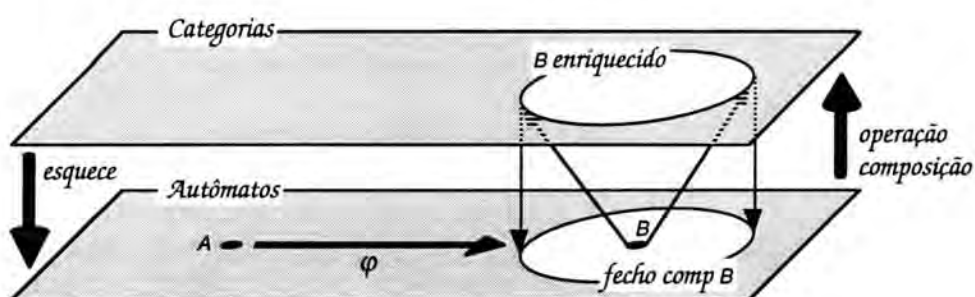




UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



Reificação de Objetos Concorrentes

Paulo Fernando Blauth Menezes

(Licenciado)

Dissertação para obtenção do Grau de Doutor em Matemática

Orientador: Doutor Amílcar dos Santos Costa Sernadas

Juri:

Presidente: Reitor da Universidade Técnica de Lisboa

Vogais: Doutora Maria Cristina Sales Viana Seródio Sernadas

Doutor Luís Pereira de Quintanilha e Mendonça Dias Torres Magalhães

Doutor José Félix Gomes da Costa

Doutor Luís Fernando Lopes Monteiro

Doutor José Manuel Cunha Leal Molarinho Carmo

Doutor Rui António Loja Fernandes

Fevereiro de 1997



UFRGS

SABI



05232807

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Paulo Fernando Blauth Menezes
Departamento de Matemática, Secção de Ciência da Computação
Doutoramento em Matemática
Orientador: Professor Doutor Amílcar dos Santos Costa Sernadas
Fevereiro de 1997

Reificação de Objetos Concorrentes

Resumo

Autômatos não-seqüenciais constituem um domínio semântico categorial do tipo não-intercalação para sistemas reativos, comunicantes e concorrentes. É baseado em sistemas de transições etiquetados, inspirado em "Redes de Petri são Monóides" de Meseguer e Montanari, onde as operações de sincronização e encapsulação são functoriais e as reificações constituem uma classe de morfismos especiais. Do que se tem conhecimento, é o primeiro modelo de concorrência a satisfazer a composicionalidade diagonal, ou seja, onde as reificações compõem (verticalmente) e distribuem-se sobre a composição paralela (verticalmente). Adjunções entre autômatos não-seqüenciais, redes de Petri e autômatos seqüenciais são introduzidas estendendo a abordagem de Winskel, Nielsen e Sassone onde é proposta uma classificação formal para modelos de concorrência. Dos passos que envolvem a passagem de um modelo para outro, pode-se inferir que os autômatos não-seqüenciais são mais concretos do que as redes de Petri e os autômatos seqüenciais. Para experimentar o domínio semântico proposto, é dada semântica a uma linguagem concorrente, baseada nos objetos, denominada Náutilus. Trata-se de uma versão simplificada e revisada da linguagem de especificação orientada aos objetos GNOME, onde são introduzidos algumas facilidades especiais, inspiradas no domínio semântico, como a reificação e a agregação. Neste contexto, a composicionalidade diagonal é uma propriedade essencial para dar a semântica.

