,\infty}^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_{\infty}$ definidas em 5.11 e 6.12. O diagrama $\mathcal{K}^{\leftrightarrow}$, representado por*

$$\mathbb{D}_0 \xleftarrow{\Pi_0^{\leftrightarrow}} \mathbb{D}_1 \longleftarrow \dots \longleftarrow \mathbb{D}_n \xleftarrow{\Pi_n^{\leftrightarrow}} \mathbb{D}_{n+1} \longleftarrow \dots$$

é definido pela família de espaços coerentes $\{\mathbb{D}_n\}_{n \in \omega} \subseteq \text{Ob}_{\text{CospLin}}$ e pela família de morfismos $\Pi_{n+1,n}^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n \subseteq \text{Mor}_{\text{CospLin}}$, indutivamente definidos em 5.10.

A partir do diagrama $\mathcal{K}^{\leftrightarrow}$, define-se a seqüência $\{z_n\}_{n \in \omega}$ de subconjuntos coerentes tal que $z_n \in \mathbb{D}_n$ e $\Pi_n^{\leftrightarrow}(z_{n+1}) = z_n$,

$$z_0 \xleftarrow{\Pi_0^{\leftrightarrow}} z_1 \longleftarrow \dots \longleftarrow z_n \xleftarrow{\Pi_n^{\leftrightarrow}} z_{n+1} \longleftarrow \dots$$

Pela Definição 6.15, para cada elemento $z_{n+1} \in \mathbb{D}_{n+1}$ da seqüência $\{z_n\}_{n \in \omega}$, existe sempre $z_n \in \mathbb{D}_n$, tal que $\Pi_n^{\leftrightarrow}(z_{n+1}) = z_n$. Portanto, se $\alpha, \beta \in \{0, 1, 2\}$, uma das seguintes condições será satisfeita

1. $\pi_n^{(\alpha 0)}(z_n) = z_{n+1}$;
2. $((\pi_{n+1}^{(\alpha 01)} \circ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)})) \cup (\pi_{n+1}^{(\beta 11)} \circ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)})))(z_n) \subseteq z_{n+1}$;
3. $((\pi_{n+1}^{(\alpha 02)} \circ (\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)})) \cup (\pi_{n+1}^{(\beta 12)} \circ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)})))(z_n) \subseteq z_{n+1}$.

Simplificando a notação, cada $z_n = \Pi_n^{\leftrightarrow}(z_{n+1})$ determina z'_n, z''_n, w'_n, w''_n em \mathbb{D}_n , definidos pelas expressões

- $(\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)})(z_n) = z'_n$ e $(\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)})(z_n) = z''_n$,
- $(\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)})(z_n) = w'_n$ e $(\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)})(z_n) = w''_n$,

tais que, uma das seguintes condições é sempre satisfeita

1. $\{\mathbf{a};00 \mid \mathbf{a} \in z_n\} = z_{n+1}$;
2. $\pi_n^{(\alpha 01)}(z_n) \cup \pi_n^{(\beta 11)}(z'_n) \subseteq z_{n+1}$;
3. $\pi_n^{(\alpha 02)}(z_n) \cup \pi_n^{(\beta 12)}(z'_n) \subseteq z_{n+1}$.

Como conseqüência, define-se logo a seguir a seqüência $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_\infty$ de aproximações finitas do conjunto coerente x , onde a projeção de cada subconjunto coerente x_n corresponde a um elemento da seqüência $\{z_n\}_{n \in \omega}$. Mais detalhadamente, uma das seguintes condições é sempre satisfeita

1. $x_{n+1} = \{\mathbf{a};00 \mid \mathbf{a} \in z_{n+1}\}$, ou
2. $z'_n \times \{ : 011; 001\} \cup z''_n \times \{ : 001; 011\} \subseteq x_{n+1}$, ou ainda
3. $w'_n \times \{ : 012; 002\} \cup w''_n \times \{ : 012; 012\} \subseteq x_{n+1}$.

Definição 6.16 Considere as imersões $\pi_{n,\infty}^{\leftrightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty$ apresentadas na Definição 6.12 como também a seqüência $\{z_n\}_{n \in \omega}$ de subconjuntos coerentes apresentada na Definição 6.15, tal que $z_n \in \mathbb{D}_n$ e $\Pi_n^{\leftrightarrow}(z_{n+1}) = z_n$. Defina-se então a seqüência $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_\infty$,

$$x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \quad x_n \text{ sempre que } \Pi_{\infty,n}^{\leftrightarrow}(x_n) = (z_n).$$

Neste caso, os subconjuntos coerentes da seqüência $\{x_n\}_{n \in \omega}$ determinam um novo objeto $x \in \mathbb{D}_\infty$, dado pela expressão

$$x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\} \quad \Leftrightarrow \quad x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \subset x$$

Cada x_n é uma aproximação finita de x onde $\Pi_{\infty,n}^{\leftrightarrow}(x_n) = z_n$ e $\pi_{n,\infty}^{\leftrightarrow}(z_n) \subseteq x_n$.

A seguir define-se os limites inversos para o diagrama $\mathcal{K}^{\leftrightarrow}$ que asseguram interpretação para todos os processos em \mathcal{P} , incluindo produtos seqüenciais que não possuem os instantes inicial e final, quando de sua execução. Além destes, também são interpretadas as correspondentes somas determinísticas, infinitas no sentido temporal.

Proposição 6.14 Considere a função $\Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_\infty \rightarrow \mathbb{D}_n$, apresentada na Definição 6.14. O par $\mathbb{D}_\infty^{\leftrightarrow} \equiv (\mathbb{D}_\infty^{\leftrightarrow}, \infty,n : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama $\mathcal{K}^{\leftrightarrow}$

$$\mathbb{D}_0 \xleftarrow{\Pi_1^{\leftrightarrow}} \mathbb{D}_1 \longleftarrow \dots \longleftarrow \mathbb{D}_n \xleftarrow{\Pi_{n+1}^{\leftrightarrow}} \mathbb{D}_{n+1} \longleftarrow \dots$$

apresentado na Definição 6.15.

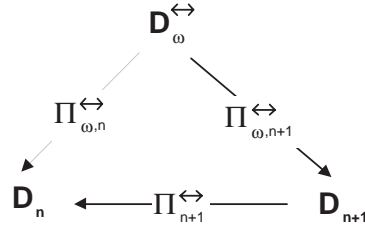
Demonstração. Considere os espaços coerentes \mathbb{D}_n e $\mathbb{D}_\infty^{\leftrightarrow}$ e as funções lineares $\Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n$ e $\Pi_{n+1,n}^{\leftrightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas nas Definições 6.14 e 5.10, respectivamente.

1. Mostra-se que $\Pi_{\infty,n}^{\leftrightarrow} = \Pi_{n+1,n}^{\leftrightarrow} \circ \Pi_{\infty,n+1}^{\leftrightarrow}$. Para tal, considere as seqüências
 - $\{z_n\}_{n \in \omega}$ de subconjuntos coerentes tal que $z_n \in \mathbb{D}_n$ e $\Pi_n^{\leftrightarrow}(z_{n+1}) = z_n$, e
 - $\{x_n\}_{n \in \omega} \in \mathbb{D}_\infty^{\leftrightarrow}$, $x_n = \Pi_{n,\infty}^{\leftrightarrow}(z_n)$,

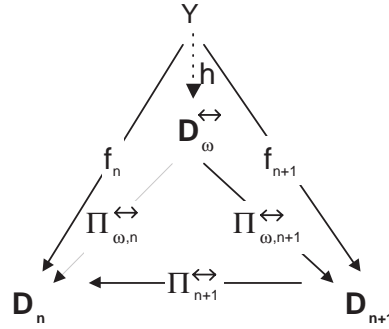
apresentadas nas Definições 6.15 e 6.16, respectivamente. Pela proposição 6.12,

$$(\Pi_{n+1}^{\leftrightarrow} \circ \Pi_{\infty,n+1}^{\leftrightarrow})(x) = (\Pi_{n+1}^{\leftrightarrow}(\Pi_{\infty,n+1}^{\leftrightarrow}(x))) = \Pi_{n+1}^{\leftrightarrow}(z_{n+1}) = z_n = \Pi_{\infty,n}^{\leftrightarrow}(x).$$

Logo $\Pi_{n+1}^{\leftrightarrow} \circ \Pi_{\infty,n+1}^{\leftrightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \{\Pi_{\infty,n}^{\leftrightarrow}(x_i)\} = \Pi_{\infty,n}^{\leftrightarrow}(x)$, ou seja, a comutatividade do próximo diagrama é satisfeita.



2. Para completar a prova, deve-se mostrar que o cone $(\mathbb{D}_\infty^{\leftrightarrow}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n)$ é um limite para os demais cones em relação ao diagrama $\mathcal{K}^{\leftrightarrow}$. Para tal, considere um outro cone $(\mathbb{Y}, f_n : \mathbb{Y} \rightarrow \mathbb{D}_n)$ para o diagrama \mathcal{K} . Seja $h : \mathbb{Y} \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ definida pela expressão $h(y) = \pi_{n,\infty}^{\leftrightarrow} \circ f_n(y)$. Considerando-se que h é definida como uma composta entre funções lineares, é fácil verificar que $h \in \text{Mor}_{\text{CospLin}}$ e que $\Pi_{\infty,n}^{\leftrightarrow} \circ h = f_n$, ou seja, h verifica a comutatividade do diagrama



Mostra-se a seguir que h é única. Suponha $h' : \mathbb{Y} \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ que também verifica a comutatividade do último diagrama, ou seja, $\Pi_{\infty,n}^{\leftrightarrow} \circ h' = f_n$, mas tal que existe $y \in \mathbb{Y}$ e $h(y) = x \neq h'(y) = x'$. Analisam-se as situações abaixo.

- (a) Suponha $(\Pi_{\infty,n}^{\leftrightarrow} \circ h)(y) = (\Pi_{\infty,n}^{\leftrightarrow} \circ h')(y)$. Neste caso, considerando-se que $\Pi_{\infty,n}^{\leftrightarrow}$ é um epimorfismo, tem-se $\bigcup_{n \in \omega} \pi_{n,\infty}^{\leftrightarrow}(z_n) = x \subset x'$. Entretanto, por hipótese, o par (\mathbb{Y}, f_n) é um cone para o diagrama \mathcal{K} , logo existe $z_i \in \{z_n\}$, tal que $\pi_{n,\infty}^{\leftrightarrow}(z_i) = x'$. Portanto, não existe $\bigcup_{n \in \omega} \{z_n\}$.
- (b) Na outra situação, $(\Pi_{\infty,n}^{\leftrightarrow} \circ h')(y) \neq (\Pi_{\infty,n}^{\leftrightarrow} \circ h)(y)$ o que leva a concluir que h_n não está bem definida, pois $\Pi_{\infty,n}^{\leftrightarrow} \circ h' = f_n = \Pi_{\infty,n}^{\leftrightarrow} \circ h$.

Portanto, como as situações acima levam a contradições, $h(y) = h'(y), \forall y \in \mathbb{Y}$.

Mostrou-se que o par $(\mathbb{D}_\infty^{\leftrightarrow}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama $\mathcal{K}^{\leftrightarrow}$. \blacktriangle

As próximas proposições formalizam as relações entre os cones já apresentados.

Definição 6.17 Considere os cones $(\mathbb{D}_\infty^{\rightarrow}, \Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \mathbb{D}_n)$ e $(\mathbb{D}_\infty^{\leftrightarrow}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n)$ apresentados nas Seções 6.3.3 e 6.4.3. Se $x = \bigcup^{\uparrow} \{x_i\}$, sempre que $x_i \in \pi_{n,\infty}^{\leftrightarrow}[\mathbb{D}_n]$, a função $\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow}, \mathbb{D}_\infty^{\leftrightarrow}} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ é definida pela expressão

$$\mathbf{H}_{\mathbb{D}_{\infty}, \mathbb{D}_{\infty}}(x) = \bigcup^{\uparrow} \mathbf{H}_{\mathbb{D}_{\infty}, \mathbb{D}_{\infty}}(x_i), \quad \text{onde}$$

$$\mathbf{H}_{\mathbb{D}_{\infty}, \mathbb{D}_{\infty}}(x_i) = \begin{cases} x_i, & \text{se } x_i \in (\pi_{n,\infty}^{\leftrightarrow} \circ \gamma_{n-1}^{(0)})[\mathbb{P}_{n-1}], \\ \{\mathbf{a}:001 \mid \mathbf{a}:001;011 \in x_i\}, & \text{se } x_i \in (\pi_{n,\infty}^{\leftrightarrow} \circ \gamma_{n-1}^{(1)})[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}], \\ \{\mathbf{a}:002 \mid \mathbf{a}:002;012 \in x_i\}, & \text{se } x_i \in (\pi_{n,\infty}^{\leftrightarrow} \circ \gamma_{n-1}^{(2)})[\mathbb{B} \prod \mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}]. \end{cases}$$

Proposição 6.15 A função $\mathbf{H}_{\mathbb{D}_{\infty}, \mathbb{D}_{\infty}} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_{\infty}^{\leftrightarrow}$ é um epimorfismo em CospLin .

Proposição 6.16 Considere os cones $\mathbb{D}^{\leftarrow} \equiv (\mathbb{D}_{\infty}^{\leftarrow}, \Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_n)$ e $\mathbb{D}^{\leftrightarrow} \equiv (\mathbb{D}_{\infty}^{\leftrightarrow}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n)$, apresentados nas Seções 6.3.3 e 6.4.3. A função linear $\mathbf{H}_{\mathbb{D}_{\infty}, \mathbb{D}_{\infty}}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_{\infty}^{\leftarrow}$ definida pela expressão

$$\mathbf{H}_{\mathbb{D}_{\infty}, \mathbb{D}_{\infty}}^{\leftarrow}(x) = \begin{cases} x, & \text{se } x \in \mathbb{P}_{\infty}^{\leftarrow}, \\ \{\mathbf{a}:011 \mid \mathbf{a}:001;011 \in x\}, & \text{se } x \in \mathbb{P}_{\infty}^{\leftarrow} \prod \mathbb{P}_{\infty}^{\leftarrow}, \\ \{\mathbf{a}:012 \mid \mathbf{a}:002;012 \in x\}, & \text{se } x \in \mathbb{B} \prod \mathbb{P}_{\infty}^{\leftarrow} \prod \mathbb{P}_{\infty}^{\leftarrow}. \end{cases}$$

é um epimorfismo em CospLin .

6.5 Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$.

De acordo com as Definições 6.1 e 6.3, considera-se aqui o espaço coerente

$$\mathbb{D}_{\infty}^{\rightarrow\leftarrow} = \mathbb{P}_{\infty}^{\rightarrow\leftarrow} \amalg (\mathbb{P}_{\infty}^{\rightarrow\leftarrow} \prod \mathbb{P}_{\infty}^{\rightarrow\leftarrow}) \amalg (\mathbb{B} \prod \mathbb{P}_{\infty}^{\rightarrow\leftarrow} \prod \mathbb{P}_{\infty}^{\rightarrow\leftarrow}),$$

onde o subespaço $\mathbb{P}_{\infty}^{\rightarrow\leftarrow}$ é a soma direta $\mathbb{P}_{\infty}^{\rightarrow\leftarrow} = \mathbb{D}_{\infty}^{\rightarrow\leftarrow} \amalg \overline{\mathbb{D}_{\infty}^{\rightarrow\leftarrow}} \amalg \overline{\mathbb{D}_{\infty}^{\rightarrow\leftarrow}}^{\perp}$.

Os objetos em $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ interpretam além de todos os processos finitos em \mathcal{P} , também aqueles processos resultantes da composição infinita dos operadores produto seqüencial e soma determinística, para os quais se conhece o instante inicial e o instante final de sua execução. Entretanto, considerando-se a construção indutiva de $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$, não é possível definir um processo cuja execução, a partir do instante inicial, alcance o instante final. No caso do operador seqüencial, é possível interpretar a aplicação simultânea da concatenação por prefixo e sufixo em $\mathbb{P}_{\infty}^{\rightarrow\leftarrow} \prod \mathbb{P}_{\infty}^{\rightarrow\leftarrow}$. Todavia, a construção por infixo não possui interpretação em $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$.

6.5.1 Imersões $\pi_{n,\infty}^{\rightarrow\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow\leftarrow}$

As imersões $\pi_{n,\infty}^{\rightarrow\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ interpretam a construção simultânea por prefixo e sufixo de processos. Na busca desta interpretação, o subespaço imagem $\pi_{n,\infty}^{\rightarrow\leftarrow}[\mathbb{D}_n]$ é determinado pela união $\pi_{n,\infty}^{\leftarrow}[\mathbb{D}_n]$ e $\pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$. Assim, primeiramente são definidas as funções $\pi_{n,\infty}^{\leftarrow}, \pi_{n,\infty}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow\leftarrow}$.

Considera-se também a notação

$$\gamma_n^{(0)} \equiv \gamma_n^{(n)}[\mathbb{P}_n], \quad \gamma_n^{(1)} \equiv \gamma_n^{(1)}[\mathbb{P}_n \prod \mathbb{P}_n], \quad \gamma_n^{(2)} \equiv \gamma_n^{(2)}[\mathbb{B} \prod \mathbb{P}_n \prod \mathbb{P}_n].$$

Definição 6.18 Considere os espaços coerentes \mathbb{D}_n e $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ como também as funções $\pi_{n,\infty}^{(00)}$ apresentadas na Definição 6.5.

Define-se a função imersão $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ pela expressão

$$\pi_{n+1,\infty}^{\leftarrow}(x) = \begin{cases} \pi_{n,\infty}^{(00)}(x), & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)})(x) \times \{(: 001; 001)\}, & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)})(x) \times \{(: 002; 002)\}, & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset, & \text{csao contrário.} \end{cases}$$

Da mesma forma $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftarrow}$ é definida pela expressão

$$\pi_{n+1,\infty}^{\leftarrow}(x) = \begin{cases} \pi_{n,\infty}^{(00)}(x) & \text{se } x \in \gamma_n^{(0)}, \\ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)})(x) \times \{(: 011; 011)\}, & \text{se } x \in \gamma_n^{(1)}, \\ (\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)})(x) \times \{(: 012; 011)\}, & \text{se } x \in \gamma_n^{(2)}, \\ \emptyset & \text{caso contrário.} \end{cases}$$

Se $n = 0$ então $\pi_{0,\infty}^{\leftarrow} = \pi_{0,\infty}^{\leftarrow} = \pi_{0,\infty}^{(00)} = \{\mathbf{a}:00 \mid \mathbf{a} \in x\}$.

Considerando-se as imersões $\pi_{n,\infty}^{(\theta)}$, $\pi_{n,\infty}^{\leftarrow}$ e $\pi_{n,\infty}^{\leftarrow}$ apresentadas nas Definições 6.5 e 6.18, seguem-se as igualdades:

1. se $x \in \gamma_n^{(0)}[\mathbb{P}_n]$, $\pi_{n,\infty}^{\leftarrow}(x) = \pi_{n,\infty}^{\leftarrow}(x) = \{\mathbf{a}:00 \mid \mathbf{a} \in x, \mathbf{a} \in \mathbf{D}_{n+1}\}$;

2. se $x \in \gamma_n^{(1)}[\mathbb{P}_n \amalg \mathbb{P}_n] \equiv \gamma_n^{(1)}$, então

$$\begin{aligned} \pi_{n+1,\infty}^{\leftarrow}(x) &= (\pi_{n+1,\infty}^{(011)} \circ (\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}))(x) \text{ e} \\ \pi_{n+1,\infty}^{\leftarrow}(x) &= (\pi_{n+1,\infty}^{(001)} \circ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}))(x); \end{aligned}$$

3. se $x \in \gamma_n^{(2)}[\mathbb{B} \amalg \mathbb{P}_n \amalg \mathbb{P}_n] \equiv \gamma_n^{(2)}$, tem-se

$$\begin{aligned} \pi_{n+1,\infty}^{\leftarrow}(x) &= (\pi_{n+1,\infty}^{(012)} \circ (\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)}))(x) \text{ e} \\ \pi_{n+1,\infty}^{\leftarrow}(x) &= (\pi_{n+1,\infty}^{(002)} \circ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)}))(x). \end{aligned}$$

Definição 6.19 Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_\infty^{\leftarrow}$ e as imersões $\pi_{n+1,\infty}^{\leftarrow}$, $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftarrow}$ apresentadas na Definição 6.18. A função estrita $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftarrow}$ é definida pela expressão

$$\pi_{n,\infty}^{\leftarrow}(x) = \pi_{n,\infty}^{\leftarrow}(x) \cup \pi_{n,\infty}^{\leftarrow}(x).$$

Proposição 6.17 Considere os espaços coerentes \mathbb{D}_n , e $\mathbb{D}_\infty^{\leftarrow}$. As funções $\pi_{n,\infty}^{\leftarrow}$, $\pi_{n,\infty}^{\leftarrow}$, $\pi_{n,\infty}^{\leftarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_\infty^{\leftarrow}$ apresentadas nas Definições 6.18 e 6.19 são morfismos na categoria CospLin .

6.5.2 Projeções $\Pi_{\infty,n}^{\leftarrow}$

Os objetos na imagem das projeções $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty^{\leftarrow} \rightarrow \mathbb{D}_n$ são definidos como a união dos subconjuntos coerentes em $\Pi_{\infty,n}^{\leftarrow}[\mathbb{D}_n]$ e $\Pi_{\infty,n}^{\leftarrow}[\mathbb{D}_n]$. Portanto, as funções $\Pi_{\infty,n}^{\leftarrow}$ e $\Pi_{\infty,n}^{\leftarrow}$ são apresentadas logo abaixo, seguidas das correspondentes proposições que asseguram sua linearidade.

Definição 6.20 Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_n \amalg \mathbb{D}_n$ e $\mathbb{B} \amalg \mathbb{D}_n \amalg \mathbb{D}_n$ e as imersões $\gamma_n^{(\beta)}$ e $\pi_{n,\infty}$ respectivamente apresentados na Proposição 5.1 e na Definição 6.12, sendo $\beta \in \{0, 1, 2\}$. Considere também $x = \bigcup_{i \in \omega}^\uparrow \{x_i\} \in \mathbb{D}_\infty$ onde cada $x_i \in \mathbb{D}_\infty^c$ é uma aproximação finita de x .

A função $\Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_\infty^{\leftarrow} \rightarrow \mathbb{D}_n$ é definida por $\Pi_{\infty,n}^{\leftarrow}(x) = \bigcup_{i \in \omega}^\uparrow \Pi_{\infty,n}^{\leftarrow}(x_i)$, sendo

$$\Pi_{\infty,n+1}^{\leftarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}:00 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\leftarrow} \circ \gamma_n^{(0)}), \\ \{\mathbf{a} \mid \mathbf{a}:001;001 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(001)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\leftarrow} \circ \gamma_n^{(1)}), \\ \{\mathbf{a} \mid \mathbf{a}:002;002 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(002)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\leftarrow} \circ \gamma_n^{(2)}), \\ \emptyset, & \text{caso contrário.} \end{cases}$$

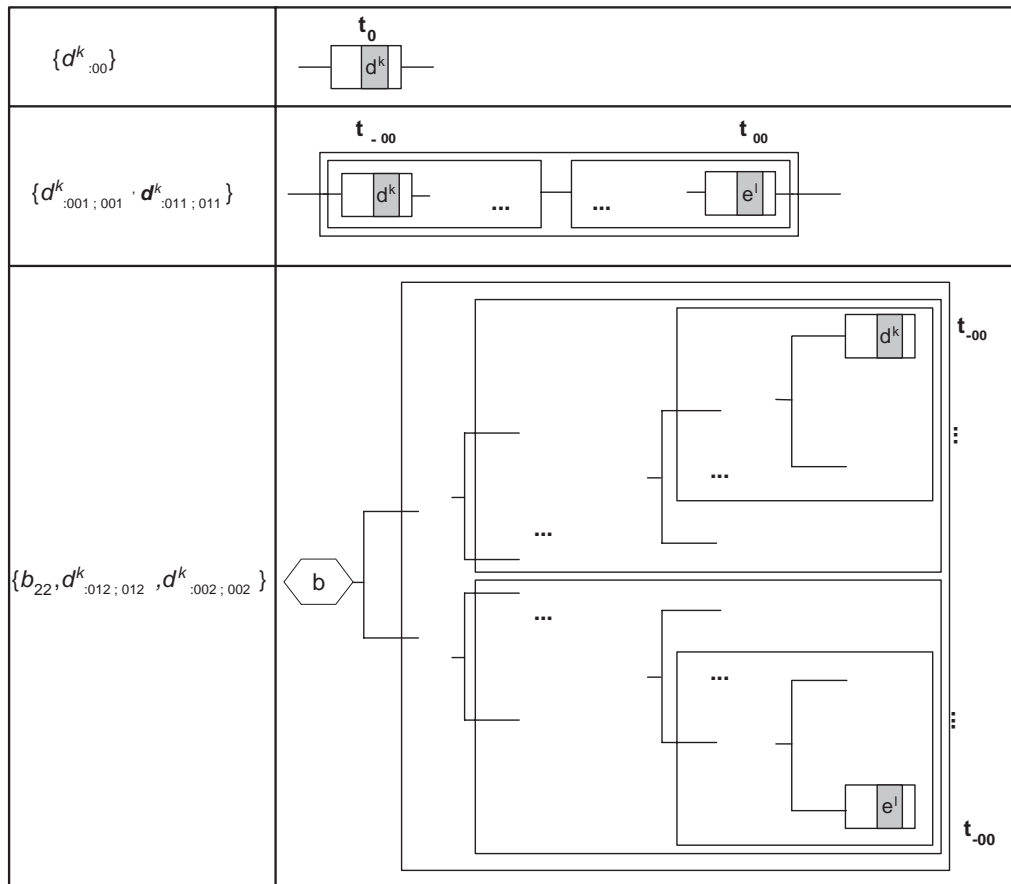


FIGURA 6.9 - Representações de Interpretações em $\pi_{n,\infty}^{\rightarrow\leftarrow}[\mathbb{D}_n]$.

Da mesma forma, $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_n$, $\Pi_{\infty,n}^{\rightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\rightarrow}(x_i)$, e

$$\Pi_{\infty,n+1}^{\rightarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\rightarrow} \circ \gamma_n^{(0)}), \\ \{\mathbf{a} \mid \mathbf{a}_{:011;011} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(011)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\rightarrow} \circ \gamma_n^{(1)}), \\ \{\mathbf{a} \mid \mathbf{a}_{:012;012} \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(012)\}, & \text{se } x_i \in (\pi_{n+1,\infty}^{\rightarrow} \circ \gamma_n^{(2)}), \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Em ambas definições, se $n = 0$ tem-se

$$\Pi_{\infty,0}^{\leftarrow}(x_i) = \Pi_{\infty,0}^{\leftarrow}(x_i) = \begin{cases} \{d^{(k)}\} & \text{se } x = \{d_{:00}^{(k)}\}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Definição 6.21 Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ e as projeções $\Pi_{n+1,\infty}^{\leftarrow}$, $\Pi_{n,\infty}^{\leftarrow} : \mathbb{D}_{\infty}^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n$ apresentadas na Definição 6.20. A função estrita $\Pi_{n,\infty}^{\rightarrow\leftarrow} : \mathbb{D}_{\infty}^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n$ é definida pela união

$$\Pi^{\rightarrow\leftarrow}(x) = \Pi_{n,\infty}^{\leftarrow}(x) \cup \Pi_{n,\infty}^{\rightarrow}(x).$$

Proposição 6.18 As projeções estritas $\Pi_{\infty,n}^{\leftarrow}$, $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n$ apresentadas nas Definições 6.20 e 6.21 são morfismos na categoria CospLin .

6.5.3 Cone Limite $(\mathbb{D}_{\infty}^{\rightarrow\leftarrow}, \Pi_{\infty,n}^{\rightarrow\leftarrow} : \mathbb{D}_{\infty}^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n)$

A seguir, define-se o limite inverso para o diagrama $\mathcal{K}^{\rightarrow\leftarrow}$ que assegura interpretação para todos os processos, incluindo os processos seqüenciais e somas determinísticas infinitos no sentido temporal, mas para o qual se pode determinar o instante inicial e o instante final de sua execução. Entretanto, o tempo entre estes instantes é infinito.

Definição 6.22 O diagrama $\mathcal{K}^{\rightarrow\leftarrow}$ representado por

$$\mathbb{D}_0 \xleftarrow{\Pi_0^{\rightarrow\leftarrow}} \mathbb{D}_1 \xleftarrow{\quad} \dots \xleftarrow{\quad} \mathbb{D}_n \xleftarrow{\Pi_n^{\rightarrow\leftarrow}} \mathbb{D}_{n+1} \xleftarrow{\quad} \dots$$

é definido pela família de subespaços coerentes $\{\mathbb{D}_n\}_{n \in \omega}$ e pelas projeções $\Pi_n^{\rightarrow\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas na Definição 5.7, de tal forma que para todo $w_{n+1} \in \mathbb{D}_{n+1}$ existe $w_n \in \mathbb{D}_n$ e $\Pi_n^{\rightarrow\leftarrow}(w_{n+1}) = w_n$.

Pela última definição, o diagrama $\mathcal{K}^{\rightarrow\leftarrow}$ determina seqüências $\{w_n\}_{n \in \omega}$ de subconjuntos coerentes $w_n \in \mathbb{D}_n$ tal que $\Pi_n^{\rightarrow\leftarrow}(w_{n+1}) = w_n$.

Como conseqüência, cada seqüência $\{w_n\}_{n \in \omega}$ também determina outra seqüência $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_{\infty}$ de aproximações finitas do conjunto coerente x , ou seja $x = \bigcup_{n \in \omega}^{\uparrow} x_n$, onde, para cada subconjunto coerente x_n existe $w_n \in \mathbb{D}_n$ tal que $\Pi_n^{\rightarrow\leftarrow}(x_n) = w_n$. Mais detalhadamente, pela Definição 6.8,

$$w_n = \Pi_n^{\rightarrow\leftarrow}(x_n) \Leftrightarrow \pi_{n,\infty}^{\rightarrow\leftarrow}(w_n) \subseteq x_n \subseteq x,$$

garantindo que uma das seguintes condições é sempre satisfeita

1. $\pi_{n,\infty}^{\rightarrow\leftarrow}(w_n) = \{\mathbf{a}_{:00} \mid \mathbf{a} \in \mathbf{D}_n\} \subseteq x_n \subseteq x$, ou
2. $\pi_{n,\infty}^{\rightarrow\leftarrow}(w_n) = ((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}))(w_n) \times \{(: 001; 001)\} \cup ((\gamma_n^{(1)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)}))(w_n) \times \{(: 011; 011)\} \subseteq x_n \subseteq x$, ou
3. $\pi_{n,\infty}^{\rightarrow\leftarrow}(w_n) = ((\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)}))(w_n) \times \{(: 002; 002)\} \cup ((\gamma_n^{(2)} \circ \psi_n^{(1)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)}))(w_n) \times \{(: 012; 012)\} \subseteq x_n \subseteq x$.

Portanto, a partir da seqüência $\{w_n\}_{n \in \omega}$, define-se a seqüência $\{x_n\}_{n \in \omega}$, cujos subconjuntos coerentes determinam um novo objeto $x \in \mathbb{D}_\infty$, dado por

$$x = \bigcup_{n \in \omega}^\uparrow \{x_n\} \Leftrightarrow x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \subseteq x \text{ e } \Pi_{\infty, n}^{\rightarrow\leftarrow}(x_n) = w_n.$$

A seguir, define-se o limite inverso para o diagrama $\mathcal{K}^{\rightarrow\leftarrow}$.

Proposição 6.19 *Considere a família de projeções $\Pi_{\infty, n}^{\rightarrow\leftarrow} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n$, apresentada na Definição 6.8. O par $\mathbb{D}^{\rightarrow\leftarrow} \equiv (\mathbb{D}_\infty^{\rightarrow\leftarrow}, \Pi_{\infty, n}^{\rightarrow\leftarrow} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama $\mathcal{K}^{\rightarrow\leftarrow}$ apresentado na Definição 6.22.*

Demonstração. A construção da demonstração é análoga àquela da Proposição 6.14.

Proposição 6.20 *Considere os cones $\mathbb{D}^{\leftrightarrow} \equiv (\mathbb{D}_\infty^{\leftrightarrow}, \Pi_{\infty, n}^{\leftrightarrow} : \mathbb{D}_\infty^{\leftrightarrow} \rightarrow \mathbb{D}_n)$ e $\mathbb{D}^{\rightarrow\leftarrow} \equiv (\mathbb{D}_\infty^{\rightarrow\leftarrow}, \Pi_{\infty, n}^{\rightarrow\leftarrow} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n)$, apresentados nas Seções 6.4.3 e 6.5.3. A função linear $\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\leftrightarrow}} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_\infty^{\leftrightarrow}$ definida por*

1. se $x \in \mathbb{P}_\infty^{\rightarrow\leftarrow}$,

$$\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\leftrightarrow}}(x) = x,$$

2. se $x \in \mathbb{P}_\infty^{\rightarrow\leftarrow} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow}$,

$$\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\leftrightarrow}}(x) = \{\mathbf{a}_{:001;011} | \mathbf{a}_{:001;001} \in x\} \cup \{\mathbf{a}_{:011;001} | \mathbf{a}_{:011;011} \in x\},$$

3. se $x \in \mathbb{B} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow}$,

$$\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\leftrightarrow}}(x) = \{\mathbf{a}_{:002;012} | \mathbf{a}_{:002;002} \in x\} \cup \{\mathbf{a}_{:012;002} | \mathbf{a}_{:012;012} \in x\}.$$

é um isomorfismo em CospLin .

O próximo corolário é consequência das Proposições 6.17 e 6.16.

Proposição 6.21 *Considere os cones $\mathbb{D}^{\rightarrow} \equiv (\mathbb{D}_\infty^{\rightarrow}, \Pi_{\infty, n}^{\rightarrow} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \mathbb{D}_n)$ e $\mathbb{D}^{\rightarrow\leftarrow} \equiv (\mathbb{D}_\infty^{\rightarrow\leftarrow}, \Pi_{\infty, n}^{\rightarrow\leftarrow} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n)$, apresentados nas Seções 6.3.3 e 6.5.3. A função linear $\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\rightarrow}} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_\infty^{\rightarrow}$ definida pela expressão*

$$\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\rightarrow}} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_\infty^{\rightarrow}(x) = \begin{cases} x, & \text{se } x \in \mathbb{P}_\infty^{\rightarrow\leftarrow}, \\ \{\mathbf{a}_{:001} | \mathbf{a}_{:001;001} \in x\}, & \text{se } x \in \mathbb{P}_\infty^{\rightarrow\leftarrow} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow}, \\ \{\mathbf{a}_{:002} | \mathbf{a}_{:002;002} \in x\}, & \text{se } x \in \mathbb{B} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow}. \end{cases}$$

é um epimorfismo em CospLin .

Proposição 6.22 *Considere os cones $\mathbb{D}^{\rightarrow} \equiv (\mathbb{D}_\infty^{\rightarrow}, \Pi_{\infty, n}^{\rightarrow} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \mathbb{D}_n)$ e $\mathbb{D}^{\rightarrow\leftarrow} \equiv (\mathbb{D}_\infty^{\rightarrow\leftarrow}, \Pi_{\infty, n}^{\rightarrow\leftarrow} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_n)$, apresentados nas Seções 6.3.3 e 6.5.3. A função linear $\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\rightarrow}} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_\infty^{\rightarrow}$ definida pela expressão*

$$\mathbf{H}_{\mathbb{D}_\infty^{\rightarrow\leftarrow}, \mathbb{D}_\infty^{\rightarrow}} : \mathbb{D}_\infty^{\rightarrow\leftarrow} \rightarrow \mathbb{D}_\infty^{\rightarrow}(x) = \begin{cases} x, & \text{se } x \in \mathbb{P}_\infty^{\rightarrow\leftarrow}, \\ \{\mathbf{a}_{:011} | \mathbf{a}_{:001;001} \in x\}, & \text{se } x \in \mathbb{P}_\infty^{\rightarrow\leftarrow} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow}, \\ \{\mathbf{a}_{:012} | \mathbf{a}_{:002;002} \in x\}, & \text{se } x \in \mathbb{B} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow} \prod \mathbb{P}_\infty^{\rightarrow\leftarrow}. \end{cases}$$

é um epimorfismo em CospLin .

A definição do espaço coerente $\mathbb{D}_\infty^{\rightleftharpoons}$, cujos processos são construídos por sufixo, prefixo e infixo, simultaneamente, pode ser obtida de forma análoga as definições já apresentadas nas seções anteriores deste capítulo. Entretanto tal construção não será formalizada neste texto.

7 Espaço Coerente $\mathbb{D}_{2\infty}^{\rightarrow}$ dos Processos Computacionais Transfinitos

A composição infinita dos morfismos apresentados no Capítulo 6, interpretando construtores e destrutores de processos em \mathcal{P} , definem novos morfismos que comprovam a versatilidade de interpretações obtidas com as construções temporal e espacial que definem a estrutura ordenada do modelo de máquina geométrica.

Neste sentido, os novos pares de funções imersão-projeção viabilizam a construção de novos espaços coerentes, capazes de prover interpretação, por exemplo, para um processo computacional executado, simultaneamente, por um conjunto enumerável \mathcal{M} de máquinas.

Desta maneira, além da noção de seqüencialidade, caracterizada na construção temporal pelos níveis da estrutura indutiva que modela cada máquina geométrica em \mathcal{M} , é possível formalizar a noção de **paralelismo temporal**, cuja estrutura matemática é alcançada pela construção do **espaço coerente $\mathbb{D}_{2\infty}$ dos processos transfinitos**.

Considerando que a indexação dos tokens dos subconjuntos coerentes em $\mathbb{D}_{2\infty}$ possui uma construção análoga àquela dos ordinais transfinitos, sugerida por Stoll em [STO 61], justifica-se portanto a denominação escolhida. Assim, os objetos em $\mathbb{D}_{2\infty}$ são subconjuntos coerentes de tokens rotulados por um conjunto I de posições de espaço geométrico e indexados a partir de um conjunto isomorfo aos ordinais transfinitos. Pela indexação dos tokens, torna-se explícito o construtor modelado e em qual máquina do conjunto enumerável \mathcal{M} o processo (parcial) será executado. As máquinas em \mathcal{M} estão sincronizadas e compartilham uma memória global.

Às observações apresentas no parágrafo anterior, acrescenta-se o fato de que o conjunto de processos elementares é o mesmo para cada máquina durante sua construção. Também neste modelo de máquina geométrica distribuída, a medida do tempo de execução de um processo seqüencial ou soma determinística, definido a partir de um determinado estado computacional, é dada pelo número de processos elementares envolvidos nesta execução. Entretanto, para os demais construtores como o produto paralelo, está análise deve considerar também o número de máquinas envolvidas nesta interpretação.

As etapas que devem ser satisfeitas para definição do espaço coerente $\mathbb{D}_{2\infty}$ estão explicitadas logo a seguir.

1. A modelagem da memória de cada máquina a partir de um conjunto de valores computacionais rotulados por posições de um espaço geométrico.
2. A escolha de um conjunto enumerável \mathcal{M} de máquinas (processadores), possivelmente distribuídas segundo um espaço geométrico. Neste sentido, cada máquina é modelada por um espaço coerente isomorfo a \mathbb{D}_{∞} .
3. A modelagem da memória da máquina geométrica distribuída, a partir de um conjunto de valores computacionais indexados por pontos do conjunto $\mathcal{M} \times I$ isomorfo aos ordinais transfinitos. Neste caso, o conjunto $\mathcal{M} \times I$ consiste na enumeração, possivelmente infinita, de cópias isomorfas do espaço geométrico rotulado por posições em I . Esta modelagem não será formalizada neste trabalho, mas pode ser obtida de forma análoga à modelagem da memória de um modelo de máquina geométrica, apresentada no Capítulo 3.
4. A construção do espaço coerente \mathbb{D}_0 de processos elementares transfinitos, a partir da escolha de um conjunto de ações rotuladas por posições de um conjunto I , indexadas por elementos de um conjunto isomorfo aos ordinais transfinitos.
5. A definição da função imersão π_n^{\rightarrow} (e sua correspondente projeção Π_n^{\rightarrow}) interpretando a construção de processos (prefixo, sufixo, infixo, ou qualquer combinação entre

estes construtores) nos subespaços coerentes \mathbb{D}_n , caracterizando a seqüencialidade no sentido temporal.

6. A completção da estrutura de máquina formalizada pelo espaço coerente $\mathbb{D}_\infty^\rightarrow$ de processos, interpretando processos infinitos quanto ao tempo de execução e caracterizando-se como a base para a construção transindutiva.
7. A definição da função imersão $\pi_{n, \infty+m}^m$, $m \in \omega$, como também de sua correspondente projeção $\Pi_{\infty+m, n}^m$, interpretando a construção de processos transfinitos que constroem os subespaços coerentes $\mathbb{D}_{\infty+m}^\rightarrow$, capaz de interpretar a distribuição de um processo infinito num subconjunto finito de \mathcal{M} , com 2^m modelos MG. Neste caso, cada um dos modelos MG pode representar um processo executado em 2^∞ utc.
8. A completção da estrutura de máquina formalizada pelo espaço coerente $\mathbb{D}_{2^\infty}^\rightarrow$, onde é possível interpretar a execução de um processo transfinito, distribuído entre um conjunto possivelmente infinito (enumerável) de modelos de máquinas geométricas.

A Figura 7.1 mostra o esquema de construção para o espaço coerente \mathbb{D}_{2^∞} de processos computacionais transfinitos, que independe do construtor modelado. Entretanto, nas próximas seções, este trabalho apresenta $\mathbb{D}_{\infty+m}^\rightarrow$ como a estrutura ordenada para o modelo MGD, ou seja, a representação temporal associada à seqüencialidade está fundamentada na construção por prefixo.

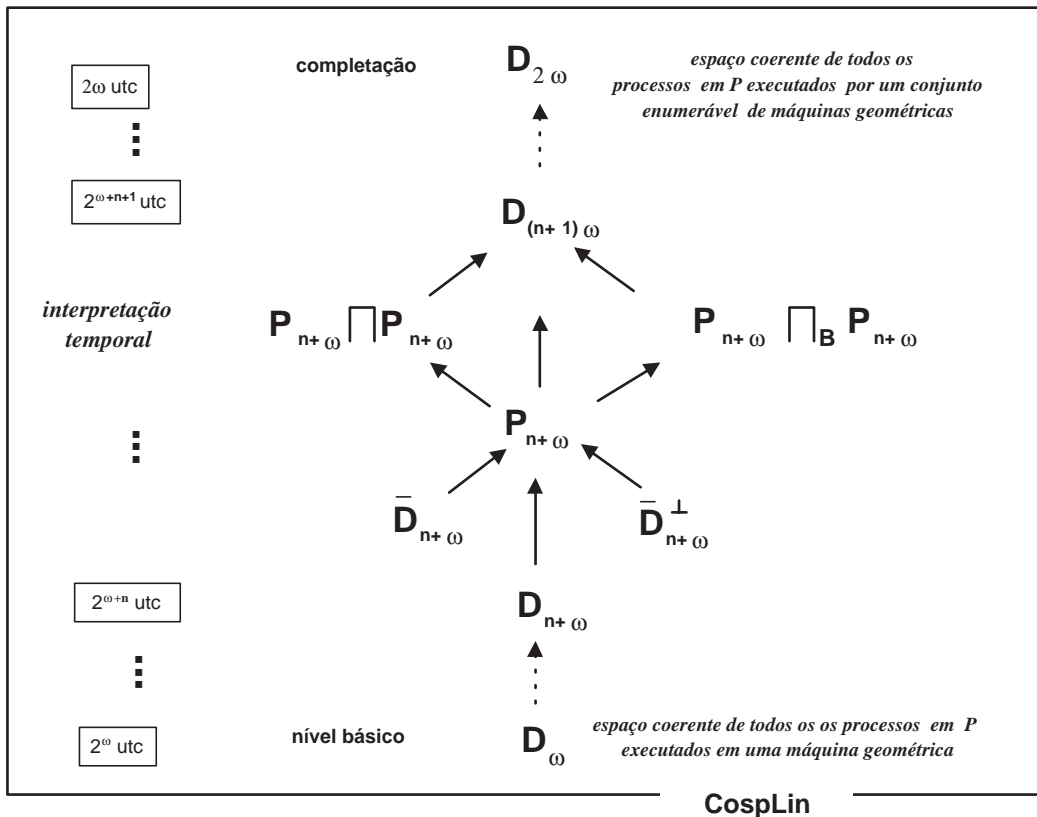


FIGURA 7.1 - Construção do Espaço Coerente \mathbb{D}_{∞^2} .

A metodologia de construção do espaço coerente $\mathbb{D}_{2\infty}^{\rightarrow}$, baseada na indução transfinita, é semelhante à indução finita aplicada nos capítulos anteriores, exceto que, neste caso, a construção inicia com o espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$, apresentado na Definição 6.1, onde a seqüencialidade não se restringe às construções finitas no sentido temporal e está caracterizada pela construção por prefixo. Conseqüentemente, $\mathbb{D}_{\infty}^{\rightarrow} - \mathbb{D}_{\infty+1}^{\rightarrow}$ constitui-se no primeiro nível da estrutura ordenada do modelo de máquina geométrica distribuída (MGD).

A mesma extensão pode ser alcançada para outros modelos de máquinas, como por exemplo $\mathbb{D}_{2\infty}^{\leftarrow}$, $\mathbb{D}_{2\infty}^{\leftrightarrow}$ e $\mathbb{D}_{2\infty}^{\overleftrightarrow{}}$, considerando-se para tal, os correspondentes domínios $\mathbb{D}_{\infty}^{\leftarrow}$, $\mathbb{D}_{\infty}^{\leftrightarrow}$ e $\mathbb{D}_{\infty}^{\overleftrightarrow{}}$, já construídos anteriormente no Capítulo 6, nas Seções 6.3, 6.4 e 6.5.

Para facilitar a compreensão dos morfismos e objetos envolvidos nas definições e proposições deste capítulo, mostra-se primeiramente, a construção indutiva do espaço coerente $\mathbb{D}_{\infty+1}^{\rightarrow}$, a partir dos subespaços coerentes \mathbb{D}_n e das projeções $\Pi_{\infty+1,n}^1$ e Π_n^1 e das correspondentes imersões. A generalização das etapas da construção, apresentada posteriormente, é obtida pela definição do domínio $\mathbb{D}_{\infty+m}^{\rightarrow}$, considerando-se as projeções $\Pi_{\infty+m,n}^m$ e Π_n^m e as suas imersões.

Na última seção, verifica-se que $\mathbb{D}_{2\infty}^{\rightarrow} \equiv (\mathbb{D}_{2\infty}^{\rightarrow}, \Pi_{2\infty,\infty+m}^{\rightarrow})$ constitui-se num cone limite para o diagrama $\mathcal{K}_{2\infty}^{\rightarrow}$, obtido a partir dos domínios $\mathbb{D}_{\infty+m}^{\rightarrow}$ e das projeções $\Pi_{\infty+m,\infty+(m+1)}^{\rightarrow} : \mathbb{D}_{\infty+(m+1)}^{\rightarrow} \rightarrow \mathbb{D}_{\infty+m}^{\rightarrow}$. Neste caso, os sub-espaços coerentes $\mathbb{D}_{\infty+1}^{\rightarrow}$ resultam da completção dos subníveis $\mathbb{D}_n - \mathbb{D}_{n+1}$ e consideradas as projeções $\Pi_n^m : \mathbb{D}_{n+1}^{\rightarrow} \rightarrow \mathbb{D}_n^{\rightarrow}$.

Para não tornar o texto repetitivo, as proposições apresentadas ao longo deste capítulo não foram demonstradas. Entretanto, as provas podem ser construídas de forma análoga às correspondentes, já construídas, nos capítulos anteriores.

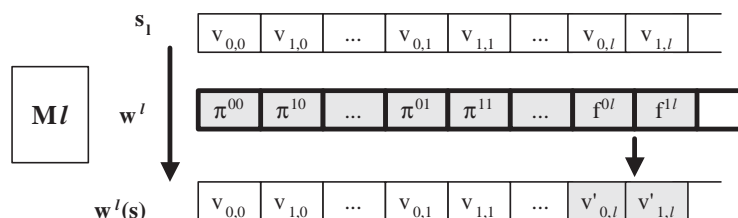


FIGURA 7.2 - Construção do Processo Transfinito W em \mathcal{M} .

Exemplificação 7.1

Esta exemplificação é construída a partir da análise do processo transfinito W apresentado na Figura 7.2. Para facilitar a representação gráfica, consideram-se \mathcal{A} um conjunto infinito de ações, I um conjunto enumerável, rotulando posições de um espaço geométrico, isomorfo aos naturais e a enumeração do conjunto de máquinas \mathcal{M} .

Para exemplificar o paralelismo temporal em uma máquina geométrica distribuída, seja o processo transfinito \mathbf{W} formado pelo produto paralelo transfinito definido sobre o conjunto de processos infinitos $w^{(l)}$. Cada processo $w^{(l)}$ executa, seqüencialmente, a mesma ação $w \in \mathcal{A}$ em todas as posições de memória rotuladas por posições em I , na mesma máquina \mathcal{M}_l .

Com base nestas considerações e a representação apresentada na Figura 7.2, tem-se as seguintes observações.

1. Cada processo w será executado por uma máquina e após sua execução todas as posições de memória desta máquina serão alterada. Na figura 7.2, optou-se por especificar que $M_l \in \mathcal{M}$ executa o processo $w^{(l)}$. Cada um destes processos inicia sua execução no instante t_0 e não se pode determinar quando terminará cada execução.
2. Assim, seguindo a especificação anterior, o processo transfinito W executa, em cada instante de tempo, todas as ações modeladas na mesma posição de memória.

Após um intervalo de tempo finito e enumerável, o processo parcial W executou o conjunto de ações, afetando a mesma posição de memória em cada um dos instantes de tempo deste intervalo.

Portanto, pode-se prever que, considerando a sincronização da construção temporal, o processo W deverá executar todas as possíveis ações modeladas, alterando todas as posições de memória. Neste caso, cada novo instante de tempo na sua execução determina uma outra posição de memória alterada.

3. O processo $W \in P$ é interpretado pelo subconjunto coerente $x \in \mathbb{D}_{2\infty}$, apresentado logo abaixo, sendo $w^{(l)} = \{ w_{:001}^{(0l)}, w_{011:001}^{(1l)}, \dots, w_{(00)^{n-1}.011:001}^{(nl)}, \dots \}$

$$x = \{ w_{10:00}^{(0)}, \dots, w_{10:00}^{(1)}, \dots, w_{10:00}^{(l)}, w_{10:00}^{(l+1)}, \dots \}_{w \in \mathcal{A}}$$

Isto é justamente o contrário do paralelismo a partir de uma única máquina, onde a execução simultânea só é possível entre processos concorrentes. Neste exemplo, tem-se um conjunto de memórias sincronizadas, mas totalmente independentes.

4. Seja $w \in \{d, e, f, \dots\} = \mathcal{A}$. O processo $W \in P$ interpretado pelo subconjunto coerente $x \in \mathbb{D}_{2\infty}$ pode ser indicado pela expressão logo abaixo.

$$x = \{ \{ d_{:001}^{(00)}, d_{011:001}^{(10)}, \dots, d_{(00)^{n-1}.011:001}^{(n0)}, \dots \}_{10:00}, \\ \{ e_{:001}^{(01)}, e_{011:001}^{(11)}, \dots, e_{(00)^{n-1}.011:001}^{(n1)}, \dots \}_{10:00}, \\ \{ f_{:001}^{(02)}, f_{011:001}^{(12)}, \dots, f_{(00)^{n-1}.011:001}^{(n2)}, \dots \}_{10:00}, \dots \}$$

O conjunto coerente x representação um produto paralelo, cuja execução modela o paralelismo temporal, e com tempo de execução infinito.

A condição que determina o paralelismo temporal é definida também por uma função posição-máquina, que determina a coerência sempre que os processos infinitos estão em máquinas possivelmente distintas do conjunto \mathcal{M} .

7.1 Espaço Coerente $\mathbb{D}_{\infty+1}^{\rightarrow}$

Com o objetivo de definir o espaço coerente $\mathbb{D}_{\infty+1}^{\rightarrow}$, segue-se a mesma metodologia de construção do espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$, apresentada na seção 6.3.

De acordo com as Definições 6.1 e 6.3, considera-se aqui o espaço coerente

$$\mathbb{D}_{\infty+1}^{\rightarrow} = \mathbb{P}_{\infty+1}^{\rightarrow} \amalg (\mathbb{P}_{\infty+1}^{\rightarrow} \amalg \mathbb{P}_{\infty+1}^{\rightarrow}) \amalg (\mathbb{P}_{\infty+1}^{\rightarrow} \amalg \mathbb{P}_{\infty+1}^{\rightarrow}) \quad ,$$

onde o subespaço $\mathbb{P}_{\infty+1}^{\rightarrow}$ corresponde à soma direta $\mathbb{P}_{\infty+1}^{\rightarrow} = \mathbb{D}_{\infty+1}^{\rightarrow} \amalg \overline{\mathbb{D}_{\infty+1}^{\rightarrow}} \amalg \overline{\mathbb{D}_{\infty+1}^{\rightarrow}}^{\perp}$.

Os objetos em $\mathbb{D}_{\infty+1}^{\rightarrow}$ interpretam todos os processos em \mathcal{P} . Também são modelados aqueles processos resultantes do produto seqüencial entre dois processos infinitos modelados em $\mathbb{D}_{\infty}^{\rightarrow}$, como é o caso dos processos seqüenciais infinitos construídos por prefixação. Estas modelagens são identificadas por objetos construídos em $\mathbb{D}_n^{\rightarrow}$ e imersos em $\mathbb{D}_{\infty+1}^{\rightarrow}$.

Neste sentido, os objetos em $\mathbb{D}_{\infty+1}^{\rightarrow}$ podem interpretar produtos seqüenciais onde ambos fatores são processos seqüenciais infinitos. Nesta modelagem, pode-se determinar em que instante, ou intervalo de tempo, o primeiro processo de cada fator será executado, e o mesmo para todos aqueles processos que o sucedem na ordem de construção temporal.

Entretanto, considerando a construção (trans)indutiva de $\mathbb{D}_{\infty+1}^{\rightarrow}$, não é possível determinar o último processo de cada fator dos processos seqüenciais interpretados. Portanto, não é possível determinar quando será executado o processo que antecede o primeiro fator de cada produto seqüencial infinito.

7.1.1 Imersões e Projeções π_n^{\rightarrow} e Π_n^{\rightarrow}

Apresenta-se agora as imersões $\pi_n^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e as correspondentes projeções $\Pi_n^{\rightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, que também interpretam construtores e destrutores de processos, respectivamente.

Neste caso, o limite para o diagrama definido a partir das projeções $\Pi_n^{\rightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ formaliza a completção da estrutura indutiva do modelo de máquina geométrica para processos computacionais transfinitos, prefixados.

No domínio $\mathbb{D}_{\infty+1}^{\rightarrow}$ é possível interpretar, entre outros, os processos seqüenciais transfinitos, onde, a partir de cada um dos instantes t_0 e t_{∞} pode-se identificar um produto seqüencial com infinitos fatores. Isto significa que também é possível determinar os processos que serão executados nos instantes que sucedem t_0 e t_{∞} . Entretanto, salienta-se que, diferentemente dos números naturais, não é possível determinar os processos que serão executados nos instantes que precedem t_0 e t_{∞} .

As imersões π_n^{\rightarrow} e as projeções Π_n^{\rightarrow} são definidas logo a seguir.

Definição 7.1 *Considere as funções $\pi_n^{\rightarrow(\beta)}$, $\pi_n^{\leftarrow(\beta)}$ e $\Pi_n^{\rightarrow(\beta)}$, $\Pi_n^{\leftarrow(\beta)}$ do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, onde o indexador $\beta \in \{0, 1, 2\}$. As imersões $\pi_n^{\rightarrow(0)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\rightarrow(1)} : \overline{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\rightarrow(2)} : \overline{\mathbb{D}}_n^{\perp} \rightarrow \mathbb{D}_{n+1}$ são definidas pela união*

$$\pi_n^{\rightarrow(\beta)}(x) = \pi_n^{\rightarrow(\beta)}(x) \cup \pi_n^{\leftarrow(\beta)}(x).$$

Neste caso, as correspondentes funções $\pi_n^{\leftarrow(0)}$, $\pi_n^{\leftarrow(1)}$: $\mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\leftarrow(1)}$, $\pi_n^{\leftarrow(2)}$: $\overline{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\leftarrow(2)}$, $\pi_n^{\leftarrow(2)}$: $\overline{\mathbb{D}}_n^{\perp} \rightarrow \mathbb{D}_{n+1}$ são dadas pelas expressões

$$\pi_{n+1}^{\rightarrow(\beta)}(x) = \begin{cases} (\pi_{n+1}^{\rightarrow(\beta)} \circ \pi_n^{\rightarrow} \circ \Pi_n^{\rightarrow})(x), & \text{se } x \notin \mathbb{P}_{n-1} \text{ e } n \geq 1, \\ \pi^{(0\beta)}(x), & \text{caso contrário;} \end{cases}$$

$$\pi_{n+1}^{\leftarrow(\beta)}(x) = \begin{cases} (\pi_{n+1}^{\leftarrow(\beta)} \circ \pi_n^{\leftarrow} \circ \Pi_n^{\leftarrow})(x), & \text{se } x \notin \mathbb{P}_{n-1} \text{ e } n \geq 1, \\ \pi^{(0\beta)}(x), & \text{caso contrário.} \end{cases}$$

Se $n = 0$ então $\pi_1^{\rightarrow(\beta)}(x) = \pi_1^{\leftarrow(\beta)}(x) = \pi^{(00)}(x)$.

Quando $\beta = 0$, utiliza-se a notação $\pi_n^{\rightarrow(0)} = \pi_n^{\rightarrow}$.

Em particular, $\pi_n^{\frac{1}{\rightarrow}(\beta)}(\emptyset) = \emptyset$.

A Exemplificação 7.2 refere-se aos morfismos introduzidos na última definição.

Exemplificação 7.2

As interpretações que ilustram a aplicação da Definição 7.1 estão representadas na Figura 7.3. Para tal, considera-se o subconjunto coerente $x \in \mathbb{D}_2 \amalg \mathbb{D}_2 \subseteq \mathbb{D}_3$,

$$x = \{d_{(001)^3}^{(k)}, d_{011.(001)^2}^{(l)}, d_{001.011.001}^{(m)}, d_{(011)^2.001}^{(j)}, e_{(001)^2.011}^{(k)}, e_{011.001.011}^{(l)}, e_{001.(011)^2}^{(m)}, e_{(011)^3}^{(j)}\},$$

e as composições $\pi_{3,n}^{\frac{1}{\rightarrow}} : \mathbb{D}_3 \rightarrow \mathbb{D}_n$, $\pi_{3,n}^{\frac{1}{\rightarrow}} = \pi_{n-1}^{\frac{1}{\rightarrow}} \circ \dots \circ \pi_4^{\frac{1}{\rightarrow}} \circ \pi_3^{\frac{1}{\rightarrow}}$. Pela Definição 7.1, obtém-se os subconjuntos coerentes

$$\begin{aligned} \cdot \pi_{3,4}^{\frac{1}{\rightarrow}}(x) &= \{d_{(001)^4}^{(k)}, d_{011.(001)^3}^{(l)}, d_{001.011.(001)^2}^{(m)}, d_{(011)^2.(001)^2}^{(j)}, \\ &\quad e_{(001)^3.011}^{(k)}, e_{011.(001)^2.011}^{(l)}, e_{001.011.001.011}^{(m)}, e_{(011)^2.001.011}^{(j)}\}, \\ \cdot \pi_{3,5}^{\frac{1}{\rightarrow}}(x) &= \{d_{(001)^5}^{(k)}, d_{011.(001)^4}^{(l)}, d_{001.011.(001)^3}^{(m)}, d_{(011)^2.(001)^3}^{(j)}, \\ &\quad e_{(001)^4.011}^{(k)}, e_{011.(001)^3.011}^{(l)}, e_{001.011.(001)^2.011}^{(m)}, e_{(011)^2.(001)^2.011}^{(j)}\}, \\ &\vdots \\ \cdot \pi_{3,n+1}^{\frac{1}{\rightarrow}}(x) &= \{d_{(001)^{n+1}}^{(k)}, d_{011.(001)^n}^{(l)}, d_{001.011.(001)^{n-1}}^{(m)}, d_{(011)^2.(001)^{n-1}}^{(j)}, \\ &\quad e_{(001)^n.011}^{(k)}, e_{011.(001)^{n-1}.011}^{(l)}, e_{001.011.(001)^{n-2}.011}^{(m)}, e_{(011)^2.(001)^{n-2}.011}^{(j)}\}, \end{aligned}$$

respectivamente representados na Figura 7.3, logo a seguir.

Definição 7.2 Considere as funções $\pi_n^{\rightarrow(\beta)}$, $\pi_n^{\leftarrow(\beta)}$ e $\Pi_n^{\rightarrow(\beta)}$, $\Pi_n^{\leftarrow(\beta)}$ do nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, onde o indexador $\beta \in \{0, 1, 2\}$. As projeções $\Pi_n^{\frac{1}{\rightarrow}(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\frac{1}{\leftarrow}(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\frac{1}{\leftarrow}(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são definidas pela união

$$\Pi_n^{\frac{1}{\rightarrow}(\beta)}(x) = \Pi_n^{\frac{1}{\rightarrow}(\beta)}(x) \cup \Pi_n^{\frac{1}{\leftarrow}(\beta)}(x).$$

Neste caso, as correspondentes funções $\Pi_n^{\frac{1}{\rightarrow}(0)}$, $\Pi_n^{\frac{1}{\leftarrow}(0)} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\frac{1}{\rightarrow}(1)}$, $\Pi_n^{\frac{1}{\leftarrow}(1)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n$ e $\Pi_n^{\frac{1}{\leftarrow}(2)}$, $\Pi_n^{\frac{1}{\leftarrow}(2)} : \mathbb{D}_{n+1} \rightarrow \bar{\mathbb{D}}_n^\perp$ são dadas pelas expressões

$$\Pi_n^{\frac{1}{\rightarrow}(\beta)}(x) = \begin{cases} (\pi_{n-1}^{\rightarrow(\beta)} \circ \Pi_{n-1}^{\rightarrow} \circ \Pi_n^{\rightarrow})(x), & \text{se } x \notin \mathbb{P}_n \text{ e } n \geq 1, \\ \Pi_n^{(0\beta)}(x), & \text{caso contrário;} \end{cases}$$

$$\Pi_n^{\frac{1}{\leftarrow}(\beta)}(x) = \begin{cases} (\pi_{n-1}^{\leftarrow(\beta)} \circ \Pi_{n-1}^{\leftarrow} \circ \Pi_n^{\leftarrow})(x), & \text{se } x \notin \mathbb{P}_n \text{ e } n \geq 1, \\ \pi_n^{(0\beta)}(x), & \text{caso contrário.} \end{cases}$$

Se $n = 0$ então $\Pi_0^{\frac{1}{\rightarrow}(\beta)}(x) = \Pi_0^{\frac{1}{\leftarrow}(\beta)}(x) = \Pi_n^{(0\beta)}(x)$.

Quando $\beta = 0$, utiliza-se a notação $\Pi_n^{\frac{1}{\rightarrow}(0)} = \Pi_n^{\frac{1}{\rightarrow}}$.

Em particular, $\Pi_n^{\frac{1}{\rightarrow}(\beta)}(\emptyset) = \emptyset$.

Proposição 7.1 As projeções $\Pi_n^{\frac{1}{\rightarrow}(\beta)}$, $\Pi_n^{\frac{1}{\leftarrow}(\beta)}$ e as imersões $\pi_n^{\frac{1}{\rightarrow}(\beta)}$, $\pi_n^{\frac{1}{\leftarrow}(\beta)}$ no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$, apresentadas nas Definições 7.1 e 7.2 são morfismos na categoria CospLin .

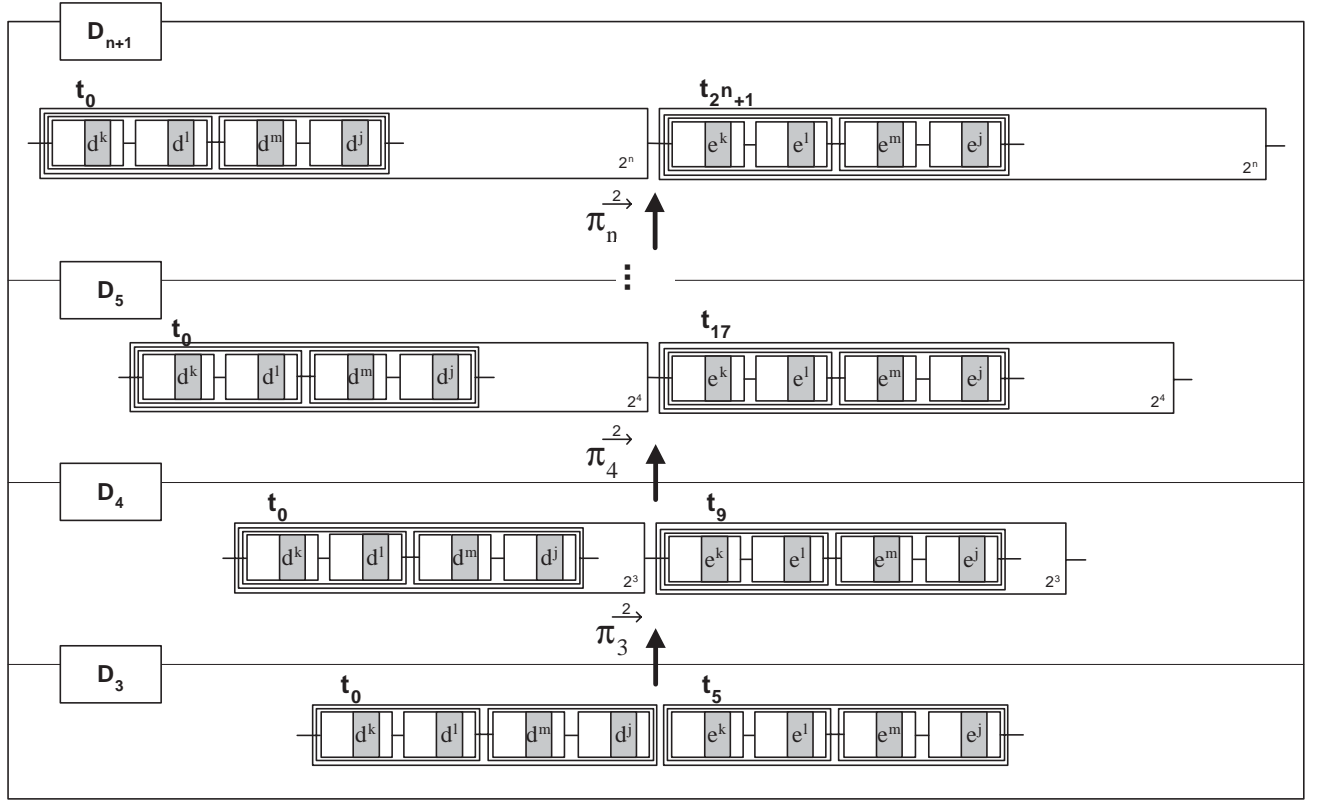


FIGURA 7.3 - Aplicações da Imersão $\pi_{n,m}^1$ sobre Objetos em $[\mathbb{D}_n]$.

7.1.2 Imersões $\pi_{n,\infty+1}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty+1}^{\rightarrow}$ e Projeções $\Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$.

Definição 7.3 Considere os espaços coerentes \mathbb{D}_n e $\mathbb{D}_{\infty+1}^{\rightarrow}$ como também as funções $\pi_{n,\infty}^{(00)}$ apresentadas na Definição 6.5 e as projeções $\Gamma_n^{(\beta)}$ e Ψ_n respectivamente apresentados na Definições 5.1 e 6.11. A função imersão $\pi_{n,\infty+1}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty+1}^{\rightarrow}$ é dada por

$$\pi_{n,\infty+1}^{\rightarrow}(x) = \pi_{n,\infty+1}^{\leftarrow}(x) \cup \pi_{n,\infty+1}^{\rightarrow}(x).$$

Neste caso, as projeções $\pi_{n,\infty+1}^{\leftarrow}, \pi_{n,\infty+1}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty+1}^{\rightarrow}$ são, respectivamente definidas pelas seguintes expressões

$$\pi_{n+1,\infty+1}^{\leftarrow}(x) = \begin{cases} \pi_{n,\infty}^{(00)}(x), & \text{se } x \in \mathbb{P}_n, \\ ((\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)}))(x) \times \{(: 001; 001)\}, & \text{se } x \in \mathbb{P}_n \prod \mathbb{P}_n, \\ ((\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(2)}))(x) \times \{(: 002; 002)\}, & \text{se } x \in \mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

$$\pi_{n+1,\infty+1}^{\rightarrow}(x) = \begin{cases} \pi_{n,\infty}^{(00)}(x) & \text{se } x \in \mathbb{P}_n, \\ (\gamma_n^{(1)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)})(x) \times \{(: 001; 011)\}, & \text{se } x \in \mathbb{P}_{n-1} \prod \mathbb{P}_n, \\ (\gamma_n^{(2)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(2)})(x) \times \{(: 002; 012)\}, & \text{se } x \in \mathbb{P}_{n-1} \prod_{\mathbb{B}} \mathbb{P}_n, \\ \emptyset & \text{caso contrário.} \end{cases}$$

Neste caso, $\pi_{0,\infty+1}^{\rightarrow}(x) = \{\mathbf{a}_{:00} \mid \mathbf{a} \in x\}$ e $\pi_{n,\infty+1}^{\rightarrow}(\emptyset) = \emptyset$.

Exemplificação 7.3

Considere as interpretações $w = \{d_{(001)^2}^{(k)}, e_{011.001}^l\}$, $v = \{d_{(002)^2}^{(k)}, e_{012.002}^l\}$ e $x \in \mathbb{D}_3$,

$$x = \{d_{(001)^3}^{(k)}, d_{011.(001)^2}^{(l)}, d_{001.011.001}^{(m)}, d_{(011)^2.001}^{(j)}, e_{(001)^2.011}^{(k)}, e_{011.001.011}^{(l)}, e_{001.(011)^2}^{(m)}, e_{(011)^3}^{(j)}\}.$$

Pela aplicação da Definição 7.3, tem-se

$$\begin{aligned} \bullet \pi_{3,\infty+1}^{\rightarrow}(x) &= \{d_{:001;001}^{(k)}, d_{011:001;001}^{(l)}, d_{001.011:001;001}^{(m)}, d_{(011)^2:001;001}^{(j)}, \\ &\quad e_{:001;011}^{(k)}, e_{011:001;011}^{(l)}, e_{001.011:001;011}^{(m)}, e_{(011)^2:001;011}^{(j)}\}. \\ \bullet \pi_{2,\infty+1}^{\rightarrow}(w) &= \{d_{:001;001}^{(k)}, e_{:001;011}^{(l)}\} \\ \bullet \pi_{2,\infty+1}^{\rightarrow}(v) &= \{d_{:002;002}^{(k)}, e_{:012;012}^{(l)}\} \subseteq \{b_{22}, d_{:002;002}^{(k)}, e_{:012;012}^{(l)}\}. \end{aligned}$$

Alguns das interpretações acima estão representadas na Figura 7.4, logo a seguir.

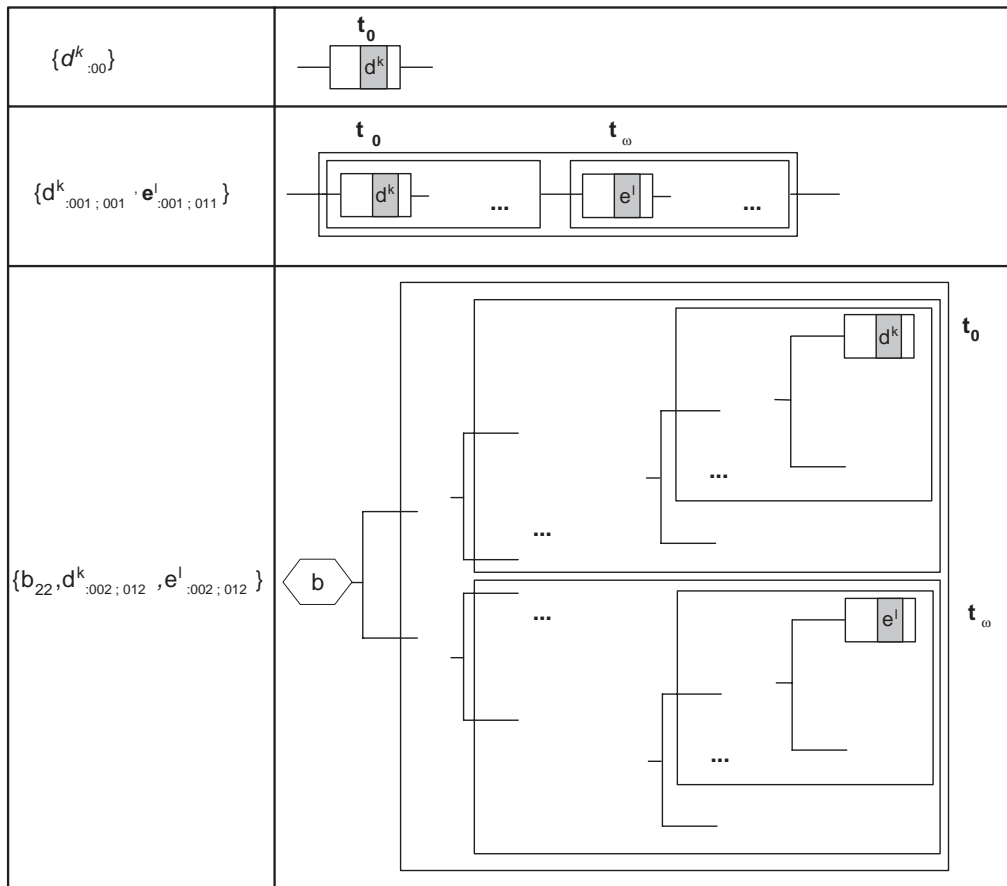


FIGURA 7.4 - Representações das Interpretações em $\pi_{\infty+1,n}^{\rightarrow}[\mathbb{D}_n]$.

Os objetos na imagem das projeções $\Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$ são definidos como a união dos subconjuntos coerentes em $\Pi_{\infty+1,n}^{\leftarrow}[\mathbb{D}_n]$ e $\Pi_{\infty+1,n}^{\rightarrow}[\mathbb{D}_n]$. Suas definições são apresentadas logo abaixo, seguidas das correspondentes proposições que asseguram a linearidade.

Definição 7.4 Considere os espaços coerentes $\mathbb{D}_n \equiv (\text{Coh } \mathbf{D}_n, \subseteq)$ e $\mathbb{D}_{\infty+1}^{\rightarrow}$, as imersões $\gamma_n^{(\beta)}$ e $\pi_{n,\infty}$ respectivamente apresentados na Proposição 5.1 e na Definição 6.12, e o indexador $\beta \in \{0, 1, 2\}$. Seja $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\} \in \mathbb{D}_{\infty}$, onde cada $x_i \in \mathbb{D}_{\infty}^{\leftarrow}$ é uma aproximação finita de x . A função $\Pi_{n,\infty+1}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$ é dada por

$$\Pi_{n,\infty+1}^{\rightarrow}(x) = \Pi_{n,\infty+1}^{\leftarrow}(x) \cup \Pi_{n,\infty+1}^{\rightarrow}(x).$$

Sendo que $\Pi_{\infty+1,n}^{\leftarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$ são definidas por $\Pi_{\infty,n}^{\leftarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\leftarrow}(x_i)$, e

$$\Pi_{\infty+1,n+1}^{\leftarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}:00 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x_i \in (\pi_{n+1,\infty+1}^{\rightarrow} \circ \gamma_n^{(0)}), \\ \{\mathbf{a} \mid \mathbf{a}:001;001 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(001)\}, & \text{se } x_i \in (\pi_{n+1,\infty+1}^{\rightarrow} \circ \gamma_n^{(1)}), \\ \{\mathbf{a} \mid \mathbf{a}:002;002 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(002)\}, & \text{se } x_i \in (\pi_{n+1,\infty+1}^{\rightarrow} \circ \gamma_n^{(2)}), \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, $\Pi_{\infty+1,n}^{\leftarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$ é dada por $\Pi_{\infty+1,n}^{\leftarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty,n}^{\leftarrow}(x_i)$, e

$$\Pi_{\infty+1,n+1}^{\leftarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}:00 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_{n+1}\}, & \text{se } x_i \in (\pi_{n+1,\infty+1}^{\rightarrow} \circ \gamma_n^{(0)}), \\ \{\mathbf{a} \mid \mathbf{a}:001;011 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(011)\}, & \text{se } x_i \in (\pi_{n+1,\infty+1}^{\rightarrow} \circ \gamma_n^{(1)}), \\ \{\mathbf{a} \mid \mathbf{a}:002;012 \in x_i \text{ e } \mathbf{a} \in \mathbf{D}_n\} \times \{(012)\}, & \text{se } x_i \in (\pi_{n+1,\infty+1}^{\rightarrow} \circ \gamma_n^{(2)}), \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Se $n = 0$ tem-se $\Pi_{\infty+1,0}^{\leftarrow}(x_i) = \Pi_{\infty+1,0}^{\leftarrow}(x_i) = \begin{cases} \{d^{(k)}\} & \text{se } x_i = \{d_{:00}^{(k)}\}, \\ \emptyset, & \text{caso contrário.} \end{cases}$

Proposição 7.2 As projeções estritas $\Pi_{\infty+1,n}^{\leftarrow}, \Pi_{\infty+1,n}^{\leftarrow}, \Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$ apresentadas nas Definições 7.3 são morfismos na categoria CospLin .

7.1.3 Cone Limite $(\mathbb{D}_{\infty+1}^{\rightarrow}, \Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n)$

A seguir, define-se o limite inverso para o diagrama \mathcal{K}^{\leftarrow} que assegura interpretação para todos os processos em P , incluindo os processos seqüenciais e somas determinísticas infinitos no sentido temporal, para os quais se pode determinar o instante inicial de sua execução.

Definição 7.5 O diagrama \mathcal{K}^{\leftarrow} representado por

$$\mathbb{D}_0 \xleftarrow{\Pi_0^{\leftarrow}} \mathbb{D}_1 \leftarrow \dots \leftarrow \mathbb{D}_n \xleftarrow{\Pi_n^{\leftarrow}} \mathbb{D}_{n+1} \leftarrow \dots$$

é definido pela família de subespaços coerentes $\{\mathbb{D}_n\}_{n \in \omega}$ e pelas projeções $\Pi_n^{\leftarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas na Definição 7.1, de tal forma que para todo $w_{n+1} \in \mathbb{D}_{n+1}$ existe $w_n \in \mathbb{D}_n$ e $\Pi_n^{\leftarrow}(w_{n+1}) = w_n$.

Pela última definição, o diagrama \mathcal{K}^{\leftarrow} determina seqüências $\{w_n\}_{n \in \omega}$ de subconjuntos coerentes $w_n \in \mathbb{D}_n$ tal que $\Pi_n^{\leftarrow}(w_{n+1}) = w_n$.

Como conseqüência, cada seqüência $\{w_n\}_{n \in \omega}$ também determina outra seqüência $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_{\infty+1}$ de aproximações finitas do conjunto coerente x . Neste caso, $x =$

$\bigcup_{n \in \omega}^{\uparrow} x_n$, onde, para cada subconjunto coerente x_n existe $w_n \in \mathbb{D}_n$ tal que $\Pi_{\infty+1,n}^{\rightarrow}(x_n) = w_n$. Mais detalhadamente, pela Definição 7.10,

$$w_n = \Pi_{\infty+1,n}^{\rightarrow}(x_n) \Leftrightarrow \pi_{n,\infty+1}^{\rightarrow}(w_n) \subseteq x_n \subseteq x.$$

Portanto, a seqüência $\{x_n\}_{n \in \omega}$ é formada por subconjuntos coerentes e determinam um novo objeto $x \in \mathbb{D}_{\infty+1}$, dado por

$$x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\} \Leftrightarrow x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \subseteq x \text{ e } \Pi_{\infty+1,n}^{\rightarrow}(x_n) = w_n.$$

A Proposição 7.3 caracteriza o domínio $\mathbb{D}_{\infty+1}^{\rightarrow}$ como o limite inverso para $\mathcal{K}^{\rightarrow}$.

Proposição 7.3 *Considere a família de projeções $\Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n$, apresentada na Definição 6.8. O par $\mathbb{D}_{\infty+1}^{\rightarrow} \equiv (\mathbb{D}_{\infty+1}^{\rightarrow}, \Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama $\mathcal{K}^{\rightarrow}$ apresentado na Definição 7.10.*

7.1.4 Relacionamento entre $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty+1}^{\rightarrow}$

Proposição 7.4 *Considere os cones $\mathbb{D}_{\infty+1}^{\rightarrow} \equiv (\mathbb{D}_{\infty+1}^{\rightarrow}, \Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n)$ e $\mathbb{D}_{\infty}^{\rightarrow} \equiv (\mathbb{D}_{\infty}^{\rightarrow}, \Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_n)$, apresentados nas Seções 7.1.3 e 6.3.3. A função linear $\Pi_{\mathbb{D}_{\infty+1}^{\rightarrow}, \mathbb{D}_{\infty}^{\rightarrow}} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ definida pela expressão*

$$\Pi_{\mathbb{D}_{\infty+1}^{\rightarrow}, \mathbb{D}_{\infty}^{\rightarrow}}(x) = \begin{cases} x, & \text{se } x \in \mathbb{P}_{\infty+1}^{\rightarrow}, \\ \{\mathbf{a}:001 \mid \mathbf{a}:001:001 \in x\}, & \text{se } x \in \mathbb{P}_{\infty+1}^{\rightarrow} \prod \mathbb{P}_{\infty+1}^{\rightarrow}, \\ \{\mathbf{a}:002 \mid \mathbf{a}:002:002 \in x\}, & \text{se } x \in \mathbb{P}_{\infty+1}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{P}_{\infty+1}^{\rightarrow}. \end{cases}$$

é um epimorfismo em CospLin .

A Figura 7.5 ilustra a proposição 7.4, representando graficamente as projeções que tornam explícito o relacionamento entre os espaços coerentes $\mathbb{D}_{\infty+1}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\rightarrow}$.

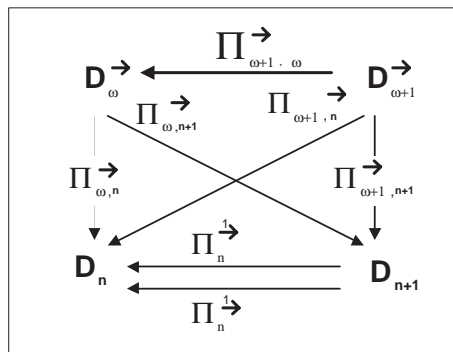


FIGURA 7.5 - Relacionamento entre $\mathbb{D}_{\infty+1}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\rightarrow}$.

7.1.5 Relacionamento entre os Domínios $\mathbb{D}_{\infty+1}^{\rightarrow}$, $\mathbb{D}_{\infty}^{\leftrightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$

Proposição 7.5 *Considere os cones $\mathbb{D}_{\infty+1}^{\rightarrow} \equiv (\mathbb{D}_{\infty+1}^{\rightarrow}, \Pi_{\infty+1,n}^{\rightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_n)$, $\mathbb{D}_{\infty}^{\leftrightarrow} \equiv (\mathbb{D}_{\infty}^{\leftrightarrow}, \Pi_{\infty,n}^{\leftrightarrow} : \mathbb{D}_{\infty}^{\leftrightarrow} \rightarrow \mathbb{D}_n)$ e $\mathbb{D}_{\infty}^{\leftarrow} \equiv (\mathbb{D}_{\infty}^{\leftarrow}, \Pi_{\infty,n}^{\leftarrow} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_n)$ apresentados nas Seções 7.1.3 e 6.4.3 e 6.5.3, respectivamente. Considere as funções lineares $H_{(00,00)}$, $H_{(11,01)} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\leftarrow}$ e $H_{(10,00)}$, $H_{(01,01)} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\leftrightarrow}$ dadas pela seguintes expressões*

$$\mathbf{H}_{(\alpha\beta,\gamma\theta)}(x) = \begin{cases} x, & \text{se } x \in \mathbb{P}_{\infty+1}^{\rightarrow}, \\ \{\mathbf{a}:0\gamma 1; 0\theta 1 \mid \mathbf{a}:0\alpha 1; 0\beta 1 \in x\}, & \text{se } x \in \mathbb{P}_{\infty+1}^{\rightarrow} \prod \mathbb{P}_{\infty+1}^{\rightarrow}, \\ \{\mathbf{a}:0\gamma 2; 0\theta 2 \mid \mathbf{a}:0\alpha 2; 0\beta 2 \in x\}, & \text{se } x \in \mathbb{P}_{\infty+1}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{P}_{\infty+1}^{\rightarrow}, \end{cases}$$

onde $\alpha, \beta, \gamma, \theta \in \{0, 1\}$. As funções lineares definidas pelas seguintes uniões

- $\mathbf{H}^{\rightarrow\leftarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\leftarrow}$, $\mathbf{H}^{\rightarrow\leftarrow}(x) = H_{(00,00)}(x) \cup H_{(11,01)}(x)$ e
- $\mathbf{H}^{\leftrightarrow} : \mathbb{D}_{\infty+1}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\leftrightarrow}$, $\mathbf{H}^{\leftrightarrow}(x) = H_{(10,00)}(x) \cup H_{(01,01)}(x)$,

são isomorfismos em CospLin .

Proposição 7.6 *Considere os espaços coerentes isomorfos $\mathbb{D}_{\infty+1}^{\rightarrow}$, $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$. As seguintes composições são satisfeitas:*

1. No diagrama (I), tem-se

- 1.1. $\Pi_{\infty,n}^{\leftarrow} \circ \mathbf{H}^{\leftrightarrow} = \Pi_{\infty+1,n+1}^{\rightarrow} \circ \Pi_n^{\leftarrow}$,
- 1.2. $\Pi_{\infty,n}^{\leftarrow} \circ \mathbf{H}^{\leftrightarrow} = \Pi_{\infty+1,n}^{\rightarrow}$, e
- 1.3. $\Pi_n^{\leftarrow} \circ \Pi_{2\infty,n}^{\rightarrow} = \Pi_{\infty+1,n}^{\rightarrow}$.

2. No diagrama (II), verificam-se as condições

- 2.1. $\Pi_{\infty,n}^{\rightarrow\leftarrow} \circ \mathbf{H}^{\rightarrow\leftarrow} = \Pi_{\infty+1,n+1}^{\rightarrow} \circ \Pi_n^{\leftarrow}$,
- 2.2. $\Pi_{\infty,n}^{\rightarrow\leftarrow} \circ \mathbf{H}^{\rightarrow\leftarrow} = \Pi_{\infty+1,n}^{\rightarrow}$, e
- 2.3. $\Pi_{\infty,n}^{\rightarrow\leftarrow} \circ \Pi_{\infty+1,n}^{\rightarrow} = \Pi_{\infty+1,n}^{\rightarrow}$.

3. Em ambos diagramas (I) e (II), tem-se

- 3.1. $\Pi_{\infty,n}^{\leftarrow} \circ \mathbf{H}^{\leftrightarrow} \circ (\mathbf{H}^{\rightarrow\leftarrow})^{-1} = \Pi_{\infty,n}^{\rightarrow\leftarrow}$, e
- 3.2. $\mathbf{H}^{\rightarrow\leftarrow} \circ (\mathbf{H}^{\leftrightarrow})^{-1} \circ \Pi_{\infty,n}^{\rightarrow\leftarrow} = \Pi_{\infty,n}^{\leftarrow}$.

A Figura 7.6 ilustra as proposições 7.4 e 7.6, representando graficamente as projeções que tornam explícitos os relacionamentos entre os espaços coerentes $\mathbb{D}_{\infty+1}^{\rightarrow}$, $\mathbb{D}_{\infty}^{\leftrightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$.

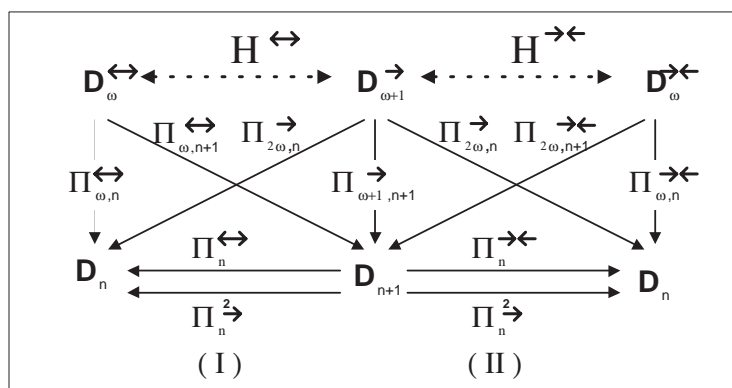


FIGURA 7.6 - Os Espaços Coerentes $\mathbb{D}_{\infty+1}^{\rightarrow}$, $\mathbb{D}_{\infty}^{\leftarrow}$ e $\mathbb{D}_{\infty}^{\leftrightarrow}$.

7.2 Espaço Coerente $\mathbb{D}_{\infty+m}^{\rightarrow}$

Com o objetivo de definir o espaço coerente $\mathbb{D}_{\infty+m}^{\rightarrow}$, segue-se a mesma metodologia de construção dos espaços coerentes $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty+1}^{\rightarrow}$, apresentados nas seções 6.3 e 7.1.

De acordo com as Definições 6.1 e 6.3, o espaço coerente $\mathbb{D}_{\infty+m}^{\rightarrow}$ é dado por

$$\mathbb{D}_{\infty+m}^{\rightarrow} = \mathbb{P}_{\infty+m}^{\rightarrow} \amalg (\mathbb{P}_{\infty+m}^{\rightarrow} \amalg \mathbb{P}_{\infty+m}^{\rightarrow}) \amalg (\mathbb{P}_{\infty+m}^{\rightarrow} \amalg_{\mathbb{B}} \mathbb{P}_{\infty+m}^{\rightarrow}),$$

onde o subespaço $\mathbb{P}_{\infty+m}^{\rightarrow}$ corresponde à soma direta $\mathbb{P}_{\infty+m}^{\rightarrow} = \mathbb{D}_{\infty+m}^{\rightarrow} \amalg \overline{\mathbb{D}}_{\infty+m}^{\rightarrow} \amalg \overline{\mathbb{D}}_{\infty+m}^{\rightarrow \perp}$.

Os objetos em $\mathbb{D}_{\infty+m}^{\rightarrow}$ interpretam além de todos os processos em \mathcal{P} , também àqueles processos resultantes do produto seqüencial com n fatores, os quais podem ser processos infinitos modelados em $\mathbb{D}_{\infty}^{\rightarrow}$. Estas modelagens são identificadas por objetos construídos em \mathbb{D}_n e imersos em $\mathbb{D}_{\infty+m}^{\rightarrow}$.

Por exemplo, os objetos em $\mathbb{P}_{\infty+m}^{\rightarrow} \amalg \mathbb{P}_{\infty+m}^{\rightarrow}$ interpretam produtos seqüenciais onde os m fatores podem interpretar processos seqüenciais infinitos e prefixados. Neste caso, pode-se determinar quando será executado o primeiro processo de cada fator assim como todos que os sucedem. Neste caso, restringindo-se a análise ao tempo associado a cada execução parcial, em cada um destes processos seqüenciais, tem-se m seqüências infinitas de instantes de tempo computacional que correspondem aos m fatores. O esquema abaixo apresenta a representação do tempo computacional de um processo seqüencial interpretado por um objeto total em $\mathbb{P}_{\infty+m}^{\rightarrow} \amalg \mathbb{P}_{\infty+m}^{\rightarrow}$.

$$\underbrace{\underbrace{t_0, t_1, t_2, \dots, t_\infty, t_{\infty+1}, t_{\infty+2}, \dots, t_{2\infty}, \dots, t_{m\infty}, t_{m\infty+1}, t_{m\infty+2}, \dots, t_{(m+1)\infty}}_{2^\infty \text{ utc}}}_{2^{\infty+1} \text{ utc}}_{2^{\infty+m} \text{ utc}}$$

Contudo, considerando a construção (trans)indutiva de $\mathbb{D}_{\infty+m}^{\rightarrow}$, não é possível determinar o último processo de cada fator de um processo seqüencial interpretado em $\mathbb{P}_{\infty+m}^{\rightarrow} \amalg \mathbb{P}_{\infty+m}^{\rightarrow}$. Portanto, não é possível determinar quais processos serão executados nos instantes que precedem $t_{m\infty}$, sempre que $m \in \omega$.

7.2.1 Imersões π_n^{\rightarrow} e Projeções Π_n^{\rightarrow} em $\mathbb{D}_n - \mathbb{D}_{n+1}$

Em relação à seqüencialidade, nos subespaços $\pi_n^{\rightarrow}[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1}$ são interpretados produtos seqüenciais finitos, cujo número máximo de fatores é igual a m e onde cada fator

é executado em 2^{n+1-m} *utc*.

Em relação às somas determinísticas, nos subespaços $\pi_n^{\overrightarrow{m}}[\mathbb{D}_n] \subseteq \mathbb{D}_{n+1}$ existem interpretações para um conjunto finito de escolhas, cuja execução se dará em 2^{n-m} *utc*.

Na busca destas interpretações, o subespaço imagem $\pi_n^{\overrightarrow{m}}[\mathbb{D}_n]$ é introduzido como uma união entre os subespaços imagens $\pi_n^{\overrightarrow{m}}[\mathbb{D}_n]$ e $\pi_n^{\overleftarrow{m}}[\mathbb{D}_n]$.

Definição 7.6 *Considere as funções $\Pi_n^{\overrightarrow{m}}$, $\Pi_n^{\overleftarrow{m}}$ e $\pi_n^{\overrightarrow{m}}$, $\pi_n^{\overleftarrow{m}}$ apresentadas na Seção 6.3 e as imersões $\pi_n^{(\beta 0)}$, $\beta \in \{0, 1, 2\}$, apresentadas na Definição 6.5. Fixado $m \in \omega$, e supondo $\pi_n^{\overrightarrow{0}}(\beta)(x) = \pi_n^{\overrightarrow{(\beta)}}(x)$, a definição das funções $\pi_n^{\overrightarrow{m}}(\beta)$, para $m \neq 0$, no nível $\mathbb{D}_n - \mathbb{D}_{n+1}$ é dada pela união*

$$\pi_n^{\overrightarrow{m}}(\beta)(x) = \pi_n^{\overrightarrow{m}}(\beta)(x) \cup \pi_n^{\overleftarrow{m}}(\beta)(x).$$

Neste caso, as imersões $\pi_n^{\overrightarrow{m}}(0)$, $\pi_n^{\overleftarrow{m}}(0) : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, $\pi_n^{\overrightarrow{m}}(1)$, $\pi_n^{\overleftarrow{m}}(1) : \bar{\mathbb{D}}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_n^{\overrightarrow{m}}(2)$, $\pi_n^{\overleftarrow{m}}(2) : \bar{\mathbb{D}}_n^\perp \rightarrow \mathbb{D}_{n+1}$ são definidas por casos, pelas expressões

• Se $n \geq m$ tem-se

$$\pi_n^{\overrightarrow{m}}(\beta)(x) = \begin{cases} (\pi_n^{\overrightarrow{(\beta)}} \circ \pi_{n-1}^{\overleftarrow{m-1}} \circ \Pi_{n-1}^{\overrightarrow{m}})(x), & \text{se } \Pi_n^{\overrightarrow{m}}(x) \notin \gamma_n^{(0)}[\mathbb{P}_{n-1}], \\ \pi_n^{(\beta 0)}(x), & \text{caso contrário;} \end{cases}$$

$$\pi_n^{\overleftarrow{m}}(\beta)(x) = \begin{cases} (\pi_n^{\overleftarrow{(\beta)}} \circ \pi_{n-1}^{\overrightarrow{m-1}} \circ \Pi_{n-1}^{\overleftarrow{m}})(x), & \text{se } \Pi_n^{\overleftarrow{m}}(x) \notin \gamma_{n-1}^{(0)}[\mathbb{P}_{n-1}], \\ \pi_n^{(\beta 0)}(x), & \text{caso contrário.} \end{cases}$$

• Se $n < m$, $\pi_n^{\overrightarrow{m}}(\beta)(x) = \pi_n^{\overleftarrow{m}}(\beta)(x) = \pi_n^{(\beta 0)}(x)$.

Quando $\beta = 0$ tem-se $\pi_n^{\overrightarrow{m}}(0)(x) = \pi_n^{\overrightarrow{m}}(x)$.

Em particular, tem-se $\pi_n^{\overrightarrow{m}}(\beta)(\emptyset) = \emptyset$ e $\pi_0^{\overrightarrow{m}}(\beta)(x) = \pi_0^{\overrightarrow{(\beta)}}(x)$.

As projeções $\Pi_n^{\overrightarrow{m}}$ em $\mathbb{D}_n - \mathbb{D}_{n+1}$ são definidas de forma análoga.

Definição 7.7 *Considere as funções $\Pi_n^{\overrightarrow{m}}$, $\Pi_n^{\overleftarrow{m}}$, $\pi_n^{\overrightarrow{m}}$, $\pi_n^{\overleftarrow{m}}$, apresentadas na Seção 6.3, e as funções $\Pi_n^{(\beta 0)}$, apresentadas na Definição 6.5, onde o indexador $\beta \in \{0, 1, 2\}$. Fixado $m \in \omega$ e supondo $\Pi_n^{\overrightarrow{0}}(\beta)(x) = \Pi_n^{\overrightarrow{(\beta)}}(x)$, a definição recursiva das projeções $\Pi_n^{\overrightarrow{m}}(\beta)$ em $\mathbb{D}_{n+1} - \mathbb{D}_n$ é dada pela expressão*

$$\Pi_n^{\overrightarrow{m}}(\beta)(x) = \Pi_n^{\overrightarrow{m}}(\beta)(x) \cup \Pi_n^{\overleftarrow{m}}(\beta)(x).$$

Neste caso, as projeções $\Pi_n^{\overrightarrow{m}}(0)$, $\Pi_n^{\overleftarrow{m}}(0) : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$, $\Pi_n^{\overrightarrow{m}}(1)$, $\Pi_n^{\overleftarrow{m}}(1) : \bar{\mathbb{D}}_{n+1} \rightarrow \mathbb{D}_n$ e $\Pi_n^{\overrightarrow{m}}(2)$, $\Pi_n^{\overleftarrow{m}}(2) : \bar{\mathbb{D}}_{n+1}^\perp \rightarrow \mathbb{D}_n$ são dadas por casos pelas expressões

• Se $n \geq m$ tem-se

$$\Pi_n^{\overrightarrow{m}}(\beta)(x) = \begin{cases} (\pi_{n-1}^{\overrightarrow{(\beta)}} \circ \Pi_{n-1}^{\overleftarrow{m-1}} \circ \Pi_n^{\overrightarrow{m}})(x), & \text{se } \Pi_n^{\overrightarrow{m}}(x) \notin \gamma_n^{(0)}[\mathbb{P}_{n-1}], \\ \Pi_n^{(\beta 0)}(x), & \text{caso contrário;} \end{cases}$$

$$\Pi_n^{\overleftarrow{m}}(\beta)(x) = \begin{cases} (\pi_{n-1}^{\overleftarrow{(\beta)}} \circ \Pi_{n-1}^{\overrightarrow{m-1}} \circ \Pi_n^{\overleftarrow{m}})(x), & \text{se } \Pi_n^{\overleftarrow{m}}(x) \notin \gamma_{n-1}^{(0)}[\mathbb{P}_{n-1}], \\ \Pi_n^{(\beta 0)}(x), & \text{caso contrário.} \end{cases}$$

• Se $n < m$, $\Pi_n^{\overrightarrow{m}(\beta)}(x) = \Pi_n^{\overrightarrow{m}(\beta)}(x) = \Pi_{n+1}^{(\beta 0)}(x)$.

Quando $\beta = 0$ $\Pi_n^{\overrightarrow{m}(0)}(x) = \Pi_n^{\overrightarrow{m}}(x)$.

Em particular, $\Pi_n^{\overrightarrow{m}(\beta)}(\emptyset) = \emptyset$ e $\Pi_0^{\overrightarrow{m}(\beta)}(x) = \Pi_0^{\overrightarrow{(\beta)}}(x)$.

Proposição 7.7 *As funções $\pi_n^{\overrightarrow{m}} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e $\Pi_n^{\overrightarrow{m}} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas nas Definições 7.6 e 7.7 são morfismos na categoria CospLin .*

Com base na linearidade, assegurada pela Proposição 7.7, a próxima proposição expressa as funções $\pi_n^{\overrightarrow{m}}$ e $\Pi_n^{\overrightarrow{m}}$ a partir dos tokens, que definem os subconjuntos coerentes nas teias \mathbf{D}_n , sob as quais se constroem os subespaços \mathbb{D}_n . Desta forma, a construção de subconjuntos nas respectivas imagens $\pi_n^{\overrightarrow{m}}[\mathbb{D}_n]$ e $\Pi_n^{\overrightarrow{m}}[\mathbb{D}_{n+1}]$ torna-se mais simplificada, veja Exemplificação 7.4. Para tal, considera-se o alfabeto finito definido pelo conjunto

$$\Sigma = \{00, 001, 011, 002, 012, 10, 101, 111, 102, 112, 20, 201, 211, 202, 212\}.$$

Seja Σ^* o conjunto de todas as palavras finitas, incluindo a palavra vazia, geradas pela concatenação de símbolos do alfabeto Σ . Por esta notação, na próxima definição, $\alpha \in \Sigma$ é um indexador e $\theta \in \Sigma^*$ é uma seqüência finita de indexadores.

Proposição 7.8 *Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_{\infty+m}^{\rightarrow}$ e as funções $\pi_n^{\overrightarrow{m}}$ e $\Pi_n^{\overrightarrow{m}}$ definidas no nível $\mathbb{D}_n - \mathbb{D}_{n+1}^{\rightarrow}$, e apresentadas nas Definições 7.6 e 7.7. Fixado $m \in \omega$, as seguintes equivalências são satisfeitas:*

• Se $n \geq m$ tem-se

$$\pi_n^{\overrightarrow{m}(\beta)}(x) = \begin{cases} \{\mathbf{a}_{\beta 0} \mid \mathbf{a} \in x\}, & \text{se } x \in \mathbb{P}_{n-1}, \\ \{\mathbf{a}_{\alpha, \beta 01, \theta} \mid \mathbf{a}_{\alpha \theta} \in x \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}, & \text{se } x \in \mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}, \\ \{\mathbf{a}_{\alpha, \beta 02, \theta} \mid \mathbf{a}_{\alpha \theta} \in x \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}, & \text{se } x \in \mathbb{P}_{n-1} \prod_{\mathbb{B}} \mathbb{P}_{n-1}; \end{cases}$$

$$\Pi_n^{\overrightarrow{m}(\beta)}(x) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}_{\beta 0} \in x\}, & \text{se } x \in \mathbb{P}_n, \\ \{\mathbf{a}_{\alpha, \theta} \mid \mathbf{a}_{\alpha, \beta 01, \theta} \in x \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}, & \text{se } x \in \mathbb{P}_n \prod \mathbb{P}_n, \\ \{\mathbf{a}_{\alpha, \theta} \mid \mathbf{a}_{\alpha, \beta 02, \theta} \in x \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}, & \text{se } x \in \mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n. \end{cases}$$

• Se $n < m$ tem-se $\pi_n^{\overrightarrow{m}(\beta)}(x) = \{\mathbf{a}_{\beta 0} \mid \mathbf{a} \in x\}$ e $\Pi_n^{\overrightarrow{m}(\beta)}(x) = \{\mathbf{a} \mid \mathbf{a}_{\beta 0} \in x\}$.

A composição finita dos morfismos $\pi_n^{\overrightarrow{m}}$ apresentados na Definição 7.6, é expressa por $\pi_{n, n+k}^{\overrightarrow{m}} = \pi_n^{\overrightarrow{m}} \circ \pi_{n+1}^{\overrightarrow{m}} \circ \dots \circ \pi_{n+k}^{\overrightarrow{m}}$, com $n, m, k \in \omega$. Pela proposição 7.8, se $x \in \mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}$, tem-se $\pi_{n, n+k}^{\overrightarrow{m}}(x) = \{\mathbf{a}_{\alpha, (001)^{k+1}\beta} \mid \mathbf{a}_{\alpha, 001, \beta} \in x \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}$.

A Figura 7.7 relaciona-se com a Exemplificação 7.4 e ilustra a Proposição 7.8.

Exemplificação 7.4

Nesta exemplificação, primeiro se considera o subconjunto $w \in \mathbb{D}_m$ dado por

$$w = \{d_{(001)_m}^{(k)}, d_{011.(001)^{m-1}}^{(k)}, \dots, d_{(011)^{m-1}.001}^{(k)}, d_{(001)^{m-1}.011}^{(k)}, d_{011.(001)^{m-2}.011}^{(k)}, \dots, d_{(011)_m}^{(k)}\}.$$

No conjunto imagem $\pi_{m, m+n}^{\overrightarrow{m}}[\mathbb{D}_m] \subseteq \mathbb{D}_{m+n}$, tem-se que

$$\pi_{m, m+n}^{\overrightarrow{m}}(w) = \{d_{(001)^{n+m}}^{(k)}, d_{(001)^n.011.(001)^{m-1}}^{(k)}, \dots, d_{(001)^n.(011)^{m-1}.001}^{(k)}, \\ d_{(001)^{n+m-1}.011}^{(k)}, d_{(001)^n.011.(001)^{m-2}.011}^{(k)}, \dots, d_{(001)^n.(011)_m}^{(k)}\}$$

Completando esta exemplificação, considere

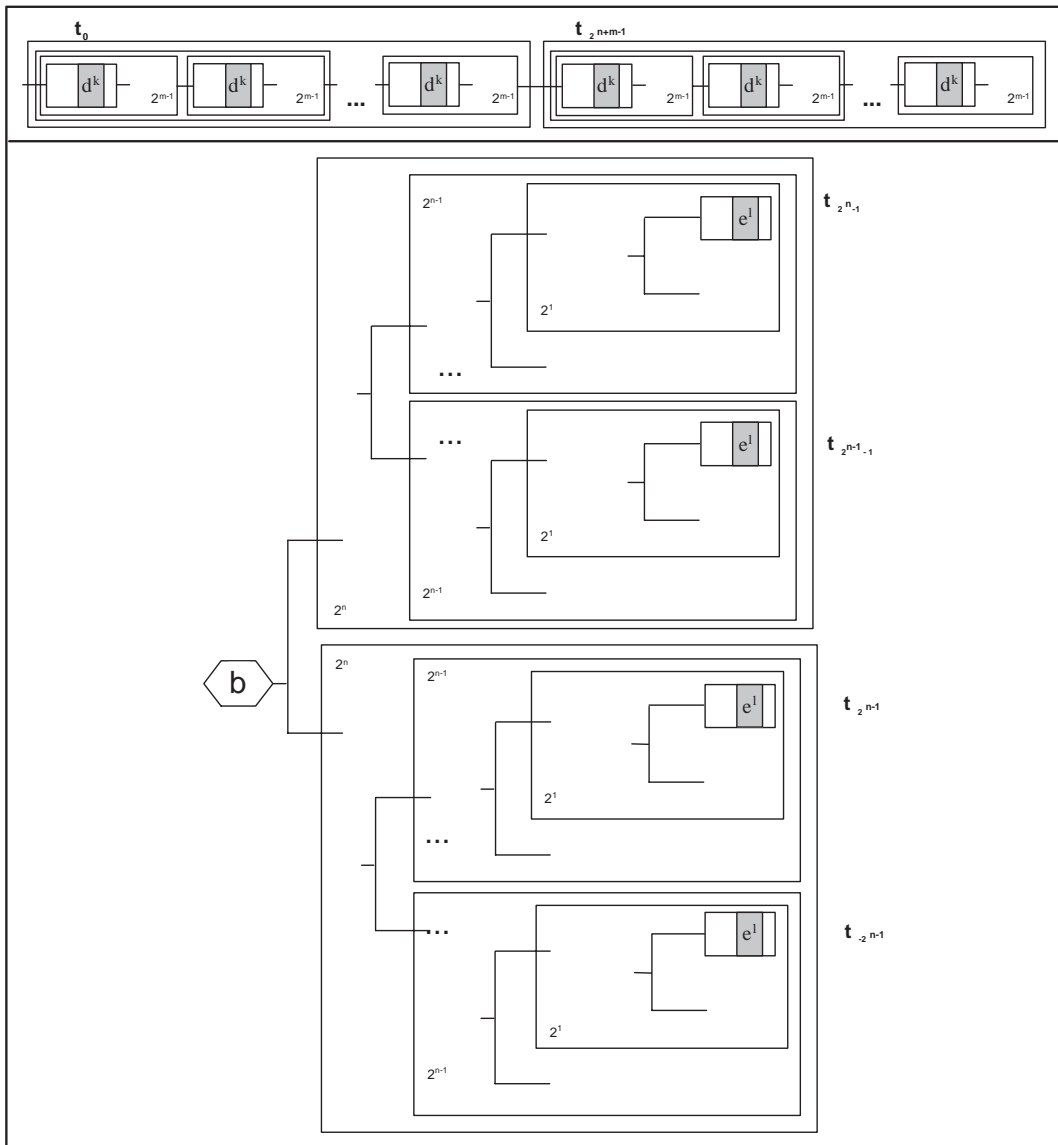


FIGURA 7.7 - Representações das Imersões $\pi_{m,m+n}^m[\mathbb{D}_m]$.

$$u = \{e_{(002)^m}^{(l)}, e_{012.(002)^{m-1}}^{(l)}, \dots, e_{002.(012)^{m-1}}^{(l)}, e_{(012)^m}^{(l)}\} \text{ em } \mathbb{D}_m.$$

Neste caso, tem-se $\pi_{m,m+n}(u) \in \mathbb{D}_{m+n}$,

$$\pi_{m,m+n}(u) = \{e_{(002)^{m+n}}^{(l)}, e_{(002)^n.012.(002)^{m-1}}^{(l)}, \dots, e_{(002)^{n+1}.(012)^{m-1}}^{(l)}, e_{(002)^n.(012)^m}^{(l)}\}.$$

7.2.2 Imersões $\pi_{n,\infty+m}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty+m}^{\rightarrow}$ e Projeções $\Pi_{\infty+m,n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$

A definição das funções $\pi_{n,\infty+m}^{\rightarrow}$ e $\Pi_{\infty+m,n}^{\rightarrow}$ é alcançada considerando-se as igualdades expressas na Proposição 7.8.

Definição 7.8 Considere os espaços coerentes \mathbb{D}_n e $\mathbb{D}_{\infty+m}^{\rightarrow}$ como também as funções $\pi_{n,\infty}^{\rightarrow}$ e $\gamma_n^{(\beta)}$. Fixado $m \in \omega$, $m \neq 0$, a função imersão $\pi_{n,\infty+m}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$ é dada por casos, pelas seguintes expressões

• Se $n \geq m$, tem-se

$$\pi_{n,\infty+m}^{\rightarrow}(x) = \begin{cases} \{\mathbf{a}_{:00} \mid \mathbf{a} \in x\}, & \text{se } x \in \gamma_n^{(0)}[\mathbb{P}_n], \\ \{\mathbf{a}_{\alpha:001;\theta} \mid \mathbf{a}_{\alpha\theta} \in x \text{ e } \mathbf{a}_{\alpha\theta} \in \mathbf{D}_{n-m}\}, & \text{se } x \in \gamma_{n-1}^{(1)}[\mathbb{P}_{n-1} \prod \mathbb{P}_{n-1}], \\ \{\mathbf{a}_{\alpha:002;\theta} \mid \mathbf{a}_{\alpha\theta} \in x \text{ e } \mathbf{a}_{\alpha\theta} \in \mathbf{D}_{n-m}\}, & \text{se } x \in \gamma_{n-1}^{(2)}[\mathbb{P}_{n-1} \prod_{\mathbb{B}} \mathbb{P}_{n-1}]. \end{cases}$$

• Se $n < m - 1$, $\pi_{n,\infty+m}^{\rightarrow}(x) = \{\mathbf{a}_{:00} \mid \mathbf{a} \in x\}$.

Em particular, $\pi_{n,m\infty}^{\rightarrow}(\emptyset) = \emptyset$.

Definição 7.9 Considere os espaços coerentes \mathbb{D}_n , $\mathbb{D}_{\infty+m}^{\rightarrow}$ e as imersões $\gamma_n^{(\beta)}$ e $\pi_{n,\infty+m}$, respectivamente apresentadas na Proposição 5.1 e na Definição 7.8. Suponha $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\} \in \mathbb{D}_{\infty+m}$, onde cada x_i é uma aproximação finita de x . Fixado $m \in \omega$, a função $\Pi_{\infty+m,n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$ é dada por

$$\Pi_{\infty+m,n}^{\rightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{\infty+m,n}^{\rightarrow}(x_i),$$

onde as funções $\Pi_{\infty+m,n}^{\rightarrow}(x_i) : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$ estão definidas pelas expressões

• Se $n \geq m$,

$$\Pi_{\infty+m,n}^{\rightarrow}(x_i) = \begin{cases} \{\mathbf{a} \mid \mathbf{a}_{:00} \in x_i\}, & \text{se } x_i \in \pi_{n,\infty+m}^{\rightarrow}[\mathbb{P}_n], \\ \{\mathbf{a}_{\alpha.\theta} \mid \mathbf{a}_{\alpha:001;\theta} \in x_i \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}, & \text{se } x_i \in \pi_{n,\infty+m}^{\rightarrow}[\mathbb{P}_n \prod \mathbb{P}_n], \\ \{\mathbf{a}_{\alpha.\theta} \mid \mathbf{a}_{\alpha:002;\theta} \in x_i \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{n-m}\}, & \text{se } x_i \in \pi_{n,\infty+m}^{\rightarrow}[\mathbb{P}_n \prod_{\mathbb{B}} \mathbb{P}_n]. \\ \emptyset, & \text{caso contrário.} \end{cases}$$

• Se $n < m$, $\Pi_{n,\infty+m}^{\rightarrow}(x_i) = \{\mathbf{a} \mid \mathbf{a} \in x_i\}$.

Proposição 7.9 As projeções estritas $\pi_{n,\infty+m}^{\rightarrow} \Pi_{\infty+m,n}^{\rightarrow}$ apresentadas nas Definições 7.8 e 7.9 são morfismos na categoria $\text{Cosp}\mathcal{L}in$.

7.2.3 Cone Limite ($\mathbb{D}_{\infty+m}^{\rightarrow}, \Pi_{\infty+m,n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$)

A seguir, define-se o limite inverso para o diagrama \mathcal{K}^m que assegura interpretação para todos os processos, incluindo a aplicação finita de construtores como o produto seqüencial e a soma determinística sobre processos infinitos no sentido temporal e modelados por $\mathbb{D}_{\infty}^{\rightarrow}$, para os quais se pode determinar o instante inicial de sua execução.

O número m de aplicações destes construtores de processos caracteriza o subespaço $\mathbb{D}_{\infty+m}^{\rightarrow}$ em que as interpretações são construídas. Entretanto, cada nível $\mathbb{D}_{\infty+m}^{\rightarrow} - \mathbb{D}_{\infty+m+1}^{\rightarrow}$ da construção admite a aplicação infinita dos construtores como o produto paralelo e a soma não-determinística. Neste caso, respeitando as condições de conflito e concorrência modeladas a partir das funções $\Upsilon_{\mathbb{D}_{\infty+m}^{\rightarrow}}$, que são extensões da função posição $\Upsilon_{\mathbb{D}_{\infty}^{\rightarrow}}$, apresentadas na Definição 6.2.

Definição 7.10 O diagrama \mathcal{K}^m representado por

$$\mathbb{D}_0 \xleftarrow{\Pi_0^m} \mathbb{D}_1 \longleftarrow \dots \longleftarrow \mathbb{D}_n \xleftarrow{\Pi_n^m} \mathbb{D}_{n+1} \longleftarrow \dots$$

é definido pela família de subespaços coerentes $\{\mathbb{D}_n\}_{n \in \omega}$ e pelas projeções $\Pi_n^m : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ apresentadas na Definição 7.8, de tal forma que para todo $w_{n+1} \in \mathbb{D}_{n+1}$ existe $w_n \in \mathbb{D}_n$ e $\Pi_n^m(w_{n+1}) = w_n$.

O diagrama \mathcal{K}^m determina seqüências $\{w_n\}_{n \in \omega}$ de subconjuntos coerentes em \mathbb{D}_n tal que $\Pi_n^m(w_{n+1}) = w_n$. Como conseqüência, tem-se as seqüências $\{x_n\}_{n \in \omega} \subseteq \mathbb{D}_{\infty+m}$ de aproximações finitas do conjunto coerente $x = \bigcup_{n \in \omega} x_n$, onde

$$w_n = \Pi_n^m(x_n) \Leftrightarrow \pi_{n, \infty}^{\rightarrow}(w_n) \subseteq x_n \subseteq x.$$

Assim, a seqüência $\{x_n\}_{n \in \omega}$ determina um novo objeto $x \in \mathbb{D}_{\infty+m}$, dado por

$$x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\} \Leftrightarrow x_0 \subseteq x_1 \subseteq \dots \subseteq x_n \subseteq \dots \subseteq x \text{ e } \Pi_{\infty+m, n}^{\rightarrow}(x_n) = w_n.$$

A seguir, caracteriza-se o domínio $\mathbb{D}_{\infty+m}^{\rightarrow}$ como o limite inverso para \mathcal{K}^m .

Proposição 7.10 Considere a família de projeções $\Pi_{\infty+m, n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n$, apresentada na Definição 6.8. O par $\mathbb{D}_{\infty+m}^{\rightarrow} \equiv (\mathbb{D}_{\infty+m}^{\rightarrow}, \Pi_{\infty+m, n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n)$ constitui-se num cone limite para o diagrama \mathcal{K}^m apresentado na Definição 7.10.

7.2.4 Relacionamento entre $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{\infty+m+1}^{\rightarrow}$

Proposição 7.11 De acordo com a Projeção 7.13, cada m determina um cone $\mathbb{D}_{\infty+m}^{\rightarrow} \equiv (\mathbb{D}_{\infty+m}^{\rightarrow}, \Pi_{\infty+m, n}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_n)$. A função projeção, linear, $\Pi_{\mathbb{D}_{\infty+m+1}^{\rightarrow}, \mathbb{D}_{\infty+m}^{\rightarrow}} : \mathbb{D}_{\infty+m+1}^{\rightarrow} \rightarrow \mathbb{D}_{\infty+m}^{\rightarrow}$, definida pela expressão

$$\Pi_{\mathbb{D}_{\infty+m+1}^{\rightarrow}, \mathbb{D}_{\infty+m}^{\rightarrow}}(x) = \begin{cases} \{\mathbf{a}_\alpha \mid \mathbf{a}_{\alpha.00} \in x\} & \text{se } x \in \mathbb{P}_{\infty+m}^{\rightarrow}, \\ \{\mathbf{a}_\alpha \mid \mathbf{a}_{\alpha.001} \in x\}, & \text{se } x \in \mathbb{P}_{\infty+m}^{\rightarrow} \prod \mathbb{P}_{\infty+m}^{\rightarrow}, \\ \{\mathbf{a}_\alpha \mid \mathbf{a}_{\alpha.002} \in x\}, & \text{se } x \in \mathbb{P}_{\infty+m}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{P}_{\infty+m}^{\rightarrow}, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

é um epimorfismo em CospLin .

A Figura 7.5 ilustra a proposição 7.4, representando graficamente as projeções já definidas em seções anteriores e que tornam explícito o relacionamento entre os espaços coerentes $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{\infty+m+1}^{\rightarrow}$. Neste caso, tem-se

1. $\Pi_{\infty+m, n}^{\rightarrow} \circ \Pi_{\infty+m+1, \infty+m}^{\rightarrow} = \Pi_{\infty+m+1, n}^{\rightarrow} = \Pi_n^m \circ \Pi_{\infty+m+1, n+1}^{\rightarrow}$;
2. $\Pi_{\infty+m, n+1}^{\rightarrow} \circ \Pi_{\infty+m+1, \infty+m}^{\rightarrow} = \Pi_{\infty+m+1, n+1}^{\rightarrow}$;
3. $\Pi_n^m \circ \Pi_{\infty+m+1, n+1}^{\rightarrow} = \Pi_{\infty+m+1, n}^{\rightarrow}$; e ainda
4. $\Pi_{\infty+m, n}^{\rightarrow} = \Pi_n^m \circ \Pi_{\infty+m, n+1}^{\rightarrow}$.

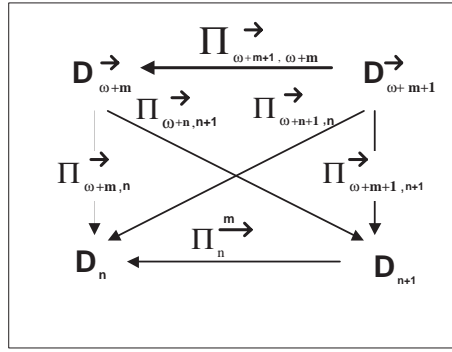


FIGURA 7.8 - Relacionamento entre $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{\infty+m+1}^{\rightarrow}$.

7.3 Espaço Coerente $\mathbb{D}_{2\infty}^{\rightarrow}$

O espaço coerente $\mathbb{D}_{2\infty}^{\rightarrow}$ dos processos computacionais transfinitos formaliza a completção para o processo de construção que a partir de $\mathbb{D}_{\infty}^{\rightarrow}$ constrói cada um dos níveis $\mathbb{D}_{\infty+m}^{\rightarrow} - \mathbb{D}_{\infty+m+1}^{\rightarrow}$ da estrutura do modelo de máquina geométrica distribuída para processos computacionais transfinitos.

Definição 7.11 *O espaço coerente $\mathbb{D}_{2\infty}^{\rightarrow}$ constitui-se no ponto fixo para a equação que define o espaço coerente dos processos finitos. Se $\mathbb{P}_{2\infty}^{\rightarrow} = \mathbb{D}_{2\infty}^{\rightarrow} \amalg \mathbb{D}_{2\infty}^{\rightarrow} \amalg \mathbb{D}_{2\infty}^{\perp}$,*

$$\mathbb{D}_{2\infty}^{\rightarrow} = \mathbb{P}_{2\infty}^{\rightarrow} \amalg (\mathbb{P}_{2\infty}^{\rightarrow} \amalg \mathbb{P}_{2\infty}^{\rightarrow}) \amalg (\mathbb{P}_{2\infty}^{\rightarrow} \amalg \prod_{\mathbb{B}} \mathbb{P}_{2\infty}^{\rightarrow}).$$

Apresenta-se primeiramente algumas representações gráficas de interpretações no domínio $\mathbb{D}_{2\infty}^{\rightarrow}$ e logo a seguir a definição dos morfismos que modelam os construtores e destrutores de processos.