Reification of Concurrent Objects

Abstract

Nonsequential automata constitute a non-interleaving categorial semantic domain for reactive, communicating and concurrent systems. It is based on labeled transition systems, inspired by Meseguer and Montanari's "Petri Nets are Monoids", where synchronization and encapsulation operations are functorial and a class of morphisms stands for reification. It is, for our knowledge, the first model for concurrency which satisfies the diagonal compositionality requirement, i. e., reifications compose (vertical) and distribute over the parallel composition (horizontal). Adjunctions between nonsequential automata, Petri nets and sequential automata are provided extending the approach of Winskel, Nielsen and Sassone where a scene for a formal classification of models for concurrency is set. The steps of abstraction involved in moving between models show that nonsequential automata are more concrete than Petri nets and sequential automata. To experiment with the proposed semantic domain, a semantics for a concurrent, object-based language named Nautilus is given. It is a simplified and revised version of the object-oriented specification language GNOME, introducing some special features inspired by the semantic domain such as reification and aggregation. The diagonal compositionality is an essential property to give semantics in this context.

Palavras Chaves

Autômato Não-Sequencial
Concorrência
Reificação
Semântica
Sincronização
Teoria das Categorias

Keywords

Category Theory
Concurrency
Reification
Semantic
Nonsequential Automaton
Synchronization

Agradecimentos

Ao Professor Amílcar Sernadas pela orientação, empenho e excepcionais discussões proporcionadas ao longo de todo o trabalho.

Ao Professor J. Félix Costa que incentivou e acompanhou grande parte das investigações apresentadas nesta dissertação.

Ao Carlos Caleiro, ao Jaime Ramos e ao Pedro Resende pelas importantes discussões que proporcionaram, à Paula Gouveia pelo apoio nas atividades finais após o meu retorno ao Brasil e a todos os integrantes da Secção de Ciência da Computação pelo excelente ambiente e apoio proporcionados.

À UFRGS - Universidade Federal do Rio Grande do Sul e ao CNPq - Conselho Nacional de Desenvolvimento e Pesquisa, no Brasil, por viabilizarem as condições e os recursos necessários.

Ao ESPRIT-III BRA WG 6071 IS-CORE (Information Systems - CORrectness and REusability) e BRA WG 6112 COMPASS (COM-Prehensive Algebraic approach to System Specification and development), por viabilizarem os recursos necessários para a participação em importantes eventos.

Dedicatória

Dedico esta dissertação às minhas filhas, Maria Lúcia e Maria Luiza e, com um sentimento muito especial, à minha esposa Maria Fernanda.

Índice

1	Introdução.....	1
1.1	Contexto e Objetivos	1
1.2	Abordagem e Estruturação	2
1.3	Sumário das Contribuições.....	7
2	Grafos e Categorias com Elementos Distinguidos	8
2.1	Grafo	8
2.2	Grafo Reflexivo.....	10
2.3	Grafo Com Nodos Distinguidos.....	12
2.4	Categoria com Objetos Distinguidos	15
3	Autômatos Seqüenciais.....	16
3.1	Autômato.....	16
3.2	Sincronização.....	19
3.3	Encapsulação.....	25
4	Reificação de Autômatos Seqüenciais	28
4.1	Computações.....	28
4.2	Composicionalidade Vertical.....	30
4.3	Composicionalidade Horizontal.....	33
5	Grafos e Categorias Internos.....	35
5.1	Grafo Interno.....	35
5.2	Grafo Reflexivo Interno	35
5.3	Grafo Interno Com Nodos Distinguidos	38
5.4	Categoria Interna.....	39
5.5	Categoria Interna com Objetos Distinguidos.....	42
6	Autômatos Não-Seqüenciais.....	44
6.1	Grafos Monoidais Comutativos.....	44
6.2	Autômato Não-Seqüencial.....	47
6.3	Sincronização.....	50
6.4	Encapsulação	52
7	Reificação de Autômatos Não-Seqüenciais	54
7.1	Categorias Internas à Categoria dos Monóides Comutativos.....	55
7.2	Computações Não-Seqüenciais	55
7.3	Composicionalidade Vertical.....	58
7.4	Composicionalidade Horizontal.....	60
7.5	Sincronização.....	62
7.6	Encapsulação	62
8	Redes de Petri.....	63
8.1	Grafos Estruturados.....	64
8.2	Rede de Petri	65
8.3	Rede de Petri Apontada	67
8.4	Rede de Petri Marcada	68
8.5	Rede de Petri Etiquetada.....	70
8.6	Sincronização e Encapsulação.....	72
9	Relacionamento entre Autômatos e Redes de Petri.....	75
9.1	Autômatos Seqüenciais e Redes de Petri Etiquetadas	76
9.2	Autômatos Seqüenciais e Não-Seqüenciais	79
9.3	Autômatos Não-Seqüenciais e Redes de Petri Etiquetadas	81
9.4	Reificação de Redes de Petri Etiquetadas.....	84
10	Linguagem Náutilus	88
10.1	Gramática	89
10.2	Exemplos	93