Exemplificação 7.5

Na Figura 7.9 estão representados alguns exemplos de interpretações construídas em $\mathbb{D}_{2\infty}^{\rightarrow}$, conforme mostra os seguintes itens:

- $\{d_{:001:001}^{(k)}\} = \pi_{\infty, 2\infty}(\{d_{:001}^{(k)}\});$
- $\{d_{:001:001}^{(k)}, d_{:001:011:001}^{(k)}, d_{:011:001:011:001}^{(k)}\} = \pi_{\infty+1, 2\infty}(\{d_{:001:001}^{(k)}, d_{:001:011}^{(k)}, d_{:011:001:011}^{(k)}\});$
- $\{d_{:002:002}^{(k)}\} = \pi_{\infty, 2\infty}(\{d_{:002}^{(k)}\});$
- $\{d_{:002:002}^{(k)}, d_{:002:012:002}^{(k)}\} = \pi_{\infty+1, 2\infty}(\{d_{:002:002}^{(k)}, d_{:002:012}^{(k)}\});$

7.3.1 Imersões $\pi_{\infty+m, 2\infty}^{\rightarrow}$ e Projeções $\Pi_{2\infty, \infty+m}^{\rightarrow}$

Definição 7.12 *Considere os espaços coerentes $\mathbb{D}_{\infty+m}^{\rightarrow}$, $\mathbb{P}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{2\infty}^{\rightarrow}$. A função $\pi_{\infty+m, 2\infty}^{\rightarrow} : \mathbb{D}_{\infty+m}^{\rightarrow} \rightarrow \mathbb{D}_{2\infty}^{\rightarrow}$ é definida pela expressão*

$$\pi_{\infty+m, 2\infty}^{\rightarrow}(x) = \begin{cases} \{\mathbf{a}_{:00} \mid \mathbf{a} \in x\}, & \text{se } x \in \mathbb{P}_{\infty+m-1}^{\rightarrow}, \\ \{\mathbf{a}_{:001} \mid \mathbf{a} \in x\}, & \text{se } x \in (\mathbb{P}_{\infty+m-1}^{\rightarrow} \amalg \prod \mathbb{P}_{\infty+m-1}^{\rightarrow}), \\ \{\mathbf{a}_{:002} \mid \mathbf{a} \in x\}, & \text{se } x_i \in (\mathbb{P}_{\infty+m-1}^{\rightarrow} \amalg \prod_{\mathbb{B}} \mathbb{P}_{\infty+m-1}^{\rightarrow}). \end{cases}$$

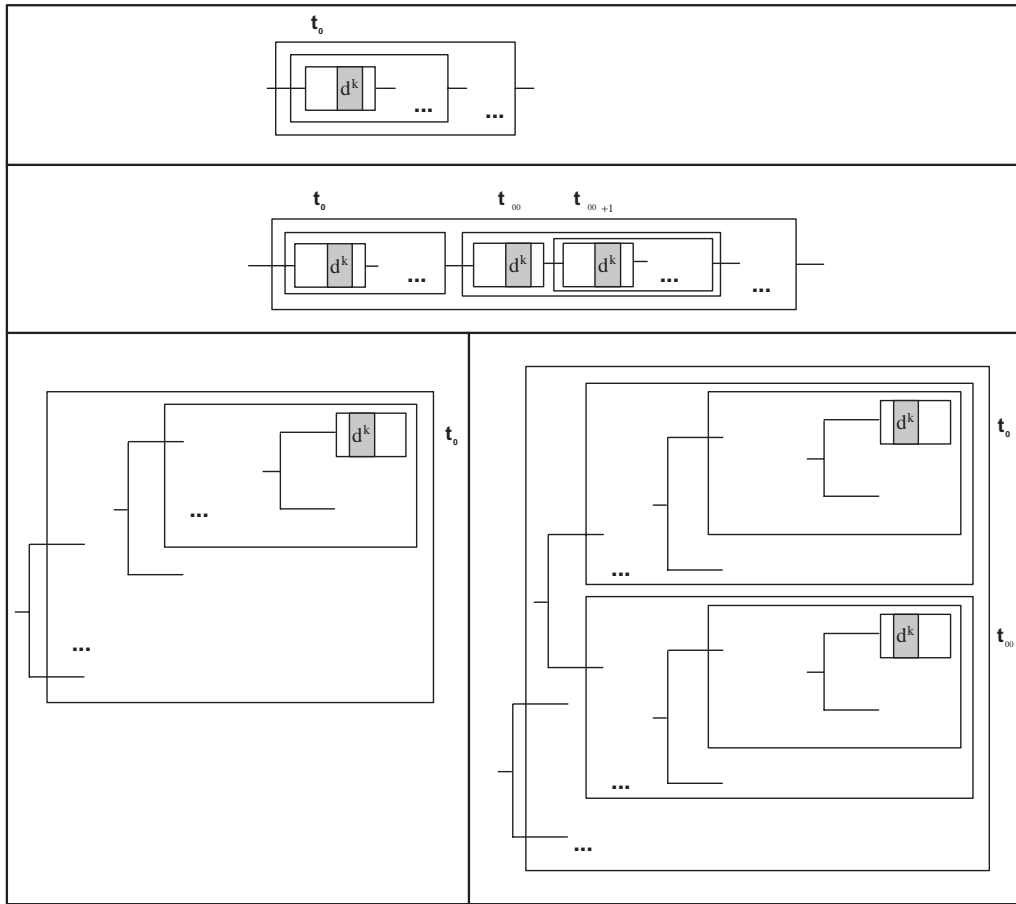


FIGURA 7.9 - Representações das Interpretações em $\pi_{\infty+m, 2\infty}^{\rightarrow}[\mathbb{D}_{\infty+m}^{\rightarrow}]$.

Definição 7.13 Considere os domínios $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_{2\infty}^{\rightarrow}$. Seja $x = \bigcup_{i \in \omega}^{\uparrow} \{x_i\} \in \mathbb{D}_{2\infty}$, onde x_i é uma aproximação finita de x . A função $\Pi_{2\infty, \infty+m}^{\rightarrow} : \mathbb{D}_{2\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty+m}$ é dada por

$$\Pi_{2\infty, \infty+m}^{\rightarrow}(x) = \bigcup_{i \in \omega}^{\uparrow} \Pi_{2\infty, \infty+m}^{\rightarrow}(x_i).$$

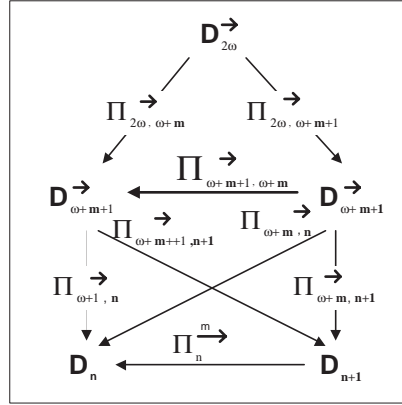
Neste caso,

$$\Pi_{2\infty, \infty+m}^{\rightarrow}(x_i) = \begin{cases} \{\mathbf{a}_{\alpha} \mid \mathbf{a}_{\alpha:00} \in x_i \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{\infty+m}\}, & \text{se } x_i \in \mathbb{P}_{2\infty}^{\rightarrow}, \\ \{\mathbf{a}_{\alpha} \mid \mathbf{a}_{\alpha:001} \in x_i \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{\infty+m}\}, & \text{se } x_i \in \mathbb{P}_{2\infty}^{\rightarrow} \prod \mathbb{P}_{2\infty}^{\rightarrow}, \\ \{\mathbf{a}_{\alpha} \mid \mathbf{a}_{\alpha:002} \in x_i \text{ e } \mathbf{a}_{\alpha} \in \mathbf{D}_{\infty+m}\}, & \text{se } x_i \in \mathbb{P}_{2\infty}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{P}_{2\infty}^{\rightarrow}. \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Proposição 7.12 As funções $\Pi_{2\infty, \infty+m}^{\rightarrow}$ e $\pi_{\infty+m, 2\infty}^{\rightarrow}$, apresentadas nas Definições 7.12 e 7.13 são morfismos na categoria CospLin .

A seguir, caracteriza-se o domínio $\mathbb{D}_{2\infty}^{\rightarrow}$ como o limite inverso para \mathcal{K}^{∞} . A Figura apresentada logo abaixo mostra os morfismos que formalizam o relacionamento entre os domínios $\mathbb{D}_{2\infty}^{\rightarrow}$, $\mathbb{D}_{\infty+m}^{\rightarrow}$ e $\mathbb{D}_n^{\rightarrow}$.

Proposição 7.13 Considere a família de projeções $\Pi_{2\infty, \infty+m}^{\rightarrow} : \mathbb{D}_{2\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty+m}$, apresentada na Definição 6.8. O par $\mathbb{D}_{2\infty}^{\rightarrow} \equiv (\mathbb{D}_{2\infty}^{\rightarrow}, \Pi_{2\infty, \infty+m}^{\rightarrow} : \mathbb{D}_{2\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty+m}^{\rightarrow})$ constitui-se num cone limite para o diagrama \mathcal{K}^{∞} apresentado na Definição 7.10.

FIGURA 7.10 - Projeções sobre $\mathbb{D}_{2\infty}^{\rightarrow}$.

Considerando-se a Proposição 7.13 e as projeções $\Pi_{(m+1)\infty, \infty+m}^{\rightarrow}$, apresentadas na seção 7.2.4, além das composições já estabelecidas, as seguintes condições também são satisfeitas:

1. $\Pi_{\infty+m+1, \infty+m}^{\rightarrow} \circ \Pi_{2\infty, \infty+m+1}^{\rightarrow} = \Pi_{2\infty, \infty+m}^{\rightarrow}$;
2. $\Pi_{\infty+m+1, n}^{\rightarrow} \circ \Pi_{2\infty, \infty+m+1}^{\rightarrow} = \Pi_{\infty+m, n}^{\rightarrow} \circ \Pi_{2\infty, \infty+m}^{\rightarrow}$;
3. $\Pi_n^m \circ \Pi_{\infty+m, n+1}^{\rightarrow} \circ \Pi_{2\infty, \infty+m}^{\rightarrow} = \Pi_n^m \circ \Pi_{\infty+m+1, n+1}^{\rightarrow} \circ \Pi_{2\infty, \infty+m+1}^{\rightarrow}$.

Exemplificação 7.6

A Figura 7.11 ilustra as proposições acima e apresenta novas interpretações nos domínios \mathbb{D}_n , $\mathbb{D}_{\infty+m}$ e $\mathbb{D}_{2\infty}$.

Seja $v = \{d_{:001:001}^{(k_0)}, d_{:001:011:001}^{(k_1)}, d_{:001:001:011:001}^{(k_2)}, d_{:001:(011)^2:001}^{(k_2)}, \dots, d_{:001:(011)^n:001}^{(k_{2^n-1})}\} \in \pi_{2\infty, \infty}^{\rightarrow}[\mathbb{D}_n]$, tem-se então

- Projeções sobre os subespaços $\mathbb{D}_{\infty+m}$ com $m \leq n$:

$$\Pi_{2\infty, \infty}(v) = \{d_{:001}^{(k_0)}\};$$

$$\Pi_{2\infty, \infty+1}(v) = \{d_{:001:001}^{(k_0)}, d_{:001:011}^{(k_1)}\};$$

$$\Pi_{2\infty, \infty+2}(v) = \{d_{:001:(001)^2}^{(k_0)}, d_{:001:011:001}^{(k_1)}, d_{:001:001:011}^{(k_2)}, d_{:001:(011)^2}^{(k_3)}\};$$

⋮

$$\Pi_{2\infty, \infty+m}(v) = \{d_{:001:(001)^m}^{(k_0)}, d_{:001:011.(001)^{m-1}}^{(k_1)}, d_{:001:001.011.(001)^{m-2}}, \dots, d_{:001:(011)^m}^{(k_{2^m-1})}\};$$

⋮

$$\Pi_{2\infty, \infty+n}(v) = \{d_{:001:(001)^n}^{(k_0)}, d_{:001:011.(001)^{n-1}}^{(k_1)}, d_{:001:001.011.(001)^{n-2}}, \dots, d_{:001:(011)^n}^{(k_{2^n-1})}\};$$

- Projeções sobre os subespaços \mathbb{D}_n .

$$\Pi_{2\infty, 0}(v) = \{d_{:001}^{(k_0)}\};$$

$$\Pi_{2\infty, 1}(v) = \{d_{:001}^{(k_0)}, d_{:011}^{(k_1)}\};$$

$$\begin{aligned} \Pi_{2\infty,2} \circ \Pi_{\infty^2,2}(v) &= \{d_{(001)^2}^{(k_0)}, d_{011.(001)}^{(k_1)}, d_{001.011}^{(k_2)}, d_{(011)^2}^{(k_3)}\}; \\ &\vdots \\ \Pi_{2\infty,m}(v) &= \{d_{(001)^m}^{(k_0)}, d_{011.(001)^{m-1}}^{(k_1)}, d_{001.011.(001)^{m-2}}^{(k_2)}, \\ &d_{(011)^2.(001)^{m-2}}, \dots, d_{(011)^m}^{(k_{2m-1})}\}; \\ &\vdots \\ \Pi_{2\infty,n} \circ \Pi_{2\infty,n}(v) &= \{d_{(001)^n}^{(k_0)}, d_{011.(001)^{n-1}}^{(k_1)}, d_{001.011.(001)^{n-2}}^{(k_2)}, \\ &d_{(011)^2.(001)^{n-2}}, \dots, d_{(011)^n}^{(k_{2n-1})}\}; \end{aligned}$$

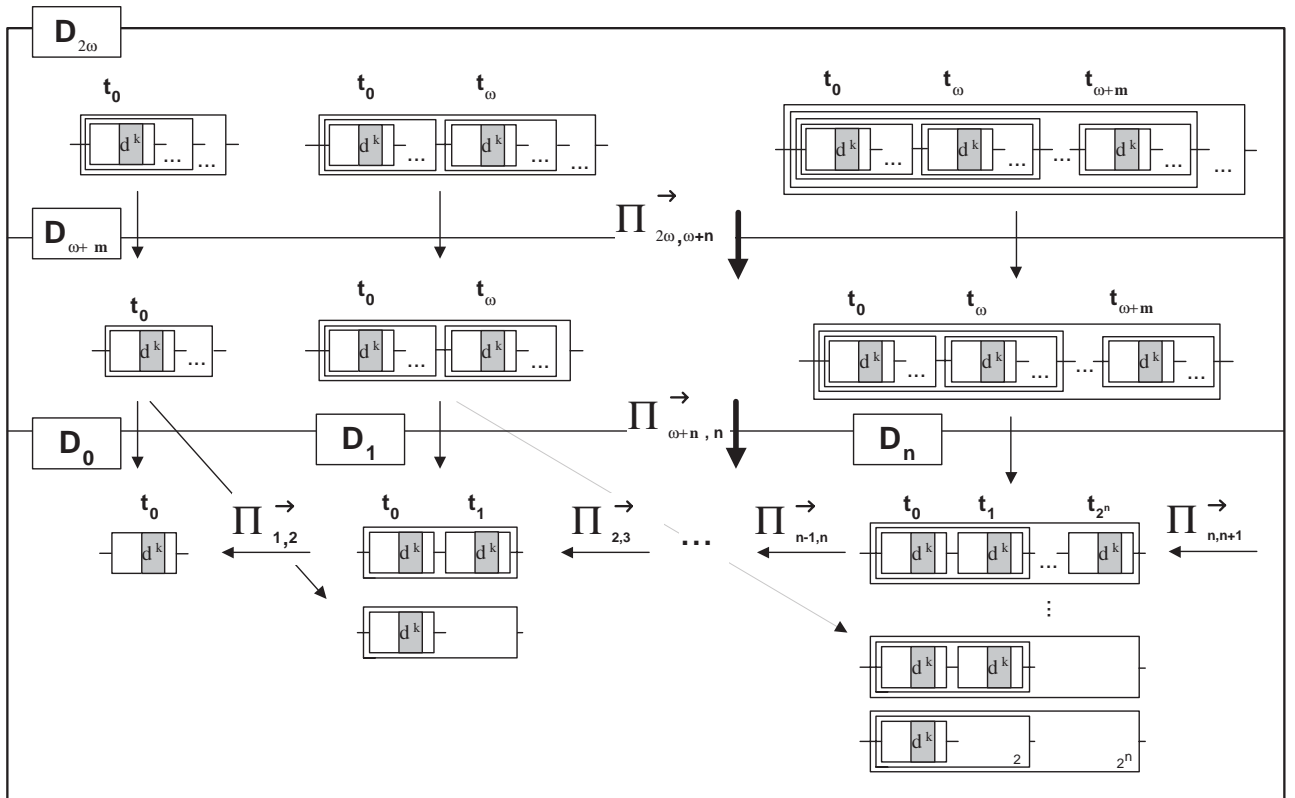


FIGURA 7.11 - Aplicações da Imersão $\pi_{\infty+m, 2\infty}^{\rightarrow}$ sobre Objetos em $\mathbb{D}_{\infty+m}^{\rightarrow}$.

Mostrou-se ao longo deste capítulo a possibilidade de se construir uma máquina geométrica distribuída, em sua concepção mais simples, a partir do modelo de uma máquina geométrica. Neste sentido, pode-se pensar que uma análise mais generalizada possa interpretar modelos mais detalhados de máquinas distribuídas.

8 Interpretação dos Construtores de Processos no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$ dos Processos Computacionais

Neste capítulo mostra-se que a interpretação de cada um dos construtores de processos é formalizada por funções lineares - morfismos na categoria $CospLin$. Além das definições com as correspondentes restrições e observações, verifica-se também que o modelo contempla interpretação para construtores recursivos, a partir da composição finita ou infinita destes morfismos.

Nas definições apresentadas neste capítulo, consideram-se as imersões $\pi_n^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ e $\pi_{n,\infty}^{\rightarrow} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ e as projeções $\Pi_n^{\rightarrow} : \mathbb{D}_{n+1} \rightarrow \mathbb{D}_n$ e $\Pi_{\infty,n}^{\rightarrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_n$, apresentadas no Capítulo 6 na Seção 6.3.

Além disso, supõe-se que $\theta, \theta_1, \theta_2 \in \{(00), (001), (011), (002), (012)\}$ correspondem a indexadores que definem as projeções e imersão, e identificam destrutores e construtores de processos.

8.1 Construtores de Processos Modelados pela Construção Espacial do Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$.

Todos os construtores de processos modelados pela construção espacial do espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$ podem ser definidos de forma análoga em $\mathbb{D}_{\infty}^{\leftarrow}$. Dependendo da aridade e definição de cada um dos operadores, consideram-se as seguintes construções sobre o espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$ de processos : $\tilde{\mathbb{D}}_{\infty}^{\rightarrow}$, $\tilde{\mathbb{D}}_{\infty}^{\perp \rightarrow}$, $\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{B} \amalg \mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow}$.

8.1.1 Operador Identidade $Id_{\mathbb{D}_{\infty}^{\rightarrow}}$

Definição 8.1 *Seja o espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$. O endomorfismo (ou função linear) $Id : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ que interpreta o construtor $Id_{\mathcal{P}}$ é definido por*

$$x \mapsto Id_{\mathbb{D}_{\infty}^{\rightarrow}}(x) = x.$$

8.1.2 Operador $F_{(00)}$

O operador $F_{(00)}$ formaliza a noção intuitiva de que um processo executado em n *utc* pode aproximar qualquer outro processo, desde que o tempo de execução deste outro seja maior ou no mínimo igual a n *utc*.

Definição 8.2 *Considere $x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\} \in \mathbb{D}_{\infty}^{\rightarrow}$ onde $x_n \in \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$. Considere também as imersões $\pi_{n,\infty}^{(00)} : \mathbb{D}_n \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ definidas na Seção 6.2. A função $F_{(00)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ que interpreta o operador **inclusão** é definida por*

$$F_{(00)}(x) = \bigcup_{n \in \omega}^{\uparrow} \{(\pi_{n+1,\infty}^{\rightarrow} \circ \pi_n^{(00)} \circ \Pi_{\infty,n}^{\rightarrow})(x_n)\}.$$

Em particular, $F_{(00)}(\emptyset) = \emptyset$.

A seguir, apresentam-se algumas exemplificações.

1. $F_{00}(\{d_{00}^{(k)}\}) = \{d_{00:00}^{(k)}\} = \{d_{00.00.00:\infty}^{(k)}\} = \dots$ representa um processo elementar em $\mathbb{D}_{\infty}^{\rightarrow}$ que, no instante t_0 , executa a ação d na k -ésima posição.

2. Em $x = \{\overline{d_{001}^{(l)} e_{011}^{(l)} 10:00}, \overline{d_{00}^{(k)} 10:00}\}$ tem-se que o subconjunto $x' = \{\overline{d_{00}^{(k)} 10:00}\}$ representa um processo computacional que, no instante 0, executa a ação d na k -ésima posição. Da mesma forma, o subconjunto coerente $x'' = \{\overline{d_{001}^{(l)} e_{011}^{(l)} 10:\infty}\}$ interpreta um processo seqüencial executado nos instantes t_0 e t_1 . Existe neste caso, uma concorrência explícita no instante t_0 .
3. Em $y = \{d_{001.001:00}^{(n)}, e_{011.001:00}^{(l)}, d_{00.011:00}^{(k)}\}$ tem-se que o subconjunto unitário $y' = \{d_{00.011:00}^{(k)}\}$ representa um processo parcial que executa a ação d na k -ésima posição, no instante t_2 . Neste exemplo, o subconjunto coerente $y'' = \{d_{001.001:\infty}^{(n)}, e_{011.001:\infty}^{(l)}\}$ interpreta a parte do processo interpretado por y que será executada seqüencialmente nos instantes t_0 e t_1 .
4. Seja $F_{00}(\{d_{001}^{(k)}, d_{011:001}^{(k)}\}) = \{d_{001.00:001}^{(k)}, d_{011.00:001}^{(k)}\}$. Neste caso, este conjunto coerente interpreta um processo seqüencial parcial, cujos dois primeiros fatores executam a ação d na k -ésima posição. Os demais (infinitos) fatores ainda não estão identificados.
5. Seja $w = \{b_{22:002}, d_{201:002}^{(k)}, e_{201:002}^{(k)}, d_{111:002}^{(k)}, d_{111:002}^{(l)}, d_{00.012:002}^{(k)}\}$. Tem-se então que w interpreta um processo parcial que resulta de um seqüência infinita de aplicações do operador soma determinística, portanto incluindo infinitos testes. Desta seqüência, apenas a escolha referente ao teste interpretado pelo conjunto coerente $\{b_{22:002}\}$ está definida. Se este teste e todos os demais testes ainda não conhecidos forem satisfeitos, executa-se o primeiro termo, definido pelo subconjunto

$$w' = \{d_{201:002}^{(k)}, e_{201:002}^{(k)}, d_{111:002}^{(k)}, d_{111:002}^{(l)}\},$$

interpretando um produto seqüencial que executa primeiro um produto paralelo e logo após uma soma não-determinística. Caso contrário, a segunda parcela interpretada por w'' será executada. Aqui, o processo elementar que executa a ação d na posição k de memória constitui-se na interpretação associada ao subconjunto coerente $w'' = \mathbf{F}_{(00)}(\{d_{00:012}^{(k)}\})$.

A próxima proposição é uma conseqüência da Definição 8.2.

Proposição 8.1 *Considere $x = \bigcup_{n \in \omega}^\uparrow \{x_n\} \in \mathbb{D}_\infty^\rightarrow$. Verifica-se a equivalência*

$$\mathbf{F}_{(00)}(x) = x \Leftrightarrow x \in \pi_{n,\infty}^{(00)}[\mathbb{D}_n].$$

Nas próximas definições utiliza-se a notação $x \sqcap y = (\{0\} \times x) \cup (\{1\} \times y)$, para indicar o conjunto coerente resultante do produto direto de x por y , onde $x, y \in \mathbb{D}_\infty^\rightarrow$. Assim, $\mathbb{D}_\infty^\rightarrow \amalg \mathbb{D}_\infty^\rightarrow$ define o espaço coerente resultante do produto direto (produto categórico) em $\mathbb{D}_\infty^\rightarrow$. Para a definição de produto categórico de espaços coerentes, veja [GIR 89].

8.1.3 Operador $\mathbf{F}_{(10)}$

O operador $\mathbf{F}_{(10)}$ interpreta o produto paralelo de processos que possuem construção temporal finita. A extensão deste operador para os demais processos só poderá ser formalizada nos domínios que constroem o espaço coerente $\mathbb{D}_{2\infty}$. A definição formal é dada logo a seguir.

Definição 8.3 *Considere $x, y \in \mathbb{D}_\infty^\rightarrow$ tais que $x = \bigcup^\uparrow \{x_n\}_{n \in \omega}$ e $y = \bigcup^\uparrow \{y_m\}_{m \in \omega}$, onde $x_n \in \pi_{n,\infty}^\rightarrow(\mathbb{D}_n)$ e $y_m \in \pi_{m,\infty}^\rightarrow(\mathbb{D}_m)$. Sempre que as próximas condições são satisfeitas*

- $x_n \in \pi_{n,\infty}^{\rightarrow} \circ \psi_{n-1}^{(0)} \circ \phi_{n-1}^0(\mathbb{D}_{n-1})$ e $y_m \in \pi_{m,\infty}^{\rightarrow} \circ \psi_{m-1}^{(0)} \circ \phi_{m-1}^0(\mathbb{D}_{m-1})$,
- $\Upsilon_{\infty}(x) \cap \Upsilon_{\infty}(y) = \emptyset$,
- $\Pi_{\infty,k}^{\rightarrow}(x), \Pi_{\infty,k}^{\rightarrow}(y) \in \gamma_{k-1}^0 \circ \phi_{k-1}^1[\mathbb{D}_{k-1}]$ onde $k = \max\{m, n\}$,

a função $\mathbf{F}_{(10)} : \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ que interpreta o **produto paralelo** de processos é definida pela expressão

$$\mathbf{F}_{(10)}(x \sqcap y) = \bigcup_{n,m \in \omega}^{\uparrow} \pi_{k,\infty}^{\rightarrow} [\Pi_{\infty,k}^{\rightarrow}(x_n) \cup \Pi_{\infty,k}^{\rightarrow}(y_m)].$$

Quando algumas das condições acima não for satisfeita, tem-se $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset$.

As próximas observações facilitam a compreensão da Definição 8.3.

Observação 8.1

(1) O processo resultante após aplicação do construtor interpretado pelo operador $\mathbf{F}_{(00)}$ é indefinido, sendo portanto interpretado pelo conjunto coerente $\emptyset \in \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$, nas seguintes situações:

1. as correspondentes aproximações de x e y são tais que $x_n \in \pi_{n,\infty}^{\theta_1}[\mathbb{D}_n]$, $y_m \in \pi_{m,\infty}^{\theta_2}[\mathbb{D}_m]$, mas $\theta_1 \neq 00$ e $\theta_2 \neq 00$;
2. os processos x e y não são concorrentes, significando que $\Upsilon_{\infty}(x) \cap \Upsilon_{\infty}(y) \neq \emptyset$;
3. $\Pi_{\infty,k}(x_n)$ (ou $\Pi_{\infty,k}(y_m)$) $\notin \gamma_{k-1}^0 \circ \phi_{k-1}^1[\bar{\mathbb{D}}_{k-1}]$;
4. $x = y = \emptyset$.

(2) Se $\theta = \theta.\theta.\dots$ pela Definição 8.3, tem-se a seguinte igualdade:

$$\pi_{n,\infty}^{(\theta)} [\Pi_{\infty,k}^{\rightarrow}(x_n) \cup \Pi_{\infty,k}^{\rightarrow}(y_m)] = \{a:\theta \mid a \in \Pi_{\infty,k}^{\rightarrow}(x_n) \cup \Pi_{\infty,k}^{\rightarrow}(y_m)\}$$

(3) Quando um dos fatores é uma aproximação finita do outro, no caso $y = x_n \subseteq x$, tem-se que $\mathbf{F}_{(10)}(x \sqcap y) = \mathbf{F}_{(10)}(x \sqcap x_n) = x$.

Proposição 8.2 Considere $x, y, z \in \mathbb{D}_{\infty}^{\rightarrow}$. Verificam-se as propriedades:

- P_1 $\mathbf{F}_{(10)}(x \sqcap y) = \mathbf{F}_{(10)}(y \sqcap x)$.
- P_2 $\mathbf{F}_{(10)}(\mathbf{F}_{(10)}(x \sqcap y) \sqcap z) = \mathbf{F}_{(10)}(x \sqcap (\mathbf{F}_{(10)}(y \sqcap z)))$.
- P_3 $\mathbf{F}_{(10)}(x \sqcap \emptyset) = \mathbf{F}_{(10)}(\emptyset \sqcap x) = x$.

Além das últimas observações e propriedades, nos próximos parágrafos apresenta-se algumas aplicações do construtor $\mathbf{F}_{(10)}$ sobre subconjuntos coerentes que interpretam processos de execução finita.

Exemplificação 8.1

Considera-se I enumerável e o indexador $\cdot : \infty = 00.00.00.\dots$

1. $\mathbf{F}_{(10)}(\{\overline{d^{(0)}}_{10:\infty}\} \sqcap \{\emptyset_{10:\infty}\}) = \{\overline{d^{(0)}}_{10:\infty}, \emptyset_{10:\infty}\}$
2. $\mathbf{F}_{(10)}(\{\overline{d^{(0)}}_{10:\infty}\} \sqcap \emptyset) = \{\overline{d^{(0)}}_{10:\infty}\}$

3. $\mathbf{F}_{(10)}(\{\overline{d^{(1)}}_{10:\infty}\} \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(0)}}_{10:\infty}\} \sqcap \emptyset)) = \{\overline{d^{(1)}}_{10:\infty}, \overline{d^{(0)}}_{10:\infty}\}$
4. $\mathbf{F}_{(10)}(\{\overline{d^{(2)}}_{10:\infty}\} \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(1)}}_{10:\infty}\} \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(0)}}_{10:\infty}\} \sqcap \emptyset))) = \{\overline{d^{(2)}}_{10:\infty}, \overline{d^{(1)}}_{10:\infty}, \overline{d^{(0)}}_{10:\infty}\}$
5. $\mathbf{F}_{(10)}(\{\overline{d^{(n)}}_{10:\infty}\} \sqcap \dots \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(2)}}_{10:\infty}\} \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(1)}}_{10:\infty}\} \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(0)}}_{10:\infty}\} \sqcap \emptyset)))) = \{\overline{d^{(n)}}_{10:\infty}, \dots, \overline{d^{(2)}}_{10:\infty}, \overline{d^{(1)}}_{10:\infty}, \overline{d^{(0)}}_{10:\infty}\}$
6. $\mathbf{F}_{(10)}(\{\dots, \overline{d^{(n+1)}}_{10:\infty}, \overline{d^{(n)}}_{10:\infty}\} \sqcap \{\overline{d^{(n-1)}}_{10:\infty}, \dots, \overline{d^{(1)}}_{10:\infty}, \overline{d^{(0)}}_{10:\infty}\}) = \{\dots, \overline{d^{(1)}}_{10:\infty}, \overline{d^{(0)}}_{10:\infty}\}$
7. $\mathbf{F}_{(10)}(\{\overline{d^{(k)}}_{201:\infty}\} \sqcap \{\overline{d^{(n)}}_{10:\infty}, \emptyset_{10:\infty}\}) = \emptyset$
8. $\mathbf{F}_{(10)}(\{\overline{d^{(k)}}_{201} e^{(k)}_{211.10:\infty}\} \sqcap \{\overline{d^{(n)}}_{10:\infty}\}) = \emptyset$
9. $\mathbf{F}_{(10)}(\{\overline{d^{(k)}}_{201} e^{(k)}_{211.10:\infty}\} \sqcap \{\overline{d^{(n)}}_{10.10:\infty}\}) = \{\overline{d^{(k)}}_{201} e^{(k)}_{211.10:\infty}, \overline{d^{(n)}}_{10.10:\infty}\}$
10. $\mathbf{F}_{(10)}(\{\overline{d^{(k_{2i})}}_{10:\infty}\}_{i \in \omega} \sqcap \{\overline{d^{(k_{2i+1})}}_{10:\infty}\}_{i \in \omega}) = \{\overline{d^{(k_{2i})}}_{10:\infty}, \overline{d^{(k_{2i+1})}}_{10:\infty}\}_{i \in \omega}$

A seguir define-se o morfismo que interpreta a soma não-determinística.

8.1.4 Operador $\mathbf{F}_{(20)}$

Definição 8.4 Considere $x, y \in \mathbb{D}_{\infty}^{\rightarrow}$ tais que $x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\}$ e $y = \bigcup_{m \in \omega}^{\uparrow} \{y_m\}$, onde $x_n \in \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ e $y_m \in \pi_{m,\infty}^{\rightarrow}[\mathbb{D}_m]$.

Sempre que as seguintes condições são satisfeitas

- $x_n \in (\pi_{n,\infty}^{\rightarrow} \circ \psi_{n-1}^{(0)} \circ \phi_{n-1}^{(0)})[\mathbb{D}_{n-1}]$ e $y_m \in (\pi_{m,\infty}^{\rightarrow} \circ \psi_{m-1}^{(0)} \circ \phi_{m-1}^{(0)})[\mathbb{D}_{m-1}]$,
- $\Upsilon_{\infty}(x) \cap \Upsilon_{\infty}(y) \neq \emptyset$ (veja Definição 6.2),
- $\Pi_{\infty,k}^{\rightarrow}(x_n), \Pi_{\infty,k}^{\rightarrow}(y_m) \in (\gamma_{k-1}^0 \circ \phi_{k-2}^2)[\overline{\mathbb{D}}_{k-1}^{\perp}]$ onde $k = \max\{m, n\}$,

a função $\mathbf{F}_{(20)} : \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ que interpreta o **soma não-determinística** de processos é definida pela expressão

$$\mathbf{F}_{(20)}(x \sqcap y) = \bigcup_{n,m \in \omega}^{\uparrow} \{\pi_{k,\infty}^{\rightarrow}(\Pi_{\infty,k}^{\rightarrow}(x_n) \cup \Pi_{\infty,k}^{\rightarrow}(y_m))\}.$$

Quando algumas das condições acima não for satisfeita, $\mathbf{F}_{(20)}(x \sqcap y) = \emptyset$.

As observações abaixo facilitam a compreensão da Definição 8.4.

Observação 8.2

(1) O processo resultante é indefinido, e neste caso interpretado pelo conjunto coerente $\emptyset \in \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$, nas seguintes situações:

1. as correspondentes aproximações de x e y são tais que $x_n \in \pi_{n,\infty}^{(\theta_1)}[\mathbb{D}_n]$, $y_m \in \pi_{m,\infty}^{(\theta_2)}[\mathbb{D}_m]$, mas $\theta_1 \neq 00$ e $\theta_2 \neq 00$;
2. os processos x e y não são conflitantes (mas concorrentes), significando que $\Upsilon_{\infty}(x) \cap \Upsilon_{\infty}(y) = \emptyset$;
3. $\Pi_{\infty,k}^{\rightarrow}(x_n)$ (ou $\Pi_{\infty,k}^{\rightarrow}(y_m)$) $\notin \gamma_{k-1}^0 \circ \phi_{k-1}^2[\overline{\mathbb{D}}_{k-1}^{\perp}]$;

4. $x = y = \emptyset$.

(2) Quanto todas as condições da Definição 8.4 são satisfeitas e tem-se ainda que $x = \bigcup_{n \in \omega}^\uparrow \{x_n\} = x_n$ e $y = \bigcup_{m \in \omega}^\uparrow \{y_m\} = y_m$ então a **soma não-determinística** dos processos interpretados por x e y é definido pela expressão

$$\mathbf{F}_{(20)}(x \sqcap y) = \mathbf{F}_{(20)}(x_n \sqcap y_m) = \pi_{k,\infty}^\theta (\Pi_{\infty,k}(x_n) \cup \Pi_{\infty,k}(y_m)).$$

(3) Quando uma das parcelas da soma não-determinística é uma aproximação finita da outra, no caso $y = x_n \subseteq x$, tem-se que $\mathbf{F}_{(20)}(x \sqcap y) = \mathbf{F}_{(20)}(x \sqcap x_n) = x$.

A próxima proposição é uma consequência da Definição 8.4.

Proposição 8.3 *Considere $x, y, z \in \mathbb{D}_{\infty}^{\rightarrow}$. Verificam-se as propriedades:*

$$P_1 \quad \mathbf{F}_{(20)}(x \sqcap y) = \mathbf{F}_{(20)}(y \sqcap x).$$

$$P_2 \quad \mathbf{F}_{(20)}(\mathbf{F}_{(20)}(x \sqcap y) \sqcap z) = \mathbf{F}_{(20)}(x \sqcap (\mathbf{F}_{(20)}(y \sqcap z))).$$

$$P_3 \quad \mathbf{F}_{(20)}(x \sqcap \emptyset) = \mathbf{F}_{(20)}(\emptyset \sqcap x) = x.$$

Nos próximos parágrafos apresenta-se a aplicação do operador $\mathbf{F}_{(20)}$ sobre alguns conjuntos coerentes que interpretam processos executados em tempo finito.

Exemplificação 8.2

Considera-se I enumerável e $\infty = 00.00.00. \dots$

1. $\mathbf{F}_{(20)}(\{\overline{d^{(0)}}_{20:\infty}\} \sqcap \emptyset_{20:\infty}) = \emptyset$
2. $\mathbf{F}_{(20)}(\{\overline{d^{(0)}}_{20:\infty}\} \sqcap \emptyset) = \{\overline{d^{(0)}}_{20:\infty}\}$
3. $\mathbf{F}_{(20)}(\{\overline{e^{(0)}}_{20:\infty}\} \sqcap \mathbf{F}_{(20)}(\{\overline{d^{(0)}}_{20:\infty}\} \sqcap \emptyset)) = \{\overline{e^{(0)}}_{20:\infty}, \overline{d^{(0)}}_{20:\infty}\}$
4. $\mathbf{F}_{(20)}(\{\overline{f^{(0)}}_{20:\infty}\} \sqcap \mathbf{F}_{(20)}(\{\overline{e^{(0)}}_{20:\infty}\} \sqcap \mathbf{F}_{(20)}(\{\overline{d^{(0)}}_{20:\infty}\} \sqcap \emptyset))) = \{\overline{f^{(0)}}_{20:\infty}, \overline{e^{(0)}}_{20:\infty}, \overline{d^{(0)}}_{20:\infty}\}$
5. $\mathbf{F}_{(20)}(\{\overline{p^{(0)}}_{20:\infty}\} \sqcap \dots$
 $\dots \sqcap \mathbf{F}_{(20)}(\{\overline{f^{(0)}}_{20:\infty}\} \sqcap \mathbf{F}_{(20)}(\{\overline{e^{(0)}}_{20:\infty}\} \sqcap \mathbf{F}_{(20)}(\{\overline{d^{(0)}}_{10:\infty}\} \sqcap \emptyset)))) = \{\overline{p^{(0)}}_{20:\infty}, \dots, \overline{f^{(0)}}_{20:\infty}, \overline{e^{(0)}}_{20:\infty}, \overline{d^{(0)}}_{20:\infty}\}$
6. $\mathbf{F}_{(20)}(\{\dots, \overline{d_{n+1}^{(0)}}_{20:\infty}, \overline{d_n^{(0)}}_{20:\infty}\}_{n \in \omega} \sqcap \{\overline{d_{n-1}^{(0)}}_{20:\infty}\}_{n \in \omega}) = \{\dots, \overline{d_{n+1}^{(0)}}_{20:\infty}, \overline{d_n^{(0)}}_{20:\infty}, \overline{d_{n-1}^{(0)}}_{20:\infty}\}_{n \in \omega}$
7. $\mathbf{F}_{(20)}(\{\overline{d^{(k)}}_{20:\infty}, \overline{e^{(l)}}_{20:\infty}\} \sqcap \{\overline{d^{(n)}}_{:\infty}\}) = \emptyset$
8. $\mathbf{F}_{(20)}(\{\overline{d_{001}^{(k)}} e_{011.20:\infty}^{(l)}\} \sqcap \{\overline{d^{(k)}}_{10} e_{1020:\infty}^{(k)}\}) = \{\overline{d_{001}^{(k)}} e_{011.20:\infty}^{(l)}, \overline{d^{(k)}}_{10} e_{1020:\infty}^{(k)}\}$
9. $\mathbf{F}_{(20)}(\{\overline{d_{001}^{(k)}} e_{011.20:\infty}^{(k)}\} \sqcap \{\overline{d_{002}^{(k)}} e_{012.20:\infty}^{(k)}\}) = \{\overline{d_{001}^{(k)}} e_{011.20:\infty}^{(k)}, \overline{d_{002}^{(k)}} e_{012.20:\infty}^{(k)}\}$
10. $\mathbf{F}_{(20)}(\{\overline{d_i^{(k)}}_{20:\infty}\}_{i \in I \subseteq \omega} \sqcap \{\overline{e_j^{(k)}}_{20:\infty}\}_{j \in J \subseteq \omega}) = \{\overline{d_i^{(k)}}_{20:\infty}, \overline{e_j^{(k)}}_{20:\infty}\}_{i \in I, j \in J}$

A proposição apresentada a seguir assegura que a interpretação dos construtores de processos modelados acima são funções lineares.

Proposição 8.4 *Os construtores identidade $Id_{\mathbb{D}_{\infty}^{\rightarrow}}$, inclusão F_{00} , produto paralelo $\mathbf{F}_{(10)}$, soma não-determinística F_{20} são morfismos na categoria $\mathbf{CospLin}$.*

Prova. Mostra-se que $\mathbf{F}_{(10)} \in \mathbf{Mor}_{\mathbf{CospLin}}$. A prova pode ser construída de forma análoga para os demais construtores.

1. Mostra-se primeiramente que $\mathbf{F}_{(10)}$ é uma função monótona.

Sejam $x, y \in \mathbb{D}_{\infty}^{\rightarrow}$ tais que $x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\}$ e $y = \bigcup_{m \in \omega}^{\uparrow} \{y_m\}$ sempre que $x_n \in \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ e $y_m \in \pi_{m,\infty}^{\rightarrow}[\mathbb{D}_m]$.

Suponha também que $x \sqcap y, x' \sqcap y' \in \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$ e $x \sqcap y \subseteq x' \sqcap y'$. Neste caso, $x \subseteq x'$ e $y \subseteq y'$.

Além disso, $\Pi_{\infty,k}^{\rightarrow}, \pi_{k+1,\infty}^{\rightarrow}$ e $\pi_{k+1,\infty}^{(00)}$ são funções monótonas.

- 1.1 Primeiramente, considere satisfeitas as condições da Definição 8.3:

- $x_n \in \pi_{n,\infty}^{(00)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ e $y_m \in \pi_{m,\infty}^{(00)}[\mathbb{D}_m] \subseteq \pi_{m,\infty}^{\rightarrow}[\mathbb{D}_m]$.
- $\Upsilon_{\infty}(x) \cap \Upsilon_{\infty}(y) = \emptyset$ como também $\Upsilon_{\infty}(x') \cap \Upsilon_{\infty}(y') = \emptyset$. Logo x e y são concorrentes da mesma forma que x' e y' .
- $\Pi_{\infty,k}^{\rightarrow}(x), \Pi_{\infty,k}^{\rightarrow}(y) \in \gamma_{k-1}^0 \circ \phi_{k-1}^1[\mathbb{D}_{k-1}]$, se $k = \max\{m, n\}$. Logo pela monotonicidade das funções $\Pi_{\infty,k}^{\rightarrow}$ e $\pi_{k+1,\infty}^{(00)}$, tem-se $\Pi_{\infty,k}^{\rightarrow}(x'), \Pi_{\infty,k}^{\rightarrow}(y') \in \gamma_{k-1}^0 \circ \phi_{k-1}^1[\mathbb{D}_{k-1}]$. Portanto, conclui-se que

$$\begin{aligned} \mathbf{F}_{(10)}(x \sqcap y) &= \pi_{k+1,\infty}^{\rightarrow} [\Pi_{\infty,k}^{\rightarrow}(x) \cup \Pi_{\infty,k}^{\rightarrow}(y)] \\ &\subseteq \pi_{k+1,\infty}^{\rightarrow} [\Pi_{\infty,k}^{\rightarrow}(x') \cup \Pi_{\infty,k}^{\rightarrow}(y')] = \mathbf{F}_{(10)}(x' \sqcap y') \end{aligned}$$

- 1.2 Caso contrário, quando alguma das condições da Definição 8.3 acima não for satisfeita, $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset$. Nesta situação analisa-se os seguintes casos:

1.2.1 Ou $x \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n)$, $y \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m)$, mas $\theta_1 \neq 00$ e $\theta_2 \neq 00$. Assim, o mesmo vale para x' e y' , ou seja, $x' \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n)$, $y' \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m)$, mas $\theta_1 \neq 00$ e $\theta_2 \neq 00$. Logo tem-se que $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset = \mathbf{F}_{(10)}(x' \sqcap y')$.

1.2.2 Ou quando x e y não são concorrentes, e conseqüentemente, pela inclusão, tem-se que x' e y' também não. Logo $\mathbf{F}_{(10)}(x \sqcap y) = \mathbf{F}_{(10)}(x' \sqcap y') = \emptyset$.

1.2.3 Ou ainda quando $\Pi_{\infty,k}^{\rightarrow}(x) \notin \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$. Neste caso, $\Pi_{\infty,k}^{\rightarrow}(x') \notin \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$. Logo $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset = \mathbf{F}_{(10)}(x' \sqcap y')$. O mesmo pode ser concluído ao se considerar $\Pi_{\infty,k}^{\rightarrow}(y) \notin \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$.

1.2.4 Ou finalmente quando $x = x' = y = y' = \emptyset$, é imediato que $\mathbf{F}_{(10)}(x \sqcap y) = \mathbf{F}_{(10)}(x \sqcap y) = \emptyset$.

Mostrou-se assim que, em ambos os casos, $\mathbf{F}_{(10)}(x \sqcap y) \subseteq \mathbf{F}_{(10)}(x' \sqcap y')$. Portanto, $\mathbf{F}_{(10)}$ é monótona.

2. Se $X \sqcap Y$ é um subconjunto dirigido de $\mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$, em relação a inclusão, então tem-se que $\mathbf{F}_{(10)}(X \sqcap Y)$ é também dirigido.

De fato, sejam $X, Y \subseteq \mathbb{D}_{\infty}^{\rightarrow}$ subconjuntos dirigidos de $\mathbb{D}_{\infty}^{\rightarrow}$ relativos à inclusão. Como $\mathbb{D}_{\infty}^{\rightarrow}$ é fechado para a uniões dirigidas, $\bigcup^{\uparrow} X \sqcap \bigcup^{\uparrow} Y \subseteq \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$ e considerando-se o fato de que $\mathbf{F}_{(10)}$ é monótona, $\mathbf{F}_{(10)}(\bigcup^{\uparrow} X \sqcap \bigcup^{\uparrow} Y)$ é dirigido.

Suponha agora que $\Pi_{\infty,k}^{\rightarrow}$ e $\pi_{k+1,\infty}^{(00)}$ são funções contínuas.

- 2.1 Se $\Upsilon_{\infty}(\bigcup^{\uparrow} X) \cap \Upsilon_{\infty}(\bigcup^{\uparrow} Y) = \emptyset$ e $\Pi_{\infty,k}^{\rightarrow}(\bigcup^{\uparrow} X), \Pi_{\infty,k}^{\rightarrow}(\bigcup^{\uparrow} Y) \in \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$, onde $k = \max\{m, n\}$ tem-se que

$$\begin{aligned} \mathbf{F}_{(10)}(\bigcup^{\uparrow} X \sqcap \bigcup^{\uparrow} Y) &= \pi_{k+1,\infty}^{(00)} [\Pi_{\infty,k}^{\rightarrow}(\bigcup^{\uparrow} X) \bigcup \Pi_{\infty,k}^{\rightarrow}(\bigcup^{\uparrow} Y)] \\ &= \pi_{k,\infty}^{(00)} [\bigcup^{\uparrow} \{\Pi_{\infty,k}^{\rightarrow}(x) \mid x \in X\} \bigcup \bigcup^{\uparrow} \{\Pi_{\infty,k}^{\rightarrow}(y) \mid y \in Y\}] \\ &\subseteq \bigcup^{\uparrow} \{\pi_{k,\infty}^{(00)} [\Pi_{\infty,k}^{\rightarrow}(y) \cup \Pi_{\infty,k}^{\rightarrow}(x)] \mid x \in X, y \in Y\} \\ &= \bigcup^{\uparrow} \{\mathbf{F}_{(10)}(x \sqcap y) \mid x \in X, y \in Y\} \end{aligned}$$

2.2 Quando alguma das hipóteses acima não for verificada, tem-se que

$$\mathbf{F}_{(10)}(\cup^\uparrow X \sqcap \cup^\uparrow Y) = \emptyset.$$

Como $\mathbf{F}_{(10)}$ é monótona, $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset, \forall x \in X, y \in Y$. Assim, provou-se que também neste caso $\mathbf{F}_{(10)}(\cup^\uparrow X \sqcap \cup^\uparrow Y) = \cup\{\mathbf{F}_{(10)}(x \sqcap y) \mid x \in X, y \in Y\}$. Portanto, provou-se que $\mathbf{F}_{(10)}$ é contínua.

3. Sejam $x \sqcap y, x' \sqcap y' \in \mathbb{D}_\infty^\rightarrow \prod \mathbb{D}_\infty^\rightarrow$, tais que $(x \sqcap y) \cup (x' \sqcap y') = (x \cup x') \sqcap (y \cup y') \in \mathbb{D}_\infty^\rightarrow \prod \mathbb{D}_\infty^\rightarrow$.

Suponha que $\Pi_{\infty,k}^\rightarrow, \pi_{k+1,\infty}^\rightarrow$ e $\pi_{k+1,\infty}^{(00)}$ são funções estáveis.

3.2 Considere primeiramente que as condições da Definição 8.3 são satisfeitas, ou seja,

- $\Upsilon_\infty(x \sqcap x') \cap \Upsilon_\infty(y \sqcap y') = \emptyset, \Upsilon_\infty(x) \cap \Upsilon_\infty(y) = \emptyset$ e $\Upsilon_\infty(x') \cap \Upsilon_\infty(y') = \emptyset$.
- $\Pi_{\infty,k}(x \sqcap x'), \Pi_{\infty,k}(y \sqcap y') \in \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$, onde $k = \max\{m, n\}$.

Assim, tem-se que

$$\begin{aligned} \mathbf{F}_{(10)}(x \sqcap y \sqcap x' \sqcap y') &= \mathbf{F}_{(10)}((x \sqcap x') \sqcap (y \sqcap y')) \\ &= \pi_{k+1,\infty}^{(00)} [\Pi_{\infty,k}^\rightarrow(x \sqcap x') \cup \Pi_{\infty,k}^\rightarrow(y \sqcap y')] \\ &= \pi_{k+1,\infty}^{(00)} [(\Pi_{\infty,k}^\rightarrow(x) \cap \Pi_{\infty,k}^\rightarrow(x')) \cup (\Pi_{\infty,k}^\rightarrow(y) \cap \Pi_{\infty,k}^\rightarrow(y'))] \\ &= \pi_{k+1,\infty}^{(00)} [\Pi_{\infty,k}^\rightarrow(x) \cup \Pi_{\infty,k}^\rightarrow(y)] \cap \pi_{k+1,\infty}^{(00)} [\Pi_{\infty,k}^\rightarrow(x') \cup \Pi_{\infty,k}^\rightarrow(y')] \\ &= \mathbf{F}_{(10)}(x \sqcap y) \sqcap \mathbf{F}_{(10)}(x' \sqcap y') \end{aligned}$$

3.2 E quando alguma das condições acima não for verificada, $\mathbf{F}_{(10)}(x \sqcap y \sqcap x' \sqcap y') = \mathbf{F}_{(10)}((x \sqcap y) \sqcap (x' \sqcap y')) = \emptyset$. Considera-se então a análise das seguintes situações:

3.2.1 Ou $x \sqcap x'$ e $y \sqcap y'$ não são concorrentes, e conseqüentemente, x e y assim como x' e y' também não são concorrentes. Logo tem-se que $\mathbf{F}_{(10)}(x \sqcap y) = \mathbf{F}_{(10)}(x' \sqcap y') = \emptyset$.

3.2.2 Ou quando $x \sqcap x' \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n), y \sqcap y' \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m)$, mas $\theta_1 \neq 00, \theta_2 \neq 00$. Assim, o mesmo vale para x, x' e y, y' , ou seja, $x, x' \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n), y, y' \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m)$, mas $\theta_1 \neq 00, \theta_2 \neq 00$. Logo tem-se que $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset = \mathbf{F}_{(10)}(x' \sqcap y')$.

3.2.3 Ou quando $\Pi_{\infty,k}^\rightarrow(x)$ (ou $\Pi_{\infty,k}^\rightarrow(x')$) $\notin \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$. Logo, $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset$ (ou $\mathbf{F}_{(10)}(x' \sqcap y') = \emptyset$). O mesmo pode ser concluído ao se considerar $\Pi_{\infty,k}^\rightarrow(y)$ (ou $\Pi_{\infty,k}^\rightarrow(y')$) $\notin \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$.

3.2.4 Ou ainda quando $x = x' = y = y' = \emptyset$. Por definição, $\mathbf{F}_{(10)}(x \sqcap y) = \mathbf{F}_{(10)}(x \sqcap y) = \emptyset$.

Mostrou-se então que $\mathbf{F}_{(10)}(x \sqcap y \sqcap x' \sqcap y') = \mathbf{F}_{(10)}(x \sqcap y) \sqcap \mathbf{F}_{(10)}(x' \sqcap y')$. Portanto $\mathbf{F}_{(10)}$ é estável.

4. Sejam $X, Y \subseteq \mathbb{D}_\infty^\rightarrow$ tais que, se para todo $x, x' \in X$ e $y, y' \in Y$ tem-se que $x \cup x' \in \mathbb{D}_\infty^\rightarrow, y \cup y' \in \mathbb{D}_\infty^\rightarrow$ e $x, x' \in \pi_{n,\infty}^{\theta_1}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\rightarrow[\mathbb{D}_n], y, y' \in \pi_{m,\infty}^{\theta_2}[\mathbb{D}_m] \subseteq \pi_{m,\infty}^\rightarrow[\mathbb{D}_m]$.

Suponha que $\Pi_{\infty,k}^\rightarrow, \pi_{\infty,k}^\rightarrow$ e $\pi_{k+1,\infty}^{(00)}$ são funções lineares.

4.1 Considere primeiramente que as condições da Definição 8.3 são satisfeitas, ou seja,

- $\Upsilon_\infty(\cup X) \cap \Upsilon_\infty(\cup Y) = \emptyset$;
- $x, x' \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n), y, y' \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m)$ e tem-se que $\theta_1 = \theta_2 = 00$;
- $\Pi_{\infty,k}(\cup X), \Pi_{\infty,k}(\cup Y) \in \gamma_k^0 \circ \phi_k^1[\mathbb{D}_k]$, onde $k = \max\{m, n\}$;

$$\begin{aligned} \mathbf{F}_{(10)}(\cup X \sqcap \cup Y) &= \pi_{k,\infty}^{(00)} [\Pi_{\infty,k}^\rightarrow(\cup X) \cup \Pi_{\infty,k}^\rightarrow(\cup Y)] \\ &= \pi_{k,\infty}^{(00)} [\cup\{\Pi_{\infty,k}^\rightarrow(x) \mid x \in X\} \cup \cup\{\Pi_{\infty,k}^\rightarrow(y) \mid y \in Y\}] \end{aligned}$$

$$\begin{aligned}
 &= \bigcup \{ \pi_{k,\infty}^{(00)} [\Pi_{\infty,k}^{\rightarrow}(x) \cup \Pi_{\infty,k}^{\rightarrow}(y)] \mid x \in X, y \in Y \} \\
 &= \bigcup \{ \mathbf{F}_{(10)}(x \sqcap y) \mid x \in X, y \in Y \}
 \end{aligned}$$

4.2 E finalmente, se nenhuma das hipóteses acima for verificada, $\mathbf{F}_{(10)}(\bigcup X \sqcap \bigcup Y) = \emptyset$. Assim, pela monotonicidade da função $\mathbf{F}_{(10)}$, já provada anteriormente, tem-se que $\mathbf{F}_{(10)}(x \sqcap y) = \emptyset$, sempre que $x \subseteq \bigcup X$ e $y \subseteq \bigcup Y$. Mostrou-se que, em ambos os casos, tem-se que

$$\mathbf{F}_{(10)}(\bigcup X \sqcap \bigcup Y) = \bigcup \{ \mathbf{F}_{(10)}(x \sqcap y) \mid x \in X, y \in Y \}.$$

Considerando-se todos os itens desta prova conclui-se que $\mathbf{F}_{(10)}$ é linear. \blacktriangle

8.2 Construtores de Processos Modelados pela Construção Temporal do Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$

O domínio $\mathbb{D}_{\infty}^{\rightarrow}$, da mesma forma que $\mathbb{D}_{\infty}^{\leftarrow}$, possui interpretação para o produto seqüencial e a soma determinística, onde um ou mais termos podem ser processos infinitos quanto ao tempo de execução. No entanto, para se alcançar esta formalização, faz-se uma análise da indexação dos tokens que compõem suas interpretações. Neste sentido, a construção temporal das interpretações de um processo se reduz à construção dos indexadores de seus tokens, podem ser entendidas como palavras de um escolhido alfabeto descrevendo os construtores envolvidos na execução parcial destes processos em intervalos de tempo finito.

Na primeira parte desta seção são apresentadas as funções deslocamento, a partir das quais, obtém-se as interpretações para o produto seqüencial e a soma determinística de processos sem restrições quanto ao tempo de execução.

8.2.1 Funções Deslocamento

Num processo seqüencial infinito, o construtor que desloca uma posição para a direita, cada um de seus fatores, será interpretado pelo operador $\vec{\mathbf{F}}$. Por analogia, o construtor responsável pelo deslocamento no sentido contrário, será interpretado pelo operador $\overleftarrow{\mathbf{F}}$.

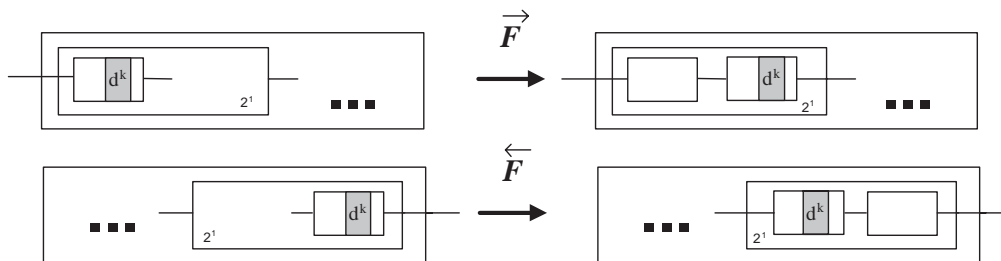


FIGURA 8.1 - As Funções Deslocamento no PP de Processos.

A Figura 8.1 é uma forma de expressar esta idéia, onde está representada a aplicação da função deslocamento $\vec{\mathbf{F}}$ sobre o objeto parcial $\{d_{:001}^{(0)}\} \in \mathbb{D}_{\infty}^{\rightarrow}$. Baseados na linearidade

de $\overrightarrow{\mathbf{F}}$, pode-se estender esta interpretação, e considerar todos os objetos totais para os quais $\{d_{:001}^{(0)}\}$ constitui-se numa aproximação finita. As mesmas considerações podem ser feitas com relação a aplicação da função deslocamento $\overleftarrow{\mathbf{F}}$ sobre o subconjunto coerente unitário $\{d_{:011}^{(k)}\} \in \mathbb{D}_{\infty}^{\leftarrow}$.

Salienta-se que, para um processo infinito, resultante de um produto ou uma soma, não existe um limite de tempo para sua execução. Portanto as imagens obtidas pelas funções deslocamento interpretam o deslocamento destes processos no tempo.

Considerando-se a soma determinística infinita, o construtor que desloca uma posição para baixo, cada um dos termos da soma, será interpretado pelo operador \mathbf{F}^{\downarrow} , da mesma forma que o operador \mathbf{F}^{\uparrow} interpreta o deslocamento oposto.

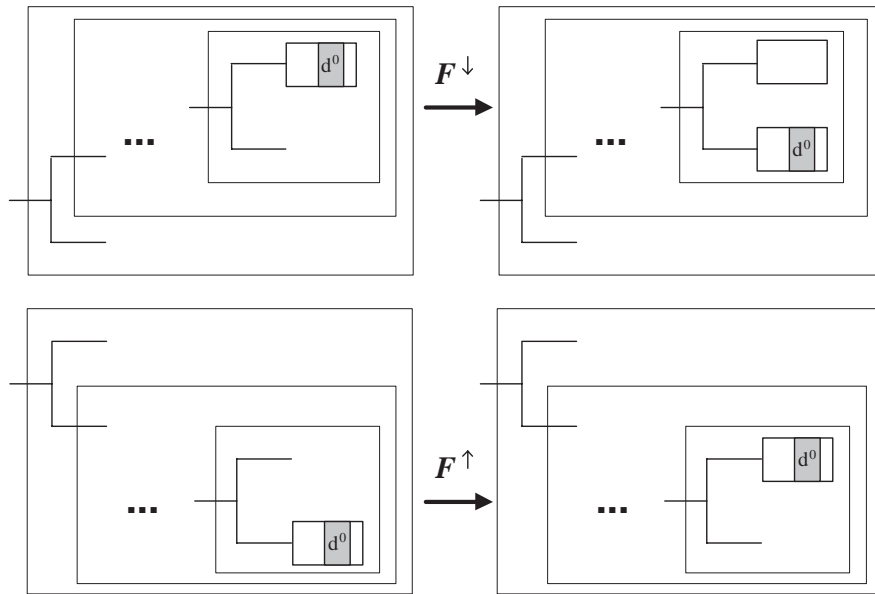


FIGURA 8.2 - As Funções Deslocamento na SS de Processos.

A Figura 8.2 é outra forma de expressar esta idéia, onde está representada a aplicação das funções \mathbf{F}^{\downarrow} e \mathbf{F}^{\uparrow} sobre os objetos parciais $\{d_{:002}^{(k)}\} \in \mathbb{D}_{\infty}^{\rightarrow}$ e $\{d_{:012}^{(k)}\} \in \mathbb{D}_{\infty}^{\leftarrow}$. Neste caso, tem-se $\mathbf{F}^{\downarrow}(\{d_{:002}^{(k)}\}) = \{d_{012:002}^{(k)}\}$ e $\mathbf{F}^{\uparrow}(\{d_{:012}^{(k)}\}) = \{d_{002:012}^{(k)}\}$.

Com base na natureza indutiva do domínio $\mathbb{D}_{\infty}^{\rightarrow}$ (ou de $\mathbb{D}_{\infty}^{\leftarrow}$), para se interpretar os construtores de processos que deslocam posições em produtos sequenciais ou em escolhas determinísticas infinitas, é preciso que tais interpretações sejam construídas, primeiramente, nos níveis \mathbb{D}_n da estrutura.

Neste sentido, a definição das funções deslocamento torna explícito um método recursivo de determinação dos novos índices dos tokens que constituem os subconjuntos na imagem de cada uma destas funções.

Após as observações apresentadas em seguida, essas idéias são formalizadas nas próximas definições e proposições.

Observação 8.3

1. Algumas observações sobre a notação são importantes pois facilitam a compreensão das próximas definições e proposições. A definição das imersões $\pi_n^{\rightarrow}, \pi_{n,\infty}^{\rightarrow}$ é dada por casos que envolvem as imersões $\pi_n^{(\theta)}, \pi_{n,\infty}^{(\theta)}$, com $\theta \in \{00, 001, 002\}$. De forma análoga, a definição das imersões $\pi_n^{\leftarrow}, \pi_{n,\infty}^{\leftarrow}$ envolve restrições relacionadas com as imersões $\pi_n^{(\theta)}, \pi_{n,\infty}^{(\theta)}$, com $\theta \in \{00, 011, 012\}$.

Portanto, para modelar o deslocamento de posições no produto seqüencial ou na soma determinística, são consideradas as imersões $\pi_n^{(\theta)}$ com indexador $\theta \in \{001, 011, 002, 012\}$.

1.1 Fixando θ , a composição finita das projeções $\Pi_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n-1}$, apresentadas na Definição 5.5, será indicada pela expressão

$$\odot_{i=n}^m \Pi_i^{(\theta)} = \Pi_n^{(\theta)} \circ \Pi_{n-1}^{(\theta)} \circ \dots \circ \Pi_m^{(\theta)},$$

se $n, m \in \omega$ e $n \geq m$. Caso contrário, quando $n < m$ tem-se que $\odot_{i=n}^m \Pi_i^{(\theta)} = Id_{\mathbb{D}_n}$.

1.2 A mesma notação é adotada para as imersões $\pi_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$, apresentadas na Definição 5.4. Além disso,

$$\odot_{i=0}^n \pi_i^{(\theta_1)} \circ \odot_{i=n+1}^{n+m+1} \pi_i^{(\theta_2)}(x) = x \times \{(\theta_1^n . \theta_2^m)\}.$$

2. Cada subconjunto coerente $x \in \mathbb{D}_{\infty}^{\rightarrow}$ que interpreta um produto seqüencial ou uma soma determinística, sem limite de tempo de execução, pode ser identificado pela expressão

$$x = \{a_{\alpha:\theta} \mid \alpha = \alpha_1 d_1 \beta_1 . \alpha_2 d_2 \beta_2 . \dots \alpha_n d_n \beta_n\},$$

sempre que, para os indexadores de tokens, tem-se

- $\alpha_n \in \{0, 1, 2\}$ indica o construtor interno interpretado por $F_{(00)}, F_{(10)}$ e $F_{(20)}$;
- $\beta_n \in \{1, 2\}$ indica o construtor externo, produto seqüencial ou soma determinística;
- $d_n \in \{0, 1\}$ indica o primeiro ou o segundo termo associado ao construtor externo;
- $\theta \in \{001, 002\}$ são indexadores das imersões de \mathbb{D}_n em \mathbb{D}_{∞} .

No caso em que $x \in \mathbb{D}_{\infty}^{\leftarrow}$, valem as mesmas considerações dos itens anteriores, exceto que agora $\theta \in \{011, 012\}$.

Sem perda de generalidade e com o objetivo de simplificar a notação, nas próximas definições considera-se $\alpha_n = 0, \forall n \in \omega$.

Definição 8.5 Considere os indexadores $\theta \in \{001, 011, 002, 012\}$ e $\beta \in \{1, 2\}$. Considere também as composições finitas de projeções e imersão, dadas pelas expressões

$$\begin{aligned} \odot_{i=n}^{n-m} \Pi_i^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n-(m+1)}, & \quad \odot_{i=n}^{n-m} \Pi_i^{(\theta)}(x) = \Pi_{n-m}^{(\theta)} \circ \dots \circ \Pi_{n-1}^{(\theta)} \circ \Pi_n^{(\theta)}(x), \\ \odot_{i=n}^{n+m} \pi_i^{(01\beta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+m}, & \quad \odot_{i=n}^{n+m} \pi_i^{(01\beta)}(x) = \pi_{n+m}^{(01\beta)} \circ \dots \circ \pi_{n+1}^{(01\beta)} \circ \pi_n^{(01\beta)}(x). \end{aligned}$$

Fixado β , a função $\mathbf{f}_n^{(0\beta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ é recursivamente definida pelas equações

$$\left\{ \begin{array}{l} \mathbf{f}_0^{(0\beta)}(x) = \pi_0^{(01\beta)}(x), \\ \mathbf{f}_n^{(0\beta)}(x) = (\pi_n^{(00\beta)} \circ \mathbf{f}_{n-1}^{(0\beta)} \circ \Pi_n^{(00\beta)})(x) \cup \\ \quad \bigcup_{i=n-2}^0 [(\pi_{n-(i+1)}^{(01\beta)} \circ \mathbf{f}_i^{(0\beta)} \circ \Pi_{i+1}^{(00\beta)} \circ \odot_{k=n}^{n-(i+1)} \Pi_k^{(01\beta)})(x)] \cup \\ \quad (\pi_n^{(01\beta)} \circ \odot_{k=0}^n \pi_k^{(00\beta)} \circ \odot_{k=n}^1 \Pi_k^{(01\beta)})(x). \end{array} \right.$$

Na definição anterior, o indexador β identifica qual o construtor de processos está sendo interpretado. Assim, $\beta = 1$ identifica o produto seqüencial, enquanto $\beta = 2$ é o identificador da soma determinística.

Fixando $\beta = 1$ tem-se $\theta \in \{001, 011\}$. Desenvolvendo as composições abreviadas na Definição 8.5, e considerando a notação onde as imersões $\pi_n^\theta : \mathbb{D}_n \rightarrow \mathbb{D}_n$ são dadas pelo produto cartesiano $\pi_n^\theta(x) = x \times \{\theta\}$ acrescida das observações apresentadas no início desta seção, pode-se explicitar cada uma das funções $\mathbf{f}_n^{(01)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$.

Tem-se então, as seguintes expressões

- $\mathbf{f}_0^{(01)}(x) = x \times \{(011)\}$
- $\mathbf{f}_1^{(01)}(x) = (\mathbf{f}_0^{(01)} \circ \Pi_1^{(001)})(x) \times \{(001)\} \cup (\Pi_1^{(011)})(x) \times \{(001.011)\}$
- $\mathbf{f}_2^{(01)}(x) = (\mathbf{f}_1^{(01)} \circ \Pi_2^{(001)})(x) \times \{(001)\} \cup$
 $(\mathbf{f}_0^{(01)} \circ \Pi_1^{(001)} \circ \Pi_2^{(011)})(x) \times \{(011)\} \cup (\Pi_1^{(011)} \circ \Pi_2^{(011)})(x) \times \{(001^2.011)\}$
- $\mathbf{f}_3^{(01)}(x) = (\mathbf{f}_2^{(01)} \circ \Pi_3^{(001)})(x) \times \{(001)\} \cup$
 $(\mathbf{f}_1^{(01)} \circ \Pi_2^{(001)} \circ \Pi_3^{(011)})(x) \times \{(011)\} \cup$
 $(\mathbf{f}_0^{(01)} \circ \Pi_1^{(001)} \circ (\Pi_2^{(011)} \circ \Pi_3^{(011)}))(x) \times \{(011^2)\} \cup$
 $(\Pi_1^{(011)} \circ \Pi_2^{(011)} \circ \Pi_3^{(011)})(x) \times \{(001^3.011)\}$
- $\mathbf{f}_4^{(01)}(x) = (\mathbf{f}_3^{(01)} \circ \Pi_4^{(001)})(x) \times \{(001)\} \cup$
 $(\mathbf{f}_2^{(01)} \circ \Pi_3^{(001)} \circ \Pi_4^{(011)})(x) \times \{(011)\} \cup$
 $(\mathbf{f}_1^{(01)} \circ \Pi_2^{(001)} \circ \Pi_3^{(011)} \circ \Pi_4^{(011)})(x) \times \{(011^2)\} \cup$
 $(\mathbf{f}_0^{(01)} \circ \Pi_1^{(001)} \circ \Pi_2^{(011)} \circ \Pi_3^{(011)} \circ \Pi_4^{(011)})(x) \times \{(011^3)\} \cup$
 $(\Pi_1^{(011)} \circ \Pi_2^{(011)} \circ \Pi_3^{(011)} \circ \Pi_4^{(011)})(x) \times \{(001^3.011)\}$
- \vdots
- $\mathbf{f}_n^{(01)}(x) = (\mathbf{f}_{n-1}^{(01)} \circ \Pi_n^{(001)})(x) \times \{(001)\} \cup$
 $\bigcup_{i=n-2}^0 [(\mathbf{f}_i^{(01)} \circ \Pi_{i+1}^{(001)} \circ \bigcirc_{k=n}^{n-(i+1)} \Pi_k^{(011)})(x) \times \{(011^{n-(i+1)})\}] \cup$
 $(\bigcirc_{k=n}^1 \Pi_k^{(011)})(x) \times \{(001^n.011)\}.$

A definição da função $\mathbf{f}_n^{(1\beta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ é construída de forma análoga. Em particular, quando $\beta = 1$, tem-se que $\mathbf{f}_n^{(11)}$ interpreta o deslocamento para a esquerda de todas as aproximações finitas - executadas em *n utc* - de objetos em $\mathbb{D}_\infty \prod \mathbb{D}_\infty$.

Definição 8.6 A função $\mathbf{f}_n^{(1\beta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ é recursivamente definida pela expressão

$$\left\{ \begin{array}{l} \mathbf{f}_0^{(1\beta)}(x) = \pi_0^{(00\beta)}(x), \\ \mathbf{f}_n^{(1\beta)}(x) = (\pi_n^{(01\beta)} \circ \mathbf{f}_{n-1}^{(1\beta)} \circ \Pi_n^{(01\beta)})(x) \cup \\ \bigcup_{i=n-2}^0 [(\pi_{n-(i+1)}^{(00\beta)} \circ \mathbf{f}_i^{(1\beta)} \circ \Pi_{i+1}^{(01\beta)} \circ \bigcirc_{k=n}^{n-(i+1)} \Pi_k^{(00\beta)})(x)] \cup \\ (\pi_n^{(00\beta)} \circ \bigcirc_{k=0}^n \pi_k^{(01\beta)} \circ \bigcirc_{k=n}^1 \Pi_k^{(00\beta)})(x). \end{array} \right.$$

Proposição 8.5 *As funções estritas $\mathbf{f}_n^{(\theta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ apresentadas na Definições 8.5 e 8.6 são morfismos na categoria CospLin , sempre que $\theta \in \{01, 02, 12, 11\}$.*

As próximas definições introduzem as funções deslocamento considerando-se as interpretações nos domínios $\mathbb{D}_\infty^\rightarrow$ e $\mathbb{D}_\infty^\leftarrow$.

Definição 8.7 *Sejam os espaços coerentes \mathbb{D}_n e \mathbb{D}_{n+1} , o indexador $\beta \in \{1, 2\}$ e as funções lineares $\mathbf{f}_n^{(0\beta)}, \mathbf{f}_n^{(1\beta)} : \mathbb{D}_n \rightarrow \mathbb{D}_{n+1}$ apresentadas anteriormente, nesta seção, nas Definições 8.5 e 8.6. Seja o indexador $\beta \in \{0, 1\}$.*

Considere as funções $\mathbf{F}_n^{(0\beta)}, \mathbf{F}_n^{(1\beta)} : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow$,

$$\mathbf{F}_n^{(0\beta)}(x) = \begin{cases} (\pi_{n+1,\infty}^{(00\beta)} \circ \mathbf{f}_n^{(0\beta)} \circ \Pi_{\infty,n}^{(00\beta)})(x), & \text{se } x = \bigcup_{n \in \omega}^\uparrow x_n, \quad x_n \in \pi_{n,\infty}^{(00\beta)} \mathbb{D}_n, \\ \emptyset, & \text{caso contrário; e} \end{cases}$$

$$\mathbf{F}_n^{(1\beta)}(x) = \begin{cases} x, & \text{se } x = \bigcup_{n \in \omega}^\uparrow x_n, \quad x_n \in \pi_{n,\infty}^{(00\beta)} \mathbb{D}_n, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

Da mesma forma, sejam as funções $\mathbf{F}_n^{(0\beta)}, \mathbf{F}_n^{(1\beta)} : \mathbb{D}_\infty^\leftarrow \rightarrow \mathbb{D}_\infty^\leftarrow$,

$$\mathbf{F}_n^{(0\beta)}(x) = \begin{cases} x, & \text{se } x = \bigcup_{n \in \omega}^\uparrow x_n, \quad x_n \in \pi_{n,\infty}^{(00\beta)} \mathbb{D}_n, \\ \emptyset, & \text{caso contrário;} \end{cases}$$

$$\mathbf{F}_n^{(1\beta)}(x) = \begin{cases} (\pi_{n+1,\infty}^{(01\beta)} \circ \mathbf{f}_n^{(1\beta)} \circ \Pi_{\infty,n}^{(01\beta)})(x), & \text{se } x = \bigcup_{n \in \omega}^\uparrow x_n, \quad x_n \in \pi_{n,\infty}^{(01\beta)} \mathbb{D}_n, \\ \emptyset, & \text{caso contrário.} \end{cases}$$

As funções deslocamento indicadas por

$$\vec{\mathbf{F}}_{\mathbb{D}_\infty^\rightarrow}, \overleftarrow{\mathbf{F}}_{\mathbb{D}_\infty^\rightarrow} : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow \quad \text{e} \quad \vec{\mathbf{F}}_{\mathbb{D}_\infty^\leftarrow}, \overleftarrow{\mathbf{F}}_{\mathbb{D}_\infty^\leftarrow} : \mathbb{D}_\infty^\leftarrow \rightarrow \mathbb{D}_\infty^\leftarrow$$

interpretam operadores que deslocam uma posição, para a direita ou para esquerda, respectivamente, cada um dos fatores num produto seqüencial cuja execução pode ser infinita. Tais funções são definidas, nos respectivos domínios, pelas expressões

$$\vec{\mathbf{F}}(x) = \bigcup_{n \in \omega} \mathbf{F}_n^{(01)}(x) \quad \text{e} \quad \overleftarrow{\mathbf{F}}(x) = \bigcup_{n \in \omega} \mathbf{F}_n^{(11)}(x).$$

Analogamente, $\mathbf{F}_{\mathbb{D}_\infty^\rightarrow}^\downarrow, \mathbf{F}_{\mathbb{D}_\infty^\rightarrow}^\uparrow : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow$ e $\mathbf{F}_{\mathbb{D}_\infty^\leftarrow}^\downarrow, \mathbf{F}_{\mathbb{D}_\infty^\leftarrow}^\uparrow : \mathbb{D}_\infty^\leftarrow \rightarrow \mathbb{D}_\infty^\leftarrow$ interpretam operadores que deslocam uma posição, para a cima ou para baixo, respectivamente, cada um dos termos numa soma determinística cuja execução pode ser infinita. Tem-se então as definições nos correspondentes domínios,

$$\mathbf{F}^\downarrow(x) = \bigcup_{n \in \omega} \mathbf{F}_n^{(02)}(x) \quad \text{e} \quad \mathbf{F}^\uparrow(x) = \bigcup_{n \in \omega} \mathbf{F}_n^{(12)}(x).$$

Proposição 8.6 *As funções apresentadas na Definição 8.7 são morfismos na categoria CospLin , sempre que $\beta \in \{0, 1\}$.*

Considerando o que foi mostrado na última seção, dado um processo elementar, é possível formalizar a geração de todos os índices relativos às posições que este processo poderá ocupar em um produto seqüencial infinito. Cada posição interpreta o tempo em que este processo será executado.

Uma interpretação mais intuitiva para a seqüencialidade, explicitada pelas sucessivas aplicações das funções deslocamento, pode ser alcançada considerando a estrutura algébrica dos sistemas n -ários, definidos essencialmente por um conjunto munido de funções sucessoras e morfismos básicos. Neste caso, os elementos do conjunto são indutivamente construídos e as operações são recursivamente definidas, consistindo portanto numa estrutura computável.

Veja a seguir, como esta interpretação é construída e como pode ser representada graficamente.

8.2.2 Abordagem Algébrica para Interpretações nos Domínios $\mathbb{D}_\infty^\rightarrow$ e $\mathbb{D}_\infty^\leftarrow$

Mostra-se a seguir uma interpretação para as imagens das sucessivas aplicações das funções deslocamento baseada na estrutura algébrica associada aos sistemas unários.

Os sistemas unários foram introduzidos por Stool [STO 61] na busca de uma abordagem algébrica para a indução e recursão nos Naturais. A posterior extensão - os sistemas binários, apresentada em Escardó [ESC 96], caracteriza a reta real por propriedades similares aos axiomas de Peano dos naturais, incluindo o princípio da indução e os correspondentes esquemas recursivos.

Em Reiser [REI 99], faz-se uma análise categórica do relacionamento entre a categoria dos sistemas unários e a categoria das ordens parciais, obtendo desta forma interpretações para os esquemas da μ -recursão nos sistemas de números naturais, utilizando construções definidas sobre os naturais parciais.

O estudo baseado nos trabalhos citados nos ajudou a mostrar que o conjunto de índices, associados aos tokens que constroem um objeto total em $\mathbb{D}_\infty^\rightarrow$ ou em $\mathbb{D}_\infty^\leftarrow$, constitui-se num sistema unário integral, mais precisamente, um sistema isomorfo ao sistema unário dos números naturais.

A aplicação sucessiva das funções deslocamento determinam sistemas unários distintos, isomorfos ao sistema unário dos naturais. O que será mostrado para as funções \overline{F} e \overline{F}^\leftarrow pode ser estendido para as outras funções deslocamento.

Embora não se mostre neste trabalho, verificou-se também que o conjunto de índices, associados aos tokens que constroem um objeto total em $\mathbb{D}_\infty^{\rightarrow\leftarrow}$, $\mathbb{D}_\infty^{\leftarrow\rightarrow}$ ou ainda em $\mathbb{D}_\infty^{\leftarrow\leftarrow}$ constitui-se num sistema binário integral, com dois centros.

Portanto, aquelas interpretações construídas nos espaços coerentes $\mathbb{D}_{\infty+m}^{\leftarrow\leftarrow}$, $\mathbb{D}_{\infty+m}^{\leftarrow\rightarrow}$ ou ainda em $\mathbb{D}_{\infty+m}^{\leftarrow\leftarrow}$ podem ser associadas à construção algébrica dos sistemas m -ários.

Definição 8.8 *Considere o alfabeto binário $\Sigma = \{0, 1\}$ e o subconjunto $\mathbb{N} \subseteq \Sigma^*$ de todas as palavras que possuem “1” como prefixo acrescido da palavra “0”. A terna ordenada $\mathbb{N} \equiv (\mathbb{N}, S : \mathbb{N} \rightarrow \mathbb{N}, 0 : \mathbf{1} \rightarrow \mathbb{N})$ é um sistema unário integral, denominado **sistema unário dos números naturais binários**. Neste caso,*

- \mathbb{N} indica o conjunto de todas as representações binárias dos naturais,
- $S : \mathbb{N} \rightarrow \mathbb{N}$ indica a função sucessora que adiciona 1 a cada número binário, e
- $0 : \mathbf{1} \rightarrow \mathbb{N}$ indica a função constante básica, a partir da qual se inicia o processo de sucessão, que gera todos os números naturais binários.

Definição 8.9 *Seja $x = \bigcup^\uparrow x_n \in \text{tot}(\mathbb{D}_\infty^\rightarrow)$, tal que $x_n \in \pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\rightarrow[\mathbb{D}_n]$ sempre que $\theta \in \{001, 002\}$. Seja também $x = \bigcup^\uparrow x_n \in \text{tot}(\mathbb{D}_\infty^\leftarrow)$, tal que $x_n \in \pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\leftarrow[\mathbb{D}_n]$ sempre que $\theta \in \{011, 012\}$. O conjunto de todos os índices associados aos tokens do subconjunto coerente x serão indicados por*

$$\mathfrak{S}_x = \{ \alpha : \theta \mid \mathbf{a}_{\alpha:\theta} \in x \},$$

onde $\alpha = \alpha_1 d_1 1 . \alpha_2 d_2 1 . \dots . \alpha_n d_n 1$, $\alpha_n \in \{0, 1, 2\}$ e $d_n \in \{0, 1\}$.

Utiliza-se uma notação diferenciada nas seguintes condições

- se $d_n = 0, \forall n \in \omega$, e $\theta = 001$, então $\alpha : 001 \equiv \mathfrak{S}_{001}$;
- se $d_n = 1, \forall n \in \omega$, e $\theta = 011$, então $\alpha : 011 \equiv \mathfrak{S}_{011}$.

Definição 8.10 *Seja \mathbb{N} o sistema unário dos números naturais binários apresentado na Definição 8.8. Defina-se, neste momento, a função $h : \mathfrak{S}_x \rightarrow \mathbb{N}$ que associa, cada $\alpha : \theta$ em \mathfrak{S}_x , uma palavra (não vazia) em \mathbb{N} da seguinte forma*

$$\begin{aligned} h(\alpha : \theta) &= d_n d_{n-1} \dots d_2 d_1, & \text{se } \theta &= 001, \\ h(\alpha : \theta) &= \tilde{d}_n \tilde{d}_{n-1} \dots \tilde{d}_2 \tilde{d}_1, & \text{onde } \tilde{d}_n &= 1 - d_n, \text{ se } \theta = 011. \end{aligned}$$

Em particular, $h(\mathfrak{S}_{001}) = h(\mathfrak{S}_{011}) = 0 \equiv 000 \dots 0$.

Sempre que $x \in \text{tot}(\mathbb{D}_\infty^{\rightarrow})$ ou $x \in \text{tot}(\mathbb{D}_\infty^{\leftarrow})$ a função $h : \mathfrak{S}_x \rightarrow \mathbb{N}$ é bijetora e portanto inversível. Define-se então a função $\beta : \mathfrak{S}_x \rightarrow \mathfrak{S}_x$ pela composição $\beta = h^{-1} \circ S \circ h$. Pela a injetividade da função β , segue-se a próxima definição.

Definição 8.11 *Seja $x = \bigcup^\uparrow x_n$, $x \in \text{tot}(\mathbb{D}_\infty^{\rightarrow})$ tal que $x_n \in \pi_{n,\infty}^{(001)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ ou $x \in \text{tot}(\mathbb{D}_\infty^{\leftarrow})$, tal que $x_n \in \pi_{n,\infty}^{(011)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\leftarrow}[\mathbb{D}_n]$. A terna ordenada*

$$\mathfrak{S}_x \equiv (\mathfrak{S}_x, \beta : \mathfrak{S}_x \rightarrow \mathfrak{S}_x, \mathfrak{S}_\theta : \mathbf{1} \rightarrow \mathfrak{S}_x)$$

constitui-se no sistema unário de indexadores dos tokens que definem cada objeto total $x \in \mathbb{D}_\infty$. No caso, diz-se que \mathfrak{S}_x é o sistema unário induzido pelo isomorfismo $h : \mathfrak{S}_x \rightarrow \mathbb{N}$ que satisfaz as equações recursivas $h \circ \mathfrak{S}_\theta = 0$ e $h \circ \beta = S \circ h$.

A partir do indexador \mathfrak{S}_θ , a consecutiva aplicação da função sucessora β descreve o processo de geração dos índices de todos os tokens de um objeto total $x = \{\mathbf{a}_{\alpha:\theta} \mid \alpha = \alpha_1 d_1 1 . \alpha_2 d_2 1 . \dots . \alpha_n d_n 1\} \in \text{tot}(\mathbb{D}_\infty^{\rightarrow})$ (ou $\text{tot}(\mathbb{D}_\infty^{\leftarrow})$), sempre que $\alpha_n \in \{0, 1, 2\}$, $d_n \in \{0, 1\}$ e $\theta \in \{001, 011, 002, 012\}$. Observe isto nos exemplos.

Exemplificação 8.3

1. Se $x = \{d_{\cdot 001}^{(0)}, d_{011 \cdot 001}^{(1)}, d_{001.011 \cdot 001}^{(2)}, d_{011.011 \cdot 001}^{(3)}, d_{001.001.011 \cdot 001}^{(4)}, \dots\} \in \text{tot}(\mathbb{D}_\infty^{\rightarrow})$ então $\mathfrak{S}_x = \{ : 001, 011 : 001, 001.011 : 001, 011.011 : 001, 001.001.011 : 001, \dots\}$ e as duas primeiras seqüências apresentadas logo a seguir, representam os elementos gerados pelos sistemas unários isomorfos \mathbb{N} e \mathfrak{S}_x .

$$\begin{array}{ccccccccc} 001:001 & \rightarrow & 011:001 & \rightarrow & 001.011:001 & \rightarrow & 011.011:001 & \rightarrow & 001.001.011:001 & \rightarrow & \dots \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \\ 0 & \rightarrow & 1 & \rightarrow & 10 & \rightarrow & 11 & \rightarrow & 100 & \rightarrow & \dots \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \\ 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & \dots \end{array}$$

Considerando as últimas definições, é possível se determinar o índice de cada token de um objeto na imagem associada à função deslocamento $\overrightarrow{\mathbf{F}}$. Pelo isomorfismo $h : \mathfrak{S}_x \rightarrow \mathbb{N}$ basta adicionar uma unidade ao dígito mais à direita em cada palavra invertida identificada com os tokens deste objeto. Isto será formalizado na próxima proposição e para o conjunto $x \in \mathbb{D}_\infty^{\rightarrow}$

$$\begin{aligned} \overrightarrow{\mathbf{F}}(x) &= \{d_{011 \cdot 001}^{(0)}, d_{001.011 \cdot 001}^{(1)}, d_{011.011 \cdot 001}^{(2)}, d_{001.001.011 \cdot 001}^{(4)}, \dots\} \text{ e} \\ \overleftarrow{\mathbf{F}}(x) &= x. \end{aligned}$$

Salienta-se por fim que é fácil de reescrever o exemplo anterior e obter considerações semelhantes relativas às funções deslocamento $\overrightarrow{\mathbf{F}}$ e $\overleftarrow{\mathbf{F}}$, quando os tokens são indexados por $\theta = 011$ e $x \in \text{tot}(\mathbb{D}_\infty^{\leftarrow})$. Veja os outros exemplos que ilustram a aplicação destas funções deslocamento.

2. Se $x = \{\overline{d^{(k)}}_{101.001.011:001}\}_{k \in I}$, $\overleftarrow{\mathbf{F}}(x) = x$ e $\overrightarrow{\mathbf{F}}(x) = \{\overline{d^{(k)}}_{111.001.011:001}\}_{k \in I}$.
3. Se $x = \{\overline{d^{(k)}}_{101:00}\}_{k \in I}$, $\overleftarrow{\mathbf{F}}(x) = \emptyset = \overleftarrow{\mathbf{F}}(\emptyset) = \overrightarrow{\mathbf{F}}(\emptyset)$

4. Se $x = \{d_{011}^{(k)}, e_{001:011}^{(l)}\}$, $\vec{\mathbf{F}}(x) = \{d_{001:011}^{(k)}, e_{011.001:011}^{(l)}\}$ e $\overleftarrow{\mathbf{F}}(x) = x$.
5. Se $x = \{d_{011}^{(0)}, d_{001:011}^{(1)}, d_{011.001:011}^{(2)}, d_{001.001:011}^{(4)}, \dots\} \in \text{tot}(\mathbb{D}_{\infty}^{\leftarrow})$ então tem-se que
- $$\vec{\mathbf{F}}(x) = x \text{ e}$$
- $$\overleftarrow{\mathbf{F}}(x) = \{d_{001:011}^{(0)}, d_{011.001:011}^{(1)}, d_{001.001:011}^{(2)}, d_{011.011.001:011}^{(4)}, \dots\}.$$

Proposição 8.7 *Considere as funções $\beta : \mathfrak{S}_x \rightarrow \mathfrak{S}_x$, $\overleftarrow{\mathbf{F}}, \vec{\mathbf{F}} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ e $\overleftarrow{\mathbf{F}}, \vec{\mathbf{F}} : \mathbb{D}_{\infty}^{\leftarrow} \rightarrow \mathbb{D}_{\infty}^{\leftarrow}$ formalmente apresentadas nas Definições 8.11 e 8.7. Sempre que $x = \bigcup^{\uparrow} x_n \in \text{tot}(\mathbb{D}_{\infty}^{\leftarrow})$ e $x_n \in \pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$, as seguintes igualdades são satisfeitas*

1. $\vec{\mathbf{F}}(x) = \begin{cases} \{\mathbf{a}_{\beta(\alpha:\theta)} \mid \mathbf{a}_{\alpha:\theta} \in x\}, & \text{se } \theta = 001, \\ \emptyset, & \text{caso contrário; e} \end{cases}$
2. $\overleftarrow{\mathbf{F}}(x) = \begin{cases} x, & \text{se } \theta = 001, \\ \emptyset, & \text{caso contrário.} \end{cases}$

Se $x = \bigcup^{\uparrow} x_n \in \text{tot}(\mathbb{D}_{\infty}^{\leftarrow})$ e $x_n \in \pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\leftarrow}[\mathbb{D}_n]$, então as seguintes igualdades são satisfeitas

1. $\vec{\mathbf{F}}(x) = \begin{cases} x, & \text{se } \theta = 001, \\ \emptyset, & \text{caso contrário.} \end{cases}$
2. $\overleftarrow{\mathbf{F}}(x) = \begin{cases} \{\mathbf{a}_{\beta(\alpha:\theta)} \mid \mathbf{a}_{\alpha:\theta} \in x\}, & \text{se } \theta = 011, \\ \emptyset, & \text{caso contrário.} \end{cases}$

A construção gráfica baseada na Definição 8.7, torna mais simples expressar a imagem obtida pela aplicação das funções deslocamento. Considerando que tais funções foram definidas por expressões recursivas, esta seção encerra com a apresentação de um método gráfico descendente, pelo qual é possível se determinar os índices dos tokens de objetos em $\mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$ e em $\mathbb{D}_{\infty}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{D}_{\infty}^{\rightarrow}$. É imediato que o mesmo pode ser obtido para objetos em $\mathbb{D}_{\infty}^{\leftarrow} \prod \mathbb{D}_{\infty}^{\leftarrow}$ e em $\mathbb{D}_{\infty}^{\leftarrow} \prod_{\mathbb{B}} \mathbb{D}_{\infty}^{\leftarrow}$.

8.2.3 Interpretação Gráfica para Funções Deslocamento nos Domínios $\mathbb{D}_{\infty}^{\rightarrow}$ e $\mathbb{D}_{\infty}^{\leftarrow}$

Para cada função-deslocamento apresentada na Definição 8.7, são construídos triângulos correspondendo às árvores binárias geradoras das palavras (infinitas) que podem ser escritas com os dígitos do alfabeto $\Sigma = \{0, 1\}$, tendo como base seqüências infinitas de 0's ou de 1's. Considerando que o construtor produto seqüencial é interpretado por um operador binário, cada dígito na árvore, interpreta o fator à esquerda (dígito 0) ou à direita (dígito 1).

O mesmo pode ser observado em relação ao primeiro e segundo termos de uma soma determinística. Neste sentido, a árvore sobre a qual pode-se interpretar a construção, no sentido temporal, dos tokens que definem os subconjuntos corentes em $\mathbb{D}_{\infty}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{D}_{\infty}^{\rightarrow}$ (ou em $\mathbb{D}_{\infty}^{\leftarrow} \prod_{\mathbb{B}} \mathbb{D}_{\infty}^{\leftarrow}$) é análoga, e será omitida.

Sobre os dígitos de cada triângulo interpretam-se deslocamentos distintos:

- o deslocamento vertical, de uma linha para alguma outra mais abaixo, determinando o conjunto de índices \mathfrak{S}_x associados a cada aproximação finita de x , construída em cada um dos níveis \mathbb{D}_n da estrutura indutiva deste modelo. Nesta representação, o deslocamento da n -ésima linha até a base define todos os possíveis índices dos tokens de um subconjunto coerente do subespaço $\pi_{n,\infty}^{(\theta)}[\mathbb{D}_n] \subseteq \mathbb{D}_{\infty}$, onde $\theta \in \{001, 002\}$.
- o deslocamento horizontal, interpretando a função sucessora $\beta : \mathfrak{S}_x \rightarrow \mathfrak{S}_x$.

Os exemplos e os triângulos da Figura 8.3 se relacionam diretamente com a função deslocamento \vec{F} . Compare estes exemplos com os outros já apresentados nesta seção.

$$\vec{F}\{d_{011.011:001}^{(k)}\} = \{d_{001.001.011:001}^{(k)}\} \quad \vec{F}\{d_{011.011.001:011}^{(k)}\} = \{d_{011.011.001:011}^{(k)}\}$$

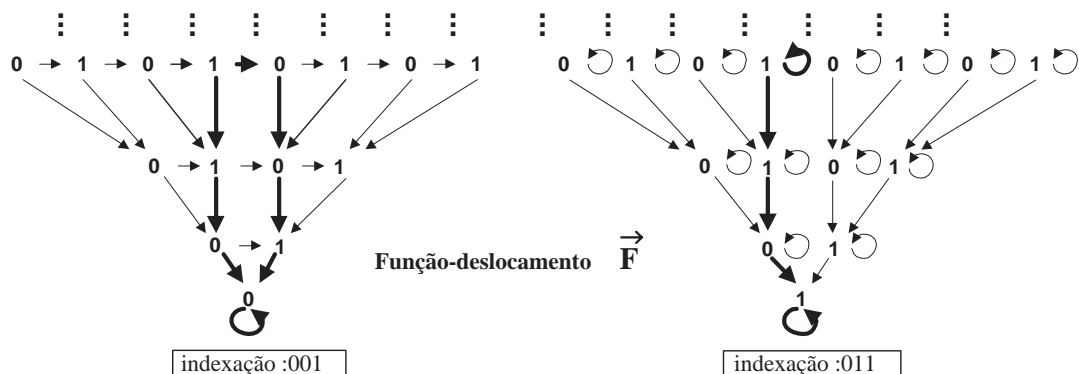


FIGURA 8.3 - Representação para a Função Deslocamento Direita.

Da mesma forma pode-se construir os cones sobre os quais pode-se interpretar graficamente o comportamento da função deslocamento \vec{F} , determinando os índices que definem os tokens de um subconjunto coerente gerado na sua imagem. Basta substituir os laços por setas horizontais com sentido de deslocamento da direita para a esquerda, no cone que tem o dígito “1” como base. No outro cone, tendo como base o dígito “0”, as setas horizontais são substituídas por laços, que representam neste caso, uma restrição da função identidade. A Figura 8.4, apresentada logo a seguir, se relaciona diretamente com a função deslocamento \overleftarrow{F} .

$$\overleftarrow{F}\{d_{011.011.001:011}^{(k)}\} = \{d_{001.011.001:011}^{(k)}\} \quad \text{e} \quad \overleftarrow{F}\{d_{011.011.001}^{(k)}\} = \{d_{011.011.001}^{(k)}\}$$

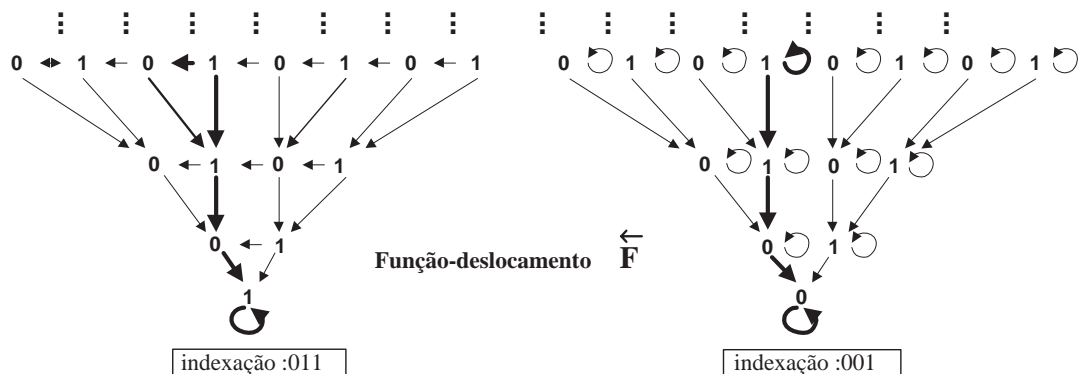


FIGURA 8.4 - Representação para Função Deslocamento Esquerda.

8.2.4 Operador $\mathbf{F}_{(01)(11)}$

Definição 8.12 Considere $x, y \in \mathbb{D}_\infty^\rightarrow$. A função que interpreta o **produto seqüencial** de processos é indicada pela expressão

$$\mathbf{F}_{(01)(11)} : \mathbb{D}_\infty^\rightarrow \prod \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow$$

e está definida por casos, conforme os itens apresentados logo a seguir.

1. Sejam $\theta_1 = \theta_2 = \theta = 00$, $x \in \pi_{n,\infty}^{\theta_1}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\rightarrow[\mathbb{D}_n]$ e $y \in \pi_{m,\infty}^{\theta_2}[\mathbb{D}_m] \subseteq \pi_{m,\infty}^\rightarrow[\mathbb{D}_m]$. Tem-se que

$$\bullet \mathbf{F}_{(01)(11)}(x \sqcap y) = \mathbf{F}_{(01)}(x) \cup \mathbf{F}_{(11)}(y).$$

Se $k = \max\{m, n\}$, $\alpha, \beta \in \{1, 2\}$ e as seguintes condições são satisfeitas:

$$(1.1) \Pi_{\infty,k}^\rightarrow(x) \in \gamma_{k-1}^0 \circ \phi_{k-1}^\alpha[\mathbb{D}_{k-1}] \quad e$$

$$(1.2) \Pi_{\infty,k}^\rightarrow(y) \in \gamma_{k-1}^0 \circ \phi_{k-1}^\beta[\mathbb{D}_{k-1}],$$

as funções $\mathbf{F}_{(01)}, \mathbf{F}_{(11)} : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow$ são respectivamente definidas por

$$\mathbf{F}_{(01)}(x) = \pi_{k,\infty}^{\theta_2} \circ (\gamma_{k-1}^{(1)} \circ \psi_{k-1}^{(0)}) \circ (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty,k}^{\theta_1})(x) = (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty,k}^{\theta_1})(x) \times \{(01 : 00)\},$$

$$\mathbf{F}_{(11)}(y) = \pi_{k,\infty}^{\theta_1} \circ (\gamma_{k-1}^{(1)} \circ \psi_{k-1}^{(1)}) \circ (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty,k}^{\theta_2})(y) = (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty,k}^{\theta_2})(y) \times \{(11 : 00)\}.$$

Senão, quando (1.1) e (1.2) não são simultâneamente satisfeitas, tem-se

$$\mathbf{F}_{(01)}(x) = (\pi_{k,\infty}^{\theta_2} \circ (\gamma_{k-1}^{(1)} \circ \psi_{k-1}^{(0)} \circ \phi_{k-1}^{(0)}) \circ \Pi_{\infty,k}^{\theta_1})(x) = \Pi_{\infty,k}^{\theta_1}(x) \times \{(001 : 00)\},$$

$$\mathbf{F}_{(11)}(y) = (\pi_{k,\infty}^{\theta_1} \circ (\gamma_{k-1}^{(1)} \circ \psi_{k-1}^{(1)} \circ \phi_{k-1}^{(0)}) \circ \Pi_{\infty,k}^{\theta_2})(y) = \Pi_{\infty,k}^{\theta_2}(y) \times \{(011 : 00)\}.$$

2. Sejam $\theta_1 = 00$, $\theta_2 = 001$ e $x \in \pi_{n,\infty}^{\theta_1}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\rightarrow[\mathbb{D}_n]$. Sejam as funções $\vec{\mathbf{F}} : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow$ apresentada na Definição 8.7 e $\mathbf{F}_{(01)} : \mathbb{D}_\infty^\rightarrow \rightarrow \mathbb{D}_\infty^\rightarrow$ definida no item anterior. Tem-se que

$$\bullet \mathbf{F}_{(01)(11)}(x \sqcap y) = \mathbf{F}_{(01)}(x) \cup \vec{\mathbf{F}}^k(y).$$

Neste caso, k indica a composição finita definida sobre $\vec{\mathbf{F}}$, logo

$$k = 2^n - 1, \text{ sempre que } \Pi_{\infty,k}^\rightarrow(x) \in \gamma_{k-1}^0 \circ \phi_{k-1}^\alpha[\mathbb{D}_{k-1}];$$

$$k = 2^n, \text{ caso contrário.}$$

3. Sejam $\theta_1 = \theta_2 = 001$. Neste caso, quando os dois argumentos do operador $\mathbf{F}_{(01)(11)}$ interpretam processos seqüenciais infinitos, tem-se que

$$\bullet \mathbf{F}_{(01)(11)}(x \sqcap y) = x.$$

4. Finalmente, quando nenhuma das situações apresentadas nos itens acima ocorrer, tem-se que

$$\bullet \mathbf{F}_{(01)(11)}(x \sqcap y) = \emptyset.$$

Observação 8.4

1. O processo resultante é indefinido, e neste caso interpretado pelo conjunto coerente $\emptyset \in \mathbb{D}_\infty^\rightarrow \amalg \mathbb{D}_\infty^\rightarrow$, nas seguintes situações:
 - as correspondentes aproximações de x e y são tais que $x_n \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n)$,
 $y_m \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m^\rightarrow)$, mas tem-se que $\theta_1, \theta_2 \in \{002, 012\}$;
 - $x = y = \emptyset$.
2. No caso em que ambos argumentos do operador $\mathbf{F}_{(01)(11)}$ interpretam processos seqüenciais infinitos, o subconjunto coerente resultante corresponde ao primeiro argumento, constituindo-se numa restrição imposta pela construção indutiva da estrutura $\mathbb{D}_\infty^\rightarrow$ do modelo de máquina geométrica que se está apresentando. Entretanto, pode-se pensar em uma interpretação mais abrangente, considerando o conjunto de interpretadores de processos definido por indução transfinita, como é o caso do espaço coerente $\mathbb{D}_{2\infty}^\rightarrow$.
3. Quando $x, y \in \mathbb{D}_\infty^\leftarrow$, a definição do subconjunto coerente $\mathbf{F}_{(01)(11)}(x \sqcap y)$ coincide em todos os itens da Definição 8.12, exceto no item 2, conforme mostra-se logo abaixo. Sejam $\theta_1 = 011$, $\theta_2 = 00$ e $y \in \pi_{m,\infty}^{\theta_1}[\mathbb{D}_m]$. Sejam as funções lineares $\overline{\mathbf{F}} : \mathbb{D}_\infty^\leftarrow \rightarrow \mathbb{D}_\infty^\leftarrow$ apresentada na Definição 8.7 e $\mathbf{F}_{(11)} : \mathbb{D}_\infty^\leftarrow \rightarrow \mathbb{D}_\infty^\leftarrow$ apresentada no primeiro item da Definição 8.12. Tem-se que

$$\bullet \mathbf{F}_{(01)(11)}(x \sqcap y) = \overline{\mathbf{F}}^k(x) \cup \mathbf{F}_{(11)}(y).$$

O número k de composições finitas sobre $\overline{\mathbf{F}}$ é dado por

$$k = 2^n - 1, \text{ sempre que } \Pi_{\infty,k}^\leftarrow(y) \in \gamma_{k-1}^0 \circ \phi_{k-1}^\beta[\mathbb{D}_{k-1}];$$

$$k = 2^n, \text{ caso contrário.}$$

Exemplificação 8.4

Nos exemplos 1–9 tem-se interpretações de processos finitos que ilustram aplicação do operador $\mathbf{F}_{(01)(11)}$. Neste caso, $:\infty = :00$. Seguem-se interpretações do produto seqüencial entre processos finitos e infinitos, em relação ao tempo computacional associado à sua execução.

1. $\mathbf{F}_{(01)(11)}(\{d_{:\infty}^{(k)}\} \sqcap \{e_{:\infty}^{(l)}\}) = \{d_{001:\infty}^{(k)}, e_{011:\infty}^{(l)}\}$
2. $\mathbf{F}_{(01)(11)}(\{\overline{d}^{(k)}_{20:\infty}\} \sqcap \emptyset) = \{\overline{d}^{(k)}_{201:\infty}\}$
3. $\mathbf{F}_{(01)(11)}(\{\overline{d}^{(k)}_{20:\infty}, \overline{e}^{(l)}_{20:\infty}\} \sqcap \{\overline{d}^{(n)}_{10:\infty}\}) = \{\overline{d}^{(k)}_{201:\infty}, \overline{e}^{(l)}_{201:\infty}, \overline{d}^{(n)}_{111:\infty}\}$
4. $\mathbf{F}_{(01)(11)}(\{\overline{d}^{(k)}_{001.20:\infty}, \overline{e}^{(k)}_{011.20:\infty}\} \sqcap \{\overline{d}^{(n)}_{101:\infty}\}) = \{\overline{d}^{(k)}_{001.201:\infty}, \overline{e}^{(k)}_{011.201:\infty}, \overline{d}^{(n)}_{101.011:\infty}\}$
5. $\mathbf{F}_{(01)(11)}(\{\overline{d}^{(k)}_{001} \overline{e}^{(k)}_{011.20:\infty}\} \sqcap \{\emptyset_{20:\infty}\}) = \{\overline{d}^{(k)}_{001} \overline{e}^{(k)}_{011.201:\infty}, \emptyset_{20.011:\infty}\}$
6. $\mathbf{F}_{(01)(11)}(\{d_{001.001:\infty}^{(k)}, d_{011.001:\infty}^{(k)}, d_{00.011:\infty}^{(k)}\} \sqcap \{e_{001.001:\infty}^{(l)}, e_{011.001:\infty}^{(l)}, e_{00.011:\infty}^{(l)}\}) =$
 $\{d_{001.001.001:\infty}^{(k)}, d_{011.001.001:\infty}^{(k)}, d_{00.011.001:\infty}^{(k)}, e_{001.001.011:\infty}^{(l)}, e_{011.001.011:\infty}^{(l)}, e_{00.011.011:\infty}^{(l)}\}$
7. $\mathbf{F}_{(01)(11)}(\{\overline{d}^{(2i)}_{10:\infty}\}_{i \in \omega} \sqcap \{\overline{d}^{(2i+1)}_{10:\infty}\}_{i \in \omega}) = \{\overline{d}^{(2i)}_{101:\infty}, \overline{d}^{(2i+1)}_{111:\infty}\}_{i \in \omega}$

8. $\mathbf{F}_{(01)(11)}(\{\overline{d_i^{(k)}}_{20:\infty}\}_{i \in I \subseteq \omega} \cap \{\overline{e_j^{(k)}}_{20:\infty}\}_{j \in J \subseteq \omega}) = \{\overline{d_i^{(k)}}_{201:\infty}, \overline{e_j^{(k)}}_{211:\infty}\}_{i \in I, j \in J}$
9. $\mathbf{F}_{(01)(11)}(\{\overline{d^{(n)}}_{10:\infty}\}_{n \in \omega} \cap \{d_{:\infty}^{(k)}\}) = \{\overline{d^{(n)}}_{101:\infty}\}_{n \in \omega} \cup \{d_{011:\infty}^{(k)}\} =$
 $\{\overline{d^{(0)}}_{101:\infty}, \overline{d^{(1)}}_{101:\infty}, \dots, \overline{d^{(k)}}_{101:\infty}, \dots, d_{011:\infty}^{(k)}\}$
10. $\mathbf{F}_{(01)(11)}(\{\overline{d^{(0)}}_{101:\infty}, \overline{d^{(1)}}_{101:\infty}\} \cap \{\overline{d^{(k)}}_{10:\infty}\}) = \{\overline{d^{(0)}}_{101.001:\infty}, \overline{d^{(1)}}_{101.001:\infty}, \overline{d^{(k)}}_{10.011:\infty}\}$
11. $\mathbf{F}_{(01)(11)}(\{d_{:00}^{(k)}\} \cap \emptyset) = \{d_{:001}^{(k)}\}$
12. $\mathbf{F}_{(01)(11)}(\{d_{:00}^{(k)}\} \cap \{e_{:001}^{(l)}\}) = \{d_{:001}^{(k)}, e_{011:001}^{(l)}\}$
13. $\mathbf{F}_{(01)(11)} = (\{\overline{d^{(n)}}_{10:00}\}_{n \in \omega} \cap \{e_n^{(0)}_{201:001}\}_{n \in \omega}) = \{\overline{d^{(n)}}_{101:001}, \overline{e_n^{(0)}}_{211:001}\}_{n \in \omega}$
14. $\mathbf{F}_{(01)(11)} = (\{\overline{d^{(n)}}_{10:00}\}_{n \in \omega} \cap \{d_{:001}^{(k)}\}) = \{\overline{d^{(n)}}_{101:001}\}_{n \in \omega} \cup \{d_{011:001}^{(k)}\} =$
 $\{\overline{d^{(0)}}_{101:001}, \overline{d^{(1)}}_{101:001}, \dots, \overline{d^{(k)}}_{101:001}, \dots, d_{011:001}^{(k)}\}$

A seguir consideram-se interpretações nos domínios $\mathbb{D}_{\infty}^{\leftarrow}$.

15. $\mathbf{F}_{(01)(11)}(\{d_{:011}^{(k)}\} \cap \emptyset) = \{d_{:011}^{(k)}\}$
16. $\mathbf{F}_{(01)(11)}(\{d_{:011}^{(k)}\} \cap \{e_{:00}^{(l)}\}) = \{d_{001:011}^{(k)}, e_{:011}^{(l)}\}$
17. $\mathbf{F}_{(01)(11)} = (\{\overline{d^{(n)}}_{111:011}\}_{n \in \omega} \cap \{e_n^{(0)}_{20:00}\}_{n \in \omega}) = \{\overline{d^{(n)}}_{101:011}, \overline{e_n^{(0)}}_{211:001}\}_{n \in \omega}$
18. $\mathbf{F}_{(01)(11)} = (\{\overline{d^{(n)}}_{111:011}\}_{n \in \omega} \cap \{d_{:00}^{(k)}\}) = \{\overline{d^{(n)}}_{101:011}\}_{n \in \omega} \cup \{d_{:011}^{(k)}\} =$
 $\{\overline{d^{(0)}}_{101:011}, \overline{d^{(1)}}_{101:011}, \dots, \overline{d^{(k)}}_{101:011}, \dots, d_{:011}^{(k)}\}$

8.2.5 Operador $\mathbf{F}_{(02)(12)(22)}$

Definição 8.13 Considere $x, y \in \mathbb{D}_{\infty}^{\rightarrow}$ tais que $x = \bigcup_{n \in \omega}^{\uparrow} \{x_n\}$ e $y = \bigcup_{m \in \omega}^{\uparrow} \{y_m\}$ onde $x_n \in \pi_{n,\infty}^{\theta_1}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ e $y_m \in \pi_{m,\infty}^{\theta_2}[\mathbb{D}_m] \subseteq \pi_{m,\infty}^{\rightarrow}[\mathbb{D}_m]$. A função

$$\mathbf{F}_{(02)(12)(22)} : \mathbb{D}_{\infty}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$$

que interpreta a **soma determinística** de processos é definida por casos, conforme os itens apresentados logo a seguir.

1. Sejam $\theta_1 = \theta_2 = \theta = 00$. Tem-se

$$\bullet \mathbf{F}_{(02)(12)(22)}(x \sqcap y \sqcap b) = \mathbf{F}_{(02)}(x) \cup \mathbf{F}_{(12)}(y) \cup \mathbf{F}_{(22)}(b).$$

Se $k = \max\{m, n\}$, $\alpha, \beta \in \{0, 1\}$ e as seguintes condições são satisfeitas:

$$(1.1) \quad \Pi_{\infty, k}(x) \in \gamma_{k-1}^0 \circ \phi_{k-1}^{\alpha}[\mathbb{D}_{k-1}] \quad e$$

$$(1.2) \quad \Pi_{\infty, k}(y) \in \gamma_{k-1}^0 \circ \phi_{k-1}^{\beta}[\mathbb{D}_{k-1}],$$

sendo as funções $\mathbf{F}_{(02)}, \mathbf{F}_{(12)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ respectivamente definidas por

$$\mathbf{F}_{(02)} = \pi^{\theta_2} \circ \gamma_{k-1}^{(2)} \circ \psi_{k-1}^{(0)} \circ (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty, k}^{\theta_1})(x) = (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty, k}^{\theta_1})(x) \times \{(02 : 00)\},$$

$$\mathbf{F}_{(12)} = \pi^{\theta_1} \circ \gamma_{k-1}^{(2)} \circ \psi_{k-1}^{(1)} \circ (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty, k}^{\theta_2})(y) = (\Gamma_{k-1}^{(0)} \circ \Pi_{\infty, k}^{\theta_2})(y) \times \{(12 : 00)\}.$$

Senão, quando (1.1) e (1.2) não são simultâneamente satisfeitas,

$$\mathbf{F}_{(02)} = \pi^{\theta_2} \circ \gamma_{k-1}^{(2)} \circ \psi_{k-1}^{(0)} \circ \phi_{k-1}^{(0)} \circ \Pi_{\infty,k}^{\theta_1}(x) = \Pi_{\infty,k}^{\theta_1}(x) \times \{(002 : 00)\},$$

$$\mathbf{F}_{(02)} = \pi^{\theta_1} \circ \gamma_{k-1}^{(2)} \circ \psi_{k-1}^{(1)} \circ \phi_{k-1}^{(0)} \circ \Pi_{\infty,k}^{\theta_2}(y) = \Pi_{\infty,k}^{\theta_2}(y) \times \{(012 : 00)\},$$

sendo que $\mathbf{F}_{(22)} : \mathbb{B} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ é dada por $\mathbf{F}_{(22)}(b) = b \times \{(22 : 00)\}$.

2. Sejam $\theta_1 = 00$, $\theta_2 = 002$. Considere a função deslocamento $\mathbf{F}^{\downarrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ apresentada na Definição 8.7 e as funções $\mathbf{F}_{(02)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$, $\mathbf{F}_{(22)} : \mathbb{B} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ apresentadas no item anterior. Tem-se

$$\bullet \mathbf{F}_{(02)(12)(22)}(x \sqcap y \sqcap b) = \mathbf{F}_{(02)}(x) \cup \mathbf{F}^{\downarrow k}(y) \cup \mathbf{F}_{(22)}(b).$$

Neste caso, k indica a composição finita definida sobre \mathbf{F}^{\downarrow} , logo

$$k = 2^n - 1, \text{ sempre que } \Pi_{\infty,k}(x) \in \gamma_{k-1}^0 \circ \phi_{k-1}^{\alpha} [\mathbb{D}_{k-1}];$$

$$k = 2^n, \text{ caso contrário.}$$

3. Sejam $\theta_1 = \theta_2 = 002$. Neste caso, os dois argumentos do operador $\mathbf{F}_{(02)(12)(22)}$ interpretam somas determinísticas infinitas. Tem-se

$$\bullet \mathbf{F}_{(02)(12)(22)}(x \sqcap y \sqcap b) = x.$$

4. Finalmente, quando nenhuma das situações apresentadas nos itens acima ocorrer, tem-se

$$\bullet \mathbf{F}_{(02)(12)(22)}(x \sqcap y \sqcap b) = \emptyset.$$

Observação 8.5

1. O processo resultante é indefinido, e neste caso interpretado pelo conjunto coerente $\emptyset \in \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}$, nas seguintes situações:

$$\bullet \text{ as correspondentes aproximações de } x \text{ e } y \text{ são tais que } x_n \in \pi_{n,\infty}^{\theta_1}(\mathbb{D}_n) \subseteq \pi_{n,\infty}^{\rightarrow}(\mathbb{D}_n),$$

$$y_m \in \pi_{m,\infty}^{\theta_2}(\mathbb{D}_m) \subseteq \pi_{m,\infty}^{\rightarrow}(\mathbb{D}_m), \text{ mas tem-se que } \theta_1 = \theta_2 = 011;$$

$$\bullet x = y = \emptyset.$$

2. No caso em que ambos argumentos do operador $\mathbf{F}_{(02)(12)(22)}$ interpretam processos seqüenciais infinitos, o subconjunto coerente resultante corresponde ao primeiro argumento, constituindo-se numa restrição imposta pela construção indutiva da estrutura $\mathbb{D}_{\infty}^{\rightarrow}$ do modelo de máquina geométrica que se está definindo. Entretanto, pode-se buscar interpretações no espaço coerente $\mathbb{D}_{2\infty}^{\rightarrow}$ de processos transfinitos, onde é possível modelar a composição infinita da soma determinística de processos infinitos.

3. A correspondente definição da soma determinística sobre interpretações do domínio $\mathbb{D}_{\infty}^{\leftarrow}$ é análoga em todos os casos, exceto no segundo, apresentado logo a seguir.

Sejam $\theta_1 = 011$, $\theta_2 = 00$. Considere a função deslocamento $\mathbf{F}^{\uparrow} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ apresentada na Definição 8.7 e as funções $\mathbf{F}_{(12)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$, $\mathbf{F}_{(22)} : \mathbb{B} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$

apresentada na Definição 8.13. Tem-se

$$\bullet \mathbf{F}_{(02)(12)(22)}(x \sqcap y) = \mathbf{F}^{\uparrow k}(x) \cup \mathbf{F}_{(12)}(y) \cup \mathbf{F}_{(22)}(b).$$

Nesta situação, o número k de composições finitas sobre \mathbf{F}^{\uparrow} é expresso por

$$k = 2^n - 1, \text{ sempre que } \Pi_{\infty, k}(y) \in \gamma_{k-1}^0 \circ \phi_{k-1}^\beta [\mathbb{D}_{k-1}];$$

$$k = 2^n, \text{ caso contrário.}$$

Veja a aplicação do construtor $\mathbf{F}_{(02)(12)(22)}$ nos próximos exemplos.

Exemplificação 8.5

Seja : $\theta = : 00$.

1. $\mathbf{F}_{(02)(12)(22)}(\{\overline{d^{(k)}}_{20:\infty}, \overline{e^{(k)}}_{20:\infty}\} \sqcap \{d^{(k)}_{:\infty}\} \sqcap \emptyset) = \{\overline{d^{(k)}}_{202:\infty}, \overline{e^{(k)}}_{202:\infty}, d^{(k)}_{012:\infty}\}$
2. $\mathbf{F}_{(02)(12)(22)}(\{\overline{d^{(k)}}_{20:\infty}, \overline{e^{(k)}}_{20:\infty}\} \sqcap \{\overline{d^{(n)}}_{10:\infty}, \overline{d^{(l)}}_{10:\infty}\} \sqcap \{b\}) =$
 $\{\overline{d^{(k)}}_{202:\infty}, \overline{e^{(k)}}_{202:\infty}, \overline{d^{(n)}}_{112:\infty}, \overline{d^{(l)}}_{112:\infty}, b_{22:\infty}\}$
3. $\mathbf{F}_{(02)(12)(22)}(\mathbf{F}_{(01)(11)}(\{d^{(k)}_{:\infty}\} \sqcap \{e^{(k)}_{:\infty}\}) \sqcap \mathbf{F}_{(10)}(\{\overline{d^{(n)}}_{:\infty}\} \sqcap \{\overline{e^{(l)}}_{:\infty}\}) \sqcap \emptyset) =$
 $\{d^{(k)}_{001.002:\infty}, e^{(k)}_{001.002:\infty}, \overline{d^{(n)}}_{10.012:\infty}, \overline{e^{(l)}}_{10.012:\infty}\}$
4. $\mathbf{F}_{(02)(12)(22)}(\{\overline{d^{(k)}}_{001} e^{(k)}_{011.20:\infty}\} \sqcap \{\overline{d^{(n)}}_{20:\infty}, \emptyset_{20:\infty}\}) =$
 $\{\overline{d^{(k)}}_{001} e^{(k)}_{011.202:\infty}, \overline{d^{(n)}}_{20.012:\infty}, \emptyset_{20.012:\infty}\}$

Seja : $\infty = : 002$.