11	Semântica da Linguagem Náutilus	101
11.1	Objeto Simples	102
11.2	Reificação	107
11.3	Visão	111
11.4	Interação ou Agregação	111
11.5	Unidade	116
12	Conclusão	118
	Referências	120
	Apêndice: Semântica Formal	125
1	Sintaxe e Semântica	125
2	Semântica Formal e Linguagens de Programação	125
3	Métodos para Semântica Formal	126
	Apêndice: Teoria das Categorias	129
1	Construções Básicas	129
2	Funtores	131
3	Transformações Naturais	132
4	Produtos e Coprodutos	133
5	Limites e Colimites	135
6	Adjunções	140
7	Construções sobre Funtores	141
8	Fibrações	143
9	Categorias Concretas	145
10	Mônadas e Categorias Kleisli	147

Índice de Figuras

1	Introdução.....	1
	Figura 1.1 Composicionalidade vertical.....	1
	Figura 1.2 Composicionalidade horizontal.....	2
	Figura 1.3 Morfismo de reificação de autômatos.....	3
	Figura 1.4 Composição de morfismos de reificação.....	4
2	Grafos e Categorias com Elementos Distinguidos.....	8
	Figura 2.1 Nodos isolados são omitidos na representação gráfica de grafos.....	9
	Figura 2.2 Diagrama comutativo para morfismos de grafos.....	9
	Figura 2.3 Coproduto e Produto de grafos.....	9
	Figura 2.4 Diagrama comutativo para a função identidade.....	10
	Figura 2.5 Diagrama comutativo adicional para morfismos de grafos reflexivos.....	10
	Figura 2.6 Arcos identidade são omitidos na representação gráfica de grafos reflexivos.....	10
	Figura 2.7 Representação gráfica de um grafo reflexivo.....	11
	Figura 2.8 Coproduto e produto de grafos reflexivos.....	11
	Figura 2.9 Representação gráfica de um grafo reflexivo de um só nodo.....	12
	Figura 2.10 Instanciação dos nodos iniciais e finais.....	13
	Figura 2.11 Representação gráfica de nodos distinguidos.....	13
	Figura 2.12 Diagrama comutativo adicional para morfismos de grafos com nodos distinguidos.....	14
	Figura 2.13 Coproduto e produto de grafos reflexivos com nodos distinguidos.....	14
	Figura 2.14 Diagrama comutativo adicional para morfismos de categorias com objetos distinguidos.....	15
3	Autômatos Sequenciais.....	16
	Figura 3.1 Autômato como um morfismo de grafos reflexivos com nodos distinguidos.....	17
	Figura 3.2 Representação gráfica de um autômato.....	18
	Figura 3.3 Diagrama comutativo para morfismos de autômatos.....	18
	Figura 3.4 Soma amalgamada.....	18
	Figura 3.5 Produto de autômatos sequenciais.....	19
	Figura 3.6 Indução do funtor de sincronização.....	20
	Figura 3.7 Produto fibrado e morfismo univocamente induzido.....	20
	Figura 3.8 Autômatos original e os correspondentes transformados.....	21
	Figura 3.9 Autômatos produtor e consumidor e a correspondente composição paralela.....	22
	Figura 3.10 Sincronização dos autômatos produtor e consumidor.....	23
	Figura 3.11 Tabela de sincronizações determinada por um colimite com a forma de um diamante.....	23
	Figura 3.12 Tabela de sincronizações determinada por um colimite.....	24
	Figura 3.13 Exemplo de tabela de sincronizações para dois autômatos.....	24
	Figura 3.14 Determinação de um colimite de um diagrama de 2-picos usando somas amalgamadas.....	25
	Figura 3.15 Indução do funtor de encapsulação.....	26
	Figura 3.16 Diagramas comutativos.....	26
	Figura 3.17 Morfismo de encapsulação determinado por uma soma amalgamada.....	27
	Figura 3.18 Encapsulação de um autômato.....	27
4	Reificação de Autômatos Sequenciais.....	28
	Figura 4.1 Autômato sequencial e o seu correspondente fecho computacional.....	30
	Figura 4.2 Composição horizontal de transformações naturais.....	30
	Figura 4.3 Composição de reificações é a composição na categoria Kleisli restrita à forma dos autômatos.....	31
	Figura 4.4 Reificação de autômatos não necessariamente preserva a etiquetação.....	32
	Figura 4.5 Reificação de autômatos com etiquetação induzida.....	32
	Figura 4.6 Composição de reificações de autômatos com etiquetação induzida.....	33
	Figura 4.7 O funtor fecho computacional não preserva limites.....	33
	Figura 4.8 A composição paralela de reificações não é, em geral, uma reificação.....	34
5	Grafos e Categorias Internos.....	35
	Figura 5.1 Diagramas comutativos para morfismos de grafos internos.....	35
	Figura 5.2 Diagramas comutativos para a categoria dos grafos reflexivos internos.....	36
	Figura 5.3 Morfismos univocamente induzidos pelo produto categorial.....	37
	Figura 5.4 Morfismos univocamente induzidos pelos igualadores.....	37
	Figura 5.5 Diagrama comutativo para morfismos de grafos internos com nodos distinguidos.....	38
	Figura 5.6 Domínio da operação de composição determinado por um produto fibrado.....	39
	Figura 5.7 Domínio, codomínio, associatividade e identidade da composição.....	40