5. $\mathbf{F}_{(02)(12)(22)}(\{d^{(k)}_{:\infty}\} \sqcap \emptyset \sqcap \emptyset) = \{\{d^{(k)}_{002:\infty}\} = \{d^{(k)}_{:\infty}\}$
6. $\mathbf{F}_{(02)(12)(22)}(\{d^{(k)}_{:\infty}\} \sqcap \{e^{(l)}_{:\infty}\} \sqcap \emptyset) = \{d^{(k)}_{:\infty}, e^{(l)}_{012:\infty}\}$
7. $\mathbf{F}_{(02)(12)(22)} = (\{d^{(n)}_{(00)^n.012:\infty}\}_{n \in \omega} \sqcap \emptyset \sqcap \{b_{22:\infty}^{(n)}\}) = \{d^{(n)}_{(00)^n.012:\infty}, b_{22:\infty}^{(n)}\}_{n \in \omega}$
8. $\mathbf{F}_{(02)(12)(22)}(\{d^{(k)}_{:00}\} \sqcap \emptyset) = \{d^{(k)}_{:002}\}$
9. $\mathbf{F}_{(02)(12)(22)}(\{d^{(k)}_{:00}\} \sqcap \{e^{(l)}_{:002}\}) = \{d^{(k)}_{:002}, e^{(l)}_{012:002}\}$
10. $\mathbf{F}_{(02)(12)(22)} = (\{\overline{d^{(n)}}_{10:00}\}_{n \in \omega} \sqcap \{\overline{e_n^{(0)}}_{201.002}\}_{n \in \omega}) =$
 $\{\overline{d^{(n)}}_{10.002:002}, \overline{e_n^{(0)}}_{201.012:002}\}_{n \in \omega}$
11. $\mathbf{F}_{(02)(12)(22)} = (\{\overline{d^{(n)}}_{10:00}\}_{n \in \omega} \sqcap \{d^{(k)}_{:002}\}) = \{\overline{d^{(n)}}_{10.002:002}\}_{n \in \omega} \cup \{d^{(k)}_{012:002}\} =$
 $\{\overline{d^{(0)}}_{10.002:002}, \overline{d^{(1)}}_{10.002:002}, \dots, \overline{d^{(k)}}_{10.002:002}, \dots, d^{(k)}_{012:002}\}$

Sejam as interpretações no domínio $\mathbb{D}_{\infty}^{\leftarrow}$ e : $\infty = : 002$.

12. $\mathbf{F}_{(02)(12)(22)}(\{\emptyset \sqcap d^{(k)}_{:012}\}) = \{d^{(k)}_{:012}\}$

13. $\mathbf{F}_{(02)(12)(22)}(\{d_{:012}^{(k)}\} \sqcap \{e_{:00}^{(l)}\}) = \{d_{002:012}^{(k)}, e_{:012}^{(l)}\}$
14. $\mathbf{F}_{(02)(12)(22)} = (\{\overline{d^{(n)}}_{112:012}\}_{n \in \omega} \sqcap \{\overline{e_n^{(0)}}_{20:00}\}_{n \in \omega}) = \{\overline{d^{(n)}}_{112:012}, \overline{e_n^{(0)}}_{102:012}\}_{n \in \omega}$
15. $\mathbf{F}_{(01)(11)} = (\{\overline{d^{(n)}}_{112:012}\}_{n \in \omega} \sqcap \{d_{:00}^{(k)}\}) = \{\overline{d^{(n)}}_{102:012}\}_{n \in \omega} \cup \{d_{:012}^{(k)}\} =$
 $\{\overline{d^{(0)}}_{102:012}, \overline{d^{(1)}}_{102:012}, \dots, \overline{d^{(k)}}_{102:012}, \dots, d_{:012}^{(k)}\}$

Proposição 8.8 *Os construtores $\mathbf{F}_{(01)(11)}$ e $\mathbf{F}_{(02)(12)(22)}$ são morfismos na categoria $\mathit{CospLin}$.*

A prova será omitida mas pode ser construída da mesma forma que àquela da Proposição 8.4.

8.3 Composição dos Construtores no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$

Na primeira seção deste capítulo, mostrou-se que os construtores de processos modelados neste trabalho são funções lineares. Portanto os construtores de processos são interpretados por morfismos na categoria $\mathit{CospLin}$. O conjunto de todos estes morfismos será, a partir de agora, indicado por

$$\mathcal{FLin} = \{Id_{\mathbb{D}_{\infty}}, \mathbf{F}_{(00)}, \mathbf{F}_{(10)}, \mathbf{F}_{(20)}, \mathbf{F}_{(01)(11)}, \mathbf{F}_{(02)(21)(22)}\},$$

e seus elementos foram apresentados nas Definições 8.1, 8.2, 8.3, 8.4, 8.12 e 8.13, respectivamente.

Neste sentido, sempre que a composição entre funções que interpretam construtores de processos estiver bem definida a função resultante será também um morfismo que dá interpretação ao novo processo construído, seja tal objeto interpretado em $\mathbb{D}_{\infty}^{\rightarrow}$ ou em $\mathbb{D}_{\infty}^{\leftarrow}$.

As próximas definições adotam a notação sugerida em [ESC 95].

Considere agora a seqüência $\{F_i\}_{i \in \omega}$ de operadores lineares definida de tal modo que $\text{cod}(F_i) = \text{dom}(F_{i+1})$, sempre que $F \in \mathcal{FLin}$. Neste caso, sempre que $X_i = \text{dom}(F_i)$ tem-se $F_i : X_i \rightarrow X_{i+1}$, e $X_i \in \{\mathbb{D}_{\infty}^{\rightarrow}, \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow}, \mathbb{D}_{\infty}^{\rightarrow} \prod_{\mathbb{B}} \mathbb{D}_{\infty}^{\rightarrow}\}$.

Definição 8.14 *Uma composição finita da seqüência $\{F_i\}_{i \in \omega}$ é uma função linear indicada por $F_0^n : X_0 \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ e definida pela expressão*

$$F_0^n \equiv \odot_{i=0}^n F_i = \odot_{i=m+1}^n F_i \circ F_m \circ \dots \circ F_1 \circ F_0, \text{ e } F_0^0 = Id_{\mathbb{D}_{\infty}^{\rightarrow}}.$$

No caso em que a seqüência $\{F_i\}_{i \in \omega}$ é constante tem-se que $F^n \equiv \odot_{i=0}^n F$.

A composição entre morfismos em \mathcal{FLin} é fechada em $\mathbb{D}_{\infty}^{\rightarrow}$, portanto a aplicação destes morfismos sobre subconjuntos coerentes resulta sempre em outro subconjunto coerente, independentemente da ordem em que se constrói a seqüência $\{F_i\}_{i \in \omega}$, podendo ocorrer, inclusive, que cada ordenação gere um novo objeto em $\mathbb{D}_{\infty}^{\rightarrow}$.

Definição 8.15 *Considere $F_i \in \mathcal{FLin}$. Uma composição infinita dos morfismos da seqüência $\{F_i\}_{i \in \omega}$ é uma função linear indicada por $F_0^{\infty} : X_0 \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ e definida pela expressão*

$$F_0^{\infty} \equiv \odot_{i=0}^{\infty} F_i = \odot_{i=n+1}^{\infty} F_i \circ F_n \circ \dots \circ F_1 \circ F_0 = \odot_{i=n+1}^{\infty} F_i \circ \odot_{i=0}^n F_i.$$

Quando a seqüência $\{F_i\}_{i \in \omega}$ é constante tem-se também que $F^{\infty} \equiv \odot_{n=0}^{\infty} F$.

O próximo parágrafo evidencia uma condição importante para a composição infinita de morfismos em $\mathbb{D}_{\infty}^{\rightarrow}$, relacionada com a interpretação de processos que não possuem uma execução em tempo finito.

Pela Proposição 6.7, que formalmente define os objetos que completam o espaço dos processos, a composição infinita quando definida por morfismos $F_i \in \mathcal{FLin}$ deve ser considerada na ordem apresentada acima. Neste sentido, a composição dada por $F_0 \circ F_1 \circ \dots \circ F_n \circ \odot_{i=n+1}^{\infty} F_i$ não é um morfismo em $\mathbb{D}_{\infty}^{\rightarrow}$.

Com base nas Definições 8.14 e 8.15, a linguagem apresentada no Capítulo 9 suporta operadores recursivos com argumentos (possivelmente finitos e infinitos) e cada composição resulta na definição (recursiva) de um novo construtor de processos. Por conseqüência, é possível também determinar as soluções (pela existência do menor ponto fixo) para equações definidas a partir dos construtores interpretados.

A composição infinita de morfismos, conforme indicada acima, é fechada em $\mathbb{D}_{\infty}^{\rightarrow}$. Nos próximos exemplos mostra-se que as soluções de equações recursivas construídas pela composição infinita destes morfismos são interpretadas por objetos em $\mathbb{D}_{\infty}^{\rightarrow}$. Na primeira exemplificação, considera-se o operador linear $\mathbf{F}_{(01)(11)}$, que interpreta o produto seqüencial. Entretanto, os exemplos podem ser facilmente redefinidos para exemplificar aplicações do operador $\mathbf{F}_{(02)(12)(22)}$ representando o construtor de somas determinísticas.

Exemplificação 8.6

Esta exemplificação é uma aplicação do teorema do ponto fixo em $\mathbb{D}_{\infty}^{\rightarrow}$. Consideram-se o processo elementar $d^{(k)} \in \mathcal{A}_k$ onde $d \in \mathcal{A}$, o indexador k em I enumerável, a Tabela 8.1 e os Exemplos 6.2 e 6.4.

- Retornando a Exemplificação 6.2 apresentada ao final da seção 6, mostra-se que cada subconjunto da seqüência $\{\pi_{n,\infty}^{(00)} w_n\}_{n \in \omega}$ é expresso como imagem do operador $\mathbf{F}_{(01)(11)}$. Para tal, define-se a seqüência $v_{n+1} = \mathbf{F}_{(01)(11)}(v_n \sqcap v'_0)$, onde $v'_0 = v_0 = \{d_{:00}^{(k)}\}$ é definida. Tem-se então

$$\begin{aligned} \bullet v_1 &= x'_0 \cup x''_0 = \{d_{001:00}^{(k)}, d_{011:00}^{(k)}\}, \\ \bullet v_2 &= x'_1 \cup x''_1 \cup x'''_1 = \{d_{001.001:00}^{(k)}, d_{011.001:00}^{(k)}, d_{00.011:00}^{(k)}\}, \\ &\vdots \\ \bullet v_n &= \bigcup_{i=0}^{n-1} x^{(i)} = \{d_{(001)^n:00}^{(k)}, d_{011.(001)^{n-1}:00}^{(k)}, \dots, d_{(00)^{n-1}.011:00}^{(k)}\}. \end{aligned}$$

A Figura 6.4 representa o processo $W00$ interpretado por v_n , onde

$$\Pi_n \circ \Pi_{\infty, n+1}^{\rightarrow}(v_{n+1}) = \Pi_{\infty, n}^{\rightarrow}(v_n), \quad \Pi_{\infty, n}^{\rightarrow}(v_n) = w_n, \quad \text{e} \quad \pi_{n,\infty}^{(00)}(w_n) = v_n.$$

- A seguir, salientam-se algumas importantes considerações resultantes da observação da primeira coluna da tabela apresentada na Figura 8.1.

• $x'_0 = \{d_{:00}^{(k)}\}$ interpreta um processo elementar que executa a ação d na k -ésima posição do espaço geométrico I .

• $\{d_{:001}^{(k)}\}$ interpreta um processo resultante da composição infinita da função $\mathbf{F}_{(01)(11)}$ que interpreta um produto seqüencial. Sendo assim, o único termo conhecido deste produto seqüencial é o primeiro fator e corresponde ao processo elementar que executa a ação d na k -ésima posição do espaço geométrico I .

- No Capítulo 9 formaliza-se uma linguagem capaz de expressar os processos interpretados em $\mathbb{D}_{\infty}^{\rightarrow}$. Antecipando um pouco estas formalizações, tem-se

• $d^{(k)}, d^{(k)} \cdot _ , \dots, d^{(k)} \cdot _2^n$ como as expressões que identificam os processos finitos interpretado por $\{d_{:00}^{(k)}\}, \{d_{001:00}^{(k)}\}, \dots, \{d_{(001)^n:00}^{(k)}\} \in \mathbb{D}_{\infty}^{\rightarrow}$, respectivas.

TABELA 8.1 - Composição Infinita do PS de Processos Elementares.

$\bullet x'_{n+1} = \mathbf{F}_{(01)(11)}(x'_n \sqcap \emptyset)$ $x'_0 = \{d_{001:00}^{(k)}\}$ $x'_1 = \{d_{(001)^2:00}^{(k)}\}$ \vdots $x'_n = \{d_{(001)^n:00}^{(k)}\}$ \vdots	$\bullet x''_{n+1} = \mathbf{F}_{(01)(11)}(x''_n \sqcap \emptyset) \quad \dots$ $x''_0 = \{d_{011:00}^{(k)}\}$ $x''_1 = \{d_{011.001:00}^{(k)}\}$ \vdots $x''_n = \{d_{011.(001)^n:00}^{(k)}\}$ \vdots	$\bullet x_{n+1}^{(i+2)} = \mathbf{F}_{(01)(11)}(x_n^{(i)} \sqcap \emptyset)$ $x_0^{(i+1)} = \{d_{(00)^i.011:00}^{(k)}\}$ $x_1^{(i+2)} = \{d_{(00)^i.011.001:00}^{(k)}\}$ \vdots $x_n^{(i+2)} = \{d_{(00)^i.011.(001)^n:00}^{(k)}\}$ \vdots
$\bullet x' = \mathbf{F}_{(01)(11)}(x' \sqcap \emptyset)$ \downarrow $x' = \{d_{:001}^{(k)}\}$	$\bullet x'' = \mathbf{F}_{(01)(11)}(x'' \sqcap \emptyset) \quad \dots$ \downarrow $x'' = \{d_{011:001}^{(k)}\}$	$\bullet x^{(i+2)} = \mathbf{F}_{(01)(11)}(x^{(i)} \sqcap \emptyset)$ \downarrow $x^{(i+2)} = \{d_{(00)^i.011:001}^{(k)}\}$

$\bullet d^{(k)} \cdot \infty$ indica o processo infinito interpretado pelo conjunto $x' = \{d_{:001}^{(k)}\}$, transcrevendo na linguagem a solução para a equação

$$\odot_{i=0}^{\infty} \mathbf{F}_{(01)(11)}(\{d_{(001)^i:00}^{(k)}\} \sqcap \emptyset)$$

apresentada na primeira linha da Tabela 8.1.

As considerações anteriores foram alcançadas observando a primeira coluna da Tabela 8.1. Pela observação das demais colunas desta tabela, construídas da mesma maneira que a primeira coluna e identificadas pela letra i , são analisadas algumas generalizações relacionadas com estas considerações.

1. Para cada i define-se uma seqüência $\{z_n^{(i)}\}_{n \in \omega} \subseteq \mathbb{D}_n$ tal que

$$z_n^{(i)} = \Pi_{\infty, n}^{\rightarrow}(x_n^{(i)}) = \Pi_n \circ \Pi_{\infty, n+1}^{\rightarrow}(x_{n+1}^{(i)}).$$

2. Pela completção tem-se que $x^{(i)} = \bigcup_{n=0}^{\infty} z_n^{(i)} \in \mathbb{D}_{\infty}^{\rightarrow}$. Verifica-se ainda que cada $x^{(i)}$ satisfaz a igualdade $x^{(i)} = \mathbf{F}_{(01)(11)}(x^{(i)} \sqcap \emptyset)$. Logo, cada subconjunto $x^{(i)} = \{d_{(00)^i.011:001}^{(k)}\}$ é um ponto fixo para a equação $x = \mathbf{F}_{(01)(11)}(x \sqcap \emptyset)$.

Neste caso, considerando-se como ordem de informação a inclusão, o conjunto coerente vazio corresponde ao menor ponto fixo, ou seja, $\emptyset = \mathbf{F}_{(01)(11)}(\emptyset \sqcap \emptyset)$.

3. Além disso, $\forall i, j$ tem-se que $\mathbf{F}_{(01)(11)}(x^{(i)} \sqcap x^{(j)}) = x^{(i)} \cup x^{(j)} \in \mathbb{D}_{\infty}^{\rightarrow}$, portanto $x = \bigcup x^{(i)} \in \mathbb{D}_{\infty}^{\rightarrow}$. Assim, é imediato que x e todas suas aproximações finitas são também soluções para esta equação. Entretanto, não se pode afirmar que x constitui-se no maior ponto fixo para a equação apresentada, e o próximo exemplo justifica isto.

4. A Figura 6.5 apresentada na Exemplificação 6.2 representa o processo parcial interpretado por $\bigcup_{i=0}^n x^{(i)} = \pi_{n,\infty}^{(001)}(w_n) = \pi_{n,\infty}^{\rightarrow}(w_n)$.

5. Por fim, considere uma nova notação, na qual $f(x) = \mathbf{F}_{(01)(11)}(x \sqcap \emptyset)$. Neste caso, $f^{(\infty)}$ indica a composição infinita de f e as seguintes igualdades são facilmente comprovadas na tabela da Figura 8.1

$$\begin{aligned} x' &= f^{(\infty)}(x'_0) = f^{(\infty)} \circ f^{(n)}(x'_0) = f^{(\infty)}(x'_n) = \dots = f(x') \text{ e} \\ x' &= f^{(\infty)}(x'_0) = f^{(n)} \circ f^{(\infty)}(x'_0) = \dots = f^2 \circ f^{(\infty)}(x'_0) = f \circ f^{(\infty)}(x'_0) \end{aligned}$$

Esta exemplificação ilustra a aplicação do Teorema do Ponto Fixo e tem-se que $f^{(\infty)}$ é essencialmente o ponto fixo de f .

Exemplificação 8.7

Esta exemplificação está dividida em duas partes sendo que ambas podem ser consideradas como uma observação mais detalhada da Exemplificação 6.1.

- O processo Q apresentado no Exemplo 6.1 executa, simultâneamente, em *lutc*, a ação $d \in \mathcal{A}$ em todas as posições de memória. Agora, retorna-se a ele para mostrar a expressão recursiva que identifica sua interpretação. Para tal, seja I o conjunto enumerável de indexadores, $d \in \mathcal{A}$ e a seqüência $\{x_{k+1}\}_{k \in I}$ dada pela expressão $x_{k+1} = \mathbf{F}_{(10)}(\{\overline{d^{(k)}}_{10:00}\} \sqcap x_k)$, onde $x_0 = \emptyset$. Assim, esta seqüência tem os seguintes subconjuntos coerentes como elementos

$$\begin{aligned} \bullet x_1 &= \{\overline{d^{(0)}}_{10:00}\}, \\ \bullet x_2 &= \{\overline{d^{(0)}}_{10:00}, \overline{d^{(1)}}_{10:00}\}, \\ &\vdots \\ \bullet x_{k+1} &= \{\overline{d^{(0)}}_{10:00}, \overline{d^{(1)}}_{10:00}, \overline{d^{(2)}}_{10:00}, \dots, \overline{d^{(k)}}_{10:00}\} = \bigcup_{n=0}^k \overline{d^{(n)}}_{10:00}. \end{aligned}$$

Neste caso, considerando as inclusões $x_k \subseteq x_{k+1}, \forall k \in I$, tem-se que

$$\begin{aligned} \bullet x &= \bigcup^\uparrow x_k = \{\overline{d^{(k)}}_{10:00}\}_{k \in I} = \{\overline{d^0}_{10:00}, \overline{d^1}_{10:00}, \dots, \overline{d^k}_{10:00}, \dots\}. \\ \bullet \parallel d^{(0)}, d^{(1)}, \dots, d^{(k)}, d^{(k+1)}, \dots \parallel &\text{ indica o processo interpretado por } x. \end{aligned}$$

Salienta-se ainda que $x \in \pi_{1,\infty}^{(00)}[\mathbb{D}_1] \subseteq \mathbb{D}_\infty^{\rightarrow}$ significando que, já em \mathbb{D}_1 , está garantida a existência de solução para a equação $x = \mathbf{F}_{(10)}(\{\overline{d^{(k)}}_{10:00}\} \sqcap x)$ a qual é inserida em \mathbb{D}_∞ pela completação. Além disso, pode-se determinar as soluções para outras equações do mesmo tipo, no caso,

$$x_{k+1} = \mathbf{F}_{(10)}(\{\overline{d_{(00)^n, 10:00}^{(k)}}\} \sqcap x_k) \text{ e } x_0 = \emptyset.$$

As primeiras linhas da Tabela 8.2 correspondem a aplicação do operador $\mathbf{F}_{(01)(11)}$ que modela o produto seqüencial sobre as soluções comentadas neste parágrafo. Assim, tem-se que

$$\mathbf{F}_{(01)(11)}(x \sqcap \emptyset) = \bigcup_{k \in I} \overline{d^{(k)}}_{101:00} = \{d_{101:00}^{(k)}\}_I.$$

- Na Tabela 8.2 apresentada logo a seguir, consideram-se cada token como conjunto e cada subconjunto coerente como uma família de conjuntos. Cada coluna apresenta as projeções -imersões e na última linha o correspondente limite.

Antes de uma análise mais abrangente, salientam-se características importantes da interpretação dos processos envolvidos.

Os processos abaixo são produtos seqüenciais parciais, pois se conhece apenas alguns de seus infinitos termos.

TABELA 8.2 - Composição Infinita do PS de Processos Paralelos.

$\bullet y'_{n+1} = \mathbf{F}_{(01)(11)}(y'_n \sqcap \emptyset)$ $y'_0 = \{\overline{d^{(k)}}_{101:00}\}_I$ $y'_1 = \{\overline{d^{(k)}}_{101.001:00}\}_I$ \vdots $y'_n = \{\overline{d^{(k)}}_{101.(001)^n:00}\}_I$ \vdots	$\bullet y''_{n+1} = \mathbf{F}_{(01)(11)}(y''_n \sqcap \emptyset) \quad \dots$ $y''_0 = \{\overline{d^{(k)}}_{111:00}\}_I$ $y''_1 = \{\overline{d^{(k)}}_{111.001:00}\}_I$ \vdots $y''_n = \{\overline{d^{(k)}}_{111.(001)^n:00}\}_I$ \vdots	$\bullet y_n^{(i+1)} = \mathbf{F}_{(01)(11)}(y_n^{(i+1)} \sqcap \emptyset)$ $y_0^{(i+1)} = \{\overline{d_{(00)^i}^{(k)}}_{.111:00}\}_I$ $y_1^{(i+1)} = \{\overline{d_{(00)^i}^{(k)}}_{.111.001:00}\}_I$ \vdots $y_n^{(i+1)} = \{\overline{d_{(00)^i}^{(k)}}_{.111.(001)^n:00}\}_I$ \vdots
$\bullet y' = \mathbf{F}_{(01)(11)}(y' \sqcap \emptyset)$ \downarrow $y' = \{\overline{d^{(k)}}_{101:001}\}_I$	$\bullet y'' = \mathbf{F}_{(01)(11)}(y'' \sqcap \emptyset)$ \downarrow $y'' = \{\overline{d^{(k)}}_{111:001}\}_I$	$\bullet y^{(i+1)} = \mathbf{F}_{(01)(11)}(y^{(i+1)} \sqcap \emptyset)$ \downarrow $y^{(i+1)} = \{\overline{d_{(00)^i}^{(k)}}_{.111:001}\}_I$

• $y'_0 = \{\overline{d^{(k)}}_{101:00}\}_{k \in I} = \{\overline{d^0}_{101:00}, \overline{d^1}_{101:00}, \dots, \overline{d^k}_{101:00}, \dots\}$ interpreta o produto seqüencial onde o primeiro fator executa a ação d em todas as posições do espaço I . Sua expressão é $(\| d^{(0)}, d^{(1)}, \dots, d^{(k)}, d^{(k+1)}, \dots \|) \cdot \underline{\infty}$.

• $y''_0 = \{\overline{d^{(k)}}_{111:001}\}_{k \in I} = \{\overline{d^0}_{111:001}, \overline{d^1}_{111:001}, \dots, \overline{d^k}_{111:001}, \dots\}$ interpreta um produto seqüencial cujo segundo fator é o processo que executa a ação d em todas as posições do espaço I . y''_0 é indicado na linguagem, pela expressão

$$(_ \cdot \| d^{(0)}, d^{(1)}, \dots, d^{(k)}, d^{(k+1)}, \dots \|) \cdot \underline{\infty}.$$

• $y_0^{(i+1)} = \{\overline{d_{(00)^i}^{(k)}}_{.111:00}\}_{k \in I} = \{\overline{d_{(00)^i}^0}_{.111:00}, \overline{d_{(00)^i}^1}_{.111:00}, \dots, \overline{d_{(00)^i}^{(k)}}_{.111:00}, \dots\}$ interpreta um produto seqüencial no qual o i -ésimo termo executa a ação d em todas as posições do espaço I . $y_0^{(i+1)}$ é indicado na linguagem, pela expressão

$$(_2^i \cdot \| d^{(0)}, d^{(1)}, \dots, d^{(k)}, d^{(k+1)}, \dots \|) \cdot \underline{\infty}.$$

• $y = \bigcup y^{(i+1)}$ é também um objeto em $\mathbb{D}_{\infty}^{\rightarrow}$ que modela a aplicação consecutiva do operador linear que interpreta o processo seqüencial. Neste caso, os fatores são constantes e correspondem ao produto paralelo que executa a ação d em todas as posições de memória (relativa ao espaço geométrico I). O processo Q'' apresentado no Exemplo 6.1 é modelado por y .

A seguir, formaliza-se as expressões recursivas destas modelagens.

1. Para cada i define-se uma seqüência $\{w_n^{(i)}\}_{n \in \omega} \subseteq \mathbb{D}_n$ tal que

$$w_n^{(i)} = \Pi_{\infty, n}^{\rightarrow}(y_n^{(i)}) = \Pi_n^{\rightarrow} \circ \Pi_{\infty, n+1}^{\rightarrow}(y_{n+1}^{(i)}).$$

2. Pela completção tem-se que $y^{(i)} = \bigcup_{n=0}^{\infty} w_n^{(i)} \in \mathbb{D}_{\infty}^{\rightarrow}$ satisfaz a igualdade $y^{(i)} = \mathbf{F}_{(01)(11)}(y^{(i)} \sqcap \emptyset)$. Logo, cada subconjunto coerente $y^{(i)} = \{\overline{d_{(00)^i}^{(k)}}_{.111:001}\}$ é também ponto fixo para a equação $x = \mathbf{F}_{(01)(11)}(x \sqcap \emptyset)$.

3. Além disso, $\forall i, j$ tem-se que $\mathbf{F}_{(01)(11)}(y^{(i)} \sqcap y^{(j)}) = y^{(i)} \cup y^{(j)} \in \mathbb{D}_\infty$, portanto $y = \bigcup y^{(i)} \in \mathbb{D}_\infty$. Da mesma forma, é imediato que y e todas suas aproximações finitas são também soluções para esta equação. Entretanto, considerando-se o conjunto corente x apresentado na Exemplificação 8.6, tem-se que $x \not\subseteq y$ e $y \not\subseteq x$.
4. A simplificação da notação de forma que $f(x) = \mathbf{F}_{(01)(11)}(x \sqcap \emptyset)$, facilita a verificação da igualdade

$$y' = f^{(\infty)}(y'_0) = f^{(\infty)} \circ f^{(n)}(y'_0) = f^{(\infty)}(y'_n) = \dots = f(y').$$

Da mesma forma, f^∞ constitui-se no ponto fixo para a função f , independente do valor inicial de f .

Exemplificação 8.8

- Considere agora a seqüência $x_{n+1} = \mathbf{F}_{(01)(11)}(x_n \sqcap \{d_{(011)^n:00}^{(k)}\})$. Neste caso, tem-se que

$$\begin{aligned} \bullet x'_0 &= x_0 = \{d_{:00}^{(k)}\}, \\ \bullet x_1 &= \{d_{001:00}^{(k)}, d_{011:00}^{(k)}\}, \\ \bullet x_2 &= \{d_{001.001:00}^{(k)}, d_{011.001:00}^{(k)}, d_{011.011:00}^{(k)}\}, \\ &\vdots \\ \bullet x_{n+1} &= \{d_{(001)^{n+1}:00}^{(k)}, d_{011.(001)^n:00}^{(k)}, \dots, d_{(011)^{n+1}:00}^{(k)}\}. \end{aligned}$$

A existência e unicidade do conjunto corente $x = \bigcup \Pi_n \circ \Pi_{\infty, n+1}(x_{n+1}) \in \mathbb{D}_\infty^{\rightarrow}$ é garantida pela completção. Da mesma forma que os dois exemplos acima, $\Pi_n^{\rightarrow} \circ \Pi_{\infty, n+1}^{\rightarrow}(x_{n+1}) = \Pi_{\infty, n}^{\rightarrow}(x_n)$. Logo, x é um ponto fixo para a equação $x = \mathbf{F}_{(01)(11)}(x \sqcap x'_0)$.

Exemplificação 8.9

- Considere $d_n \in \mathcal{A}$ nas equações $x_{n+1} = \mathbf{F}_{(20)}(\{\overline{d_{n+1}^{(k)}}_{20:00}\} \sqcap x_n)$, onde $x_0 = \{\overline{d_0^{(k)}}_{20:00}\}$. Tem-se então a seqüência de conjuntos corentes

$$\begin{aligned} \bullet x_1 &= \{\overline{d_0^{(k)}}_{20:00}, \overline{d_1^{(k)}}_{20:00}\}, \\ \bullet x_2 &= \{\overline{d_0^{(k)}}_{20:00}, \overline{d_1^{(k)}}_{20:00}, \overline{d_2^{(k)}}_{20:00}\}, \\ &\vdots \\ \bullet x_{n+1} &= \{\overline{d_0^{(k)}}_{20:00}, \overline{d_1^{(k)}}_{20:00}, \overline{d_2^{(k)}}_{20:00}, \dots, \overline{d_{n+1}^{(k)}}_{20:00}\} \end{aligned}$$

cujos ponto fixo é indicado por $x = \bigcup^\uparrow x_n \in \pi_{1, \infty}^{(00)}[D_1] \subseteq \mathbb{D}_\infty$.

Nas próximas exemplificações são definidas equações que possuem solução em $\mathbb{D}_\infty^{\leftarrow}$ e $\mathbb{D}_{2\infty}^{\rightarrow}$. Salienta-se que, o modelo computacional proposto por Scott [SCO 72a] não possui interpretação computacional para estes casos.

Exemplificação 8.10

Veja a Exemplificação 6.3 apresentada ao final da seção 6. Mostra-se a seguir que cada subconjunto da seqüência $\{\pi_{n, \infty}^{(00)}(w'_n)\}_{n \in \omega} \subseteq \mathbb{D}_\infty^{\leftarrow}$ é expresso como imagem do operador $\mathbf{F}_{(01)(11)}$.

- Considere a seqüência $x_{n+1} = \mathbf{F}_{(01)(11)}(x'_0 \sqcap x_n)$, tem-se que
 - $x'_0 = x_0 = \{d_{:00}^{(k)}\}$,
 - $x_1 = \{d_{001:00}^{(k)}, d_{011:00}^{(k)}\}$,
 - $x_2 = \{d_{00.001:00}^{(k)}, d_{001.011:00}^{(k)}, d_{011.011:00}^{(k)}\}$,
 - \vdots
 - $x_{n+1} = \{d_{(00)^n.001:00}^{(k)}, d_{(00)^{n-1}.001.011:00}^{(k)}, \dots, d_{001.(011)^n:00}^{(k)}, d_{(011)^{n+1}:00}^{(k)}\}$,

Sabendo-se que $\Pi_{\infty,n}^{\leftarrow}(x_n) = w'_n$, ou ainda $\pi_{n,\infty}^{(00)}(w'_n) = x_n$, a Figura 6.4 representa o processo interpretado por x_n . Verifica-se facilmente que $\Pi_n^{\leftarrow} \circ \Pi_{\infty,n+1}^{\leftarrow}(x_{n+1}) = \Pi_{\infty,n}^{\leftarrow}(x_n)$. Portanto, tem-se

$$x = \bigcup \Pi_n^{\leftarrow} \circ \Pi_{\infty,n+1}^{\leftarrow}(x_{n+1}) \in \mathbb{D}_{\infty}^{\leftarrow} \quad \text{e} \quad x = \mathbf{F}_{(01)(11)}(x \sqcap x'_0).$$

Exemplificação 8.11

O próximo exemplo é uma aplicação da composição infinita do operador linear $\mathbf{F}_{(10)}$ e modelado em $\mathbb{D}_{2\infty}$ pode ser redefinido, considerando-se o operador $\mathbf{F}_{(20)}$. Salienta-se que o modelo computacional proposto por Scott, em [SCO 72a], não possui interpretação computacional para estes casos. Neste sentido, $\mathbb{D}_{2\infty}$ é uma extensão do modelo proposto por Scott, que se constitui um trabalho pioneiro dentro da semântica denotacional das linguagens de programação.

- Considere a Exemplificação 8.6 apresentada no início desta seção. Suponha uma seqüência de soluções para a equação:

$$x_0 = \{d_{:001}^{(k)}\}, \quad x_{n+1} = \mathbf{F}_{(01)(11)}(x_n \sqcap \{d_{(00)^n:001}^{(k)}\}) \quad (8.1)$$

onde os termos variam de acordo com o valor atribuído ao indexador $k \in I$. Tem-se então:

$$\begin{aligned} \bullet \mathbf{0} &= \{d_{:001}^{(0)}, d_{011:001}^{(0)}, \dots, d_{(00)^n.011:001}^{(0)}, \dots\}, \\ \bullet \mathbf{1} &= \{d_{:001}^{(1)}, d_{011:001}^{(1)}, \dots, d_{(00)^n.011:001}^{(1)}, \dots\}, \\ &\vdots \\ \bullet \mathbf{k} &= \{d_{:001}^{(k)}, d_{011:001}^{(k)}, \dots, d_{(00)^n.011:001}^{(k)}, \dots\}. \end{aligned}$$

Cada subconjunto coerente $\mathbf{k} \in \text{Coh } \mathbf{D}_{\infty}$ constitui-se num objeto total que interpreta o processo que executa a ação d seqüencialmente, na mesma posição de memória k , sem restrição quanto ao tempo de execução.

A seguir, mostra-se a construção de novos subconjuntos em $\mathbb{D}_{2\infty}$ a partir da definição de uma nova teia, modelando a concorrência entre processo sem restrição quanto ao tempo de execução. Neste caso, os tokens são os subconjuntos \mathbf{k} . Pela Definição a aplicação da função posição será indicada por $\Upsilon_{\infty}(\mathbf{k}) = \{k\}$.

Para tal, considera-se uma outra equação definida pela expressão

$$y_0 = \emptyset, \quad y_{n+1} = \mathbf{F}_{(10)}(y_n \sqcap \{\mathbf{n}_{10:00}\}). \quad (8.2)$$

Tem-se então os seguintes termos

$$\begin{aligned} \bullet y_1 &= \{\mathbf{0}_{10:00}\}, \\ \bullet y_2 &= \{\mathbf{0}_{10:00}, \mathbf{1}_{10:00}\}, \\ &\vdots \\ \bullet y_k &= \{\mathbf{0}_{10:00}, \mathbf{1}_{10:00}, \dots, \mathbf{k}_{10:00}, \dots\}. \end{aligned}$$

- O conjunto coerente $y = \bigcup_{k \in I} y_k \in \overline{\mathbb{D}}_\infty$ interpreta o processo R que executa, simultaneamente, todos os produtos seqüenciais interpretados pelos subconjuntos \mathbf{k} que satisfazem a equação 6.1. Portanto, y constitui-se num ponto fixo para a equação 6.2. Na Figura 8.5 apresenta-se uma representação para o processo R .

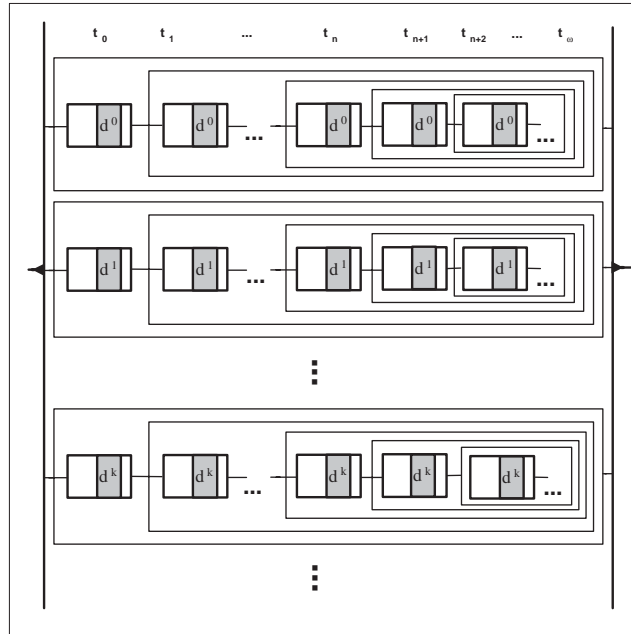


FIGURA 8.5 - O Processo R .

9 Linguagem de Processos Baseada no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$ dos Processos Computacionais

Considera-se neste capítulo uma linguagem de processos baseada em interpretações definidas sobre os objetos e morfismos do espaço coerente de processos $\mathbb{D}_{\infty}^{\rightarrow}$. Primeiramente, apresenta-se o conjunto de constantes e símbolos funcionais que podem aparecer na especificação da linguagem que se está propondo. Tais conjuntos definem sua assinatura. A seguir introduz-se a especificação equacional da linguagem com a definição do conjunto Ω de todas as expressões bem formadas na assinatura Σ baseada na função de representação definida de $\mathbb{D}_{\infty}^{\rightarrow}$ para Ω . Ao final, apresenta-se alguns exemplos de aplicação dos construtores de processos na definição e solução de equações recursivas.

9.1 Assinatura da linguagem

Seja I um conjunto de rótulos indicando as posições de um espaço geométrico e \mathcal{A} a família de ações rotuladas por posições em I . O conjunto K dos símbolos constantes é dado pela união

$$K = Ind_P \cup Ind_T.$$

Nesta caso, o conjunto $Ind_T = Test \cup \{bool\}$ indica a união do conjunto de indicadores de testes com o conjunto unitário cujo único elemento denota o teste totalmente indefinido. Em analogia, o conjunto $Ind_P = \mathcal{A} \times I \cup \{skip\}$ denota a união do conjunto de indicadores de processos elementares com o conjunto unitário cujo único elemento indica o processo *skip* totalmente indefinido, para o qual não existe ação associada.

De acordo com o que foi apresentado na Seção 3.3, define-se o conjunto

$$FOp = \{Id, (), \parallel, |, \cdot, +\}$$

dos identificadores operações ou dos símbolos funcionais. Neste caso, tem-se:

- $Id, () : Ind_P \rightarrow Ind_P$ são símbolos funcionais unários (aridade 1),
- $\parallel, |, \cdot : Ind_P \times Ind_P \rightarrow Ind_P$ são símbolos binários, e
- $+ : Ind_P \times Ind_P \times Ind_T \rightarrow Ind_P$ é um símbolo funcional de aridade 3.
Em particular, ao fixar $b \in Ind_T$, obtém-se $+_b : Ind_P \times Ind_P \rightarrow Ind_P$.

Sejam $F_n \in FOp, \forall n \in \omega$. A igualdade $\odot_{i=0}^n(F_i) = F_n \circ \dots \circ F_1 \circ F_0$ indica um processo resultante de uma composição finita dos construtores representados pelos símbolos funcionais do conjunto FOp . Assim, o conjunto de símbolos funcionais recursivos é indicado por

$$FRec = \{\odot_{i=0}^n(F_i) \mid F_i \in FOp\}.$$

Em analogia, a igualdade $\odot_{i=0}^{\infty}(F_i) = \dots \circ F_n \circ \dots \circ F_1 \circ F_0$, indica um processo resultante de uma composição infinita dos construtores representados pelos símbolos funcionais do conjunto F . O conjunto de símbolos funcionais infinitos é dado por

$$FInf = \{\odot_{i=0}^{\infty}(F_i) \mid F_i \in \{\cdot, +\}\}.$$

Definição 9.1 A assinatura para uma linguagem de processos baseada em \mathbb{D}_{∞} é definida pela união

$$\Sigma = K \cup FOp \cup FRec \cup FInf.$$

Supõe-se ainda que Σ contém tantas variáveis quantas forem necessárias, e essas serão indicadas pelas letras $X, Y, Z, T, \dots x, y, z, t \dots$, possivelmente subscritas. Além disso, tem-se que:

- d, e, f, g, \dots são variáveis que indicam ações em \mathcal{A} , e
- k, l, m, n, \dots são variáveis que indicam posições (rótulos) em I .

9.1.1 Termos Bem Formados da Assinatura Σ

Seja Ω o conjunto de todas as expressões bem formadas na assinatura Σ .

1. Variáveis e símbolos constantes (identificadores de processos ou identificadores de testes) são termos na assinatura Σ , portanto elementos do conjunto Ω .
2. Se $*$ $\in \{Id, (), \|\|, | |, \cdot, +_b\}$ e $t_0, t_1, \dots, t_n, t_{n+1}, \dots, b \in \Omega$ então

$$*_{i=0}^n(t_{i+1}) = t_{n+1} * t_n * \dots * t_1 * t_0 \quad \text{e} \quad *_{i=0}^n(t_{i+1}) = t_0 * \dots * t_n * t_{n+1}$$

são também termos (finitos) recursivamente definidos em Ω . Neste casos, tais termos podem ser também identificados pelas respectivas equações

$$\begin{cases} T'_0 = t_0, \\ T'_{n+1} = t_{n+1} * T'_n. \end{cases} \quad \text{e} \quad \begin{cases} T_0 = t_0, \\ T_{n+1} = T_n * t_{n+1}. \end{cases}$$

3. Além destes, tem-se que $T_\infty = *_{n=0}^\infty t_n = t_0 * t_1 * \dots * t_{n+1} * \dots$ são termos infinitos em Ω .

Seja $\Omega^* \subseteq \Omega$ o subconjunto de todas as expressões bem formadas na assinatura Σ que interpretam processos executados em tempo finito.

1. Ω^* está identificado com o subespaço coerente $\pi_{n+1, \infty}^{\rightarrow}[\mathbb{D}_{n+1}] \subseteq \mathbb{D}_\infty^{\rightarrow}$ das interpretações construídas nos subníveis \mathbb{D}_n e imersas no domínio $\mathbb{D}_\infty^{\rightarrow}$.
2. Cada $t \in \Omega^*$ está identificado com um conjunto coerente $x \in \pi_{n+1, \infty}^\theta[\mathbb{D}_{n+1}]$, sempre que $\theta \in \{(00), (001), (002)\}$. Tal identificação é formalizada na próxima seção pela função de representação \mathfrak{R}^* ou pela correspondente inversa.
3. Considerando que em $\pi_{n+1, \infty}^{\rightarrow}[\mathbb{D}_{n+1}]$ não há interpretação para processos cuja execução não se efetiva em tempo computacional finito, sempre que $*$ $\in FOp$ tem-se que $T_\infty = *_{i=0}^\infty(t_i) \notin \Omega^*$. Isto significa que, a composição infinita $\odot_{i=0}^\infty(F_i)$, definida $\forall i \in \omega$ não é fechada em Ω^* .

9.2 Especificação Equacional da Linguagem

Considere o conjunto Ω de todas as expressões bem formadas na assinatura Σ que se identificam com o subconjunto de \mathcal{P} de todos os processos interpretados em $\mathbb{D}_\infty^{\rightarrow}$.

Define-se primeiramente, a função de representação $\mathfrak{R}^* : \pi_{n+1, \infty}^{\rightarrow}[\mathbb{D}_{n+1}] \rightarrow \Omega^*$ que formaliza a identificação de cada conjunto coerente interpretando um processo, com execução finita no sentido temporal, com sua correspondente representação - a expressão analítica do processo interpretado por cada subconjunto coerente como um termo válido em Σ .

Posteriormente, com a definição de uma ordem parcial em Ω^* induzida pela inclusão em $\pi_{n, \infty}^{\rightarrow}[\mathbb{D}_n]$, estende-se a definição da função de representação, no caso indicada por $\mathfrak{R} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \Omega$.

9.2.1 Função de Representação $\mathfrak{R}^* : \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n] \subseteq \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \Omega^*$

Seja Ω^* o conjunto de todas as expressões bem formadas na assinatura Σ que se identificam com o subespaço coerente $\pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ de todos os subconjuntos coerentes em $\mathbb{D}_{\infty}^{\rightarrow}$ que interpretam processos executados em tempo finito. Os objetos em $\pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ são construídos em \mathbb{D}_n , portanto são imagens de imersões $\pi_{n,\infty}^{\theta}[\mathbb{D}_n]$, sempre que $\theta \in \{(00), (001), (002)\}$.

Definição 9.2 *Sejam $x_n \in \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n] \subseteq \mathbb{D}_{\infty}^{\rightarrow}$, onde $\theta \in \{(00), (001), (002)\}$.*

Seja o espaço coerente $\mathbb{B} \equiv (\text{Coh}\mathbf{B}, \subseteq)$. A função de representação booleana $\mathfrak{R}^{\text{bool}} : \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n] \rightarrow \Omega$ que associa cada teste interpretado no subespaço coerente $\pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$ a um termo bem formado em $\text{Ind}_T \subseteq \Omega$ é definida pela expressão $\mathfrak{R}^{\text{bool}}(x) = \{b \mid b_{:\theta} \in x \text{ e } b \in \mathbf{B}\}$.

A função $\mathfrak{R}^ : \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n] \subseteq \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \Omega^*$ é recursivamente definida nos diferentes casos apresentados logo a seguir.*

1 *Seja $x_n \in \mathbb{D}_{\infty}^{\rightarrow}$ tal que $x_n \in \pi_{n,\infty}^{(00)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n]$.*

$$\bullet \mathfrak{R}^*(x_0) = \begin{cases} \text{skip} & \text{se } x_0 = \emptyset, \\ d^{(k)} & \text{se } x_0 = \{d_{:00}^{(k)}\}. \end{cases}$$

$$\bullet \mathfrak{R}^*(x_{n+1}) = (\mathfrak{R}^*(x_n))$$

se $x_{n+1} \in F_{00}(\mathbb{D}_{\infty}^{\rightarrow})$ e a seguinte condição é satisfeita:

$$\bullet (\gamma_n^{(0)} \circ \phi_n^{(0)} \circ \Pi_{\infty,n}^{\rightarrow})(x_n) = \Pi_{\infty,n+1}^{\rightarrow}(x_{n+1}).$$

$$\bullet \mathfrak{R}^*(x_{n+1}) = \| \text{\textit{-}}_{2^n}, \mathfrak{R}^*(x_n^{(0)}), \mathfrak{R}^*(x_n^{(1)}), \dots, \mathfrak{R}^*(x_n^{(k)}) \text{\textit{-}}_{2^n} \|$$

se $x_{n+1} \in \mathbf{F}_{(10)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$, $k \in \Upsilon_{\infty}(x_{n+1}) \subset I$ e verifica-se que

$$\bullet (\Phi_n^{(1)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty,n+1}^{\rightarrow})(x_{n+1}) = \bigcup_{k \in \Upsilon_{\infty}(x_{n+1})} (\gamma_n^{(0)} \circ \phi_n^{(1)})(\overline{\Pi_{\infty,n}^{\rightarrow}(x_n^{(k)})})$$

$$\bullet \mathfrak{R}^*(x_{n+1}) = \| \mathfrak{R}^*(x_n^{(0)}), \mathfrak{R}^*(x_n^{(1)}), \dots, \mathfrak{R}^*(x_n^{(k)}), \dots \|$$

se $x_{n+1} \in \mathbf{F}_{(10)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$, $k \in \Upsilon_{\infty}(x_{n+1}) = I$ e verifica-se que

$$\bullet (\Phi_n^{(1)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty,n+1}^{\rightarrow})(x_{n+1}) = \bigcup_{k \in \Upsilon_{\infty}(x_{n+1})} (\gamma_n^{(0)} \circ \phi_n^{(1)})(\overline{\Pi_{\infty,n}^{\rightarrow}(x_n^{(k)})})$$

$$\bullet \mathfrak{R}^*(x_{n+1}) = \| \text{skip} \|$$

se $x_{n+1} \in \mathbf{F}_{(10)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$, $\Upsilon_{\infty}(x_{n+1}) = \emptyset$ e verifica-se que

$$\bullet (\Phi_n^{(1)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty,n+1}^{\rightarrow})(x_{n+1}) = \{\emptyset\}$$

$$\bullet \mathfrak{R}^*(x_{n+1}) = | \text{\textit{-}}_{2^n} \mid \mathfrak{R}^*(x_n^{(0)}) \mid \mathfrak{R}^*(x_n^{(1)}) \mid \dots \mid \mathfrak{R}^*(x_n^{(k)}) \mid \text{\textit{-}}_{2^n} |$$

se $x_{n+1} \in \mathbf{F}_{(20)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$, $k \in \Upsilon_{\infty}(x_{n+1}) \subset I$ e verifica-se que:

$$\bullet (\Phi_n^{(2)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty,n+1}^{\rightarrow})(x_{n+1}) = \bigcup_{k \in \Upsilon_{\infty}(x_{n+1})} (\gamma_n^{(0)} \circ \phi_n^{(2)})(\overline{\Pi_{\infty,n}^{\rightarrow}(x_n^{(k)})}).$$

$$\bullet \mathfrak{R}^*(x_{n+1}) = | \mathfrak{R}^*(x_n^{(0)}) \mid \mathfrak{R}^*(x_n^{(1)}) \mid \dots \mid \mathfrak{R}^*(x_n^{(k)}) \mid \dots |$$

se $x_{n+1} \in \mathbf{F}_{(20)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$, $k \in \Upsilon_{\infty}(x_{n+1}) = I$ e verifica-se que:

$$\bullet (\Phi_n^{(2)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty,n+1}^{\rightarrow})(x_{n+1}) = \bigcup_{k \in K \subseteq \omega} (\gamma_n^{(0)} \circ \phi_n^{(2)})(\overline{\Pi_{\infty,n}^{\rightarrow}(x_n^{(k)})}).$$

- $\mathfrak{R}^*(x_{n+1}) = | \text{skip} |$
 se $x_{n+1} \in \mathbf{F}_{(20)}(\mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$, $k \in \Upsilon_{\infty}(x_{n+1}) = \emptyset$ e verifica-se que:
 - $(\Phi_n^{(2)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \{\emptyset\}$.

- $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(y_n) \cdot -_{2^n}$
 se $x_{n+1} \in \mathbf{F}_{(01)(11)}(\mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$, e verificam-se as seguintes condições:
 - $(\Psi_n^{(0)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(y_n)$
 - $(\Psi_n^{(1)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \emptyset$,

- $\mathfrak{R}^*(x_{n+1}) = -_{2^n} \cdot \mathfrak{R}^*(y_n)$
 se $x_{n+1} \in \mathbf{F}_{(01)(11)}(\mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$, e verificam-se as seguintes condições:
 - $(\Psi_n^{(0)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \emptyset$,
 - $(\Psi_n^{(1)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(y_n)$

- $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(y_n) \cdot \mathfrak{R}^*(z_n)$
 se $x_{n+1} \in \mathbf{F}_{(01)(11)}(\mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$ e verificam-se as seguintes condições:
 - $(\Psi_n^{(0)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(y_n)$ e
 - $(\Psi_n^{(1)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(z_n)$.

- $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(y_n) +_{\mathfrak{R}^{bool(b)}} -_{2^n}$
 se $x_{n+1} \in \mathbf{F}_{(02)(12)(22)}(\mathbb{B} \prod \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$ e verificam-se as condições:
 - $(\Lambda_n^{(0)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(y_n)$ e
 - $(\Lambda_n^{(1)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \emptyset$.

- $\mathfrak{R}^*(x_{n+1}) = -_{2^n} +_{\mathfrak{R}^{bool(b)}} \mathfrak{R}^*(z_n)$
 se $x_{n+1} \in \mathbf{F}_{(02)(12)(22)}(\mathbb{B} \prod \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$ e verificam-se as condições:
 - $(\Lambda_n^{(0)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \emptyset$ e
 - $(\Lambda_n^{(1)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(z_n)$.

- $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(y_n) +_{\mathfrak{R}^{bool(b)}} \mathfrak{R}^*(z_n)$
 se $x_{n+1} \in \mathbf{F}_{(02)(12)(22)}(\mathbb{B} \prod \mathbb{D}_{\infty}^{\rightarrow} \prod \mathbb{D}_{\infty}^{\rightarrow})$ e verificam-se as condições:
 - $(\Lambda_n^{(0)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(y_n)$ e
 - $(\Lambda_n^{(1)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(z_n)$.

- 2 Seja $x_{n+1} \in \mathbb{D}_{\infty}^{\rightarrow}$ tal que $x_{n+1} \in \pi_{n+1, \infty}^{(001)}[\mathbb{D}_{n+1}] \subseteq \pi_{n, \infty}^{\rightarrow}[\mathbb{D}_n]$.
- $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(\pi_{n+1, \infty}^{(00)} \circ \Pi_{\infty, n+1}^{\rightarrow}(x_{n+1})) \cdot \underline{\infty}$.

- 3 Seja $x_{n+1} \in \mathbb{D}_{\infty}^{\rightarrow}$ tal que $x_{n+1} \in \pi_{n+1, \infty}^{(002)}[\mathbb{D}_{n+1}] \subseteq \pi_{n, \infty}^{\rightarrow}[\mathbb{D}_n]$.

- $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(\pi_{n+1,\infty}^{(00)} \circ \Pi_{\infty,n+1}^{\rightarrow}(x_{n+1})) +_{\mathfrak{R}^{bool}(x)} \underline{\infty}$.

Proposição 9.1 *A função $\mathfrak{R}^* : \pi_{n,\infty}^{\rightarrow}[\mathbb{D}_n] \rightarrow \Omega^*$ é um isomorfismo em $\mathcal{S}et$.*

Prova.

Considere as seguintes notações :

- $:\infty = \theta.\theta.\theta\dots$
 - $\mathcal{FLin} = \{\mathbf{F}_{(00)}, \mathbf{F}_{(10)}, \mathbf{F}_{(20)}, \mathbf{F}_{(01)(11)}, \mathbf{F}_{(02)(12)(22)}, Id_{\pi_{n,\infty}^{\rightarrow}}\}$
 - $\mathcal{FOP} = \{Id, (), \parallel, |, \cdot, +_b\}$
- Além disso, suponha também $x_{n+1}, x'_{n+1} \in \pi_{n+1,\infty}^{\theta}[\mathbb{D}_{n+1}] \subseteq \mathbb{D}_{\infty}^{\rightarrow}$.

1. Mostra-se primeiramente que \mathfrak{R}^* está bem definida. Suponha o contrário, existem $t, t' \in \Omega^*$, $t \neq t'$ tais que $t = \mathfrak{R}^*(x_{n+1})$, $t' = \mathfrak{R}^*(x'_{n+1})$ e $x_{n+1} = x'_{n+1}$. Neste caso, deve-se analisar as três situações apresentadas logo a seguir.

- Se t é o processo totalmente indefinido, $t = \mathfrak{R}^*(x_{n+1}) = skip$, então $x_{n+1} = \emptyset$ e neste caso, como $t' \neq \underline{\infty}$ tem-se que $x'_{n+1} \neq \emptyset$. Portanto $x'_{n+1} \neq x_{n+1}$. O mesmo ocorre se t' é o processo totalmente indefinido.

- Se $t = \mathfrak{R}^*(x_0)$, $t' = \mathfrak{R}^*(x'_0)$ são processos elementares distintos então tem-se $t = d^{(k)}$ e $t' = e^{(l)}$ e verificam-se as seguintes possibilidades: $d \neq e$ e $k = l$, ou $d = e$ e $k \neq l$ ou $d \neq e$ e $k \neq l$. Assim, conclui-se que $x_0, x'_0 \in \pi_{0,\infty}^{\theta}[\mathbb{D}_0]$ e $x_0 = \{d_{\cdot\theta}^{(k)}\} \neq x'_0 = \{e_{\cdot\theta}^{(l)}\}$.

- Finalmente, se t, t' não são processos elementares e nenhum deles é indefinido, então tais processos ou resultam da aplicação de construtores (operadores) distintos ou são imagens do mesmo construtor mas com argumentos (operandos) distintos:

No primeiro caso, $x_{n+1} \in F[X]$, $x'_{n+1} \in G[Y]$ e considera-se $F \neq G$, $F, G \in \mathcal{FLin}$ e $X, Y \in \{\pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \subseteq \mathbb{D}_{\infty}^{\rightarrow}, \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}], \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \mathbb{B}\}$, dependendo da aridade dos construtores.

No segundo caso, $x_{n+1}, x'_{n+1} \in F[X]$ e tem-se que existem $y, y' \in X$, $y \neq y'$ tais que $x_{n+1} = F[y]$ e $x'_{n+1} = F[y']$.

Sabendo-se que F, G são funções (lineares) em ambos os casos $x_{n+1} \neq x'_{n+1}$.

Logo, todas as situações contrariam a hipótese da igualdade $x_{n+1} \neq x'_{n+1}$. Conclui-se que $t = \mathfrak{R}^*(x_{n+1}) \neq t' = \mathfrak{R}^*(x'_{n+1})$, tem-se $x_{n+1} \neq x'_{n+1}$ e pode-se afirmar que a função de representação \mathfrak{R}^* apresentada na Definição 9.4 está bem definida.

2. Agora mostra-se que a função \mathfrak{R}^* é um isomorfismo em $\mathcal{S}et$.

- Para mostrar a injetividade de \mathfrak{R} supõe-se da mesma forma que existem $x_{n+1}, x'_{n+1} \in \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \subseteq \mathbb{D}_{\infty}^{\rightarrow}$ tais que $x_{n+1} \neq x'_{n+1}$ mas $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(x'_{n+1}) = t$.

Se t é o processo indefinido então é imediato que $x_0 = x'_0 = \emptyset$. Da mesma forma, se $t = d^{(k)}$ é um processo elementar, então $x_0 = x'_0 = \{d_{\cdot\theta}^{(k)}\}$;

Agora, caso t não seja um processo elementar e nem o processo indefinido, então suponha primeiramente que existe u_n tal que $x_{n+1} = F(u_n)$ e $x'_{n+1} = G(u_n)$ sempre que F, G são funções lineares e $u_n \in \{\pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}], \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \pi_{n+1,\infty}^{\rightarrow}[\mathbb{D}_{n+1}] \prod \mathbb{B}\}$, dependendo da aridade dos construtores F e G .

Por hipótese, se $x_{n+1} \neq x'_{n+1}$ então $F \neq G$. De acordo com o que se mostrou na primeira parte desta prova, \mathfrak{R} está bem definida, portanto $\mathfrak{R}^*(x_{n+1}) \neq \mathfrak{R}^*(x'_{n+1})$,

o que é uma contradição. Assim sendo, sempre que $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(x'_{n+1})$ tem-se $x_{n+1} = x'_{n+1}$.

Quando $x_{n+1}, x'_{n+1} \in F[X]$ e tem-se que existem $y_n, y'_n \in X$, $y_n \neq y'_n$ tais que $x_{n+1} = F(y_n)$ e $x'_{n+1} = F(y'_n)$. Neste caso, se a função linear F tem como imagem pela função \mathfrak{R} o operador $T \in FRec \cup FOp$ e por hipótese $\mathfrak{R}^*(x_{n+1}) = \mathfrak{R}^*(x'_{n+1})$ então conclue-se também que $x_{n+1} = x'_{n+1}$.

• Para mostrar que \mathfrak{R}^* é sobrejetora supõe-se de novo o contrário. Existe $T \in \Omega^*$ tal que não existe $x_{n+1} \in \pi_{n+1, \infty}^{-1}[\mathbb{D}_{n+1}]$ que satisfaz a igualdade $\mathfrak{R}^*(x_{n+1}) = T$. Isto contraria a forma com que se definem os termos válidos em Ω^* , representando processos executados em tempo finito - os processos elementares, o processo indefinido *skip* ou aqueles construídos pela aplicação finita (recursiva) dos construtores que são elementos do conjunto $FOp \cup FRec$. Assim, considerando-se esta hipótese, sempre que $* \in FOp$ tem-se que $T = T'_\infty = *_{i=0}^\infty(t_i) \notin \Omega^*$. Logo $T \notin \mathbb{D}_\infty^c$.

Provou-se que $\mathfrak{R}^* : \pi_{n, \infty}^{-1}[\mathbb{D}_n] \rightarrow \Omega^*$ é bijetora, ou seja, um isomorfismo em *Set*. ▲

Considerando-se a proposição anterior, pode-se definir a correspondente inversa da função de representação, denominada função de interpretação, neste caso indicada por $\mathfrak{S}^* : \Omega^* \rightarrow \pi_{n+1, \infty}^{-1}[\mathbb{D}_{n+1}]$ e definida por $\mathfrak{S}^*(t) = \mathfrak{R}^{*-1}(t) = x$, sempre que $\mathfrak{R}^*(x) = t$.

9.2.2 Relação de Ordem em Ω^*

Definição 9.3 *Sejam o subespaço coerente $\pi_{n, \infty}^{-1}[\mathbb{D}_n]$ e o subconjunto Ω^* de termos bem formados de Σ que representam processos executados em tempo finito. Dados $t, t' \in \Omega$, define-se a relação*

$$t \sqsubseteq^* t' \Leftrightarrow \exists x, x' \in \mathbb{D}_\infty^c \mid \mathfrak{R}^*(x) = t, \mathfrak{R}^*(x') = t' \text{ e } x \subseteq x'.$$

Proposição 9.2 \sqsubseteq^* é uma relação de ordem.

De fato, a prova é imediata considerando-se que a inclusão é uma relação de ordem, portanto reflexiva, simétrica e transitiva. Neste sentido, diz-se que a relação \sqsubseteq é uma relação de ordem induzida pela inclusão em $\pi_{n, \infty}^{-1}[\mathbb{D}_n]$.

Como consequência da Proposição 9.3 tem-se que $(\Omega^*, \sqsubseteq^*, \textit{skip})$ é uma ordem parcial cujo menor elemento é o processo totalmente indefinido.

Entretanto o conjunto parcialmente ordenado $(\Omega^*, \sqsubseteq^*, \textit{skip})$ não é completo. Ou seja, com base na noção de subconjunto dirigido $X \subseteq \Omega^*$ como uma seqüência generalizada, a completude de Ω^* é alcançada quando cada uma destas seqüências X converge para um único elemento, indicado por $\bigsqcup X$.