Figura 5.8	Morfismos univocamente induzidos.....	40
Figura 5.9	Diagramas comutativos para funtores internos.....	40
Figura 5.10	Morfismos univocamente induzidos.....	41
Figura 5.11	Morfismos univocamente induzidos.....	42
Figura 5.12	Diagrama comutativo para morfismos de categorias internas com objetos distinguidos.....	43
6	Autômatos Não-Sequenciais.....	44
Figura 6.1	Representação gráfica de nodos e arcos estruturados.....	45
Figura 6.2	Simplificação da representação gráfica de grafos monoidais.....	45
Figura 6.3	Representação gráfica de um grafo monoidal.....	45
Figura 6.4	Produto ou coproduto de grafos monoidais.....	46
Figura 6.5	Representação gráfica de um grafo monoidal com um único nodo.....	46
Figura 6.6	Produto ou coproduto de grafos monoidais com nodos distinguidos.....	47
Figura 6.7	Representação gráfica de um autômato não-sequencial.....	48
Figura 6.8	Quadrado de independência.....	49
Figura 6.9	Produto fibrado.....	50
Figura 6.10	Tabela de sincronizações determinada por um colimite.....	51
Figura 6.11	Sincronização dos autômatos produtor e consumidor.....	52
Figura 6.12	Morfismo de encapsulação determinado por uma soma amalgamada.....	53
Figura 6.13	Encapsulação de um autômato não-sequencial.....	53
7	Reificação de Autômatos Não-Sequenciais.....	54
Figura 7.1	Autômato não-sequencial e o seu correspondente fecho computacional.....	58
Figura 7.2	Composição de reificações é a composição na categoria Kleisli restrita à forma dos autômatos.....	59
Figura 7.3	Reificação de autômatos não-sequenciais com etiquetagem induzida.....	59
Figura 7.4	Composição de reificações de autômatos não-sequenciais com etiquetagem induzida.....	59
Figura 7.5	Fecho computacional não-sequencial preserva a composição paralela de autômatos.....	61
Figura 7.6	Morfismos univocamente induzidos.....	61
Figura 7.7	Sincronização de autômatos não-sequenciais reificados.....	62
Figura 7.8	Encapsulação de autômatos não-sequenciais reificados.....	62
8	Redes de Petri.....	63
Figura 8.1	Diagramas comutativos para morfismos de grafos estruturados.....	64
Figura 8.2	Diagrama comutativo para morfismos de grafos estruturados com nodos distinguidos.....	65
Figura 8.3	Representação gráfica de transições de redes.....	65
Figura 8.4	Tipos de transições.....	65
Figura 8.5	Simplificação na representação gráfica de redes.....	66
Figura 8.6	Representação gráfica de uma rede de Petri.....	66
Figura 8.7	Coproduto e produto de redes de Petri.....	67
Figura 8.8	Produto de redes de Petri.....	67
Figura 8.9	Simplificação na representação gráfica de redes apontadas.....	68
Figura 8.10	Produto de redes de Petri apontadas.....	69
Figura 8.11	Coproduto e produto de redes de Petri marcadas.....	70
Figura 8.12	Coproduto de redes em Meseguer e Montanari.....	70
Figura 8.13	Rede de Petri etiquetada.....	71
Figura 8.14	Produto fibrado.....	72
Figura 8.15	Tabela de sincronizações.....	73
Figura 8.16	Sincronização de redes.....	73
Figura 8.17	Morfismo de encapsulação determinado por uma soma amalgamada.....	74
9	Relacionamento entre Autômatos e Redes de Petri.....	75
Figura 9.1	Relacionamento entre as diversas categorias.....	75
Figura 9.2	Um autômato sequencial e a correspondente rede de Petri etiquetada.....	77
Figura 9.3	Funções totais de origem e destino.....	78
Figura 9.4	Escolha.....	78
Figura 9.5	Concorrência.....	78
Figura 9.6	Ciclo.....	79
Figura 9.7	Um autômato sequencial e as passagens para não-sequencial e, novamente, para sequencial.....	80
Figura 9.8	Diagramas comutativos para autômatos não-sequenciais apontados.....	82
Figura 9.9	Diagrama comutativo para morfismos entre autômatos não-sequenciais apontados.....	82
Figura 9.10	Morfismo univocamente induzido.....	83
Figura 9.11	Escolha.....	83
Figura 9.12	Concorrência.....	84
Figura 9.13	Ciclo.....	84
Figura 9.14	Um autômato não-sequencial e a correspondente rede de Petri etiquetada.....	85
Figura 9.15	Composição de reificações é a composição na categoria Kleisli sem considerar as etiquetas.....	86
Figura 9.16	O funtor fecho computacional não preserva produtos.....	86
10	Linguagem Náutilus.....	88
Figura 10.1	Jantar dos cinco filósofos.....	97
Figura 10.2	Relacionamento entre os objetos componentes de uma fábrica.....	97