No caso, considerando-se a ordem estendida \sqsubseteq a completação é caracterizada pelo conjunto Ω que, além de incluir todos os termos bem formados em Ω^* , admite também termos que correspondem aos supremos para estas seqüências generalizadas, representando os processos que não podem ser executados em tempo finito.

Desde que o espaço coerente $\mathbb{D}_\infty^{\rightarrow}$ é obtido pela a completação do subespaço coerente $\pi_{n, \infty}^{-1}[\mathbb{D}_n]$, apresenta-se a seguir a extensão (contínua) da função de representação $\mathfrak{R}^* : \pi_{n, \infty}^{-1}[\mathbb{D}_n] \subseteq \mathbb{D}_\infty^{\rightarrow} \rightarrow \Omega^*$ indicada por $\mathfrak{R} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \Omega$.

9.2.3 Extensão da Função de Representação

Definição 9.4 *Considere o espaço coerente $\mathbb{D}_\infty^{\rightarrow}$ e o conjunto Ω de todas as expressões válidas da assinatura Σ , incluindo as expressões para processos sem limite quanto ao tempo de execução. Considere também $x = \bigcup^\uparrow x_n$ sempre que $x_n \in \pi_{n, \infty}^{-1}[\mathbb{D}_n] \subseteq \mathbb{D}_\infty^{\rightarrow}$ e a função de representação apresentada na Definição 9.4. A função de representação estendida $\mathfrak{R} : \mathbb{D}_\infty^{\rightarrow} \rightarrow \Omega$ é definida pela expressão*

$$\mathfrak{R}(x) = \mathfrak{R}(\bigcup^\uparrow x_n) = \bigsqcup \mathfrak{R}^*(x_n).$$

A seguir, define-se a relação de ordem em Ω .

Definição 9.5 *Sejam o espaço coerente $\mathbb{D}_\infty^\rightarrow$ e o conjunto Ω de termos válidos em Σ . Dados $t, t' \in \Omega$, define-se a relação*

$$t \sqsubseteq t' \Leftrightarrow \exists x, x' \in \mathbb{D}_\infty^\rightarrow \mid \mathfrak{R}(x) = t, \mathfrak{R}(x') = t' \quad e \quad x \subseteq x'.$$

Proposição 9.3 \sqsubseteq *é uma relação de ordem.*

Prova é imediata.

Na próxima proposição mostra-se que $(\Omega, \sqsubseteq, skip)$ é uma ordem parcial completa (*cpo*) e algébrica. Considera-se então outra noção fundamental, a de aproximações compactas, relativa a algebricidade de subconjuntos parcialmente ordenados.

De acordo com o que já foi colocado na Seção 6.1.1, as aproximações compactas de um elemento t são aproximações significativas no sentido em que não existe maneira de se aproximar de t sem levar em conta tais elementos. Um elemento t é compacto se, sempre que ele precede o supremo de um conjunto dirigido $X \subseteq \Omega$, existe outro elemento em tal conjunto X tal que t é uma de suas aproximações.

Proposição 9.4 $(\Omega, \sqsubseteq, skip)$ *é um cpo algébrico.*

Prova

1. Mostra-se primeiramente que, no conjunto parcialmente ordenado $(\Omega, \sqsubseteq, skip)$ com menor elemento *skip*, para todo subconjunto dirigido $X \subseteq \Omega$ tem-se que $\bigsqcup X \in \Omega$. Neste caso, $(\Omega, \sqsubseteq, \infty)$ é uma ordem parcial completa.

Para tal, considere o subconjunto dirigido $X \subseteq \Omega$. $\mathcal{X} = \{x \mid \mathfrak{R}(x) = t \in X\}$ é também um subconjunto dirigido de \mathbb{D}_∞ . De fato, para todo $x', x'' \in \mathcal{X}$ existe $t' = \mathfrak{R}(x')$, $t'' = \mathfrak{R}(x'') \in X \subseteq \Omega$. Além disso, existe também $t \in X$ tal que $t' \sqsubseteq t$ e $t'' \sqsubseteq t$. Logo, pela ordem induzida, existe $x \in \mathbb{D}_\infty^\rightarrow$, com $t = \mathfrak{R}(x)$, e tem-se que $x' \subseteq x$ e $x'' \subseteq x$.

Além disso, desde que $\mathbb{D}_\infty^\rightarrow$ é um espaço coerente, $\bigcup^\uparrow \mathcal{X} \in \mathbb{D}_\infty^\rightarrow$ e $\mathfrak{R}(\bigcup^\uparrow \mathcal{X}) \in \Omega$. Suponha agora que $\mathfrak{R}(\bigcup^\uparrow \mathcal{X}) \neq \bigsqcup X = u$. Então existe $y \neq \bigcup^\uparrow \mathcal{X} \in \mathbb{D}_\infty^\rightarrow$, $\mathfrak{R}(y) = u$ e tal que $x \subseteq u$, para todo $x \in \mathcal{X}$. O que é absurdo, pois o $\bigcup^\uparrow \mathcal{X}$ é o único elemento maximal do subconjunto dirigido \mathcal{X} . Portanto, $\mathfrak{R}(\bigcup^\uparrow \mathcal{X}) = \bigsqcup X = u$.

2. Mostra-se a seguir que $(\Omega, \sqsubseteq, skip)$ é um *cpo* algébrico.

Define-se então conjunto $approx(t) = \{u \in \Omega^* \mid u \sqsubseteq t\}$. Desde que $\infty \sqsubseteq t$, $approx(t) \neq \emptyset$. Pela função de representação, para todo $t', t'' \in approx(t)$ existe $x', x'' \in \mathbb{D}_\infty^{\rightarrow c} \subseteq \mathbb{D}_\infty$ tal que $\mathfrak{R}(x') = t'$ e $\mathfrak{R}(x'') = t''$. Pela completude binária em $\mathbb{D}_\infty^\rightarrow$ existe $x \in \mathbb{D}_\infty^\rightarrow$ tal que $x' \subseteq x$ e $x'' \subseteq x$. Além disso, como $\mathbb{D}_\infty^\rightarrow$ é algébrico, $x', x'' \in approx(x)$ e tem-se que $\bigsqcup approx(x) = x$.

Considerando-se a ordem induzida, existe $t \in \Omega$ tal que $t' \sqsubseteq t$ e $t'' \sqsubseteq t$ e mostrou-se portanto que $approx(t)$ é um subconjunto dirigido de Ω . Suponha-se entretanto, que $\bigsqcup approx(t) \neq t$. Logo, para todo $t' \in \Omega$ existe $\bar{t} \in \Omega$, $\bar{t} \neq t$ e tem-se que $t' \sqsubseteq \bar{t}$. Entretanto isto implicaria que, pela definição da função de representação, $\exists \bigcup^\uparrow approx(x)$, o que é absurdo. Logo tem-se que $\bigsqcup approx(t) = t$. O que caracteriza a algebricidade da ordem parcial completa $(\Omega, \sqsubseteq, skip)$. \blacktriangle

A próxima proposição mostra que o *cpo* algébrico $(\Omega, \sqsubseteq, skip)$ é consistentemente completo, ou seja, constitui-se num domínio.

A consistência de qualquer subconjunto X em Ω garante que este conjunto não contém informação contraditória. Ou ainda, existe sempre um termo comum em Ω para o qual todos os termos de X são aproximações.

Proposição 9.5 *O cpo algébrico $(\Omega, \sqsubseteq, skip)$ é um domínio de Scott.*

Prova

Considerando-se que a ordem parcial completa $(\Omega, \sqsubseteq, skip)$ é algébrica, basta mostrar que todo subconjunto de elementos compactos que é consistente possui supremo em $\mathbb{D}_\infty^\rightarrow$.

Suponha o contrário. $X \subseteq \Omega^*$ é um subconjunto consistente, entretanto tem-se que $\bigsqcup X \notin \Omega$.

Considere então o subconjunto $\mathcal{X} = \{x \mid \mathfrak{R}(x) \in X\} \subseteq \mathbb{D}_\infty^\rightarrow{}^c$. Mostra-se agora que \mathcal{X} é consistente. De fato, para todo $x \in \mathcal{X}$ tem-se que existe $t \in X \subseteq \omega^*$ tal que $\mathfrak{R}(x) = t$. Como por hipótese X é consistente, existem $\bar{t} \in \Omega$ tal que, para cada $t \in X$ tem-se que $t \sqsubseteq \bar{t}$. Logo existe $\bar{x} \in \mathbb{D}_\infty^\rightarrow$, $\bar{t} = \mathfrak{R}(\bar{x})$ tal que $x \sqsubseteq \bar{x}$, $\forall x \in \mathcal{X}$. Ou seja, o conjunto \mathcal{X} possui cota superior em $\mathbb{D}_\infty^\rightarrow$.

Desde que $\mathbb{D}_\infty^\rightarrow$ é consistentemente completo, $\bigsqcup \mathcal{X} \in \mathbb{D}_\infty^\rightarrow$ o que implica em absurdo, pois garante a existência e unicidade de uma cota superior para $X \in \Omega^*$.

Mostrou-se então que se $X \subseteq \Omega^*$ então $\bigsqcup X \in \Omega$ e conclui-se que $(\Omega, \sqsubseteq, skip)$ é consistentemente completo. \blacktriangle

A Proposição apresentada logo a seguir assegura a continuidade da função de representação \mathfrak{R} . Uma função é contínua se ela preserva o supremo de subconjuntos dirigidos de elementos compactos do domínio, $\mathfrak{R}(x) = \bigsqcup \{f(a) \mid a \in approx(x)\}$. Neste sentido, tal propriedade se refere ao comportamento da função quando aplicada sobre os processos executados em tempo finito em \mathbb{D}_∞ e à análise das expressões que correspondem às imagens em Ω .

Proposição 9.6 *A função de representação estendida $\mathfrak{R} : \mathbb{D}_\infty^\rightarrow \rightarrow \Omega$ é contínua.*

Prova.

Considerando-se que $\mathbb{D}_\infty^\rightarrow$ e Ω são domínios, e que por definição a função $\mathfrak{R}^* : \mathbb{D}_\infty^\rightarrow{}^c \rightarrow \Omega^*$ é monótona, é imediato que a função de representação $\mathfrak{R} : \mathbb{D}_\infty^\rightarrow \rightarrow \Omega$ constitui-se na única extensão contínua da função \mathfrak{R}^* . Veja [STV 94]. \blacktriangle

Considerando-se a proposição anterior, pode-se provar a continuidade da correspondente inversa da função de representação estendida, denominada função de interpretação estendida, neste caso indicada por $\mathfrak{S} : \Omega \rightarrow \mathbb{D}_\infty^\rightarrow$ e definida por $\mathfrak{S}(t) = \mathfrak{R}^{-1}(t) = x$, sempre que $\mathfrak{R}(x) = t$.

A definição da relação de equivalência em Ω é induzida pela função de interpretação \mathfrak{S} e apresentada logo a seguir.

Definição 9.6 *Sejam o espaço coerente $\mathbb{D}_\infty^\rightarrow$ e o conjunto de termos válidos em Ω . Seja também $\mathfrak{S} : \Omega \rightarrow \mathbb{D}_\infty^\rightarrow$ definida por $\mathfrak{S}(t) = \mathfrak{R}^{-1}(t) = x$, sempre que $\mathfrak{R}(x) = t$. Dados $t, t' \in \Omega$, define-se a relação*

$$t \equiv_\Omega t' \Leftrightarrow \exists x \in \mathbb{D}_\infty^\rightarrow \mid \mathfrak{S}(t) = x = \mathfrak{S}(t').$$

Uma classe de equivalência de um termo $t \in \Omega$ é o conjunto de todos os termos que estão relacionados com o termo t pela relação de equivalência \equiv_Ω , ou seja, $cl(t) = \{t' \mid t' \equiv_\Omega t\}$. Pela proposição 9.1, para cada termo em Ω tem-se $cl(t) = \{t\}$. Isto está formalizado na próxima proposição.

Proposição 9.7 *O conjunto Ω de termos válidos da assinatura Σ , apresentado na seção 9.1.1 e a correspondente família $CL[\Omega] = \{cl(t) \mid t \in \omega\}$ de todas as classes de equivalência definidas sobre Ω são subconjuntos isomorfos.*

Observa-se ainda que, pela indexação dos tokens de cada interpretação, tem-se que $:\theta = \theta : \theta$, sempre que $\theta \in \{00, 001, 002\}$. Logo, tem-se a igualdade entre os conjuntos $\{\emptyset_{201.001:001}\} = \{\emptyset_{201:001}\}$. Além disso, pela Definição 9.4, tem-se

- $\mathfrak{R}\{\{\emptyset_{201.001:001}\}\} = ((\mid skip \mid \cdot _1) \cdot _2) \cdot \underline{\infty}$ e
- $\mathfrak{R}\{\{\emptyset_{201:001}\}\} = ((\mid skip \mid \cdot _1) \cdot \underline{\infty})$.

Portanto, $((\mid skip \mid \cdot _1) \cdot _2) \cdot \underline{\infty} = ((\mid skip \mid \cdot _1) \cdot \underline{\infty})$. Contudo, esta situação não contraria a Proposição 9.7, pois neste caso, os subconjuntos são iguais no domínio $\mathbb{D}_\infty^\rightarrow$.

9.2.4 Aplicação da Função de Representação no Domínio $\mathbb{D}_\infty^\rightarrow$

A aplicação da função de representação $\mathfrak{R} : \mathbb{D}_\infty^\rightarrow \rightarrow \Omega$ sobre os construtores de processos apresentados na Seção 8 é dada na próxima definição.

Definição 9.7 *Sejam $x, x' \in \mathbb{D}_\infty^\rightarrow$ tais que $\mathfrak{R}(x) = t$ e $\mathfrak{R}(x') = t'$. Tem-se que*

1. $\mathfrak{R}[Id_{\mathbb{D}_\infty} : \mathbb{D}_\infty \rightarrow \mathbb{D}_\infty] = Id_\Omega : \Omega \rightarrow \Omega$ tal que
 $\mathfrak{R}(Id_{\mathbb{D}_\infty}(x)) = \mathfrak{R}(x) = t = Id_\Omega(t);$
2. $\mathfrak{R}[\mathbf{F}_{(00)} : \mathbb{D}_\infty \rightarrow \mathbb{D}_\infty] = ()_\Omega : \Omega \rightarrow \Omega$ tal que
 $\mathfrak{R}(\mathbf{F}_{(00)}(x)) = (t) = ()_\Omega(t);$
3. $\mathfrak{R}[\mathbf{F}_{(10)} : \mathbb{D}_\infty \prod \mathbb{D}_\infty \rightarrow \mathbb{D}_\infty] = \parallel \parallel_\Omega : \Omega \times \Omega \rightarrow \Omega$ tal que
 $\mathfrak{R}(\mathbf{F}_{(10)}(x \sqcap x')) = \parallel _, \mathfrak{R}(x), \mathfrak{R}(x'), _ \parallel = \parallel _, t, t', _ \parallel = \parallel (t, t') \parallel;$
4. $\mathfrak{R}[\mathbf{F}_{(20)} : \mathbb{D}_\infty \prod \mathbb{D}_\infty \rightarrow \mathbb{D}_\infty] = \mid \mid_\Omega : \Omega \times \Omega \rightarrow \Omega$ tal que
 $\mathfrak{R}(\mathbf{F}_{(20)}(x \sqcap x')) = \mid _, \mathfrak{R}(x) \uparrow \mathfrak{R}(x'), _ \mid = \mid _, t \uparrow t', _ \mid = \mid (t, t') \mid;$
5. $\mathfrak{R}[\mathbf{F}_{(01)(11)} : \mathbb{D}_\infty \prod \mathbb{D}_\infty \rightarrow \mathbb{D}_\infty] = \cdot \cdot_\Omega : \Omega \times \Omega \rightarrow \Omega$ tal que
 $\mathfrak{R}(\mathbf{F}_{(01)(11)}(x \sqcap x')) = \mathfrak{R}(x) \cdot \mathfrak{R}(x') = t \cdot t' = \cdot \cdot_\Omega(t, t');$
6. $\mathfrak{R}[\mathbf{F}_{(02)(12)(22)} : \mathbb{D}_\infty \prod \mathbb{D}_\infty \prod \mathbb{B} \rightarrow \mathbb{D}_\infty] = +_\Omega : \Omega \times \Omega \times \Omega \rightarrow \Omega$ tal que
 $\mathfrak{R}(\mathbf{F}_{(02)(12)(22)}(x \sqcap x' \sqcap b)) = \mathfrak{R}(x) +_{\mathfrak{R}^{bool}(b)} \mathfrak{R}(x') = t +_b t' = +_\Omega(t, t', b).$

É interessante ressaltar algumas situações apresentadas por neste modelo.

- Se $x \in \pi_{n,\infty}^{(00)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\rightarrow[\mathbb{D}_n]$ e $y \in tot(\mathbb{D}_\infty^\rightarrow)$, então
 $F_{00,10}(x \sqcap y) \in tot(\mathbb{D}_\infty^\rightarrow)$ e $\mathfrak{R}[F_{00,10}(x \sqcap y)] \in tot(\Omega);$
 $F_{00,10}(y \sqcap x) = y \in tot(\mathbb{D}_\infty^\rightarrow)$ e $\mathfrak{R}[F_{00,10}(y \sqcap x)] = \mathfrak{R}[y] \in tot(\Omega);$
- Se $x \in \pi_{n,\infty}^{(001)}[\mathbb{D}_n] \subseteq \pi_{n,\infty}^\rightarrow[\mathbb{D}_n]$ e $y \in \pi_{m,\infty}^{(001)}[\mathbb{D}_m] \subseteq \pi_{m,\infty}^\rightarrow[\mathbb{D}_m]$, então
 $F_{00,10}(x \sqcap y) = x \in \pi_{n,\infty}^{(001)}[\mathbb{D}_n]$ e $\mathfrak{R}[F_{00,10}(x \sqcap y)] = \mathfrak{R}[x] \in \Omega;$
 $F_{00,10}(y \sqcap x) = y \in \pi_{m,\infty}^{(001)}[\mathbb{D}_m]$ e $\mathfrak{R}[F_{00,10}(y \sqcap x)] = \mathfrak{R}[y] \in \Omega;$

TABELA 9.1 - Equações Satisfeitas em Ω .

0.	$\ \ (t', t) \in \Omega$
1.	$\ \ (skip, t) = \ \ (t, skip) = t$
3.	$\ \ (\ \ (t, t'), t'') = (\ \ (t, \ \ (t', t''))$
4.	$\ \ (t, t') = \ \ (t', t)$
0.	$ (t', t) \in \Omega$
2.	$ (t, t) = t$
3.	$ ((t, t'), t'') = ((t, (t', t''))$
4.	$ (t, t') = (t', t)$
0.	$\cdot (t', t) \in \Omega$
1.	$\cdot (skip, t) = \cdot (t, skip) = t$
0.	$+_{bool}(t', t) \in \Omega$
2.	$+_{bool}(t, t) = t$

0 \rightarrow Fechamento		
1 \rightarrow Elemento Neutro	2 \rightarrow Idempotência	
3 \rightarrow Associatividade	4 \rightarrow Comutatividade	

9.2.5 Propriedades dos operadores em Ω .

Nesta seção apresentam-se as propriedades satisfeitas pelos construtores de processos modelados em $\mathbb{D}_{\infty}^{\rightarrow}$ e representados por operadores em Ω .

A especificação equacional apresentada na Figura 9.1 descreve o conjunto de propriedades satisfeitas pelos processos representados pelas expressões válidas em Ω , induzidas pela função de interpretação $\mathfrak{R} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \Omega$. Tal figura resume as conclusões apresentadas na Proposição 9.8, logo a seguir.

Proposição 9.8 *Sejam os conjuntos de operadores $FOp = \{Id, (), \| \|, | |, \cdot, +\}$, $FRec = \{\odot_{i=0}^n(F_i) \mid F_i \in FOp\}$ e $FInf = \{\odot_{i=0}^{\infty}(F_i) \mid F_i \in \{\cdot, +\}\}$. As seguintes propriedades são satisfeitas:*

1. se $F \in FOp \cup FRec \cup FInf$ então F verifica a propriedades de fechamento;
2. se $F \in \{+, | |\}$ então F verifica a propriedade de idempotência;
3. se $F \in \{\| \|, | |\}$ então F verifica a comutatividade, a associatividade e a existência de elemento neutro.

Observa-se que o operador produto seqüencial, assim como os correspondentes construtores definidos pela composição finita ou infinita sobre este operador, não verificam a propriedade associativa. Isto ocorre pela construção temporal da estrutura do modelo de máquina geométrica, onde operadores categóricos como produto direto e soma direta não são associativos. Esta propriedade será alcançada, como é de se esperar, na interpretação semântica associada a estes operadores.

TABELA 9.2 - Processos Elementares.

0	$z_0 := s \mapsto s[i_0 := 0]$	$u_0 := s \mapsto s[i_0 := 1]$
1	$z_1 := s \mapsto s[i_1 := 0]$	$u_1 := s \mapsto s[i_1 := 1]$
\vdots	\vdots	\vdots
n	$z_n := s \mapsto s[i_n := 0]$	$u_n := s \mapsto s[i_n := 1]$

Pode-se pensar ainda em definir uma gramática que torne explícita as regras de construção de expressões em Ω , compatível com a construção das correspondentes interpretações no domínio $\mathbb{D}_\infty^\rightarrow$, a exemplo do que foi feito em [ISL 2001].

Em anexo, são apresentadas algumas representações, que correspondem à aplicação da função de representação sobre objetos em $\mathbb{D}_\infty^\rightarrow$ construídos nas primeiras etapas da estrutura ordenada do modelo de máquina geométrica. Veja

- Tabela A.1, com interpretações envolvendo o conjunto coerente vazio e definidas pela composição infinita de $\mathbf{F}_{(00)}$;
- Tabela A.2 com representações envolvendo o conjunto coerente vazio sendo definidas pela composição infinita de $\mathbf{F}_{(00)(10)}$;
- Tabela A.3, com as representações envolvendo também o conjunto coerente vazio e mas estão definidas pela composição infinita de $\mathbf{F}_{(00)(20)}$;
- Tabela A.4, com as representações de subconjuntos coerentes unitários construídos nos níveis $\mathbb{D}_0 - \mathbb{D}_1$ e $\mathbb{D}_1 - \mathbb{D}_2$ e imersos em $\mathbb{D}_\infty^\rightarrow$ pela composição infinita de $\mathbf{F}_{(00)}$;
- Tabela A.5, também com as representações de subconjuntos coerentes unitários construídos nos níveis $\mathbb{D}_0 - \mathbb{D}_1$ e $\mathbb{D}_1 - \mathbb{D}_2$, mas imersos em $\mathbb{D}_\infty^\rightarrow$ pela composição infinita de $\mathbf{F}_{(00)(01)}$;

9.3 Equações Recursivas em Ω

Os próximos exemplos de aplicação dos construtores de processos e da função de representação na definição e solução de equações recursivas encerram este capítulo.

Observação 9.1

Seja $I = \{i_0, i_1, \dots, i_n, \dots\}_{\{n \in \omega\}}$ um conjunto enumerável de rótulos e $V = \{0, 1\}$ o conjunto de valores que as variáveis que compõe cada estado computacional podem assumir. Considera-se agora os processos elementares apresentados na Tabela 9.2.

- Pela equação recursiva $X_n \equiv \| z_n, X_{n+1} \|$, tem-se que X_0 descreve o processo que executa, simultaneamente, todos os processos elementares que aparecem na primeira coluna da Tabela 9.2. Ou ainda, a aplicação deste processo resulta no estado computacional constante que tem *zero* em todas as suas posições de memória.

- Da mesma forma, da equação $Y_n \equiv \| u_n, Y_{n+1} \|$ tem-se que Y_0 descreve o processo que executa, simultaneamente, todos os processos elementares que aparecem na segunda coluna da Tabela 9.2. Neste caso, todo estado computacional é transformado no estado constante que tem *um* em todas as posições de memória.
- Pela equação $Z_n \equiv \| \| z_n, u_n \|, X_{n+1} \|$ o processo X_0 descreve o processo que executa, simultaneamente, uma escolha aleatoriamente entre os valores *zero* e *um* para cada uma das variáveis do estado computacional de saída.
- Considere a expressão $K_n \equiv (n == k \Rightarrow u_n, z_n)$. Se $k < n$ então a equação recursiva $W_{n,k} \equiv \| K_n, W_{n+1,k} \|$ que define o processo que coloca o valor *um* na k -ésima posição do estado computacional final, e o valor *zero* nas demais posições, a partir da posição n .
- Considere a expressão $K_n \equiv (n <= k \Rightarrow u_n, z_n)$. A equação recursiva $W_{n,k} \equiv \| K_n, W_{n+1,k} \|$ que define o processo que coloca o valor *um* desde a primeira posição até a k -ésima posição do estado computacional final, e o valor *zero* nas demais posições.
- Considere a expressão $K_n \equiv (n >= k \Rightarrow u_n, z_n)$. A equação recursiva $W_{n,k} \equiv \| K_n, W_{n+1,k} \|$ que define o processo que coloca o valor *um* a partir da k -ésima posição do estado computacional final, e o valor *zero* nas posições que precedem esta.

Observação 9.2

Considera-se agora o processo que executa rotações de comprimento p , deslocando o valor de cada variável uma posição para a direita no vetor que representa um estado computacional. Neste caso, o conteúdo (valor da variável) da posição i do estado computacional de entrada é copiado e constitui-se no conteúdo da posição $i + 1$ do estado computacional de saída, até completar um período p de posições.

Pode-se pensar que o vetor que representa um estado computacional é dividido em grupos que contém p posições consecutivas. Este processo garante que cada grupo será rotacionado para a direita, uma posição, simultaneamente.

Para formalizar tal processo, algumas expressões recursivas são necessárias:

- $shift_right(i, j) \equiv \| (i <= j \Rightarrow NIL, copy(i, i + 1), shift_right(i + 1, j)) \|$ que define o processo que copia o valor da variável que ocupa a posição i no estado computacional de entrada e o coloca como o valor associado à variável que ocupa a posição $i + 1$ no estado computacional de saída. Tal processo deve ser executado enquanto i é menor que j .
- $rot_right(i, j) \equiv \| copy(j, i), shift_right(i, j) \|$ que corresponde a parte do processo que copia o valor da variável que ocupa a última posição e o coloca como o valor da variável que ocupa a primeira posição, em cada período, e simultaneamente, executa o processo $shift_right(i, j)$ apresentado no subitem anterior.
- $period_rot_right(i, p) \equiv \| rot - right(i, i + p - 1), period_rot_right(i + p, p) \|$ que corresponde a expressão recursiva do processo que executa uma rotação para a direita dos valores associados as variáveis de um estado computacional de entrada, considerando que o parâmetro p indica o período de cada rotação. Neste caso, se o vetor que representa cada estado computacional tem comprimento $n \times p$ este processo deverá fazer n rotações de comprimento p , todas executada ao mesmo tempo.

Assim sendo, ao substituir $i := 0$ nesta última expressão recursiva obtem-se

$$R_0 \equiv \text{period_rot_right}(0, p)$$

que define o processo que a partir da primeira posição (i_0), rotaciona uma posição para a direita cada um dos valores atribuídos as variáveis do vetor que representa o estado computacional de entrada. Simultaneamente, este procedimento se repete, de $p+1$ em $p+1$ posições, até esgotar todas as posições de tal estado computacional.

Salienta-se ainda que a execução seqüencial do processo R_0 gera ciclos de comprimento p . De fato,

$$\underbrace{R_0 \cdot R_0 \cdot \dots \cdot R_0}_{p+1} \equiv (\cdot R_0)^{p+1} = Id \quad e \quad (\cdot R_0)^{p+2} = R_0.$$

Observação 9.3

Considere agora o processo que executa as rotações de período p no sentido contrário, ou seja, para a esquerda. Neste caso, tem-se que

- $\text{shift_left}(i, j) \equiv \| (i \leq j \Rightarrow \text{skip}, \text{copy}(i+1, i), \text{shift_left}(i+1, j)) \|$
- $\text{rot_left}(i, j) \equiv \| (\text{copy}(i, j), \text{shift_left}(i+1, j)) \|$
- $\text{period_rot_left}(i, p) \equiv \| \text{rot_left}(i+p-1, i), \text{period_rot_left}(i+p, p) \|$

Assim sendo,

$$L_0 \equiv \text{period_rot_left}(0, p)$$

corresponde a expressão recursiva do processo que executa a rotação de uma posição para esquerda dos valores associados as variáveis do estado computacional de entrada (com dimensão finita), a partir da primeira posição (i_0). Considera-se que o parâmetro p indica o período de cada rotação. Neste caso, se o vetor que representa cada estado computacional tem comprimento $n \times p$ este processo deverá fazer n rotações de comprimento p , todas executada ao mesmo tempo.

Salienta-se também que a execução seqüencial do processo L_0 gera ciclos de comprimento p . De fato,

$$\underbrace{L_0 \cdot L_0 \cdot \dots \cdot L_0}_{p+1} \equiv (\cdot L_0)^{p+1} = Id \quad e \quad (\cdot L_0)^{p+2} = L_0.$$

Observação 9.4

Considere as expressões recursivas dos processos apresentados nas Observações 9.2 e 9.3. Define-se um novo processo cujo período é de $2p$. Neste caso, cada período é dividido em dois subperíodos, o primeiro que executa rotações para a direita e o segundo que executa rotações para a esquerda. As expressões envolvidas em cada processo são dadas por:

- $W(i, p) \equiv \| \text{wperiod_rot_right}(i, p), \text{wperiod_rot_left}(i+p, p) \|$
- $\text{wperiod_rot_right}(i, p) \equiv \| \text{rot_right}(i, i+p-1), \text{wperiod_rot_right}(i+2p, p) \|$
- $\text{wperiod_rot_left}(i+p, p) \equiv \| \text{rot_left}(i+p-1, i), \text{wperiod_rot_right}(i+2p, p) \|$

Portanto,

$$W_{0,p} \equiv \| \text{wperiod_rot_right}(0, p), \text{wperiod_rot_left}(p, p) \|$$

corresponde a expressão recursiva do processo periódico que executa, simultaneamente, rotações para a direita das posições que compõe a primeira metade de cada período e rotações para a esquerda nas posições da outra metade.

Considera-se que o parâmetro $2p$ indica o período de cada rotação. Neste caso, se o vetor (de dimensão finita) que representa cada estado computacional tem comprimento $n \times 2p$ o processo $W(0, p)$ deverá fazer n chamada recursivas de comprimento $2p$, todas executada ao mesmo tempo.

Salienta-se também que a execução seqüencial do processo $W_{0,p}$ gera ciclos de comprimento p . De fato,

$$\underbrace{W_{0,p} \cdot W_{0,p} \cdot \dots \cdot W_{0,p}}_{p+1} \equiv (\cdot W_{0,p})^{p+1} = Id \text{ e } (\cdot W_{0,p})^{p+2} = W_{0,p}.$$

10 Interpretação Semântica Baseada no Espaço Coerente $\mathbb{D}_{\infty}^{\rightarrow}$ dos Processos Computacionais

10.1 Espaço Coerente \mathbb{S} dos Estados Computacionais

Da mesma forma que no domínio E dos fluxogramas proposto por Scott, os objetos no espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$, interpretando processos computacionais, também são entes matemáticos estáticos. Contudo, apesar da noção dinâmica de transformação associada aos processos, com entrada e (possível) saída de estados computacionais, tal objeto ou interpretação permanece impassível, ou ainda, imutável. Isto traduz um ato freqüente e intuitivo de se deslizar os dedos ou deslocar o olhar, sobre cada representação gráfica de um processo, no sentido do seu início para o seu final, sem que tal atitude altere sua representação.

Conforme explicitado na introdução, o modelo de máquina apresentada neste trabalho está centrado na noção de processo como transformação de estados, capaz de modelar o funcionamento de sistemas dinâmicos. Com base nesta noção intuitiva, e seguindo a solução proposta por Scott, a formalização desta idéia de modelar as transformações entre estados a fim de se analisar o comportamento dos processos é alcançada no espaço de funções definidas sobre o domínio de todos os estados de computação.

Pode-se verificar ao longo deste capítulo, que tal formalização define o domínio semântico $\mathbb{S} \multimap \mathbb{S}$, capaz de avaliar o comportamento dos processos interpretados no espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$.

A interpretação semântica do modelo proposto é alcançada com a definição de uma função linear, chamada função de avaliação, capaz de associar cada conjunto coerente em $\mathbb{D}_{\infty}^{\rightarrow}$ a um conjunto coerente de traços lineares em $\mathbb{S} \multimap \mathbb{S}$. Neste sentido, o comportamento dos processos modelados como objetos do espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$ é interpretado por funções lineares - objetos do espaço coerente $\mathbb{S} \multimap \mathbb{S}$.

Conseqüentemente, considerando a função de representação $\mathfrak{R} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \Omega$, apresentada nas Definições 9.2 e 9.4, este modelo de máquina geométrica é capaz de interpretar o comportamento de todos os processos definidos por termos válidos da linguagem proposta no Capítulo 9, usando como estrutura os objetos e morfismos da categoria $\mathit{CospLin}$.

O gráfico na próxima figura mostra os morfismos que definem o relacionamento entre estes domínios.

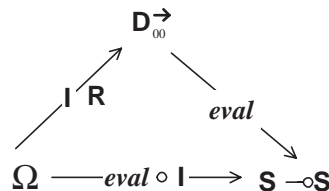


FIGURA 10.1 - O Relacionamento entre $\mathbb{D}_{\infty}^{\rightarrow}$, Ω e $\mathbb{S} \multimap \mathbb{S}$.

Com base nos comentários apresentados nos parágrafos anteriores, este capítulo primeiramente introduz o espaço coerente \mathbb{S} dos estados computacionais, o qual interpreta o conjunto S de estados computacionais apresentado na Definição 3.1.

Entretanto, o espaço funcional de todas as funções lineares definidas sobre \mathbb{S} não constitui-se num espaço coerente, conforme Troelstra [TRO 92]. Uma prova detalhada deste fato pode ser encontrada em Dimuro [DIM 98]. Para contornar este obstáculo, seguiu-se o sugerido nestas últimas referências, que provam que a coleção de todas as funções lineares definidas sobre \mathbb{S} indicada por $[\mathbb{S} \rightarrow \mathbb{S}]$ é ordem isomorfa ao conjunto dos traços lineares de tais funções.

Assim, baseado no espaço coerente \mathbb{S} , o espaço coerente $\mathbb{S} \multimap \mathbb{S}$ cujos objetos são os grafos especiais de funções lineares definidas sobre \mathbb{S} é definido, cujos objetos modelam o comportamento dos processos interpretados em $\mathbb{D}_\infty^\rightarrow$.

10.1.1 Espaço Coerente dos Nomes das Variáveis Computacionais

O espaço coerente \mathbb{I} dos nomes das variáveis computacionais que definem um estado computacional dá interpretação para os elementos do conjunto I .

Definição 10.1 *Considere I um conjunto de índices. Seja $\mathbf{I} \equiv (I, \cong)$ a teia discreta definida pela coleção de todos os índices $i \in I$ juntamente com a relação de coerência, que nesta definição coincide com a relação de igualdade e é dada por*

$$i \cong j \iff i = j,$$

sempre que $i, j \in I$. Assim, $\mathbb{I} = (\text{Coh}(\mathbf{I}), \subseteq)$ indica o domínio plano de Girard dos nomes das variáveis computacionais e $\text{Coh}(\mathbf{I}) = \{\emptyset\} \cup \{\{i\} \mid i \in I\}$.

Quando I é um conjunto enumerável de índices, por exemplo $I = \{i_n\}_{n \in \omega}$, tem-se então que a teia \mathbf{I} é definida pela coleção de todos os índices $i_n \in I$ juntamente com a relação de coerência $i_n \cong i_k \iff n = k$. Neste caso, $\text{Coh}(\mathbf{I}) = \{\emptyset\} \cup \{\{i_n\}\}_{n \in \omega}$.

10.1.2 Espaço Coerente dos Valores das Variáveis Computacionais

Os subconjuntos coerentes do espaço coerente \mathbb{V} dos valores das variáveis computacionais, que definem cada estado computacional, interpretam os elementos do conjunto $V = \text{Val} \cup \{\#\}$.

Definição 10.2 *Considere um conjunto de valores Val . Seja $\mathbf{V} \equiv (V, \cong)$ uma teia definida pela coleção de todos os valores juntamente com uma relação reflexiva e simétrica chamada relação de coerência. $\mathbb{V} = (\text{Coh}(\mathbf{V}), \subseteq)$ indica o domínio de Girard dos valores das variáveis computacionais.*

Pela definição anterior, $\emptyset \in \mathbb{V}$ interpreta o valor computacional $\# \in V$ da Definição 3.1 e cada valor $v \in \text{Val}$ é interpretado por um conjunto coerente não-vazio, no espaço coerente \mathbb{V} . Além disso, subconjuntos de conjuntos coerentes que interpretam valores em Val são conjuntos coerentes que interpretam valores parciais, não necessariamente presentes em Val .

10.1.3 Espaço Coerente $\mathbb{I} \multimap \mathbb{V}$

Nesta seção se apresenta formalmente o espaço coerente de todas as funções lineares definidas do espaço coerente \mathbb{I} dos indexadores de nomes para o espaço coerente \mathbb{V} dos valores das variáveis que definem os estados de computação, onde a ordem de informação é a inclusão. Primeiramente, algumas considerações já comprovadas na literatura específica serão ressaltadas no próximo parágrafo e estão diretamente relacionadas com a próxima definição. Veja [GIR 89] [TRO 92] [DIM 98].

Seja o conjunto $\text{Lin} = \{s : \mathbb{I} \rightarrow \mathbb{V} \mid s \text{ é linear}\}$ de todas as funções (estritas) monótonas, contínuas, estáveis e lineares de \mathbb{I} em \mathbb{V} .

1. Cada função $s \in Lin$ é identificada por um subconjunto especial de seu grafo, denominado traço linear e dado por

$$ltr(s) = \{(i, v) \in |\mathbb{I}| \times |\mathbb{V}| \mid v \in s(\{i\})\},$$

onde $|\mathbb{I}| = \bigcup \mathbb{I} = \{i \mid \{i\} \in \mathbb{I}\} = I$ e $|\mathbb{V}| = \bigcup \mathbb{V} = \{v \mid \{v\} \in \mathbb{V}\}$. Neste caso, $Ltr = \{ltr(s) \mid s \in Lin\}$ indica o conjunto de todos os traços lineares de funções $s : \mathbb{I} \rightarrow \mathbb{V}$.

Dada $s \in Lin$, sempre que $(i, v), (i', v') \in ltr(s)$ duas condições são satisfeitas:

- 1.1 $i \cong_{\mathbf{I}} i' \Rightarrow v \approx_{\mathbf{V}} v'$ (ou ainda $v \not\approx_{\mathbf{V}} v' \Rightarrow i \not\cong_{\mathbf{I}} i'$) e
 1.2 $v = v'$ e $i \cong_{\mathbf{I}} i' \Rightarrow i = i'$.

2. Por outro lado, todo subconjunto $X \subseteq I \times V$ que satisfaz as duas condições acima determina uma função linear definida pela expressão $s : \mathbb{I} \rightarrow \mathbb{V}$, $s(x) = \{v \mid \exists i \in x \text{ e } (i, v) \in X\}$ e tal que $X = ltr(s)$.

Com base nestas considerações, é definido o isomorfismo que garante a equivalência $Lin \equiv Ltr$.

O conjunto Ltr de todos os traços lineares de funções $s : \mathbb{I} \rightarrow \mathbb{V}$, ordenado pela inclusão, constitui-se num espaço coerente, conforme a próxima definição.

Definição 10.3 Considere os espaços coerentes \mathbb{I} e \mathbb{V} . $\mathbf{I} \multimap \mathbf{V} = (|\mathbb{I}| \times |\mathbb{V}|, \approx_{\multimap})$ indica a teia definida pelo produto cartesiano $I \times V$ e pela relação de coerência \approx_{\multimap} , segundo a qual, $(i, v) \approx_{\multimap} (i', v')$ se, e somente se,

$$i \cong_{\mathbf{I}} i' \Rightarrow (v \approx_{\mathbf{V}} v' \text{ e } (v = v' \Rightarrow i = i')).$$

A coleção de todos os subconjuntos coerentes da teia $\mathbf{I} \multimap \mathbf{V}$,

$$Coh(\mathbf{I} \multimap \mathbf{V}) \equiv \{x \subseteq I \times V \mid (i, v) \approx_{\multimap} (i', v'), \forall (i, v), (i', v') \in x\},$$

ordenados pela inclusão, define o espaço coerente $\mathbb{I} \multimap \mathbb{V} \equiv (Coh(\mathbf{I} \multimap \mathbf{V}), \subseteq)$.

Pela definição anterior, cada conjunto coerente de pares $s = \{(i, v)\}_{i \in J \subseteq I}$ tal que $\{v\} = s(\{i\})$, ou simplesmente $v = s(i)$, corresponde ao traço linear da função $s : \mathbb{I} \multimap \mathbb{V}$ pelo isomorfismo entre os conjuntos Lin e $Coh(\mathbf{I} \multimap \mathbf{V})$.

Note-se, entretanto, que nem todo subconjunto coerente em $\mathbb{I} \multimap \mathbb{V}$ modela um estado de computação em $I \rightarrow V$. Apenas os subconjunto maximais fazem isso, por mapearem objetos totais em objetos totais.

Salienta-se neste momento, que todos os subconjuntos coerentes em $\mathbb{I} \multimap \mathbb{V}$ são grafos especiais, chamados traços lineares, de funções totais onde a indefinição é associada ao conjunto vazio em \mathbb{V} . Justifica-se assim a inclusão $J \subseteq I$ na indicação de um elemento $s \in \mathbb{I} \multimap \mathbb{V}$ pela expressão $s = \{(i, v)\}_{i \in J \subseteq I}$.

Em particular, quando o traço linear de s é representado pelo conjunto vazio, tem-se que s é a função constante que associa todos os conjuntos coerentes em \mathbb{I} ao conjunto coerente vazio em \mathbb{V} .

10.1.4 Espaço Coerente dos Estados Computacionais

Neste trabalho, a noção de estado de computação é modelada de tal forma que todos os processos em \mathcal{P} executados em tempo finito, inclusive os não-determinísticos, possam ser interpretados como funções que sempre, a partir de um estado inicial, alcançam um estado final. Esta noção é posteriormente estendida para os demais processos.

Suponha então que cada conjunto coerente em $\mathbb{I} \multimap \mathbb{V}$ interpreta um pedaço consistente de informação sobre algum estado computacional. Ou ainda, todo conjunto coerente em $\mathbb{I} \multimap \mathbb{V}$ é um token da teia \mathbf{S} , sobre a qual se constrói o espaço coerente \mathbb{S} dos estados computacionais.

Considerando que a relação de coerência desta teia \mathbf{S} é a trivial, todos seus tokens são, dois a dois, coerentes. Portanto a união de quaisquer tokens determina sempre um novo conjunto coerente, que por sua vez, interpreta um novo estado computacional.

Essa abordagem possibilita a usarmos interpretação clássica do estado não-determinístico, qual seja, a de que um estado não-determinístico é uma coleção de estados determinísticos, cada um representando uma alternativa possível de realização do não-determinismo.

Assim, seja a família $Coh(\mathbb{I} \multimap \mathbb{V})$ dos subconjuntos coerentes em $\mathbb{I} \multimap \mathbb{V}$. Defina-se a seguir, o espaço coerente cuja teia \mathbf{S} consiste no conjunto de todas as partes de $Coh(\mathbb{I} \multimap \mathbb{V})$. Neste caso, cada token da teia \mathbf{S} é um conjunto coerente em $\mathbb{I} \multimap \mathbb{V}$, ou seja, é um traço linear, finito ou infinito, de uma função linear de \mathbb{I} para \mathbb{V} .

Definição 10.4 *Sejam $J, L \subseteq I$. Considere o espaço coerente $\mathbb{I} \multimap \mathbb{V}$. Na teia $\mathbf{S} = (Coh(\mathbb{I} \multimap \mathbb{V}), \approx_{\mathbf{S}})$ a relação de coerência é dada por*

$$\{(i, v)\}_{i \in J} \approx_{\mathbf{S}} \{(i', v')\}_{i' \in L}, \quad \forall \{(i, v)\}_{i \in J}, \{(i', v')\}_{i' \in L} \in Coh(\mathbb{I} \multimap \mathbb{V}).$$

A coleção $Coh(\mathbf{S}) = Coh(Coh(\mathbb{I} \multimap \mathbb{V}), \approx_{\mathbf{S}})$ de todos os subconjuntos coerentes da teia \mathbf{S} , ordenados pela inclusão, constitui o **espaço coerente dos estados computacionais**, indicado por

$$\mathbb{S} \equiv (Coh\mathbf{S}, \subseteq) \equiv (Coh(Coh(\mathbb{I} \multimap \mathbb{V}), \approx_{\mathbf{S}}), \subseteq).$$

Em particular, $\emptyset \in Coh(\mathbf{S})$ interpreta o estado computacional *skip*, denominado estado totalmente indefinido.

Em outras palavras, no modelo proposto neste trabalho, cada estado computacional (possivelmente não-determinístico) σ é interpretado por um conjunto coerente de traços lineares de funções definidas de \mathbb{I} para \mathbb{V} . Como consequência, o comportamento dos processos é interpretado por transformações que mapeiam conjunto de traços lineares.

A seguir, apresenta-se o espaço coerente de todos as funções lineares definidas sobre \mathbb{S} , através do qual se formaliza a interpretação semântica dos processos modelados em $\mathbb{D}_{\infty}^{\rightarrow}$.

10.1.5 Espaço Coerente $\mathbb{S} \multimap \mathbb{S}$

Definição 10.5 *Considere o espaço coerente \mathbb{S} dos estados computacionais e o conjunto $|\mathbb{S}| = \bigcup \mathbb{S} = \{\sigma \mid \{\sigma\} \in \mathbb{S}\} = S$. $\mathbf{S} \multimap \mathbf{S} = (|\mathbb{S}| \times |\mathbb{S}|, \approx_{\multimap})$ indica a teia definida pelo produto cartesiano $S \times S$ e pela relação de coerência \approx_{\multimap} , onde tem-se que $(\sigma, \varsigma) \approx_{\multimap} (\sigma', \varsigma') \text{ sss } \sigma \cong_{\mathbf{S}} \sigma' \Rightarrow (\varsigma \approx_{\mathbf{S}} \varsigma' \text{ e } (\varsigma = \varsigma' \Rightarrow \sigma = \sigma'))$. A coleção de todos os subconjuntos coerentes da teia, indicada por $\mathbf{S} \multimap \mathbf{S}$*

$$Coh(\mathbf{S} \multimap \mathbf{S}) \equiv \{x \subseteq S \times S \mid (\sigma, \varsigma) \approx_{\multimap} (\sigma', \varsigma'), \forall (\sigma, \varsigma), (\sigma', \varsigma') \in x\}$$

ordenados pela inclusão, define o espaço coerente $\mathbb{S} \multimap \mathbb{S} \equiv (Coh(\mathbf{S} \multimap \mathbf{S}), \subseteq)$.

Considerando agora as Definições 10.4, 10.3 e 10.5 é possível interpretar cada processo $p \in \mathcal{P}$ pelo traço linear de uma função $\mathbf{p} : \mathbb{S} \rightarrow \mathbb{S}$, no caso, um conjunto de pares $\{(\sigma, \varsigma) \mid \varsigma = \mathbf{p}(\sigma)\} \in \mathbb{S} \multimap \mathbb{S}$, tal que a seguinte notação é estabelecida:

- se $\sigma = \{s^{(m)}\}_{m \in M \subseteq \omega} \in \mathbb{S}$ então $\varsigma = \mathbf{p}(\sigma) = \{z^{(l)} \mid z^{(l)} \in \mathbf{p}(s^{(n)}), s^{(n)} \in \sigma\}_{l \in L \subseteq \omega} \in \mathbb{S}$;
- $s^{(m)}, z^{(l)} \in \mathbb{I} \multimap \mathbb{V}$ são conjuntos dados pelas expressões

$$s^{(m)} = \{(i, v) \mid v = s^{(m)}(i), i \in J \subseteq I\}, \quad \text{e}$$

$$z^{(l)} = \{(i', v') \mid v' = z^{(l)}(i'), i' \in J' \subseteq I\}.$$

Esta interpretação é estabelecida primeiramente para os processos elementares, sendo depois estendida sobre todos os processos construídos a partir destes pela aplicação dos construtores modelados em $\mathbb{D}_{\infty}^{\rightarrow}$.

Esta seção encerra com a definição do espaço coerente $\mathbb{S} \multimap \mathbf{Bool}$ que expressa o comportamento de cada teste computacional modelado em \mathbb{B} por uma função linear. Conforme Definição 3.7, pode-se dizer que o conjunto $\mathcal{B} = [S \rightarrow B]$ dos estados booleanos é interpretado pelo espaço coerente $\mathbb{S} \multimap \mathbf{Bool}$, definido logo a seguir.

Definição 10.6 *Considere o conjunto B de valores Booleanos na Definição 3.7. Considere também a teia discreta $\mathbf{Bool} \equiv (B, \cong)$, dada pelo conjunto de valores em B juntamente com a relação de igualdade. Defina-se o **domínio plano de Girard dos valores Booleanos** indicado por $\mathbf{Bool} = (\mathbf{Coh}(\mathbf{Bool}), \subseteq)$*

Em particular, tem-se $\mathbf{Coh}(\mathbf{Bool}) = \{\emptyset\} \cup \{\{V\}, \{F\}\}$ e $\emptyset \in \mathbf{Bool}$ interpreta o valor $bool \in B$.

Definição 10.7 *Considere os espaços coerentes \mathbf{Bool} e \mathbb{S} . Considere também a teia $\mathbf{S} \multimap \mathbf{Bool} = (|\mathbb{S}| \times |\mathbf{Bool}|, \approx_{\multimap})$ definida pelo produto cartesiano $S \times B$ e pela relação de coerência \approx_{\multimap} segundo a qual $(\sigma, \beta) \approx_{\multimap} (\sigma', \beta')$ se, e somente se, $\sigma \cong_{\mathbf{Bool}} \sigma' \Rightarrow (\beta \approx_{\mathbf{S}} \beta' \text{ e } (\beta = \beta' \Rightarrow \text{sigma} = \sigma'))$. O espaço coerente $\mathbf{S} \multimap \mathbf{Bool} \equiv (\mathbf{Coh}(\mathbf{S} \multimap \mathbf{Bool}), \subseteq)$ é construído pela coleção de todos os subconjuntos coerentes da teia $\mathbf{S} \multimap \mathbf{Bool}$ ordenados pela inclusão. Tem-se então que*

$$\mathbf{Coh}(\mathbf{S} \multimap \mathbf{Bool}) \equiv \{x \subseteq S \times B \mid (\sigma, \beta) \approx_{\multimap} (\sigma', \beta'), \forall (\sigma, \beta), (\sigma', \beta') \in x\}$$

Considerando as duas últimas definições, é possível interpretar cada teste computacional $b : S \rightarrow B \in \mathcal{B}$ pelo traço linear de uma função $\mathbf{b} : \mathbb{S} \rightarrow \mathbf{Bool}$, no caso, um conjunto de pares $\{(\sigma, \beta) \mid \beta = \mathbf{b}(\sigma)\} \in \mathbf{S} \multimap \mathbf{Bool}$.

10.2 Função de Avaliação

Define-se a seguir a função de avaliação que associa cada conjunto coerente em $\mathbb{D}_{\infty}^{\rightarrow}$ a uma função linear no conjunto coerente $\mathbb{S} \multimap \mathbb{S}$ das transformações de estados computacionais. Para tal, consideram-se as seguintes notações, apresentadas em seções anteriores.

1. \mathbb{I} indica o espaço coerente plano dos nomes das variáveis de cada estado computacional.
2. \mathbb{V} indica o espaço coerente dos valores das variáveis de cada estado computacional.
3. $\mathbb{I} \multimap \mathbb{V}$ indica o espaço coerente dos traços lineares de funções $s : \mathbb{I} \rightarrow \mathbb{V}$ onde

$$\bullet s \equiv \{(i, s(i))\}_{i \in J \subseteq I} \in \mathbb{I} \multimap \mathbb{V} \quad \text{e} \quad (i, s(i)) \in \text{ltr}(s) \subseteq I \times V.$$

4. $[\mathbb{I} \multimap \mathbb{V}] \multimap \mathbb{V}$ indica também um espaço coerente de traços lineares que modela o comportamento das ações do conjunto $\mathcal{A} = [\tilde{S} \rightarrow (I \times V)]$ apresentado na Seção 3.3.6. Neste caso, salienta-se que
 - $ltr[f]$ indica o traço linear da função $f : [\mathbb{I} \multimap \mathbb{V}] \rightarrow \mathbb{V}$ cujos elementos são indicados por pares $(s, v) = (\{(i, s(i))\}_{i \in J \subseteq I}, v) \in ltr[f]$;
 - para cada $s : \mathbb{I} \multimap \mathbb{V}$ existe $v \in \mathbb{V}$ e $f[s] = f[\{(i, s(i))\}_{i \in J \subseteq I}] = v \in \mathbb{V}$.
5. $\mathbb{S} \equiv (Coh(Coh(\mathbb{I} \multimap \mathbb{V}), \approx_{\mathbb{S}}), \subseteq)$ indica o espaço coerente dos estados computacionais. Qualquer família de subconjuntos coerentes em $\mathbb{I} \multimap \mathbb{V}$ determina um subconjunto coerente $\sigma \in \mathbb{S}$ e caracteriza um estado computacional possivelmente não-determinístico. Tem-se então que
 - $\sigma = \{s^{(m)} \mid s^{(m)} : \mathbb{I} \multimap \mathbb{V}, m \in M \subseteq \omega\} \subseteq Coh(\mathbb{S})$; e
 - $s^{(m)} = \{(i, s^{(m)}(i))\}_{i \in J \subseteq I} \in Coh(\mathbb{I} \multimap \mathbb{V})$.
6. $\mathbb{S} \multimap \mathbb{S}$ indica o espaço coerente dos traços lineares de funções $\mathbf{f} : \mathbb{S} \rightarrow \mathbb{S}$, sendo que cada conjunto coerente é dado pela expressão $\{(\sigma, \varsigma) \mid \mathbf{f}(\sigma) = \varsigma\} = ltr[\mathbf{f}]$. Assim, tem-se que para cada $\sigma \in \mathbb{S}$ existe $\varsigma \in \mathbb{S}$ tal que
 - $\mathbf{f}[\sigma] = f[\{s^{(m)} \mid s^{(m)} \in \sigma\}] = \varsigma = \{z^{(l)} = f(s^{(m)}) \mid z^{(l)} \in \varsigma\}$.
7. \mathbb{Bool} indica o espaço coerente plano dos valores booleanos.
8. $\mathbb{S} \multimap \mathbb{Bool}$ indica o espaço coerente que interpreta o comportamento dos testes computacionais modelados pelo espaço coerente \mathbb{B} .

Os espaços coerentes indicados na lista anterior são importantes na interpretação do comportamento dos processos modelados no domínio $\mathbb{D}_{\infty}^{\rightarrow}$.

Desde que estes processos são definidos por ações rotuladas, primeiramente se mostra como interpretar o conjunto das ações como os objetos do espaço coerente $[\mathbb{I} \multimap \mathbb{V}] \multimap \mathbb{V}$.

Definição 10.8 *Considere as funções lineares $pr^{(i)}, d : [\mathbb{I} \multimap \mathbb{V}] \multimap \mathbb{V}$ definidas pelas respectivas expressões $pr^{(i)}[s] = pr^{(i)}[\{(i, s(i))\}_{i \in J \subseteq I}] = s(i)$ e $d[s] = v$. A função $\mathbf{d}^k : [\mathbb{I} \multimap \mathbb{V}] \multimap [\mathbb{I} \multimap \mathbb{V}]$ é dada pela expressão*

$$\mathbf{d}^k[s] = \mathbf{d}^k[\{(i, s(i))\}_{i \in J \subseteq \omega}] = \{(i, z(i))\}_{i \in J' \subseteq J} = z,$$

sempre que cada função linear $z : \mathbb{I} \multimap \mathbb{V}$ é definida pela expressão

$$z(i) = \begin{cases} pr^{(i)}[s] = s(i) & \text{se } i \neq k, \\ d[s] = v & \text{se } i = k. \end{cases}$$

Proposição 10.1 *A função $\mathbf{d}^{(k)} : [\mathbb{I} \multimap \mathbb{V}] \multimap [\mathbb{I} \multimap \mathbb{V}]$ apresentada na Definição 10.1 é um morfismo na categoria $CospLin$.*

Prova. Sejam $s' = \mathbf{d}^{(k)}(s)$ e $z' = \mathbf{d}^{(k)}(z)$. Pelo que foi apresentado na seção 10.1.3, deve-se mostrar que se $(s, s'), (z, z') \in ltr(\mathbf{d}^{(k)}) \in [\mathbb{I} \multimap \mathbb{V}] \multimap [\mathbb{I} \multimap \mathbb{V}]$, as seguintes condições são satisfeitas:

1. $s \approx_{\multimap} z \Rightarrow s' \approx_{\multimap} z'$. De fato, suponha $s \approx_{\multimap} z$.
 - 1.1 Se $i \neq k$ tem-se que $\mathbf{d}^{(k)}(s)(i) = s(i) \approx_{\mathbb{V}} z(i) = \mathbf{d}^{(k)}(z)(i)$;
 - 1.2 Se $i = k$ tem-se que $\mathbf{d}^{(k)}(s)(k) = d(s)(k) \approx_{\multimap} d(z)(k) = \mathbf{d}^{(k)}(z)(k)$, pois a função $d : [\mathbb{I} \multimap \mathbb{V}] \multimap \mathbb{V}$ é linear, por hipótese.

Portanto, pelos dois itens acima tem-se que se $s \approx_{\multimap} z$ então $s' \approx_{\multimap} z'$.

2. $s' = z'$ e $s \approx_{\rightarrow} z \Rightarrow s = z$. Suponha que $s' = z'$ e $s \approx_{\rightarrow} z$.

2.1 Se $i \neq k$ então $s(i) = \mathbf{d}^{(k)}(s)(i) = s(i) = z'(i) = \mathbf{d}^{(k)}(z)(i) = z(i)$. Logo tem-se que $s = z$;

2.2 Se $i = k$ então $\mathbf{d}^{(k)}(s)(i) = \mathbf{d}^{(k)}(s)(k) = \mathbf{d}^{(k)}(s)(i) = d(s)(k) = d(z)(k) = \mathbf{d}^{(k)}(z)(k) = \mathbf{d}^{(k)}(z)(i)$ e $s = z$.

Portanto $\forall i \in I$ mostrou-se que $s = z$.

Mostrou-se assim que a função $\mathbf{d}^{(k)} : [\mathbb{I} \rightarrow \mathbb{V}] \rightarrow [\mathbb{I} \rightarrow \mathbb{V}]$ é linear. \blacktriangle

Definição 10.9 Considere o espaço coerente \mathbb{B} , que interpreta testes computacionais, apresentado na Definição 4.17. Considere também o espaço coerente $\mathbb{S} \multimap \mathbf{Bool}$ dos traços lineares das funções definidas do espaço coerente de estados computacionais para o espaço coerente de valores booleanos, apresentada na Definição 10.7.

A função $eval_{bool} : \mathbb{B} \rightarrow [\mathbb{S} \multimap \mathbf{Bool}]$ associa cada conjunto coerente $b \in \mathbb{B}$ a uma função linear $\mathbf{b} : \mathbb{S} \multimap \mathbf{Bool}$ de tal forma que

$$eval_{bool}(b)(\sigma) = \mathbf{b}(\sigma) = \beta \in \mathbf{Bool}.$$

Proposição 10.2 A função $eval_{bool} : \mathbb{B} \rightarrow [\mathbb{S} \multimap \mathbf{Bool}]$ apresentada na Definição 10.9 é um morfismo na categoria $CospLin$.

Prova. Imediata, considerando-se o domínio plano \mathbb{B} . \blacktriangle

A definição da função de avaliação é dada a seguir.

Definição 10.10 Considere os espaços coerentes $\mathbb{S} \multimap \mathbb{S}$, $\mathbb{D}^{\rightarrow \infty}$ e as funções $\Pi_{\infty, n+1}^{\rightarrow}$, $\Pi_{\infty, n+1}^{\rightarrow}$ e $\pi_{n+1, \infty}^{(00)}[\mathbb{D}_{n+1}] \subseteq \pi_{n+1, \infty}^{\rightarrow}$. Considere também $x = \bigcup^{\uparrow} x_n$ onde $x_n \in \pi_{n, \infty}^{(\theta)}[\mathbb{D}_n]$ e $\theta \in \{00, 001, 012\}$. A **função de avaliação**, indicada por

$$Eval : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{S} \multimap \mathbb{S},$$

interpreta o comportamento dos processos modelados por subconjuntos em $\mathbb{D}_{\infty}^{\rightarrow}$, e sua definição é apresentada nos próximos três itens, determinados pelo indexador θ .

1 Suponha primeiramente que $\theta = 00$. Tem-se

$$Eval(x) = \bigcup^{\uparrow} eval(x_n)$$

onde $eval : \pi_{n, \infty}^{\rightarrow}[\mathbb{D}_n] \subseteq \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{S} \multimap \mathbb{S}$ é uma função recursivamente definida pelas expressões apresentadas logo abaixo.

1.1 Seja $x_0 \in \mathbb{D}_{\infty}^{\rightarrow}$ tal que $x_0 \in \pi_{0, \infty}^{(00)}[\mathbb{D}_0]$.

• Se $x_0 = \emptyset$,

$$\bullet eval(\emptyset) = Id_{\mathbb{S} \multimap \mathbb{S}}, \text{ logo } eval(\emptyset)(\sigma) = \sigma, \forall \sigma \in \mathbb{S}.$$

• Se $x_0 = \{d_{;00}^{(k)}\}$ interpreta um processo elementar e a função $\mathbf{d}^{(k)} \in [\mathbb{I} \multimap \mathbb{V}] \multimap [\mathbb{I} \multimap \mathbb{V}]$ apresentada na Definição 10.8, satisfaz a Proposição 10.1, tem-se

$$\bullet eval(\{d^{(k)}\}) : \mathbb{S} \multimap \mathbb{S}, \text{ onde } eval(\{d^{(k)}\})(\sigma) = \{\mathbf{d}^{(k)}(s^{(m)}) \mid s^{(m)} \in \sigma\}$$

1.2 Seja $x_{n+1} \in \mathbb{D}_{\infty}^{\rightarrow}$ tal que $x_{n+1} \in \pi_{n+1, \infty}^{(00)}[\mathbb{D}_{n+1}]$.

- Se $x_{n+1} \in \mathbf{F}_{00}(\mathbb{D}_{\infty}^{\rightarrow})$ e verifica-se a condição

$$(\gamma_n^{(0)} \circ \phi_n^{(0)} \circ \Pi_{\infty, n}^{\rightarrow})(x_n) = \Pi_{\infty, n+1}^{\rightarrow}(x_{n+1}),$$

significando que $x_{n+1} = \pi_n^{(00)}(x_n)$

- $eval(x_{n+1}) = eval(x_n)$

- Se $x_{n+1} \in \mathbf{F}_{(10)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$ e verifica-se a condição

$$(\Phi_n^{(1)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \bigcup_{k \in \omega} \overline{\Pi_{\infty, n}^{\rightarrow}(x_n^{(k)})},$$

significando que $x_{n+1} = \{x_n^{(k)}_{10:00} \mid \bigcap_{k \in \omega} \Upsilon_n(x_n^{(k)}) = \emptyset\}$, tem-se

- $eval(x_{n+1}) : \mathbb{S} \rightarrow \mathbb{S}$, $eval(x_{n+1})(\sigma) = \{\xi_{(10)}(s^{(m)}) \mid s^{(m)} \in \sigma\}$.

Neste caso, $\xi_{(10)}(s^{(m)}) : \mathbb{I} \rightarrow \mathbb{V}$ denota a função linear dada por

- $\xi_{(10)}(s^{(m)})(i) = z^{(m)}(i) = \begin{cases} eval(x_n^{(k)})(s^{(m)})(i) & \text{se } i \in \Upsilon_n(x_n^{(k)}), \\ Id_{\mathbb{V}}(s^{(m)})(i) & \text{caso contrário.} \end{cases}$

- Se $x_{n+1} \in \mathbf{F}_{(20)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$ e verifica-se a condição

$$(\Phi_n^{(2)} \circ \Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = \bigcup_{k \in \omega} (\gamma_n^{(0)} \circ \phi_n^{(2)}) (\overline{\Pi_{\infty, n}^{\rightarrow}(x_n^{(k)})}),$$

significando que $x_{n+1} = \{x_n^{(k)}_{20:00} \mid \bigcap_{k \in \omega} \Upsilon_n(x_n^{(k)}) \neq \emptyset\}$, tem-se

- $eval(x_{n+1}) : \mathbb{S} \rightarrow \mathbb{S}$, $eval(x_{n+1})(\sigma) = \{eval(x_n^{(k)})(s^{(m)}) \mid s^{(m)} \in \sigma\}$.

Salienta-se que nos dois últimos itens, k indica uma coleção enumerável e possivelmente infinita de subconjuntos x_n .

- Se $x_{n+1} \in \mathbf{F}_{(01)(11)}(\mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$ tal que as seguintes condições são satisfeitas

$$(1) (\Lambda_n^{(0)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_n^{(0)}) \quad e$$

$$(2) (\Lambda_n^{(1)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_n^{(1)}),$$

significando que $x_{n+1} = \mathbf{F}_{(01)(11)}(x_n^{(0)} \sqcap x_n^{(1)})$, tem-se que

- $eval(x_{n+1}) = eval(x_n^{(1)}) \circ (eval x_n^{(0)})$.

Por outro lado, quando são consideradas as funções lineares

$$(1) \xi_{n+1}^{(01)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}, \quad \xi_{n+1}^{(01)} = (\pi_{n+1, \infty}^{(00)} \circ \gamma_n^{(0)}) \circ (\Psi_n^{(0)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow}) \quad e$$

(2) $\xi_{n+1}^{(11)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$, $\xi_{n+1}^{(11)} = (\pi_{n+1, \infty}^{(00)} \circ \gamma_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow})$, $eval(x_{n+1}) : \mathbb{S} \rightarrow \mathbb{S}$ é definida pela expressão

- $eval(x_{n+1}) = eval((\xi_{n+1}^{(11)})(x_{n+1}) \circ eval(\xi_{n+1}^{(01)})(x_{n+1}))$.

Comparando-se as definições apresentadas neste item, tem-se as igualdades

$$((\xi_{n+1}^{(01)})(x_{n+1})) = x_n^{(0)} \quad e \quad ((\xi_{n+1}^{(11)})(x_{n+1})) = x_n^{(1)}.$$

- Se $x_{n+1} \in \mathbf{F}_{(02)(12)(22)}(\mathbb{B} \amalg \mathbb{D}_{\infty}^{\rightarrow} \amalg \mathbb{D}_{\infty}^{\rightarrow})$ satisfazendo as condições

$$(1) (\Lambda_n^{(0)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_n^{(0)}),$$

$$(2) (\Lambda_n^{(1)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Gamma_n^{(0)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_n^{(1)}) \quad e$$

$$(3) (\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1}) = (\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(b),$$

significando que $x_{n+1} = \mathbf{F}_{(02)(12)(22)}(x_{n+1}^{(0)} \sqcap x_n^{(1)} \sqcap b)$, tem-se que

$$\bullet \text{ eval}(x_{n+1}) = \begin{cases} \text{eval}(x_n^{(0)}) & \text{se } \text{eval}_{\text{bool}}(\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(b) = \{V\}, \\ \text{eval}(x_n^{(1)}) & \text{se } \text{eval}_{\text{bool}}(\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(b) = \{F\}, \\ \text{Id}_{\mathbb{S} \rightarrow \mathbb{S}} & \text{se } \text{eval}_{\text{bool}}(\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(b) = \emptyset. \end{cases}$$

Por outro lado, quando

$$(1.1) \quad \text{eval}_{\text{bool}}(\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1})(\sigma) = \{V\} \text{ e}$$

$$(1.2) \quad \xi_{n+1}^{(02)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}, \xi_{n+1}^{(02)} = (\pi_{n, \infty}^{(00)} \circ \gamma_n^{(0)}) \circ (\Lambda_n^{(0)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow}), \text{ então}$$

$$\bullet \text{ eval}(x_{n+1}) = \text{eval}(\xi_{n+1}^{(02)})(x_n).$$

Se as seguintes condições são válidas,

$$(2.1) \quad \text{eval}_{\text{bool}}(\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1})(\sigma) = \{F\} \text{ e}$$

$$(2.2) \quad \xi_{n+1}^{(12)} : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}, \xi_{n+1}^{(12)} = (\pi_{n, \infty}^{(00)} \circ \gamma_n^{(0)}) \circ (\Lambda_n^{(1)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow}), \text{ então}$$

$$\bullet \text{ eval}(x_{n+1}) = \text{eval}(\xi_{n+1}^{(12)})(x_n).$$

Por fim, quando

$$(3) \quad \text{eval}_{\text{bool}}(\Lambda_n^{(2)} \circ \Gamma_n^{(2)} \circ \Pi_{\infty, n+1}^{\rightarrow})(x_{n+1})(\sigma) = \emptyset \text{ então}$$

$$\bullet \text{ eval}(x_{n+1}) = \text{Id}_{\mathbb{S} \rightarrow \mathbb{S}}.$$

Comparando as expressões apresentadas nesta definições, tem-se as igualdades

$$(\xi_{n+1}^{(02)})(x_n) = x_{n+1}^{(0)} \quad \text{e} \quad (\xi_{n+1}^{(12)})(x_n) = x_{n+1}^{(1)}.$$

2 Seja $\theta = 001$.

$$\bullet \mathcal{E}val(x) = (\odot_{i=1}^{\infty} \text{eval} \xi_i^{(1)} \circ \text{eval} \xi_0)(x), \quad \text{ou ainda}$$

$$\mathcal{E}val(x) = (\dots \circ \text{eval} \xi_n^{(1)} \circ \dots \circ \text{eval} \xi_1^{(1)} \circ \text{eval} \xi_0)(x),$$

sempre que as funções $\xi_n : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{D}_{\infty}^{\rightarrow}$ são definidas pelas expressões:

$$\bullet \xi_0 = (\pi_{0, \infty}^{(00)} \circ \Pi_{\infty, 0}^{\rightarrow})(x),$$

$$\bullet \xi_1 = \text{eval}((\pi_{1, \infty}^{(00)} \circ \psi_0^{(0)}) \circ (\Psi_0^{(1)} \circ \Gamma_0^{(1)} \circ \Pi_{\infty, 1}^{\rightarrow}))(x),$$

\vdots

$$\bullet \xi_{n+1} = \text{eval}((\pi_{n+1, \infty}^{(00)} \circ \psi_n^{(0)}) \circ (\Psi_n^{(1)} \circ \Gamma_n^{(1)} \circ \Pi_{\infty, n+1}^{\rightarrow}))(x).$$

Em particular, se $x \in \pi_{n+1, \infty}^{(001)}[\mathbb{D}_{n+1}]$, tem-se

$$\bullet \mathcal{E}val(x) = (\odot_{i=1}^{n+1} \text{eval} \xi_i) \circ (\text{eval} \xi_0)(x).$$

3 Seja $\theta = 002$.

$$\bullet \mathcal{E}val(x) = \odot_{i=0}^{\infty} \text{eval}_{\text{bool}}(\Lambda_i^{(2)} \circ \Gamma_i^{(2)} \circ \Pi_{\infty, i+1}^{\rightarrow})(x).$$

Proposição 10.3 A função de avaliação $\mathcal{E}val : \mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{S} \rightarrow \mathbb{S}$, apresentada na Definição 10.10 constituiu-se num morfismo na categoria $\text{Cosp}\mathcal{L}in$.

A prova desta proposição não será apresentada neste texto, como também não se mostra que a função de avaliação $\mathcal{E}val$ é o menor ponto fixo para um operador Θ sobre a família de funções $[\mathbb{D}_{\infty}^{\rightarrow} \rightarrow \mathbb{S} \rightarrow \mathbb{S}]$.

11 Considerações Finais

Utilizando a Teoria dos Domínios, mais precisamente os domínios qualitativos caracterizados pela compatibilidade binária e denominados espaços coerentes, este trabalho introduz o modelo de máquina geométrica.

A máquina geométrica é um modelo abstrato, admitindo memória infinita com tempo de acesso à memória constante, características não compatíveis com a realidade. Entretanto, mostra-se que este modelo de máquina constitui-se num eficiente instrumento para estudo da estrutura lógica e ordenada de computações paralelas e não-determinísticas. Ou seja, neste contexto, o desenvolvimento de algoritmos para a máquina geométrica pode fundamentar o desenvolvimento de algoritmos para máquinas reais.

Uma dos principais objetivos alcançados neste trabalho, e que lhe confere um caráter criativo, foi a formalização do modelo de máquina geométrica distribuída, como uma extensão desta estrutura ordenada e indutiva, capaz de interpretar também processos computacionais transfinitos.

11.1 Conclusões

O resumo das principais conclusões alcançadas e algumas sugestões para a continuidade deste trabalho são apresentadas logo a seguir.

11.1.1 Modelo MG

Inspirada na teoria dos sistemas dinâmicos, a idéia intuitiva deste trabalho é a modelagem de uma máquina por uma estrutura matemática, onde os conceitos básicos como memória e processadores são construídos a partir de um espaço geométrico. Mais precisamente, o conjunto S de estados e o conjunto \mathcal{P} de processos são, respectivamente, definidos pelos conjuntos de valores e de ações elementares, ambos rotulados por posições de um espaço geométrico.

Os pontos do espaço geométrico podem ser entendidos como posições de memória a partir dos quais são definidos os estados computacionais sem restrição quanto ao seu tamanho. Baseado nisto, a noção de processo computacional está relacionada com a noção de transformação entre estados de computação. Compatível com a idéia de que a memória não é necessariamente limitada, a máquina modela processadores sem restrição quanto ao seu tamanho.

Quando, a partir de um estado inicial cada transformação alcança um estado final, o tempo de execução associado a tal transformação é indicado por um número finito de unidades de tempo computacional. Cada unidade de tempo computacional, indicada pela expressão *utc* modela a noção de um ciclo do relógio. O modelo de máquina que se está propondo contempla interpretação tanto para estas transformações como para as interpretações infinitas.

A interpretação da computação executada por um processo infinito é representada por uma coleção de funções, sendo que cada função desta coleção interpreta o comportamento deste processo em um determinado tempo computacional e constitui-se numa aproximação ou representação parcial. Entretanto, o estado final só é alcançado em tempo infinito, formalmente definido como o limite de suas computações parciais, caracterizando a completação do modelo de máquina.

A noção fundamental de processo computacional não se restringe a condição unidimensional, podendo ser estendido para manipulação de estruturas multi-dimensionais, geometricamente representadas.

O conjunto \mathcal{P} de processos modelados é obtido a partir do conjunto dos processos elementares, cuja execução altera apenas uma posição de memória, e pelo conjunto dos construtores de processos, identificados como os produtos paralelo e seqüencial e as somas determinísticas e não-determinísticas.

A estrutura ordenada que modela a construção do conjunto \mathcal{P} de processos é indutivamente construída seguindo a metodologia sugerida por Scott [SCO 71]. Nesta abordagem, o domínio semântico base da estrutura construtiva deste modelo de máquina é o espaço coerente \mathbb{D}_0 dos processos elementares, construído a partir do relacionamento baseado na coerência binária entre unidades atômicas de informações da teia \mathbf{D}_0 - grafo não-dirigido cujos tokens são os processos elementares rotulados por pontos do espaço geométrico I . Neste sentido, cada token da teia \mathbf{D}_0 é uma estrutura matemática interpretando um processo computacional elementar.

Tomando-se como base o espaço coerente \mathbb{D}_0 dos processos elementares, define-se o espaço coerente \mathbb{D}_1 , no qual são modelados todos os processos em \mathcal{P} executados em no máximo duas unidades de tempo computacional (2^1 utc).

A metodologia de construção do primeiro nível $\mathbb{D}_0 - \mathbb{D}_1$ é repetida na construção dos demais níveis. Em $\mathbb{D}_n - \mathbb{D}_{n+1}$ são interpretados os processos que executam no máximo 2^{n+1} operações externas (produto seqüencial ou soma determinística) ou um número arbitrário de operações internas (soma não-determinística ou produto paralelo) interpretadas como objetos construídos em \mathbb{P}_n . Neste caso, \mathbb{P}_n é obtido pela aplicação do operador categórico soma direta, que justapõe os subespaços coerentes \mathbb{D}_n , $\bar{\mathbb{D}}_n$ e $\bar{\mathbb{D}}_n^\perp$.

A relação de coerência que constrói a teia sobre a qual se define o espaço coerente $\bar{\mathbb{D}}_n$ é obtida a partir da definição recursiva da função posição $\Upsilon_{\mathbb{D}_n}$, ao explicitar o conjunto de posições de memória alteradas por cada processo, identificando a concorrência verdadeira. E no espaço dual, $\bar{\mathbb{D}}_n^\perp$ tem-se a interpretação para a relação de conflito de acesso a posições na memória, característica das escolhas não-determinísticas.

Por fim, a aplicação do operador produto direto modela os construtores produto seqüencial e, considerando-se o espaço coerente \mathbb{B} dos testes computacionais, a escolha determinística.

11.1.2 Diversidade dos Modelos MG

A extensão do espaço coerente \mathbb{D}_n de processos finitamente gerados é formalizada com a construção e definição dos objetos e morfismos (imersões e projeções) do espaço coerente \mathbb{D}_∞ e pode ser modelada de diferentes maneiras, cada uma delas correspondendo a construção temporal que modela a seqüencialidade.

Este novo espaço coerente, obtido pela completção, também provê interpretação para todos os processos em \mathcal{P} , seja pela reconstrução dos objetos já definidos nos níveis \mathbb{D}_n , seja pela construção de novos objetos, aqueles que interpretam processos cuja execução não pode ser limitada no tempo.

No que se refere à construção, todos os objetos em \mathbb{D}_∞ são definidos por suas aproximações - subconjuntos coerentes que interpretam processos parciais cuja execução se efetiva em tempo finito. Entretanto, isto não significa que tais objetos devam possuir representação finita no espaço geométrico ou no conjunto de ações.

Neste contexto, os novos objetos são obtidos como o limite inverso de seqüências de projeções lineares definidas do espaço coerente \mathbb{D}_∞ sobre cada subespaço \mathbb{D}_n .

Usufruindo desta construção geométrica e essencialmente posicional, a construção temporal que leva à completção da estrutura indutiva deste modelo de máquina pode admitir diferentes abordagens, introduzindo diferentes domínios. As diferenças significativas entre tais domínios relacionam-se diretamente com as interpretações da seqüencialidade e da escolha determinística dos processos computacionais. Salienta-se ainda que cada domínio identifica uma enumeração distinta dos indexadores dos tokens de seus objetos e

de seus esquemas de recursão.

Embora a ênfase neste texto seja na definição e estudo mais detalhado do espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$, onde são modelados os processos cuja construção é obtida a partir de prefixos, mostra-se também outros domínios. Por exemplo,

- em $\mathbb{D}_{\infty}^{\leftarrow}$ tem-se a interpretação da construção de processos por sufixo;
- em $\mathbb{D}_{\infty}^{\leftrightarrow}$ interpretam-se os processos construídos por infixo;
- em $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$ dos processos pela construção por sufixo e prefixo, simultaneamente.
- e em $\mathbb{D}_{\infty}^{\leftrightarrow\leftarrow}$, cujos processos são construídos por sufixo, prefixos e infixo, simultaneamente.

Assim como estes, muitos outros domínios podem ser definidos, comprovando a diversidade dos modelos de máquinas geométricas. Esta diversidade não se restringe apenas ao número de processos elementares ou ao espaço geométrico associado aos mesmos, mas pode estender-se muito amplamente, de acordo com o tipo de construção temporal que se tenciona modelar. Isto é uma consequência imediata da forma criativa de definir as imersões e projeções que formalizam o relacionamento entre os níveis da construção indutiva.

11.1.3 Modelo MGD

Considerando que a indexação dos tokens dos subconjuntos coerentes em $\mathbb{D}_{2\infty}$ possui uma construção análoga àquela dos ordinais transfinitos, sugerida por Stoll em [STO 61], este modelo é capaz de formalizar a noção de **seqüencialidade espacial**, relacionada com a modelagem de processos distribuídos a partir do conjunto enumerável \mathcal{M} de máquinas localizadas num espaço \mathcal{E} , cuja estrutura matemática é formalizada pelo **espaço coerente $\mathbb{D}_{2\infty}$ dos processos transfinitos**.

Assim, os objetos em $\mathbb{D}_{2\infty}$ são subconjuntos coerentes de tokens indexados a partir de um conjunto isomorfo aos ordinais transfinito. Pela indexação dos tokens torna-se explícito quais posições de memória são afetadas pela ação associada, qual o construtor modelado e ainda mais, em que máquina o processo parcial será executado.

Nesta nova abordagem, os processos são determinados pelas respectivas posições no conjunto I de indexadores do espaço geométrico que modela a memória de cada máquina e pelos indexadores indentificando cada uma das máquinas no conjunto enumerável \mathcal{M} .

As máquinas em \mathcal{M} estão **sincronizadas** e seus estados de memória interpretados por subconjuntos coerentes de espaços coerentes isomorfos. Isto indica que neste modelo, os estados computacionais nas diferentes máquinas devem ser modelados por objetos isomorfos nos domínios de interpretação de cada máquina.

11.2 Propostas para Continuação dos Trabalhos

Na última seção são sugeridos diversos temas para a continuidade deste trabalho de pesquisa.

11.2.1 Relacionamento entre as Memórias Infinita e Transfinita no modelo MG

Este trabalho de pesquisa mostrou que a definição de máquina com memória distribuída pode ser alcançada por generalização do processo de construção das máquinas com memória compartilhada, na busca de um modelo mais intuitivo para os computadores paralelos, que executem computações concorrentes e não-determinísticas.

Baseando-se na intuitiva idéia desta construção, centrada na construção dos ordinais transfinitos como generalização do processo de geração dos ordinais, surge naturalmente a possibilidade de simulação da máquina geométrica distribuída caracterizada por uma memória transfinita, por uma máquina geométrica distribuída com memória infinita.

Esta possibilidade está fundamentada no processo de diagonalização de Cantor, onde se prova, a partir do isomorfismo entre $2^N \equiv \wp(N)$, que o conjunto 2^N das funções definidas de N para $\{0, 1\}$ não é enumerável.

Em termos computacionais, estas simulações são justificadas considerando-se os diferentes objetivos na modelagem de algoritmos. Ou seja, a característica das máquinas distribuídas de múltipla memória pode tornar mais flexível o desenvolvimento de algoritmos para máquinas paralelas, mas, em contrapartida, pode dificultar a análise e a medida de sua complexidade.

Estas considerações se relacionam com a possibilidade de aplicação deste modelo para fundamentação da análise de algoritmos, independentemente da arquitetura das diversas máquinas paralelas (SIMD, MIMD, SISD) conforme comentado na Seção 2.1.2 do Capítulo 2. Dentro da teoria clássica da computação, o modelo de computação para análise da computabilidade de algoritmos está fundamentada na Máquina de Turing, veja Seção 2.1.1 do Capítulo 2.

Com base nas considerações dos parágrafos anteriores, parece sugestivo o estudo e a análise mais detalhada de simulações envolvendo os modelos de máquinas tradicionais na literatura (incluindo as versões da máquina de Turing) e seus relacionamentos com o modelo de máquina geométrica (distribuída).

11.2.2 Classificação do Modelo MG na Hierarquia dos Níveis de Abstração para Modelos Semânticos

Numerosos e diferenciados são os modelos sugeridos e analisados para fundamentação e desenvolvimento em sistemas de computação distribuídos, desde a pesquisa até a aplicação prática. Em [WHI 95], algumas características importantes são analisadas e definem uma hierarquia dos níveis de abstração destes sistemas, possibilitando uma classificação formal que cresce no sentido dos modelos mais concretos para os mais abstratos. Nesta hierarquia, a magnitude de um modelo é medida pela sua abstração, associada a três fundamentais características:

- a modelagem do sistema em termos dos padrões seqüenciais das ações, expressando o paralelismo por escolhas não-determinísticas (*interleaving*);
- a modelagem do tempo associado às computações do sistema, classificando-o em linear ou ramificado;
- a modelagem explícita dos estados do sistema, não se concentra apenas na análise do comportamento das ações deslocadas no tempo.

Com base nestas características, o modelo de máquina geométrica, de acordo com o que foi mostrado nos capítulos que antecedem esta conclusão, se caracteriza por modelar a concorrência síncrona, por conflito de acesso a memória. Além disso, a modelagem temporal é explicitada na construção indutiva dos níveis da estrutura ordenada, interpretada pelos subespaços coerentes dos processos cuja execução é limitada no tempo. Portanto uma modelagem ramificada, mesmo se consideradas as interpretações para processos mutuamente exclusivos ou não-determinísticos. Por fim, é imediato que o modelo torna explícito os estados, definidos por um conjunto de valores de variáveis rotuladas por pontos de um espaço geométrico.

Por todas estas considerações, o modelo de máquina geométrica constitui-se numa construção essencialmente concreta, pois caracteriza-se pela concorrência síncrona, com

memória indexada por pontos de um espaço geométrico e modelagem temporal expressa na construção indutiva.

A formalização do relacionamento com outros modelos como a máquina *PRAM*, ou ainda aqueles mais abstratos onde a estrutura de memória não esta modelada - as estruturas de eventos, os sistemas de transição, as árvores de sincronização, as redes de Petri e tantos outros - constitui-se num interessante objeto de pesquisa, que necessariamente implica na construção da Categoria das Máquinas Geométricas, cujos objetos são os modelos de máquinas geométricas e os morfismos são mapeamentos entre tais modelos máquinas.

De acordo com o sugerido em [WHI 95] , a linguagem categórica constitui-se numa poderosa ferramenta matemática de abstração e formalização de relacionamentos entre diversas estruturas relevantes para a Teoria da Computação, incluindo-se os diferentes e mais fundamentais modelos de computação distribuídos.

11.2.3 Interpretações para Construções Recursivas

Sempre que o espaço geométrico I , a partir do qual se definem os processos e a memória de cada máquina geométrica, pode ser descrito por um processo de geração, a partir de funções básicas e funções sucessoras, a concorrência responsável pelo definição do paralelismo espacial, pode ser explicitada por equações recursivas, que correspondem ao menor ponto fixo para o operador produto paralelo.

Por outro lado, a natureza indutiva do modelo e sua completação asseguram interpretação para os construtores produto seqüencial e soma determinística, através de operadores lineares recursivamente definidos sobre o espaço coerente $\mathbb{D}_{\infty}^{\rightarrow}$. Portanto, o modelo também contempla interpretação para recursão no sentido temporal.

Finalmente, considerando-se a indução transfinita, é possível estender a interpretação para a concorrência temporal. No espaço coerente $\mathbb{D}_{\infty}^{\rightarrow m}$, os construtores de processos distribuídos são interpretados por operadores recursivos, supondo-se um sistema de computação formado por um conjunto finito e enumerável de m máquinas geométricas. Para dar continuidade a esta interpretação, pode-se pensar que em $\mathbb{D}_{2\infty}^{\rightarrow}$ têm-se um modelo de máquina distribuída flexível o bastante para suportar a inclusão de novas máquinas ao sistema , ou de um subconjunto enumerável e infinito de máquinas. Generalizando-se, pode formalizar uma estrutura indutiva de construção de sistemas de computação distribuídos, modelando o paralelismo temporal, onde cada unidade do sistema constitui-se numa máquina geométrica, modelando o paralelismo temporal.

Neste trabalho, apresentou-se a abordagem mais simplificada, onde a indexação dos tokens de um conjunto coerente modelando a seqüencialidade de um processo interpretado em $\mathbb{D}_{\infty}^{\rightarrow}$ está relacionada com o sistema unário dos números naturais. De forma similar, a enumeração das máquinas também pode ser pensada em $\mathbb{D}_{2\infty}^{\rightarrow}$. Entretanto, quando esta análise é feita sobre outros espaços coerentes, $\mathbb{D}_{\infty}^{\leftrightarrow}$, $\mathbb{D}_{\infty}^{\rightarrow\leftarrow}$, $\mathbb{D}_{\infty}^{\leftarrow\rightarrow}$, torna-se necessário considerar outros sistemas algébricos como por exemplo, os sistemas binários com dois centros.

Pelo que foi exposto ao longo deste trabalho, parece ser interessante e até mesmo relevante, um estudo em termos categóricos, detalhando o relacionamento entre as estruturas ordenadas que definem os diversos modelos de máquinas geométricas (incluindo as distribuídas) e os sistemas algébricos n -ários que formalizam os esquemas de recursão visando uma abordagem mais generalizada para a definição das interpretações dos construtores de processos.

11.2.4 Abordagem Semântica Baseada na Teoria dos Tipos

Girard [GIR 86] introduz os espaços coerente como modelo semântico para os sistemas \mathbf{F} , fazendo uso de uma linguagem de tipos variáveis para expressar tais interpretações. Motivados por esta idéia, e a correspondente semântica denotacional apresentada por Abramsky [ABR 90], apresenta-se no Capítulo 10 a função de avaliação, capaz de expressar as propriedades das estruturas computacionais modeladas pelo espaço coerente de processos, fundamentada no mesmo paradigma seguido por Girard - o isomorfismo de Curry-Howard.

Considerando-se a natureza indutiva da construção do modelo de máquina apresentado neste texto, e a abordagem sugerida no parágrafo anterior, pode-se construir um operador linear definido sobre o espaço funcional, um morfismo na categoria $CopsLin$, para o qual a função de avaliação $Eval$ constitui-se no menor ponto fixo.

Pode-se pensar numa extensão da semântica denotacional para o modelo de máquina geométrica distribuída, com um particular objetivo, a análise do comportamento de processos caracterizados pela distribuição.

11.2.5 Extensão da Linguagem com Representação para Processos Distribuídos

A utilização da Teoria dos Domínios para a fundamentação e interpretação semântica de uma linguagem de programação de alto nível, se justifica pela construção de algoritmos compatíveis com as máquinas paralelas reais e de validação mais sistemáticas.

Os resultados apresentados no Capítulo 9 constituem-se nos primeiros passos para a definição de uma linguagem de programação construída a partir das interpretações obtidas da estrutura ordenada da máquina geométrica.

Embora não se tenha, até o presente momento, priorizado a construção da gramática capaz de gerar todas as expressões válidas e nem a construção do processo de validação para estas expressões, tais situações são objetivos importantes e devem merecer uma análise, constituindo-se em novas propostas para continuidade deste trabalho.

11.2.6 Aplicações do Modelo na Computação Científica

A máquina de Turing (padrão) constitui-se num dos modelos de computação irrestritos, e portanto uma poderosa e bem sucedida ferramenta na fundamentação da teoria clássica da computação, referência para o conceito de função computável. Entretanto, o mesmo não se verifica quando se trata da teoria moderna da computação científica.

Alguns trabalhos na literatura específica [BLU 98] buscam uma teoria da computação que integre os fundamentos da teoria clássica mas que, simultaneamente, sejam diretamente aplicáveis aos problemas da matemática, da análise numérica e da computação científica. Esta trabalho desenvolveu-se no sentido de contribuir nesta busca.

A transformação de uma máquina RAM em uma máquina de Turing (determinística) é uma procedimento imediato, da mesma forma que sua transformação inversa, embora seja mais detalhada [BOV 94]. O mesmo pode ser alcançado para a máquina PRAM. Portanto, parece que este procedimento de transformação pode ser formalizado considerando-se o modelo de máquina geométrica.

Além disso, como instrumento essencial de comprovação da relevância dos estudos teóricos sugeridos neste trabalho está a aplicação da linguagem e da correspondente semântica associada às interpretações do espaço coerente \mathbb{D}_∞ na interpretação de algoritmos da matemática intervalar que usam processamento vetorial em sua execução, e na teoria dos autômatos e suas aplicações [AGU 2001].

Anexo Aplicações da Função de Representação

TABELA A.1 - Representações Finitas com Processo *skip*

-
0. $\mathfrak{R}[\emptyset] = skip$
- $\star : 00 = 00.00. \dots$
1. $\mathfrak{R}[\{\emptyset_{10:00}\}] = \parallel_{-1}, skip, -1 \parallel$
2. $\mathfrak{R}[\{\emptyset_{101:00}\}] = \parallel_{-1}, skip, -1 \parallel \cdot_{-1}$
3. $\mathfrak{R}[\{\emptyset_{111:00}\}] = -_{-1} \cdot \parallel_{-1}, skip, -1 \parallel$
4. $\mathfrak{R}[\{\emptyset_{102:00}\}] = \parallel_{-1}, skip, -1 \parallel +_{-bool} -_{-1}$
5. $\mathfrak{R}[\{\emptyset_{112:00}\}] = -_{-1} +_{\perp} \parallel_{-1}, skip, -1 \parallel$
6. $\mathfrak{R}[\{\emptyset_{20:00}\}] = | skip |$
7. $\mathfrak{R}[\{\emptyset_{201:00}\}] = | skip | \cdot_{-1}$
8. $\mathfrak{R}[\{\emptyset_{211:00}\}] = -_{-1} \cdot | skip |$
9. $\mathfrak{R}[\{\emptyset_{202:00}, b_{22:00}\}] = | skip | +_b -$
10. $\mathfrak{R}[\{\emptyset_{212:00}, b_{22:00}\}] = -_{-1} +_b | skip |$
11. $\mathfrak{R}[\{\emptyset_{10.001:00}\}] = \parallel_{-1}, skip, -1 \parallel \cdot_{-2}$
12. $\mathfrak{R}[\{\emptyset_{101.001:00}\}] = (\parallel_{-1}, skip, -1 \parallel \cdot_{-1}) \cdot_{-2}$
13. $\mathfrak{R}[\{\emptyset_{111.001:00}\}] = (-_{-1} \cdot \parallel_{-1}, skip, -1 \parallel) \cdot_{-2}$
14. $\mathfrak{R}[\{\emptyset_{102.001:00}\}] = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -_{-1}) \cdot_{-2}$
15. $\mathfrak{R}[\{\emptyset_{112.001:00}\}] = (-_{-1} +_{-bool} \parallel_{-1}, skip, -1 \parallel) \cdot_{-2}$
16. $\mathfrak{R}[\{\emptyset_{20.001:00}\}] = | skip | \cdot_{-2}$
17. $\mathfrak{R}[\{\emptyset_{201.001:00}\}] = (| -skip | \cdot_{-1}) \cdot_{-2}$
18. $\mathfrak{R}[\{\emptyset_{211.001:00}\}] = (-_{-1} \cdot | skip |) \cdot_{-2}$
19. $\mathfrak{R}[\{\emptyset_{202.001:00}\}] = (| -skip | +_{-bool} -) \cdot_{-2}$
20. $\mathfrak{R}[\{\emptyset_{212.001:00}\}] = (-_{-1} +_{-bool} | skip |) \cdot_{-2}$
21. $\mathfrak{R}[\{\emptyset_{10.011:00}\}] = -_{-2} \cdot \parallel_{-1}, skip, -1 \parallel$
22. $\mathfrak{R}[\{\emptyset_{101.011:00}\}] = -_{-2} \cdot (\parallel_{-1}, skip, -1 \parallel \cdot_{-1})$
23. $\mathfrak{R}[\{\emptyset_{111.011:00}\}] = -_{-2} \cdot (-_{-1} \cdot \parallel_{-1}, skip, -1 \parallel)$
24. $\mathfrak{R}[\{\emptyset_{102.011:00}\}] = -_{-2} \cdot (\parallel_{-1}, skip, -1 \parallel +_{-bool} -_{-1})$
25. $\mathfrak{R}[\{\emptyset_{112.011:00}\}] = -_{-2} \cdot (-_{-1} +_{-bool} \parallel_{-1}, skip, -1 \parallel)$
26. $\mathfrak{R}[\{\emptyset_{20.011:00}\}] = -_{-2} \cdot | skip |$
27. $\mathfrak{R}[\{\emptyset_{201.011:00}\}] = -_{-2} \cdot (| skip | \cdot_{-1})$
28. $\mathfrak{R}[\{\emptyset_{211.011:00}\}] = -_{-2} \cdot (-_{-1} \cdot | skip |)$
29. $\mathfrak{R}[\{\emptyset_{202.011:00}\}] = -_{-2} \cdot (| skip | +_{-bool} -)$
30. $\mathfrak{R}[\{\emptyset_{212.011:00}\}] = -_{-2} \cdot (-_{-1} +_{-bool} | skip |)$

31. $\Re[\{\emptyset_{10.002:00}\}] = \parallel_{-1}, skip,_{-1} \parallel +_{-bool} \text{ }_{-2}$
32. $\Re[\{\emptyset_{101.002:00}\}] = (\parallel_{-1}, skip,_{-1} \parallel \cdot_{-1}) +_{-bool} \text{ }_{-2}$
33. $\Re[\{\emptyset_{111.002:00}\}] = (_{-1} \cdot \parallel_{-1}, skip,_{-1} \parallel) +_{-bool} \text{ }_{-2}$
34. $\Re[\{\emptyset_{102.002:00}\}] = (\parallel_{-1}, skip,_{-1} \parallel +_{-bool} \text{ }_{-1}) +_{-bool} \text{ }_{-2}$
35. $\Re[\{\emptyset_{112.002:00}\}] = (_{-1} +_{-bool} \parallel_{-1}, skip,_{-1} \parallel) +_{-bool} \text{ }_{-2}$

36. $\Re[\{\emptyset_{20.002:00}\}] = | skip | +_{-bool} \text{ }_{-2}$
37. $\Re[\{\emptyset_{201.002:00}\}] = (| \text{ }_{-}skip | \cdot_{-1}) +_{-bool} \text{ }_{-2}$
38. $\Re[\{\emptyset_{211.002:00}\}] = (_{-1} \cdot | skip |) +_{-bool} \text{ }_{-2}$
39. $\Re[\{\emptyset_{202.002:00}\}] = (| \text{ }_{-}skip | +_{-bool} \text{ }_{-}) +_{-bool} \text{ }_{-2}$
40. $\Re[\{\emptyset_{212.002:00}\}] = (_{-1} +_{-bool} | skip |) +_{-bool} \text{ }_{-2}$

41. $\Re[\{\emptyset_{10.012:00}\}] = \text{ }_{-2} +_{-bool} , \parallel_{-1}, skip,_{-1} \parallel$
42. $\Re[\{\emptyset_{101.012:00}\}] = \text{ }_{-2} +_{-bool} (\parallel_{-1}, skip,_{-1} \parallel \cdot_{-1})$
43. $\Re[\{\emptyset_{111.012:00}\}] = \text{ }_{-2} +_{-bool} (_{-1} \cdot \parallel_{-1}, skip,_{-1} \parallel)$
44. $\Re[\{\emptyset_{102.012:00}\}] = \text{ }_{-2} +_{-bool} (\parallel_{-1}, skip,_{-1} \parallel +_{-bool} \text{ }_{-1})$
45. $\Re[\{\emptyset_{112.012:00}\}] = \text{ }_{-2} +_{-bool} (_{-1} +_{-bool} \parallel_{-1}, skip,_{-1} \parallel)$

46. $\Re[\{\emptyset_{20.012:00}\}] = \text{ }_{-2} +_{-bool} | skip |$
47. $\Re[\{\emptyset_{201.012:00}\}] = \text{ }_{-2} +_{-bool} (| skip | \cdot_{-1})$
48. $\Re[\{\emptyset_{211.012:00}\}] = \text{ }_{-2} +_{-bool} (_{-1} \cdot | skip |)$
49. $\Re[\{\emptyset_{202.012:00}\}] = \text{ }_{-2} +_{-bool} (| skip | +_{-bool} \text{ }_{-})$
50. $\Re[\{\emptyset_{212.012:00}\}] = \text{ }_{-2} +_{-bool} (_{-1} +_{-bool} | skip |)$

TABELA A.2 - Representações para PS com Processo *skip*

$\star : 001 = 001.001. \dots$ e $001 : 001 = : 001$

1. $\mathfrak{R}[\{\emptyset_{10:001}\}] = \parallel_{-1}, skip, -1 \parallel \cdot \underline{\infty}$
2. $\mathfrak{R}[\{\emptyset_{101:001}\}] = (\parallel_{-1}, skip, -1 \parallel \cdot -1) \cdot \underline{\infty}$
3. $\mathfrak{R}[\{\emptyset_{111:001}\}] = (-_{-1} \cdot \parallel_{-1}, skip, -1 \parallel) \cdot \underline{\infty}$
4. $\mathfrak{R}[\{\emptyset_{102:001}\}] = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1) \cdot \underline{\infty}$
5. $\mathfrak{R}[\{\emptyset_{112:001}\}] = (-1 +_{\perp} \parallel_{-1}, skip, -1 \parallel) \cdot \underline{\infty}$

6. $\mathfrak{R}[\{\emptyset_{20:001}\}] = | skip | \cdot \underline{\infty}$
7. $\mathfrak{R}[\{\emptyset_{201:001}\}] = (| skip | \cdot -1) \cdot \underline{\infty}$
8. $\mathfrak{R}[\{\emptyset_{211:001}\}] = (-1 \cdot | skip |) \cdot \underline{\infty}$
9. $\mathfrak{R}[\{\emptyset_{202:001}, b_{22:001}\}] = (| skip | +_b -) \cdot \underline{\infty}$
10. $\mathfrak{R}[\{\emptyset_{212:001}, b_{22:001}\}] = (-1 +_b | skip |) \cdot \underline{\infty}$

11. $\mathfrak{R}[\{\emptyset_{10.001:001}\}] = \mathfrak{R}[\{\emptyset_{10:001}\}] = \parallel_{-1}, skip, -1 \parallel \cdot \underline{\infty}$
12. $\mathfrak{R}[\{\emptyset_{101.001:001}\}] = \mathfrak{R}[\{\emptyset_{101:001}\}] = ((\parallel_{-1}, skip, -1 \parallel \cdot -1) \cdot \underline{\infty})$
13. $\mathfrak{R}[\{\emptyset_{111.001:001}\}] = \mathfrak{R}[\{\emptyset_{111:001}\}] = (-_{-1} \cdot \parallel_{-1}, skip, -1 \parallel) \cdot \underline{\infty}$
14. $\mathfrak{R}[\{\emptyset_{102.001:001}\}] = \mathfrak{R}[\{\emptyset_{102:001}\}] = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1) \cdot \underline{\infty}$
15. $\mathfrak{R}[\{\emptyset_{112.001:001}\}] = \mathfrak{R}[\{\emptyset_{112:001}\}] = (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel) \cdot \underline{\infty}$

16. $\mathfrak{R}[\{\emptyset_{20.001:001}\}] = \mathfrak{R}[\{\emptyset_{20:001}\}] = | skip | \cdot \underline{\infty}$
17. $\mathfrak{R}[\{\emptyset_{201.001:001}\}] = \mathfrak{R}[\{\emptyset_{201:001}\}] = (| skip | \cdot -1) \cdot \underline{\infty}$
18. $\mathfrak{R}[\{\emptyset_{211.001:001}\}] = \mathfrak{R}[\{\emptyset_{211:001}\}] = (-1 \cdot | skip |) \cdot \underline{\infty}$
19. $\mathfrak{R}[\{\emptyset_{202.001:001}\}] = \mathfrak{R}[\{\emptyset_{202:001}\}] = (| skip | +_{-bool} -) \cdot \underline{\infty}$
20. $\mathfrak{R}[\{\emptyset_{212.001:001}\}] = \mathfrak{R}[\{\emptyset_{212:001}\}] = (-1 +_{-bool} | skip |) \cdot \underline{\infty}$

21. $\mathfrak{R}[\{\emptyset_{10.011:001}\}] = (-_2 \cdot \parallel_{-1}, skip, -1 \parallel) \cdot \underline{\infty}$
22. $\mathfrak{R}[\{\emptyset_{101.011:001}\}] = (-_2 \cdot (\parallel_{-1}, skip, -1 \parallel \cdot -1)) \cdot \underline{\infty}$
23. $\mathfrak{R}[\{\emptyset_{111.011:001}\}] = (-_2 \cdot (-1 \cdot \parallel_{-1}, skip, -1 \parallel)) \cdot \underline{\infty}$
24. $\mathfrak{R}[\{\emptyset_{102.011:001}\}] = (-_2 \cdot (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1)) \cdot \underline{\infty}$
25. $\mathfrak{R}[\{\emptyset_{112.011:001}\}] = (-_2 \cdot (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel)) \cdot \underline{\infty}$

26. $\mathfrak{R}[\{\emptyset_{20.011:001}\}] = (-_2 \cdot | skip |) \cdot \underline{\infty}$
27. $\mathfrak{R}[\{\emptyset_{201.011:001}\}] = (-_2 \cdot (| skip | \cdot -1)) \cdot \underline{\infty}$
28. $\mathfrak{R}[\{\emptyset_{211.011:001}\}] = (-_2 \cdot (-1 \cdot | skip |)) \cdot \underline{\infty}$
29. $\mathfrak{R}[\{\emptyset_{202.011:001}\}] = (-_2 \cdot (| skip | +_{-bool} -)) \cdot \underline{\infty}$
30. $\mathfrak{R}[\{\emptyset_{212.011:001}\}] = (-_2 \cdot (-1 +_{-bool} | skip |)) \cdot \underline{\infty}$

31. $\mathfrak{R}[\{\emptyset_{10.002:001}\}] = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -_2) \cdot \underline{\infty}$
32. $\mathfrak{R}[\{\emptyset_{101.002:001}\}] = ((\parallel_{-1}, skip, -1 \parallel \cdot -1) +_{-bool} -_2) \cdot \underline{\infty}$
33. $\mathfrak{R}[\{\emptyset_{111.002:001}\}] = ((-1 \cdot \parallel_{-1}, skip, -1 \parallel) +_{-bool} -_2) \cdot \underline{\infty}$
34. $\mathfrak{R}[\{\emptyset_{102.002:001}\}] = ((\parallel_{-1}, skip, -1 \parallel +_{-bool} -1) +_{-bool} -_2) \cdot \underline{\infty}$
35. $\mathfrak{R}[\{\emptyset_{112.002:001}\}] = ((-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel) +_{-bool} -_2) \cdot \underline{\infty}$

36. $\mathcal{R}[\{\emptyset_{20.002:001}\}] = (| \textit{skip} | +_{-bool} \textit{-}_2) \cdot \underline{\infty}$
37. $\mathcal{R}[\{\emptyset_{201.002:001}\}] = ((| \textit{-skip} | \cdot \textit{-}_1) +_{-bool} \textit{-}_2) \cdot \underline{\infty}$
38. $\mathcal{R}[\{\emptyset_{211.002:001}\}] = ((\textit{-}_1 \cdot | \textit{skip} |) +_{-bool} \textit{-}_2) \cdot \underline{\infty}$
39. $\mathcal{R}[\{\emptyset_{202.002:001}\}] = ((| \textit{-skip} | +_{-bool} \textit{-}) +_{-bool} \textit{-}_2) \cdot \underline{\infty}$
40. $\mathcal{R}[\{\emptyset_{212.002:001}\}] = ((\textit{-}_1 +_{-bool} | \textit{skip} |) +_{-bool} \textit{-}_2) \cdot \underline{\infty}$
41. $\mathcal{R}[\{\emptyset_{10.012:001}\}] = (\textit{-}_2 +_{-bool} , \| \textit{-}_1, \textit{skip}, \textit{-}_1 \|) \cdot \underline{\infty}$
42. $\mathcal{R}[\{\emptyset_{101.012:001}\}] = (\textit{-}_2 +_{-bool} (\| \textit{-}_1, \textit{skip}, \textit{-}_1 \| \cdot \textit{-}_1)) \cdot \underline{\infty}$
43. $\mathcal{R}[\{\emptyset_{111.012:001}\}] = (\textit{-}_2 +_{-bool} (\textit{-}_1 \cdot \| \textit{-}_1, \textit{skip}, \textit{-}_1 \|)) \cdot \underline{\infty}$
44. $\mathcal{R}[\{\emptyset_{102.012:001}\}] = (\textit{-}_2 +_{-bool} (\| \textit{-}_1, \textit{skip}, \textit{-}_1 \| +_{-bool} \textit{-}_1)) \cdot \underline{\infty}$
45. $\mathcal{R}[\{\emptyset_{112.012:001}\}] = (\textit{-}_2 +_{-bool} (\textit{-}_1 +_{-bool} \| \textit{-}_1, \textit{skip}, \textit{-}_1 \|)) \cdot \underline{\infty}$
46. $\mathcal{R}[\{\emptyset_{20.012:001}\}] = (\textit{-}_2 +_{-bool} | \textit{skip} |) \cdot \underline{\infty}$
47. $\mathcal{R}[\{\emptyset_{201.012:001}\}] = (\textit{-}_2 +_{-bool} (| \textit{skip} | \cdot \textit{-}_1)) \cdot \underline{\infty}$
48. $\mathcal{R}[\{\emptyset_{211.002:001}\}] = (\textit{-}_2 +_{-bool} (\textit{-}_1 \cdot | \textit{skip} |)) \cdot \underline{\infty}$
49. $\mathcal{R}[\{\emptyset_{202.012:001}\}] = (\textit{-}_2 +_{-bool} (| \textit{skip} | +_{-bool} \textit{-})) \cdot \underline{\infty}$
50. $\mathcal{R}[\{\emptyset_{212.012:001}\}] = (\textit{-}_2 +_{-bool} (\textit{-}_1 +_{-bool} | \textit{skip} |)) \cdot \underline{\infty}$

TABELA A.3 - Representações para SD com Processo *skip*

-
- $\star : 0012 = 002.002. \dots$ e $002 : 002 = : 002$
1. $\mathfrak{R}\{\{\emptyset_{10:002}\}\} = \parallel_{-1}, skip, -1 \parallel +_{-bool} \underline{\infty}$
 2. $\mathfrak{R}\{\{\emptyset_{101:002}\}\} = (\parallel_{-1}, skip, -1 \parallel \cdot_{-1}) +_{-bool} \underline{\infty}$
 3. $\mathfrak{R}\{\{\emptyset_{111:002}\}\} = (-_{-1} \cdot \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\infty}$
 4. $\mathfrak{R}\{\{\emptyset_{102:002}\}\} = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1) +_{-bool} \underline{\infty}$
 5. $\mathfrak{R}\{\{\emptyset_{112:002}\}\} = (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\infty}$

 6. $\mathfrak{R}\{\{\emptyset_{20:002}\}\} = | skip | +_{-bool} \underline{\infty}$
 7. $\mathfrak{R}\{\{\emptyset_{201:002}\}\} = (| skip | \cdot_{-1}) +_{-bool} \underline{\infty}$
 8. $\mathfrak{R}\{\{\emptyset_{211:002}\}\} = (-1 \cdot | skip |) +_{-bool} \underline{\infty}$
 9. $\mathfrak{R}\{\{\emptyset_{202:002}, b_{22:002}\}\} = (| skip | +_b -1) +_{-bool} \underline{\infty}$
 10. $\mathfrak{R}\{\{\emptyset_{212:002}, b_{22:002}\}\} = (-1 +_b | skip |) +_{-bool} \underline{\infty}$

 11. $\mathfrak{R}\{\{\emptyset_{10.001:002}\}\} = \parallel_{-1}, skip, -1 \parallel +_{-bool} \underline{\infty}$
 12. $\mathfrak{R}\{\{\emptyset_{101.001:002}\}\} = ((\parallel_{-1}, skip, -1 \parallel \cdot_{-1}) +_{-bool} \underline{\infty})$
 13. $\mathfrak{R}\{\{\emptyset_{111.001:002}\}\} = (-_{-1} \cdot \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\infty}$
 14. $\mathfrak{R}\{\{\emptyset_{102.001:002}\}\} = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1) +_{-bool} \underline{\infty}$
 15. $\mathfrak{R}\{\{\emptyset_{112.001:002}\}\} = (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\infty}$

 16. $\mathfrak{R}\{\{\emptyset_{20.001:002}\}\} = | skip | +_{-bool} \underline{\infty}$
 17. $\mathfrak{R}\{\{\emptyset_{201.001:002}\}\} = (| skip | \cdot_{-1}) +_{-bool} \underline{\infty}$
 18. $\mathfrak{R}\{\{\emptyset_{211.001:002}\}\} = (-1 \cdot | skip |) +_{-bool} \underline{\infty}$
 19. $\mathfrak{R}\{\{\emptyset_{202.001:002}\}\} = (| skip | +_{-bool} -) +_{-bool} \underline{\infty}$
 20. $\mathfrak{R}\{\{\emptyset_{212.001:002}\}\} = (-1 +_{-bool} | skip |) +_{-bool} \underline{\infty}$

 21. $\mathfrak{R}\{\{\emptyset_{10.011:002}\}\} = (-_2 \cdot \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\infty}$
 22. $\mathfrak{R}\{\{\emptyset_{101.011:002}\}\} = (-_2 \cdot (\parallel_{-1}, skip, -1 \parallel \cdot_{-1})) +_{-bool} \underline{\infty}$
 23. $\mathfrak{R}\{\{\emptyset_{111.011:002}\}\} = (-_2 \cdot (-1 \cdot \parallel_{-1}, skip, -1 \parallel)) +_{-bool} \underline{\infty}$
 24. $\mathfrak{R}\{\{\emptyset_{102.011:002}\}\} = (-_2 \cdot (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1)) +_{-bool} \underline{\infty}$
 25. $\mathfrak{R}\{\{\emptyset_{112.011:002}\}\} = (-_2 \cdot (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel)) +_{-bool} \underline{\infty}$

 26. $\mathfrak{R}\{\{\emptyset_{20.011:002}\}\} = (-_2 \cdot | skip |) +_{-bool} \underline{\infty}$
 27. $\mathfrak{R}\{\{\emptyset_{201.011:002}\}\} = (-_2 \cdot (| skip | \cdot_{-1})) +_{-bool} \underline{\infty}$
 28. $\mathfrak{R}\{\{\emptyset_{211.011:002}\}\} = (-_2 \cdot (-1 \cdot | skip |)) +_{-bool} \underline{\infty}$
 29. $\mathfrak{R}\{\{\emptyset_{202.011:002}\}\} = (-_2 \cdot (| skip | +_{-bool} -1)) +_{-bool} \underline{\infty}$
 30. $\mathfrak{R}\{\{\emptyset_{212.012:002}\}\} = (-_2 \cdot (-1 +_{-bool} | skip |)) +_{-bool} \underline{\infty}$

31. $\mathfrak{R}[\{\emptyset_{10.002:002}\}] = \mathfrak{R}[\{\emptyset_{10:002}\}] = \parallel_{-1}, skip, -1 \parallel +_{-bool} \underline{\underline{\infty}}$
32. $\mathfrak{R}[\{\emptyset_{101.002:002}\}] = \mathfrak{R}[\{\emptyset_{101:002}\}] = (\parallel_{-1}, skip, -1 \parallel \cdot_{-1}) +_{-bool} \underline{\underline{\infty}}$
33. $\mathfrak{R}[\{\emptyset_{111.002:002}\}] = \mathfrak{R}[\{\emptyset_{111:002}\}] = (-1 \cdot \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\underline{\infty}}$
34. $\mathfrak{R}[\{\emptyset_{102.002:002}\}] = \mathfrak{R}[\{\emptyset_{102:002}\}] = (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1) +_{-bool} \underline{\underline{\infty}}$
35. $\mathfrak{R}[\{\emptyset_{112.002:002}\}] = \mathfrak{R}[\{\emptyset_{112:002}\}] = (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\underline{\infty}}$

36. $\mathfrak{R}[\{\emptyset_{20.002:002}\}] = \mathfrak{R}[\{\emptyset_{20:002}\}] = | skip | +_{-bool} \underline{\underline{\infty}}$
37. $\mathfrak{R}[\{\emptyset_{201.002:002}\}] = \mathfrak{R}[\{\emptyset_{201:002}\}] = (| skip | \cdot_{-1}) +_{-bool} \underline{\underline{\infty}}$
38. $\mathfrak{R}[\{\emptyset_{211.002:002}\}] = \mathfrak{R}[\{\emptyset_{211:002}\}] = (-1 \cdot | skip |) +_{-bool} \underline{\underline{\infty}}$
39. $\mathfrak{R}[\{\emptyset_{202.002:002}\}] = \mathfrak{R}[\{\emptyset_{202:002}\}] = (| skip | +_{-bool} -) +_{-bool} \underline{\underline{\infty}}$
40. $\mathfrak{R}[\{\emptyset_{212.002:002}\}] = \mathfrak{R}[\{\emptyset_{212:002}\}] = (-1 +_{-bool} | skip |) +_{-bool} \underline{\underline{\infty}}$

41. $\mathfrak{R}[\{\emptyset_{10.012:002}\}] = (-2 +_{-bool} \parallel_{-1}, skip, -1 \parallel) +_{-bool} \underline{\underline{\infty}}$
42. $\mathfrak{R}[\{\emptyset_{101.012:002}\}] = (-2 +_{-bool} (\parallel_{-1}, skip, -1 \parallel \cdot_{-1})) +_{-bool} \underline{\underline{\infty}}$
43. $\mathfrak{R}[\{\emptyset_{111.012:002}\}] = (-2 +_{-bool} (-1 \cdot \parallel_{-1}, skip, -1 \parallel)) +_{-bool} \underline{\underline{\infty}}$
44. $\mathfrak{R}[\{\emptyset_{102.012:002}\}] = (-2 +_{-bool} (\parallel_{-1}, skip, -1 \parallel +_{-bool} -1)) +_{-bool} \underline{\underline{\infty}}$
45. $\mathfrak{R}[\{\emptyset_{112.012:002}\}] = (-2 +_{-bool} (-1 +_{-bool} \parallel_{-1}, skip, -1 \parallel)) +_{-bool} \underline{\underline{\infty}}$

46. $\mathfrak{R}[\{\emptyset_{20.012:002}\}] = (-2 +_{-bool} | skip |) +_{-bool} \underline{\underline{\infty}}$
47. $\mathfrak{R}[\{\emptyset_{201.012:002}\}] = (-2 +_{-bool} (| skip | \cdot_{-1})) +_{-bool} \underline{\underline{\infty}}$
48. $\mathfrak{R}[\{\emptyset_{211.012:002}\}] = (-2 +_{-bool} (-1 \cdot | skip |)) +_{-bool} \underline{\underline{\infty}}$
49. $\mathfrak{R}[\{\emptyset_{202.012:002}\}] = (-2 +_{-bool} (| skip | +_{-bool} -)) +_{-bool} \underline{\underline{\infty}}$
50. $\mathfrak{R}[\{\emptyset_{212.012:002}\}] = (-2 +_{-bool} (-1 +_{-bool} | skip |)) +_{-bool} \underline{\underline{\infty}}$

TABELA A.4 - Representações Finitas de Construções com Processo $\mathbf{d}^{(k)}$

★ Nível $\mathbb{D}_0 - \mathbb{D}_1$
1. $\mathfrak{R}[\{d_{:00}^{(k)}\}] = d^{(k)}$
2. $\mathfrak{R}[\{d_{001:00}^{(k)}\}] = d^{(k)} \cdot {}_{-1}$
3. $\mathfrak{R}[\{d_{011:00}^{(k)}\}] = {}_{-1} \cdot d^{(k)}$
4. $\mathfrak{R}[\{d_{002:00}^{(k)}\}] = d^{(k)} + {}_{-bool} {}_{-1}$
5. $\mathfrak{R}[\{d_{012:00}^{(k)}\}] = {}_{-1} + {}_{-bool} d^{(k)}$
6. $\mathfrak{R}[\{\overline{d^{(k)}}_{10:00}\}] = \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel$
7. $\mathfrak{R}[\{\overline{d^{(k)}}_{101:00}\}] = \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel \cdot -$
8. $\mathfrak{R}[\{\overline{d^{(k)}}_{111:00}\}] = {}_{-1} \cdot \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel$
9. $\mathfrak{R}[\{\overline{d^{(k)}}_{102:00}\}] = \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel + {}_{-bool} {}_{-1}$
10. $\mathfrak{R}[\{\overline{d^{(k)}}_{112:00}\}] = {}_{-1} + {}_{-bool} \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel$
11. $\mathfrak{R}[\{\overline{d^{(k)}}_{20:00}\}] = {}_{-1} d^{(k)} {}_{-1} $
12. $\mathfrak{R}[\{\overline{d^{(k)}}_{201:00}\}] = {}_{-1} d^{(k)} {}_{-1} \cdot {}_{-1}$
13. $\mathfrak{R}[\{\overline{d^{(k)}}_{211:00}\}] = {}_{-1} \cdot {}_{-1} d^{(k)} {}_{-1} $
14. $\mathfrak{R}[\{\overline{d^{(k)}}_{202:00}, b_{22:00}\}] = {}_{-1} d^{(k)} {}_{-1} + b {}_{-1}$
15. $\mathfrak{R}[\{\overline{d^{(k)}}_{212:00}, b_{22:00}\}] = {}_{-1} + b {}_{-1} d^{(k)} {}_{-1} $
★ Nível $\mathbb{D}_1 - \mathbb{D}_2$
16. $\mathfrak{R}[\{d_{00.001:00}^{(k)}\}] = d^{(k)} \cdot {}_{-2}$
17. $\mathfrak{R}[\{d_{001.001:00}^{(k)}\}] = (d^{(k)} \cdot {}_{-1}) \cdot {}_{-2}$
18. $\mathfrak{R}[\{d_{011.001:00}^{(k)}\}] = ({}_{-1} \cdot d^{(k)}) \cdot {}_{-2}$
19. $\mathfrak{R}[\{d_{002.001:00}^{(k)}\}] = (d^{(k)} + {}_{-bool} {}_{-1}) \cdot {}_{-2}$
20. $\mathfrak{R}[\{d_{012.001:00}^{(k)}\}] = ({}_{-1} + {}_{-bool} d^{(k)}) \cdot {}_{-2}$
21. $\mathfrak{R}[\{\overline{d^{(k)}}_{10.001:00}\}] = (\parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel) \cdot {}_{-2}$
22. $\mathfrak{R}[\{\overline{d^{(k)}}_{101.001:00}\}] = (\parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel \cdot -) \cdot {}_{-2}$
23. $\mathfrak{R}[\{\overline{d^{(k)}}_{111.001:00}\}] = ({}_{-1} \cdot \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel) \cdot {}_{-2}$
24. $\mathfrak{R}[\{\overline{d^{(k)}}_{102.001:00}\}] = (\parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel + {}_{-bool} {}_{-1}) \cdot {}_{-2}$
25. $\mathfrak{R}[\{\overline{d^{(k)}}_{112.001:00}\}] = ({}_{-1} + {}_{-bool} \parallel {}_{-1}, d^{(k)}, {}_{-1} \parallel) \cdot {}_{-2}$
26. $\mathfrak{R}[\{\overline{d^{(k)}}_{20.001:00}\}] = ({}_{-1} d^{(k)} {}_{-1}) \cdot {}_{-2}$
27. $\mathfrak{R}[\{\overline{d^{(k)}}_{201.001:00}\}] = ({}_{-1} d^{(k)} {}_{-1} \cdot {}_{-1}) \cdot {}_{-2}$
28. $\mathfrak{R}[\{\overline{d^{(k)}}_{211.001:00}\}] = ({}_{-1} \cdot {}_{-1} d^{(k)} {}_{-1}) \cdot {}_{-2}$
29. $\mathfrak{R}[\{\overline{d^{(k)}}_{202.001:00}, b_{22:00}\}] = ({}_{-1} d^{(k)} {}_{-1} + b {}_{-1}) \cdot {}_{-2}$
30. $\mathfrak{R}[\{\overline{d^{(k)}}_{212.001:00}, b_{22:00}\}] = ({}_{-1} + b {}_{-1} d^{(k)} {}_{-1}) \cdot {}_{-2}$

31. $\mathfrak{R}\{d_{00.011:00}^{(k)}\} = \neg_2 \cdot d^{(k)}$
32. $\mathfrak{R}\{d_{001.011:00}^{(k)}\} = \neg_2 \cdot (d^{(k)} \cdot \neg_1)$
33. $\mathfrak{R}\{d_{011.011:00}^{(k)}\} = \neg_2 \cdot (\neg_1 \cdot d^{(k)})$
34. $\mathfrak{R}\{d_{002.011:00}^{(k)}\} = \neg_2 \cdot (d^{(k)} +_{\neg_{bool}} \neg_1)$
35. $\mathfrak{R}\{d_{012.011:00}^{(k)}\} = \neg_2 \cdot (\neg_1 +_{\neg_{bool}} d^{(k)})$

36. $\mathfrak{R}\{\overline{d^{(k)}}_{10.011:00}\} = \neg_2 \cdot (\| \neg_1, d^{(k)}, \neg_1 \|)$
37. $\mathfrak{R}\{\overline{d^{(k)}}_{101.011:00}\} = \neg_2 \cdot (\| \neg_1, d^{(k)}, \neg_1 \| \cdot \neg_1)$
38. $\mathfrak{R}\{\overline{d^{(k)}}_{111.011:00}\} = \neg_2 \cdot (\neg_1 \cdot \| \neg_1, d^{(k)}, \neg_1 \|)$
39. $\mathfrak{R}\{\overline{d^{(k)}}_{102.001:00}\} = \neg_2 \cdot (\| \neg_1, d^{(k)}, \neg_1 \| +_{\neg_{bool}} \neg_1)$
40. $\mathfrak{R}\{\overline{d^{(k)}}_{112.011:00}\} = \neg_2 \cdot (\neg_1 +_{\neg_{bool}} \| \neg_1, d^{(k)}, \neg_1 \|)$

41. $\mathfrak{R}\{\overline{d^{(k)}}_{20.011:00}\} = \neg_2 \cdot (| \neg_1 | d^{(k)} | \neg_1 |)$
42. $\mathfrak{R}\{\overline{d^{(k)}}_{201.011:00}\} = \neg_2 \cdot (| \neg_1 | d^{(k)} | \neg_1 | \cdot \neg_1)$
43. $\mathfrak{R}\{\overline{d^{(k)}}_{211.011:00}\} = \neg_2 \cdot (\neg_1 \cdot | \neg_1 | d^{(k)} | \neg_1 |)$
44. $\mathfrak{R}\{\overline{d^{(k)}}_{202.011:00}, b_{22.011:00}\} = \neg_2 \cdot (| \neg_1 | d^{(k)} | \neg_1 | +_b \neg_1)$
45. $\mathfrak{R}\{\overline{d^{(k)}}_{212.011:00}, b_{22.011:00}\} = \neg_2 \cdot (\neg_1 +_b | \neg_1 | d^{(k)} | \neg_1 |)$

46. $\mathfrak{R}\{d_{00.002:00}^{(k)}\} = d^{(k)} +_{\neg_{bool}} \neg_2$
47. $\mathfrak{R}\{d_{001.002:00}^{(k)}\} = (d^{(k)} \cdot \neg_1) +_{\neg_{bool}} \neg_2$
48. $\mathfrak{R}\{d_{011.002:00}^{(k)}\} = (\neg_1 \cdot d^{(k)}) +_{\neg_{bool}} \neg_2$
49. $\mathfrak{R}\{d_{002.002:00}^{(k)}\} = (d^{(k)} +_{\neg_{bool}} \neg_1) +_{\neg_{bool}} \neg_2$
50. $\mathfrak{R}\{d_{012.002:00}^{(k)}\} = (\neg_1 +_{\neg_{bool}} d^{(k)}) +_{\neg_{bool}} \neg_2$

51. $\mathfrak{R}\{\overline{d^{(k)}}_{10.002:00}\} = (\| \neg_1, d^{(k)}, \neg_1 \|) +_{\neg_{bool}} \neg_2$
52. $\mathfrak{R}\{\overline{d^{(k)}}_{101.002:00}\} = (\| \neg_1, d^{(k)}, \neg_1 \| \cdot \neg_1) +_{\neg_{bool}} \neg_2$
53. $\mathfrak{R}\{\overline{d^{(k)}}_{111.002:00}\} = (\neg_1 \cdot \| \neg_1, d^{(k)}, \neg_1 \|) +_{\neg_{bool}} \neg_2$
54. $\mathfrak{R}\{\overline{d^{(k)}}_{102.002:00}\} = (\| \neg_1, d^{(k)}, \neg_1 \| +_{\neg_{bool}} \neg_1) +_{\neg_{bool}} \neg_2$
55. $\mathfrak{R}\{\overline{d^{(k)}}_{112.002:00}\} = (\neg_1 +_{\neg_{bool}} \| \neg_1, d^{(k)}, \neg_1 \|) +_{\neg_{bool}} \neg_2$

56. $\mathfrak{R}\{\overline{d^{(k)}}_{20.002:00}\} = (| \neg_1 | d^{(k)} | \neg_1 |) +_{\neg_{bool}} \neg_2$
57. $\mathfrak{R}\{\overline{d^{(k)}}_{201.002:00}\} = (| \neg_1 | d^{(k)} | \neg_1 | \cdot \neg_1) +_{\neg_{bool}} \neg_2$
58. $\mathfrak{R}\{\overline{d^{(k)}}_{211.002:00}\} = (\neg_1 \cdot | \neg_1 | d^{(k)} | \neg_1 |) +_{\neg_{bool}} \neg_2$
59. $\mathfrak{R}\{\overline{d^{(k)}}_{202.002:00}, b_{22.002:00}\} = (| \neg_1 | d^{(k)} | \neg_1 | +_b \neg_1) +_{\neg_{bool}} \neg_2$
60. $\mathfrak{R}\{\overline{d^{(k)}}_{212.002:00}, b_{22.002:00}\} = (\neg_1 +_b | \neg_1 | d^{(k)} | \neg_1 |) +_{\neg_{bool}} \neg_2$

61. $\mathfrak{R}\{d_{00.012:00}^{(k)}\} = \neg_2 +_{\neg_{bool}} d^{(k)}$
62. $\mathfrak{R}\{d_{001.012:00}^{(k)}\} = \neg_2 +_{\neg_{bool}} (d^{(k)} \cdot \neg_1)$
63. $\mathfrak{R}\{d_{011.012:00}^{(k)}\} = \neg_2 +_{\neg_{bool}} (\neg_1 \cdot d^{(k)})$
64. $\mathfrak{R}\{d_{002.012:00}^{(k)}\} = \neg_2 +_{\neg_{bool}} (d^{(k)} +_{\neg_{bool}} \neg_1)$
65. $\mathfrak{R}\{d_{012.012:00}^{(k)}\} = \neg_2 +_{\neg_{bool}} (\neg_1 +_{\neg_{bool}} d^{(k)})$

66. $\Re[\{\overline{d^{(k)}}_{10.012:00}\}] = {}_{-2} +_{-bool} (\| {}_{-1}, d^{(k)}, {}_{-1} \|)$
67. $\Re[\{\overline{d^{(k)}}_{101.012:00}\}] = {}_{-2} +_{-bool} (\| {}_{-1}, d^{(k)}, {}_{-1} \| \cdot {}_{-1})$
68. $\Re[\{\overline{d^{(k)}}_{111.012:00}\}] = {}_{-2} +_{-bool} ({}_{-1} \cdot \| {}_{-1}, d^{(k)}, {}_{-1} \|)$
69. $\Re[\{\overline{d^{(k)}}_{102.002:00}\}] = {}_{-2} +_{-bool} (\| {}_{-1}, d^{(k)}, {}_{-1} \| +_{-bool} {}_{-1})$
70. $\Re[\{\overline{d^{(k)}}_{112.012:00}\}] = {}_{-2} +_{-bool} ({}_{-1} +_{-bool} \| {}_{-1}, d^{(k)}, {}_{-1} \|)$
71. $\Re[\{\overline{d^{(k)}}_{20.012:00}\}] = {}_{-2} +_{-bool} (| {}_{-1} | d^{(k)} | {}_{-1} |)$
72. $\Re[\{\overline{d^{(k)}}_{201.012:00}\}] = {}_{-2} +_{-bool} (| {}_{-1} | d^{(k)} | {}_{-1} | \cdot {}_{-1})$
73. $\Re[\{\overline{d^{(k)}}_{211.012:00}\}] = {}_{-2} +_{-bool} ({}_{-1} \cdot | {}_{-1} | d^{(k)} | {}_{-1} |)$
74. $\Re[\{\overline{d^{(k)}}_{202.012:00}, b_{22.011:00}\}] = {}_{-2} +_{-bool} (| {}_{-1} | d^{(k)} | {}_{-1} | + b {}_{-1})$
75. $\Re[\{\overline{d^{(k)}}_{212.012:00}, b_{22.011:00}\}] = {}_{-2} +_{-bool} ({}_{-1} + b | {}_{-1} | d^{(k)} | {}_{-1} |)$
76. $\Re[\{\overline{d^{(k)}}_{00.10:00}\}] = \| {}_{-2}, d^{(k)}, {}_{-2} \|$
77. $\Re[\{\overline{d^{(k)}}_{001.10:00}\}] = \| {}_{-2}, d^{(k)} \cdot {}_{-1}, {}_{-2} \|$
78. $\Re[\{\overline{d^{(k)}}_{011.10:00}\}] = \| {}_{-2}, {}_{-1} \cdot d^{(k)}, {}_{-2} \|$
79. $\Re[\{\overline{d^{(k)}}_{002.10:00}\}] = \| {}_{-2}, d^{(k)} +_{-bool} {}_{-1}, {}_{-2} \|$
80. $\Re[\{\overline{d^{(k)}}_{012.10:00}\}] = \| {}_{-2}, {}_{-1} +_{-bool} d^{(k)}, {}_{-2} \|$
81. $\Re[\{\overline{\overline{d^{(k)}}}_{10.10:00}\}] = \| {}_{-2}, \| {}_{-1}, d^{(k)}, {}_{-1} \|, {}_{-2} \|$
82. $\Re[\{\overline{\overline{d^{(k)}}}_{101.10:00}\}] = \| {}_{-2}, \| {}_{-1}, d^{(k)}, {}_{-1} \| \cdot {}_{-1}, {}_{-2} \|$
83. $\Re[\{\overline{\overline{d^{(k)}}}_{111.10:00}\}] = \| {}_{-2}, {}_{-1} \cdot \| {}_{-1}, d^{(k)}, {}_{-1} \|, {}_{-2} \|$
84. $\Re[\{\overline{\overline{d^{(k)}}}_{102.10:00}\}] = \| {}_{-2}, \| {}_{-1}, d^{(k)}, {}_{-1} \| +_{-bool} {}_{-1}, {}_{-2} \|$
85. $\Re[\{\overline{\overline{d^{(k)}}}_{112.10:00}\}] = \| {}_{-2}, {}_{-1} +_{-bool} \| {}_{-1}, d^{(k)}, {}_{-1} \|, {}_{-2} \|$
86. $\Re[\{\overline{\overline{\overline{d^{(k)}}}_{20.10:00}\}] = \| {}_{-2}, | {}_{-1} | d^{(k)} | {}_{-1} |, {}_{-2} \|$
87. $\Re[\{\overline{\overline{\overline{d^{(k)}}}_{201.10:00}\}] = \| {}_{-2}, | {}_{-1} | d^{(k)} | {}_{-1} | \cdot {}_{-1}, {}_{-2} \|$
88. $\Re[\{\overline{\overline{\overline{d^{(k)}}}_{211.10:00}\}] = \| {}_{-2}, {}_{-1} \cdot | {}_{-1} | d^{(k)} | {}_{-1} |, {}_{-2} \|$
89. $\Re[\{\overline{\overline{\overline{d^{(k)}}}_{202.10:00}\}] = \| {}_{-2}, | {}_{-1} | d^{(k)} | {}_{-1} | + b {}_{-1}, {}_{-2} \|$
90. $\Re[\{\overline{\overline{\overline{d^{(k)}}}_{212.10:00}\}] = \| {}_{-2}, {}_{-1} + b | {}_{-1} | d^{(k)} | {}_{-1} |, {}_{-2} \|$

TABELA A.5 - Representações para PS com Processo $\mathbf{d}^{(k)}$

★ Nível $\mathbb{D}_0 - \mathbb{D}_1$

1. $\Re[\{d_{:001}^{(k)}\}] = d^{(k)} \cdot \underline{\infty}$
2. $\Re[\{d_{001:001}^{(k)}\}] = \Re[\{d_{:001}^{(k)}\}] = d^{(k)} \cdot \underline{\infty}$
3. $\Re[\{d_{011:001}^{(k)}\}] = ({}_{-1} \cdot d^{(k)}) \cdot \underline{\infty}$
4. $\Re[\{d_{002:001}^{(k)}\}] = (d^{(k)} + {}_{-bool} {}_{-1}) \cdot \underline{\infty}$
5. $\Re[\{d_{012:001}^{(k)}\}] = ({}_{-1} + {}_{-bool} d^{(k)}) \cdot \underline{\infty}$

6. $\Re[\{\overline{d^{(k)}}_{10:001}\}] = (\| {}_{-1}, d^{(k)}, {}_{-1} \|) \cdot \underline{\infty}$
7. $\Re[\{\overline{d^{(k)}}_{101:001}\}] = (\| {}_{-1}, d^{(k)}, {}_{-1} \| \cdot -) \cdot \underline{\infty}$
8. $\Re[\{\overline{d^{(k)}}_{111:001}\}] = ({}_{-1} \cdot \| {}_{-1}, d^{(k)}, {}_{-1} \|) \cdot \underline{\infty}$
9. $\Re[\{\overline{d^{(k)}}_{102:00}\}] = (\| {}_{-1}, d^{(k)}, {}_{-1} \| + {}_{-bool} {}_{-1}) \cdot \underline{\infty}$
10. $\Re[\{\overline{d^{(k)}}_{112:001}\}] = ({}_{-1} + {}_{-bool} \| {}_{-1}, d^{(k)}, {}_{-1} \|) \cdot \underline{\infty}$

11. $\Re[\{\overline{d^{(k)}}_{20:001}\}] = (| {}_{-1} | d^{(k)} | {}_{-1} |) \cdot \underline{\infty}$
12. $\Re[\{\overline{d^{(k)}}_{201:001}\}] = (| {}_{-1} | d^{(k)} | {}_{-1} | \cdot -) \cdot \underline{\infty}$
13. $\Re[\{\overline{d^{(k)}}_{211:001}\}] = ({}_{-1} \cdot | {}_{-1} | d^{(k)} | {}_{-1} |) \cdot \underline{\infty}$
14. $\Re[\{\overline{d^{(k)}}_{202:00}, b_{22:00}\}] = (| {}_{-1} | d^{(k)} | {}_{-1} | + b {}_{-1}) \cdot \underline{\infty}$
15. $\Re[\{\overline{d^{(k)}}_{212:00}, b_{22:00}\}] = ({}_{-1} + b | {}_{-1} | d^{(k)} | {}_{-1} |) \cdot \underline{\infty}$

★ Nível $\mathbb{D}_1 - \mathbb{D}_2$

 16. $\Re[\{d_{00.001:001}^{(k)}\}] = \Re[\{d_{00:001}^{(k)}\}]$
 17. $\Re[\{d_{001.001:001}^{(k)}\}] = \Re[\{d_{:001}^{(k)}\}]$
 18. $\Re[\{d_{011.001:001}^{(k)}\}] = \Re[\{d_{011:001}^{(k)}\}]$
 19. $\Re[\{d_{002.001:001}^{(k)}\}] = \Re[\{d_{002:001}^{(k)}\}]$
 20. $\Re[\{d_{012.001:001}^{(k)}\}] = \Re[\{d_{012:001}^{(k)}\}]$

 21. $\Re[\{\overline{d^{(k)}}_{10.001:001}\}] = \Re[\{\overline{d^{(k)}}_{10:001}\}]$
 22. $\Re[\{\overline{d^{(k)}}_{101.001:001}\}] = \Re[\{\overline{d^{(k)}}_{101:001}\}]$
 23. $\Re[\{\overline{d^{(k)}}_{111.001:001}\}] = \Re[\{\overline{d^{(k)}}_{111:001}\}]$
 24. $\Re[\{\overline{d^{(k)}}_{102.001:001}\}] = Re[\{\overline{d^{(k)}}_{102:001}\}]$
 25. $\Re[\{\overline{d^{(k)}}_{112.001:001}\}] = \Re[\{\overline{d^{(k)}}_{112:001}\}]$

 26. $\Re[\{\overline{d^{(k)}}_{20.001:001}\}] = \Re[\{\overline{d^{(k)}}_{20:001}\}]$
 27. $\Re[\{\overline{d^{(k)}}_{201.001:001}\}] = \Re[\{\overline{d^{(k)}}_{201:001}\}]$
 28. $\Re[\{\overline{d^{(k)}}_{211.001:001}\}] = \Re[\{\overline{d^{(k)}}_{211:001}\}]$
 29. $\Re[\{\overline{d^{(k)}}_{202.001:001}, b_{22:001}\}] = \Re[\{\overline{d^{(k)}}_{202:001}, b_{22:001}\}]$
 30. $\Re[\{\overline{d^{(k)}}_{212.001:001}, b_{22:001}\}] = \Re[\{\overline{d^{(k)}}_{212:001}, b_{22:001}\}]$

31. $\mathfrak{R}[\{d_{00.011:001}^{(k)}\}] = (-_2 \cdot d^{(k)}) \cdot \underline{\infty}$
32. $\mathfrak{R}[\{d_{001.011:001}^{(k)}\}] = (-_2 \cdot (d^{(k)} \cdot -_1)) \cdot \underline{\infty}$
33. $\mathfrak{R}[\{d_{011.011:001}^{(k)}\}] = (-_2 \cdot (-_1 \cdot d^{(k)})) \cdot \underline{\infty}$
34. $\mathfrak{R}[\{d_{002.011:001}^{(k)}\}] = (-_2 \cdot (d^{(k)} +_{-bool} -_1)) \cdot \underline{\infty}$
35. $\mathfrak{R}[\{d_{012.011:001}^{(k)}\}] = (-_2 \cdot (-_1 +_{-bool} d^{(k)})) \cdot \underline{\infty}$

36. $\mathfrak{R}[\{\overline{d^{(k)}}_{10.011:001}\}] = (-_2 \cdot (\| -_1, d^{(k)}, -_1 \|)) \cdot \underline{\infty}$
37. $\mathfrak{R}[\{\overline{d^{(k)}}_{101.011:001}\}] = (-_2 \cdot (\| -_1, d^{(k)}, -_1 \| \cdot -_1)) \cdot \underline{\infty}$
38. $\mathfrak{R}[\{\overline{d^{(k)}}_{111.011:001}\}] = (-_2 \cdot (-_1 \cdot \| -_1, d^{(k)}, -_1 \|)) \cdot \underline{\infty}$
39. $\mathfrak{R}[\{\overline{d^{(k)}}_{102.001:001}\}] = (-_2 \cdot (\| -_1, d^{(k)}, -_1 \| +_{-bool} -_1)) \cdot \underline{\infty}$
40. $\mathfrak{R}[\{\overline{d^{(k)}}_{112.011:001}\}] = (-_2 \cdot (-_1 +_{-bool} \| -_1, d^{(k)}, -_1 \|)) \cdot \underline{\infty}$

41. $\mathfrak{R}[\{\overline{d^{(k)}}_{20.011:001}\}] = (-_2 \cdot (| -_1 | d^{(k)} | -_1 |)) \cdot \underline{\infty}$
42. $\mathfrak{R}[\{\overline{d^{(k)}}_{201.011:001}\}] = (-_2 \cdot (| -_1 | d^{(k)} | -_1 | \cdot -_1)) \cdot \underline{\infty}$
43. $\mathfrak{R}[\{\overline{d^{(k)}}_{211.011:001}\}] = (-_2 \cdot (-_1 \cdot | -_1 | d^{(k)} | -_1 |)) \cdot \underline{\infty}$
44. $\mathfrak{R}[\{\overline{d^{(k)}}_{202.011:001}, b_{22.011:001}\}] = (-_2 \cdot (| -_1 | d^{(k)} | -_1 | +_b -_1)) \cdot \underline{\infty}$
45. $\mathfrak{R}[\{\overline{d^{(k)}}_{212.011:001}, b_{22.011:001}\}] = (-_2 \cdot (-_1 +_b | -_1 | d^{(k)} | -_1 |)) \cdot \underline{\infty}$

46. $\mathfrak{R}[\{d_{00.002:001}^{(k)}\}] = (d^{(k)} +_{-bool} -_2) \cdot \underline{\infty}$
47. $\mathfrak{R}[\{d_{001.002:001}^{(k)}\}] = ((d^{(k)} \cdot -_1) +_{-bool} -_2) \cdot \underline{\infty}$
48. $\mathfrak{R}[\{d_{011.002:001}^{(k)}\}] = ((-_1 \cdot d^{(k)}) +_{-bool} -_2) \cdot \underline{\infty}$
49. $\mathfrak{R}[\{d_{002.002:001}^{(k)}\}] = ((d^{(k)} +_{-bool} -_1) +_{-bool} -_2) \cdot \underline{\infty}$
50. $\mathfrak{R}[\{d_{012.002:001}^{(k)}\}] = ((-_1 +_{-bool} d^{(k)}) +_{-bool} -_2) \cdot \underline{\infty}$

51. $\mathfrak{R}[\{\overline{d^{(k)}}_{10.002:001}\}] = ((\| -_1, d^{(k)}, -_1 \|) +_{-bool} -_2) \cdot \underline{\infty}$
52. $\mathfrak{R}[\{\overline{d^{(k)}}_{101.002:001}\}] = ((\| -_1, d^{(k)}, -_1 \| \cdot -_1) +_{-bool} -_2) \cdot \underline{\infty}$
53. $\mathfrak{R}[\{\overline{d^{(k)}}_{111.002:001}\}] = ((-_1 \cdot \| -_1, d^{(k)}, -_1 \|) +_{-bool} -_2) \cdot \underline{\infty}$
54. $\mathfrak{R}[\{\overline{d^{(k)}}_{102.002:001}\}] = ((\| -_1, d^{(k)}, -_1 \| +_{-bool} -_1) +_{-bool} -_2) \cdot \underline{\infty}$
55. $\mathfrak{R}[\{\overline{d^{(k)}}_{112.002:001}\}] = ((-_1 +_{-bool} \| -_1, d^{(k)}, -_1 \|) +_{-bool} -_2) \cdot \underline{\infty}$

56. $\mathfrak{R}[\{\overline{d^{(k)}}_{20.002:001}\}] = ((| -_1 | d^{(k)} | -_1 |) +_{-bool} -_2) \cdot \underline{\infty}$
57. $\mathfrak{R}[\{\overline{d^{(k)}}_{201.002:001}\}] = ((| -_1 | d^{(k)} | -_1 | \cdot -_1) +_{-bool} -_2) \cdot \underline{\infty}$
58. $\mathfrak{R}[\{\overline{d^{(k)}}_{211.002:001}\}] = ((-_1 \cdot | -_1 | d^{(k)} | -_1 |) +_{-bool} -_2) \cdot \underline{\infty}$
59. $\mathfrak{R}[\{\overline{d^{(k)}}_{202.002:001}, b_{22.002:001}\}] = ((| -_1 | d^{(k)} | -_1 | +_b -_1) +_{-bool} -_2) \cdot \underline{\infty}$
60. $\mathfrak{R}[\{\overline{d^{(k)}}_{212.002:001}, b_{22.002:001}\}] = ((-_1 +_b | -_1 | d^{(k)} | -_1 |) +_{-bool} -_2) \cdot \underline{\infty}$

61. $\mathfrak{R}[\{d_{00.012:001}^{(k)}\}] = (-_2 +_{-bool} d^{(k)}) \cdot \underline{\infty}$
62. $\mathfrak{R}[\{d_{001.012:001}^{(k)}\}] = (-_2 +_{-bool} (d^{(k)} \cdot -_1)) \cdot \underline{\infty}$
63. $\mathfrak{R}[\{d_{011.012:001}^{(k)}\}] = (-_2 +_{-bool} (-_1 \cdot d^{(k)})) \cdot \underline{\infty}$
64. $\mathfrak{R}[\{d_{002.012:001}^{(k)}\}] = (-_2 +_{-bool} (d^{(k)} +_{-bool} -_1)) \cdot \underline{\infty}$
65. $\mathfrak{R}[\{d_{012.012:001}^{(k)}\}] = (-_2 +_{-bool} (-_1 +_{-bool} d^{(k)})) \cdot \underline{\infty}$

66. $\mathfrak{R}[\{\overline{d^{(k)}}_{10.012:001}\}] = (\neg_2 +_{-bool} (\| \neg_1, d^{(k)}, \neg_1 \|)) \cdot \underline{\infty}$
67. $\mathfrak{R}[\{\overline{d^{(k)}}_{101.012:001}\}] = (\neg_2 +_{-bool} (\| \neg_1, d^{(k)}, \neg_1 \| \cdot \neg_1)) \cdot \underline{\infty}$
68. $\mathfrak{R}[\{\overline{d^{(k)}}_{111.012:00}\}] = (\neg_2 +_{-bool} (\neg_1 \cdot \| \neg_1, d^{(k)}, \neg_1 \|)) \cdot \underline{\infty}$
69. $\mathfrak{R}[\{\overline{d^{(k)}}_{102.002:001}\}] = (\neg_2 +_{-bool} (\| \neg_1, d^{(k)}, \neg_1 \| +_{-bool} \neg_1)) \cdot \underline{\infty}$
70. $\mathfrak{R}[\{\overline{d^{(k)}}_{112.012:001}\}] = (\neg_2 +_{-bool} (\neg_1 +_{-bool} \| \neg_1, d^{(k)}, \neg_1 \|)) \cdot \underline{\infty}$
71. $\mathfrak{R}[\{\overline{d^{(k)}}_{20.012:001}\}] = (\neg_2 +_{-bool} (| \neg_1 | d^{(k)} | \neg_1 |)) \cdot \underline{\infty}$
72. $\mathfrak{R}[\{\overline{d^{(k)}}_{201.012:001}\}] = (\neg_2 +_{-bool} (| \neg_1 | d^{(k)} | \neg_1 | \cdot \neg_1)) \cdot \underline{\infty}$
73. $\mathfrak{R}[\{\overline{d^{(k)}}_{211.012:001}\}] = (\neg_2 +_{-bool} (\neg_1 \cdot | \neg_1 | d^{(k)} | \neg_1 |)) \cdot \underline{\infty}$
74. $\mathfrak{R}[\{\overline{d^{(k)}}_{202.012:001}, b_{22.011:001}\}] = (\neg_2 +_{-bool} (| \neg_1 | d^{(k)} | \neg_1 | +_b \neg_1)) \cdot \underline{\infty}$
75. $\mathfrak{R}[\{\overline{d^{(k)}}_{212.012:001}, b_{22.011:001}\}] = (\neg_2 +_{-bool} (\neg_1 +_b | \neg_1 | d^{(k)} | \neg_1 |)) \cdot \underline{\infty}$
76. $\mathfrak{R}[\{\overline{d^{(k)}}_{00}{}_{10:001}\}] = (\| \neg_2, d^{(k)}, \neg_2 \|) \cdot \underline{\infty}$
77. $\mathfrak{R}[\{\overline{d^{(k)}}_{001}{}_{10:001}\}] = (\| \neg_2, d^{(k)} \cdot \neg_1, \neg_2 \|) \cdot \underline{\infty}$
78. $\mathfrak{R}[\{\overline{d^{(k)}}_{011}{}_{10:001}\}] = (\| \neg_2, \neg_1 \cdot d^{(k)}, \neg_2 \|) \cdot \underline{\infty}$
79. $\mathfrak{R}[\{\overline{d^{(k)}}_{002}{}_{10:00}\}] = (\| \neg_2, d^{(k)} +_{-bool} \neg_1, \neg_2 \|) \cdot \underline{\infty}$
80. $\mathfrak{R}[\{\overline{d^{(k)}}_{012}{}_{10:00}\}] = \| \neg_2, \neg_1 +_{-bool} d^{(k)}, \neg_2 \|$
81. $\mathfrak{R}[\{\overline{d^{(k)}}_{10}{}_{10:001}\}] = (\| \neg_2, \| \neg_1, d^{(k)}, \neg_1 \|, \neg_2 \|) \cdot \underline{\infty}$
82. $\mathfrak{R}[\{\overline{d^{(k)}}_{101}{}_{10:001}\}] = (\| \neg_2, \| \neg_1, d^{(k)}, \neg_1 \| \cdot \neg_1, \neg_2 \|) \cdot \underline{\infty}$
83. $\mathfrak{R}[\{\overline{d^{(k)}}_{111}{}_{10:001}\}] = (\| \neg_2, \neg_1 \cdot \| \neg_1, d^{(k)}, \neg_1 \|, \neg_2 \|) \cdot \underline{\infty}$
84. $\mathfrak{R}[\{\overline{d^{(k)}}_{102}{}_{10:00}\}] = (\| \neg_2, \| \neg_1, d^{(k)}, \neg_1 \| +_{-bool} \neg_1, \neg_2 \|) \cdot \underline{\infty}$
85. $\mathfrak{R}[\{\overline{d^{(k)}}_{112}{}_{10:00}\}] = (\| \neg_2, \neg_1 +_{-bool} \| \neg_1, d^{(k)}, \neg_1 \|, \neg_2 \|) \cdot \underline{\infty}$
86. $\mathfrak{R}[\{\overline{d^{(k)}}_{20}{}_{10:00}\}] = (\| \neg_2, | \neg_1 | d^{(k)} | \neg_1 |, \neg_2 \|) \cdot \underline{\infty}$
87. $\mathfrak{R}[\{\overline{d^{(k)}}_{201}{}_{10:00}\}] = (\| \neg_2, | \neg_1 | d^{(k)} | \neg_1 | \cdot \neg_1, \neg_2 \|) \cdot \underline{\infty}$
88. $\mathfrak{R}[\{\overline{d^{(k)}}_{211}{}_{10:00}\}] = (\| \neg_2, \neg_1 \cdot | \neg_1 | d^{(k)} | \neg_1 |, \neg_2 \|) \cdot \underline{\infty}$
89. $\mathfrak{R}[\{\overline{d^{(k)}}_{202}{}_{10:00}, \}] = (\| \neg_2, | \neg_1 | d^{(k)} | \neg_1 | +_b \neg_1, \neg_2 \|) \cdot \underline{\infty}$
90. $\mathfrak{R}[\{\overline{d^{(k)}}_{212}{}_{10:00}\}] = (\| \neg_2, \neg_1 +_b | \neg_1 | d^{(k)} | \neg_1 |, \neg_2 \|) \cdot \underline{\infty}$

Bibliografia

- [ABR 87] ABRAMSKY, S. **Domain Theory in Logical Form**. London: Department of Computer Science/Imperial College, 1987.
- [ABR 90] ABRAMSKY, S. Computational interpretation of linear logic. **Theoretical Computer Science**, Amsterdam, v.111, n.1, p.3-57, Apr. 1993. Trabalho apresentado no Workshop on the Mathematical Foundations of programming Semantics, 1990, Kingston, Canada.
- [ABR 94] ABRAMSKY, S.; JUNG, A. Domain Theory. In: **Handbook of Logic in Computer Science**. New York : Oxford University Press, 1995. v.3, p.1-168.
- [ACI 91] ACIÓLY, B. M. **Fundamentação Computacional da Matemática Intervalar**. Porto Alegre: CPGCC da UFRGS, 1991. 253p.
- [ACI 91a] ACIÓLY, B. M.; DIMURO, G. P.; CLAUDIO, D. M. Toward a Computational Interval Mathematics. In: INTERNATIONAL SYMPOSIUM ON COMPUTER ARITHMETIC AND SCIENTIFIC COMPUTING, 1991, Oldenburg. **Proceedings...**Oldenburg: Universität Oldenburg, 1991.
- [AGU 2001] AGUIAR, M. S.; COSTA, A. C. R. Autômatos Celulares para Análise da Monotonicidade da Declividade de Áreas Geológicas. In: WORKSHOP BRASILEIRO DE GEOINFORMÁTICA, 3., 2001, Rio de Janeiro. **Anais...** Porto Alegre: Sociedade Brasileira de Computação, 2001. p. 87-94.
- [ALB 97] ALBRECHT, A. **Theoretical Computer Science**, Amsterdam, v.185, n.1, p.217-235, Oct. 1997.
- [BAE 90] BAETEN, J. C.; WEIJLAND, W. P. **Process Algebra**. Cambridge: Cambridge University Press, 1990.
- [BAR 84] BARENFREGT, H. P. **The Lambda Calculus: its Syntax and Semantics**. Amsterdam: North-Holland, 1984.
- [BEA 94] BENABOU, J.; LOISEAU, B. Orbits and monoids in topos. **Journal of Pure and applied algebra**,[S.l.], p.29-54, 1984.
- [BED 94] BEDREGAL, B. R. C. **Sistemas de Informação Contínuos**. Recife: CPGCC/UFPE, 1994. 109p.
- [BER 78] BERRY, G. Stable Models of Typed Lambda-Calculi. In: ICALP CONFERENCE, 5., 1978, Udine. **Proceedings...** Udine: [s.n.], 1978.
- [BIR 76] BIRD, R. **Programs and Machines**. [S.l.:sn], c 1976.
- [BLA 97] BLANCK, J. Domain Representability of Metric Spaces. **Pure and Applied Logic**, [S.l.], v.83, p.225-247, 1997.
- [BLU 98] BLUM, L.; CUCKER, F.; SCHUB, M.; SMALE, S. **Complexity and Real Computation**. New York: Springer-Verlag, 1998. p.453.
- [BOA 90] BOAS, P. E. Machine Models and Simulations. In: **Handbook of Theoretical Computer Science**. Amsterdam: Elsevier Science Publishers B. V., 1990. p.1-66.
- [BON 95] BONSANGUE, F.; BREUGEL, F.; RUTTEN, J. **Generalized Ultrametric Spaces: Completion, Topology, and Powerdomains via the Yoneda Embedding**. Amsterdam: Centrum voor Wiskunde en Informatica, 1995. 43p. (Report, CS-R9560).

- [BOV 94] BOVET, D. P.; CRESCENZI, P. **Introduction to the Theory of Complexity**. New York: Prentice Hall, 1994. p.282.
- [COO 73] COOK, S. A. ; RECKHOW, R. A. Time bounded random access machine. **Journal of Computer and System Sciences**, New York, n.7, p.354-375, 1973.
- [COS 94] COSTA, A. C. R., MOREIRA, A.F. **Resource-flow interpretation of sequent calculi**. Porto Alegre: CPGCC da UFRGS, 1994. 23p. Technical report - Projeto SIAD.
- [DIM 91] DIMURO, G. P. **Domínios Intervalares da Matemática Computacional**. Porto Alegre: CPGCC da UFRGS, 1991. 321p.
- [DIM 91a] DIMURO, G. P.; ACIÓLY, B. M.; CLAUDIO, D. M. About Range of Functions - A Domain Approach. In: INTERNATIONAL AMSZE CONFERENCE INFORMATION & SYSTEM, 1991, Hangzhou. **Proceedings...** Beijing: International Academic Publishers, 1991. v.2, p. 553-556.
- [DIM 91b] DIMURO, G. P.; ACIÓLY, B. M.; CLAUDIO, D. M. The Partial Real Interval Domain. In: INTERNATIONAL SYMPOSIUM ON COMPUTER ARITHMETIC AND SCIENTIFIC COMPUTAION, 1991, Oldenburg. **Proceedings...** Oldenburg: Universität Oldenburg, 1991.
- [DIM 96] DIMURO, G. P. **Uma Construção dos Reais Computáveis utilizando Espaços Coerentes de Intervalos**. Porto Alegre: CPGCC da UFRGS, 1996. 150 p. (E.Q. - 05).
- [DIM 96a] DIMURO, G. P.; COSTA, A. C. R.; CLAUDIO, D. M. A Coherence Space of Rational Intervals. In: WORKSHOP ON COMPUTER ARITHMETIC, INTERVAL AND SYMBOLIC COMPUTATION, 2., 1996, Recife. **Anais...** Recife: UFPE-DI, 1996. p.26-28.
- [DIM 98] DIMURO, G. P. **A Global Constructive Representation for Second Ordered Systems in Interval Bi-Structured Coherence Spaces,with an Application in Interval Mathematics**. Porto Alegre: CPGCC/UFRGS, 1998. 312 p. Tese de Doutorado.
- [DIM 99] DIMURO, G. P.; COSTA, A. C. R.; CLAUDIO, D. M. A Construção de Representações Globais em Espaços Coerentes Bi-estruturados, com aplicação em Computação Científica. In: CLEI - CONFERENCIA LATINOAMERICANA DE INFORMATICA ,25.,1999 , Asunción, Paraguai. **Anales...** Asunción: Universidad Autónoma de Asunción,1999.v. 2,p.648-673.
- [DIM 2000] DIMURO, G. P.; COSTA, A. C. R.; CLAUDIO, D. M. A Coherence Space of Rational Intervals for a Construction of IR. **Journal of Reliable Computing**, Dordrecht, v.6, n.2, p.139-178, 2000.
- [EDA 93] EDALAT, A.; SMYTH, M. B. Information Categories.**Applied Categorical Structures**, [S.l.], v.1, p. 197-232, 1993.
- [EDA 95] EDALAT, A. Domain Theory and Integration. **Theoretical Computer Science**, Amsterdam, v.151, n.1, p.163-193, Nov. 1995. Trabalho apresentado no Workshop on Topology and Completion in Semantics, 1993, Chartres, France.
- [EDA 97] EDALAT, A. **Domains for Computation in Mathematics, Physics and Exact Real Arithmetic**. London: Department of Computing/ Imperial College of Science, Technology and Medicine, 1997. Trabalho apresentado na Summer School on New Paradgms for Computation on Classical Spaces; Workshop on Computation and Approximation, 3.,1997, Birmingham.

- [EDA 95A] EDALAT, A.; ESCARDÓ, M. H. **Integration in Real PCF**. London: Department of Computing/Imperial College, 1995. 10p. Trabalho apresentado em Logic in Computer Science, 1996.
- [EDA 95b] EDALAT, A. Dynamical Systems, Measures and Fractals via Domain Theory. **Information and Computation**, Cambridge, v.120, n.1, p.32-48, July 1995.
- [EDA 96] EDALAT, A. Power Domains and Iterated Functions Systems. **Information and Computation**, Cambridge, v.124, n.2, p.182-197, Feb. 1996.
- [EDA 97] EDALAT, A.; SÜNDERHAUF, P. **A Domain Theoretic Approach to Computability on the Real Line**. London: Department of Computing/Imperial College of Science, Technology and Medicine, 1997. Disponível em: <http://theory.doc.ic.ac.uk/people/Edalat/reals.ps.Z>. Acesso em: mar. 2001.
- [EDA 98] EDALAT, A.; HECKMANN, R. A Computational Model for Metric Spaces. **Theoretical Computer Science**, Amsterdam, v. 193, p.53-73, 1998.
- [EDA 98a] EDALAT, A.; SÜNDERHAUF, P. A Domain Theoretic Approach to Computability on the Real Line. **Theoretical Computer Science**, Amsterdam, v. 210, n.1, p.73-98, 1998.
- [EDA 98b] EDALAT, A.; POTTS, J. P.; SÜNDERHAUF, P. Lazy Computation with Exact Real Numbers. In: ACM SIGPLAN INTERNATIONAL CONFERENCE ON FUNCTIONAL PROGRAMMING, 3., 1998. **Proceeding...** [S.l.]: ACM, 1998. p.185-194.
- [ERS 72] ERSHOV, Y. L. Computational Functionals of Finite Types. **Algebra and Logic**, [S.l.], v.11, n.4, p. 367-437, 1972.
- [ESC 94] ESCARDÓ, M. H. **Towards a Notion of Primitive recursive Real Function**. London: Department of Computing/Imperial College, 1994. 23p.
- [ESC 95] ESCARDÓ, M. H. **Induction and Recursion on the Real Line**. London: Department of Computing/Imperial College, 1995. 26p.
- [ESC 96] ESCARDÓ, M. H. PCF Extended With Real Numbers. **Theoretical Computer Science**, Amsterdam, v. 162, n.1, p.79-115, 1996.
- [ESC 97] ESCARDÓ, M. H.; STREICHER, T. Induction and Recursion on the Partial Real Line via Biquotients of Bifree Algebras. In: IEEE SYMPOSIUM ON LOGIC IN COMPUTER SCIENCE, 12., 1997. **Proceedings...** [S.l.]: IEEE, 1997.
- [ESC 99] ESCARDÓ, M. H.; STREICHER, T. Induction and recursion on the partial real line with applications to Real PCF. **Theoretical Computer Science**, Amsterdam, v. 210, n., p. 121-157, 1999.
- [ESC 99a] ESCARDÓ, M. H. A Metric Model of PCF. In: WORKSHOP ON REALIZABILITY SEMANTICS AND APPLICATIONS/FEDERATED LOGIC CONFERENCE, 1999, Trento. **Proceedings...**[S.l.:s.n.], 1999.
- [FLO 67] FLOYD, R. Assigning meanings to programs in Mathematical Aspects of Computer Science. In: AMERICAN MATHEMATICAL SOCIETY, New York. **Proceedings...** [S.l.:s.n.], 1967. p.19-32.
- [GIA 93] GIANANTONIO, P. **A functional Approach to Real Number Computation**. Pisa: University of Pisa, 1993.
- [GIA 96] GIANANTONIO, P. Real Number Computability and Domain Theory. **Information and Computation**, Cambridge, v.127, n.1, p.11-25, 1996.

- [GIR 86] GIRARD, J. The System F of Variable Types, Fifteen Years Later. **Theoretical Computer Science**, Amsterdam, n.45, p.159-192, 1986.
- [GIR 87] GIRARD, J. Linear Logic. **Theoretical Computer Science**, Amsterdam, n.50, p.1-102, 1987.
- [GIR 89] GIRARD, J. ; LAFONT, Y.; TAYLOR, P. **Proofs and Types**. Cambridge: Cambridge University Press, 1989. 176p.
- [GIR 92] GIRARD, J. **Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur**. Thèse d'État, Iniversity of Paris VII, 1992.
- [GIR 96] GIRARD, J. **Coherent Banach Spaces: A Continuous Denotational Semantics**. Marseille: Institut de Mathématique de Luminy, 1996. 23p.
- [GIE 80] GIERS, K. et al. **A Compedium of Continuous Lattice**. Berlin: Springer-Verlag, 1980.
- [GOG 77] GOGUEN, J.A.; THATCHER, J. W.; WAGNER, E. G. An initial algebra approach to the specification, correctness and implementation of abstract data types. **Current Trents in Programming Methodology**. New York: Springer-Verlag, 1993. p.77-84.
- [GUN 92] GUNTER, C. A. **Semantics of Programming Languages**. Cambridge: MIT Press, 1992.
- [GUN 94] GUNTER, C. A. The Semantics of Types in Programming Languages. In: **Handbook of Logic in Computer Science**, New York : Oxford University Press, 1995. v.3, p.394-489.
- [HOA 69] HOARE, C. A. R. An axiomatic basis for computer programming In: COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY. 12., 1969. **Proceedings...** [S.l.:s.n.], 1969. p.576-580.
- [HOA 85] HOARE, C. A. R. **Communicating Sequential Processes**. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [HOO 93] HOOFFMAN, R. Continuous Information Systems. **Information and Computation**, Cambridge, v.105, p.42-71, 1993.
- [HUT 95] HUTH, M. A Maximal Monoidal Closed Category of Distributive Algebraic Domains. **Information and Computation**, Cambridge, v.116, n.1, p.10-25, Jan. 1995.
- [ISL 2001] ISLABAO, M. H. **Definição de uma Linguagem de Programação Baseada no Modelo de Espaços Coerentes para Processos Concorrentes**. 2001. Projeto de Diplomação, UCPel, Pelotas.
- [JUN 89] JUNG, A. **Cartesian Closed Categories of Domains**. Amsterdam: Center for Mathematics and Computer Science, 1989.
- [JOH 82] JOHNSTONE, P. T. **Stone Spaces**. Cambridge: Cambridge University Press, 1982. (Cambridge Studies in Advanced Mathematics, v.3).
- [KAM 84] KAMIMURA, T.; TANG, A. Total Objects of Domains. **Theoretical Computer Science**, Amsterdam, n.29, p.155-166, 1984.
- [KAM 84a] KAMIMURA, T.; TANG, A. Effectively Given Spaces. **Theoretical Computer Science**, Amsterdam, n.34, p.275-288, 1984.
- [KEY 98] KEYE, M. Domain Theoretic Models of Topological Spaces. **Theoretical Computer Science**, n.13, 1998. 9p. Trabalho apresentado no Workshop

- on Computation and Approximation, 3.,1997, Birmingham. Disponível em (<http://citeseer.nj.com/martin98domain.html>) Acessado em: mai.2000.
- [KEY 99] KEYE, M. **A Foundation for Computation**. New Orleans: Department of Mathematics/Tulane University, 1999. PhD Thesis.
- [KEY 99a] KEYE, M. **The Measurement Process in Domain Theory**. New Orleans: Department of Mathematics/Tulane University, 1999. 41p.
- [LAC 55] LACOMBE, D. Extension de la Notion de Fonction Récursive aux Fonctions d'une ou Plusieurs Variable Réelles I. **Comptes Rendus**, [S.l.], v. 240, p.2478-80, 1955.
- [LAF 88] LAFONT, Y. **Introduction to Linear Logic**. Isle of Thorns: [s.n.], 1988. Lecture Notes for the Summer School on Constructive Logics and Category Theory.
- [LAW 87] LAWSON, J. **The Versatile Continuous Order**. Baton Range: Department of Mathematics/ Louisiana State University, 1987.
- [LAW 97] LAWSON, J. **Elements of Domain Theory**. Baton Rouge: Department of Mathematics/ Louisiana State University, 1997. Trabalho apresentado na Summer School on New Paradigms for Computation on Classical Spaces; Workshop on Computation and Approximation, 3.,1997, Birmingham.
- [LAW 73] LAWVERE, F. W. Metric Spaces, Generalized Logic, and Closed Categories. In: SEMINARIO MATEMATICO E FISICO DE MILANO, 43., 1973, Milão. **Rendiconti**. [S.l: s.n.], 1973. p.135-166.
- [MAN 86] MANES, E. G.; ARBID, M. A. **Algebraic Approach to program Semantics**. New York: Springer-Verlag, 1986. 351p.
- [MES 74] MESEGUER, J.; GOGUEN, J. A. **Algebraic Methods in semantics**. Initiality, induction and computability. Cambridge: University of Cambridge, 1985.
- [MII 74] MILNE, R. E. **The formal semantics of computer languages and their implementations**. Cambridge: University of Cambridge, 1974. PhD. Thesis.
- [MIL 75] MILNER, R. Process: A mathematical model of computing agents. In: LOGIC COLLOC, 1973, Amsterdam. **Proceedings...** [S.l.: s.n.], 1975. p.157-174.
- [MIL 90] MILNER, R. Operational and algebraic semantics of concurrent processes. In: **Handbook of Theoretical in Computer Science**. Amsterdam: Elsevier Science Publishers B. V., 1990. v.2, p.1203-1242.
- [MIL 80] MILNER, R. **A Calculus of Communicating Systems**. Berlin: Springer-Verlag, 1980. (Lecture Notes in Computer Science, v.92).
- [MIN 67] MINSKY, M. **Computation, Finite and Infinite Machines**. Prentice Hall, Englewood Cliffs, NY: Prentice Hall, 1967.
- [MIS 99] MISLOVE, M. **Generalizing Domain Theory**. New Orleans: Tulane University, 1999. 19p.
- [MIT 90] MITCHELL, J. C. Types systems for programming languages. In: **Handbook of Theoretical Computer Science**. Amsterdam: Amsterdam: Elsevier Science Publishers B. V., 1990. v.2, p.365-458.
- [NIE 81] NIELSON, M.; PLOTKIN, G.; WHINSKEL, G. Petri Nets, event structures and domains, Part I. **Theoretical Computer science**, Amsterdam, v.13, p.85-108, 1981.
- [PLO 77] PLOTKIN. G. LCF considered as a Programming Language. **Theoretical Computer Science**, Amsterdam, v.5, n.1, p.233-255, 1977.

- [PLO 81] PLOTKIN, G. **Post-Graduate Lecture Notes in Advanced Domain Theory**. Edinburgh: Department of Computer Science/University of Edinburgh, 1981.
- [PHO 94] PHOA, W. From Terms Models to Domains. **Information and Computation**, Orlando, v.109, n.1-2, p.211-255, Feb./Mar. 1994.
- [PLU 98] PLUNME, D. **A Calculator for Exact Real Number Computation**. Edinburgh: Department of Computer Science/University of Edinburgh, 1981. (4th Year Project Report).
- [POT 96] POTTS, P. J.; EDALAT, A. **Exact Real Arithmetic based on Linear Fractional Transformations**. London: Department of Computing/ Imperial College of Science, Technology and Medicine, 1996. Disponível em: <http://www-tfm.doc.ic.ac.uk/~pjp>. Acesso em: out 1999.
- [POT 97] POTTS, P. J. **Efficient and Strict Algorithms for Exact Real Arithmetic**. London: Department of Computing/ Imperial College of Science, Technology and Medicine, 1997. Trabalho apresentado na Summer School on New Paradigms for Computation on Classical Spaces/Workshop on Computation and Approximation, 3. , 1997, Birmingham.
- [POT 97a] POTTS, P. J.; EDALAT, A. **Exact Real Computer Arithmetic**. London: Department of Computing/ Imperial College of Science, Technology and Medicine, 1997. Disponível em: <http://www-tfm.doc.ic.ac.uk/~pjp>. Acesso em: out 1999.
- [POT 97b] POTTS, P. J.; EDALAT, A.; ESCARDO, M. Semantics fo Exact Real Arithmetic. In: IEEE SYMPOSIUM ON LOGIC IN COMPUTER SCIENCE, 12., 1997. **Proceedings...** [S.l.]: IEEE, 1997.
- [REI 97] REISER, R. H. S. **Estudo da Categoria Computável dos Espaços Coerentes Gerados por Conjuntos Básicos com Aplicação na Análise Real**. Porto Alegre: CPGCC da UFRGS, 1997. 121p.
- [REI 98] REISER, R. H. S.; COSTA, A. C.; CLAUDIO, D. M. **The category of unary systems - a formal construction to explicate induction and recursion on natural numbers**. Porto Alegre: PPGC/UFRGS, 1998. 114p. (EQ - 28).
- [REI 98a] REISER, R. H. S.; COSTA, A. C. R.; DIMURO, G. P. A category study of primitive recursion on natural numbers, In: WORKSHOP ON FORMAL METHODS, 1., 1998, Porto Alegre. **Anais ...** Porto Alegre: PPGC/UFRGS, 1998. p.157-162.
- [REI 99] REISER, R. H. S.; COSTA, A. C. R.; DIMURO, G. P. Deriving μ -recursive functions from some categories of unary system, In: WORKSHOP ON FORMAL METHODS, 2., 1999, Florianópolis. **Anais ...** Florianópolis: DI/UFSC, 1999. p.89-97.
- [REY 74] REYNOLDS, J. C. **Towards a theory of type structure**. New York: Springer-Verlag, 1974. p.408-425.(Lecture Notes in Computer Science, v.19).
- [REY 77] REYNOLDS, J. C. Semantics of the domain of Flow Diagrams. **Journal of Association for Computing Machinery**, New York, v. 24, n.3, p.484-503,1977.
- [REY 84] REYNOLDS, J. C. **Polimorphism is not a set-theoretic**. New York: Springer-Verlag, 1984. p.145-156. (Lecture Notes in Computer Science v. 173).
- [RUT 93] RUTTEN, R. **Initial algebra and final coalgebra for cuncurrency**. New York: Springer Verlag, 1993. p.530-582. (Lecture Notes in Computer Science, v. 803).

- [SAS 96] SASASONE, V.; NIELSON, M.; WHINSKEL, G. Models for concurrency: Towards a classification. **Theoretical Computer Science**, Amsterdam, v.170, p.297-348, 1996.
- [SCO 67] SCOTT, D. Some definitional suggestions for Automata Theory. **Journal of Computer and System Sciences**, New York, n.1, p.187-212, 1967.
- [SCO 70] SCOTT, D. Outline of a Mathematical Theory of Computation. In: ANNUAL PRINCETON CONFERENCE ON INFORMATION SCIENCE AND SYSTEMS, 1., 1970, Princeton. **Proceedings...** [S.l.: s.n.], 1970. p.169-176.
- [SCO 71] SCOTT, D.; STRACHEY, C. Toward a mathematical semantics for computer languages. In: SYMP. ON COMPUTER AND AUTOMATA, Microwave Research Institute Symposia Series Princeton. **Proceedings...** New York: Polytechnic Institute of Brooklyn, p.19-46, 1971.
- [SCO 71] SCOTT, D. **The lattice of flow diagrams**. Berlin: Springer Verlag, 1971. p.311-372.(Lecture Notes in Mathematics, 188).
- [SCO 72] SCOTT, D. Continuous Lattices. In: TOPOSES, ALGEBRAIC, GEOMETRY AND LOGIC, 1972. **Proceedings...** Berlin: Springer-Verlag, 1972. p.97-136. (Lecture Notes in Mathematics, n. 274).
- [SCO 72a] SCOTT, D. **Lattice Theory, Data Types and Semantics. Formal Semantics and Programming Languages**. Englewood Cliffs: Prentice Hall, 1972. p.65-106.
- [SCO 73] SCOTT, D. Models for Various Type-Free Calculi. In: LOGIC, METHODOLOGY AND PHILOSOPHY OF SCIENCE, 4., 1973. **Proceedings...** Amsterdam: North-Holland, 1973. p.157-187.
- [SCO 76] SCOTT, D. Data Types as Lattices. **SIAM Journal of Computing**, Philadelphia, v.5, n.1, p.522-587, 1976.
- [SCO 82] SCOTT, D. Domains for Denotational Semantics. In: COLLOQUIUM ON AUTOMATA LANGUAGES AND PROGRAMMING. **Proceedings...** Berlin: Springer-Verlag, 1982. p.577-613. (Lecture Notes in Computer Science, v.140).
- [SCO 82a] SCOTT, D. Lectures on a Mathematical Theory of Computation. In: **Theoretical Foundations of Programming Methodology**. Dordrecht: D. Reidel, 1982. p.145-292. (Nato Advanced Study Institutes Series, Series C, v. 91).
- [SCO 90] SCOTT, D., GUNTER, C. A. Semantic Domains. In: **Handbook of Theoretical Computer Science**. Amsterdam: Elsevier Science Publishers B. V., 1990. p.635-674.
- [SEL 96] SELLANES, R. G. S. **Estratégias de Computação Seqüenciais e Paralelas sobre Espaços Coerentes**. Porto Alegre: CPGCC da UFRGS, 1996. 143p.
- [SHE 63] SHEPHERDSON, J. C.; STURGIS, H. E. Computability of recursive functions. **Journal of Association for Computing Machinery**, New York, n.10, p.217-255, 1963.
- [SMY 77] SMYTH, M. B. Effectively given Domains. **Theoretical Computer Science**, Amsterdam, v.5, n.1, p.257-274, 1977.
- [SMY 83] SMYTH, M. B. Powerdomains and Predicate Transformers: a Topological View. In: COLLOQUIUM ON AUTOMATA LANGUAGES AND PROGRAMMING, 10., 1983, Barcelona. **Proceedings...** Berlin: Springer-Verlag, 1983. p.662-675. (Lecture Notes in Computer Science, v. 154).
- [SMY 83a] SMYTH, M. B. The largest cartesian closed category of domains. **Theoretical Computer Science**, Amsterdam, v.27, p.109-119, 1983.

- [SMY 87] SMYTH, M. B. **Quasi Uniformities**: Reconciling Domains with Metric Spaces. Berlin: Springer-Verlag, 1987. p.236-253 (Lectures Notes in Computer Science, v. 298). Trabalho apresentado no Workshop on Mathematical Foundations of Programming Language Semantics, 3., 1987, New Orleans.
- [SMY 90] SMYTH, M. B. Topology. In: **Handbook of Logic in Computer Science**. New York : Oxford University Press, 1995. v.1, p.641-751.
- [SMY 91] SMYTH, M. B. Totally Bounded Spaces and Compact Ordered Spaces as Domains of Computation. **Topology and Category Theory in Computer Science**, New York, p.207-229, 1991.
- [STA 85] STATMAN, R. Logical relations and the typed λ -calculus. **Information and Control**. [S.l.], v.65, p.17-26, 1985.
- [STO 61] STOLL, R. **Set Theory and Logic**. New York: Dover Publications, 1961. p.474.
- [STV 88] STOLTENBERG-HANSEN, V. ; TUCKER, J. V. Complete Local rings as Domains. **Journal of Symbolic Logic**, [S.l.], v.53, p.603-624, 1988.
- [STV 94] STOLTENBERG-HANSEN, V. ; LINDSTRÖM, I.; GRIFFOR, E. R. **Mathematical Theory of Domains**. Cambridge: Cambridge University Press, 1994. 349p.
- [STV 95] STOLTENBERG-HANSEN, V. ; TUCKER, J. V. Effective Algebras. In: **Handbook of Logic in Computer Science**, New York : Oxford University Press, 1995. v.4, p.357-526.
- [STY 77] STOY, J. E. **Denotational Semantics**: The Scott-Strachey Approach to Programming languages. Cambridge: MIT Press, 1977.
- [TRO 92] TROELSTRA, A. S. **Lectures on Linear Logic**. Stanford: CSLI/Leland Stanford Junior University, 1992. (Lecture Notes, n.29).
- [TRO 92] TROELSTRA, A. S. Natural deduction for intuitionistic linear logic. **Annals of Pure Applied Logic**, [S.l.] v.73, p.79-108, 1997.
- [VIC 87] VICKERS, S. **Topology via Logic**. Cambridge: Cambridge University Press, 1987.
- [VIC 87] VICKERS, S. Information Systems for Continuous Posets. **Theoretical Computer Science**, Amsterdam, v.114, p.201-229, 1993.
- [VOE 92] VOEVODIN, V. V. **Mathematical Foundations of Parallel Computing**. Singapore: World Scientific Publishing Co, 1992. p.343.
- [WAG 90] WAGNER, E. G. Algebraic Semantics. In: **Handbook of Logic in Computer Science**, Oxford: Clarendon Press, 1990. v.4, p.323-386.
- [WEI 80] WEIHRAUCH, K.; SCHREIBER, U. **Berechenbarkeit auf CPO's**. [S.l.]: Rheinisch-Westfälische Hochschule Aachen, 1980.
- [WEI 81] WEIHRAUCH, K.; DEIL, T. Embedding Metric Spaces into CPO's. **Theoretical Computer Science**, Amsterdam, v.16, p.5-24, 1981.
- [WHI 77] WHINSKEL, G. **An introduction to event structures**. New York: Springer-Verlag, 1988. p.365-398. (Lectures Notes in Computer Science, v. 354).
- [WHI 82] WHINSKEL, G. **Event structure semantics for CCS and related languages**. New York: Springer-Verlag, 1982. p.561-576. (Lectures Notes in Computer Science, v. 140).
- [WHI 88] WHINSKEL, G. **Event structures**. New York: Springer-Verlag, 1988. p.325-392. (Lectures Notes in Computer Science, v. 255).

- [WHI 85] WHINSKEL, G. On Powerdomains and modality. **Theoretical Computer Science**, Amsterdam, v.36, p.127-137, 1985.
- [WHI 95] WHINSKEL, G.; NIELSON, M. Models for concurrency. In: **Handbook of Logic in Computer Science**, New York : Oxford University Press, 1995. v.4, p.1-148.
- [ZHA 89] ZHANG, G. **d-I Domains as Information Systems**. New York: Springer-Verlag, 1989. p.773-788 (Lectures Notes in Computer Science, v. 372). Trabalho apresentado no International Colloquium in Automata, Languages and Programming, 16., 1989, Stresa.
- [ZHA 91] ZHANG, G. **Logic of Domains**. Boston: Birkhäuser, 1991. Progress in Theoretical Computer Science.
- [ZHA 91a] ZHANG, G. Q. A monoidal closed category of event structures. In: CONFERENCE ON MATHEMATICS FOUNDATIONS OF PROGRAMMING SEMANTICS, 7., 1991, Pittsburg. **Proceedings...** [S.l.:s.n.], 1991.
- [ZHA 92] ZHANG, G. Stable Neighbourhoods. **Theoretical Computer Science**, Amsterdam, v.93, n.1, p.143-157, Feb. 1992.
- [ZHA 92a] ZHANG, G. d-I Domains as Prime Information Systems. **Information and Computation**, Orlando, v.100, n.2, p.151-177, Oct. 1992.
- [ZHA 96] ZHANG, G. Quasi-Prime Algebraic Domains. **Theoretical Computer Science**, Amsterdam, v.155, n.1, p.221-264, Feb. 1996.
- [ZHA 96a] ZHANG, G. The Largest Cartesian Closed Category of Stable Domains. **Theoretical Computer Science**, Amsterdam, v.166, n.1-2, p.203-219, Oct. 1996.