11	Semântica da Linguagem Náutilus	101
	Figura 11.1 Semântica de um objeto em Náutilus como um morfismo de reificação	107
	Figura 11.2 Representação gráfica da semântica de uma reificação.....	110
	Figura 11.3 Representação gráfica da semântica de uma visão.....	112
	Figura 11.4 Relacionamento entre autômatos componentes de uma interação e de uma agregação.....	115
	Figura 11.5 Autômato (origem) resultante de uma interação e de uma agregação.....	116
12	Conclusão	118
	Referências.....	120
	Apêndice: Semântica Formal.....	125
	Apêndice: Teoria das Categorias.....	129
	Figura 1.1 Diagrama comutativo para fatoração de morfismos	131
	Figura 2.1 Fibra.....	131
	Figura 3.1 Diagrama comutativo para transformações naturais.....	132
	Figura 3.2 Composição vertical de transformações naturais.....	133
	Figura 3.3 Composição horizontal de transformações naturais.....	133
	Figura 4.1 Diagrama comutativo para produtos binários.....	133
	Figura 4.2 Diagrama comutativo para produtos finitos/arbitrários.....	134
	Figura 4.3 Diagrama comutativo para coprodutos binários.....	134
	Figura 4.4 Diagrama comutativo para coprodutos finitos/arbitrários	135
	Figura 5.1 Diagrama comutativo para igualador.....	135
	Figura 5.2 Diagrama comutativo para co-igualador.....	135
	Figura 5.3 Diagrama comutativo para produto fibrado.....	136
	Figura 5.4 Diagrama comutativo.....	137
	Figura 5.5 Produto fibrado.....	137
	Figura 5.6 Diagrama comutativo.....	137
	Figura 5.7 Diagrama comutativo para soma amalgamada.....	138
	Figura 5.8 Diagrama comutativo.....	138
	Figura 5.9 Diagrama comutativo.....	138
	Figura 5.10 Cone.....	139
	Figura 5.11 Diagrama comutativo para morfismos de cones.....	139
	Figura 5.12 Limite	140
	Figura 6.1 Adjunção.....	141
	Figura 7.1 Diagrama comutativo para as categorias das setas.....	142
	Figura 7.2 Categoria das setas.....	142
	Figura 7.3 Funtor produto fibrado.....	143
	Figura 7.4 Adjunto esquerdo do funtor produto fibrado.....	143
	Figura 8.1 Morfismo cartesiano	144
	Figura 8.2 Funtor induzido por uma fibração	144
	Figura 8.3 Diagrama comutativo para funtor induzido por uma fibração.....	145
	Figura 8.4 Diagrama comutativo para funtor induzido por uma cofibração.....	145
	Figura 9.1 Fonte inicial.....	146
	Figura 10.1 Diagramas comutativos para monóides	147
	Figura 10.2 Diagramas comutativos para mônadas.....	148
	Figura 10.3 Construção da multiplicação de uma mônadas	148
	Figura 10.4 Composição de morfismos em uma categoria Kleisli	148

1 Introdução

1.1 Contexto e Objetivos

Esta dissertação prossegue as investigações dos fundamentos matemáticos da orientação aos objetos concorrentes e originou-se da necessidade de mostrar que o problema da reificação de sistemas reativos, concorrentes e comunicantes tem solução.

Trabalhos como [Sernadas & Ehrich 90], [Ehrich & Sernadas 90], [Ehrich *et al* 91], [Costa 91], [Costa *et al* 92], [Costa & Sernadas 92], [Costa *et al* 93] e [Costa *et al* 94], entre outros, fornecem meios para um adequado entendimento do que é um objeto e de como podem ser interconectados de forma a construir objetos mais complexos. Entretanto, em relação à reificação, os resultados atingidos não foram plenamente satisfatórios. Assim, esta dissertação foi proposta com os seguintes objetivos:

- construir um domínio semântico para sistemas distribuídos reativos e comunicantes o qual deve satisfazer a uma propriedade especial em relação à reificação denominada *composicionalidade diagonal*;
- comparar a expressividade do domínio semântico proposto com outros domínios conhecidos e representativos;
- aplicar o domínio semântico proposto a uma linguagem orientada aos objetos.

Em um sistema dito reativo, o comportamento é determinado por um relacionamento de estímulos e respostas e não em termos de uma função com entradas e saídas. Por composicionalidade diagonal entende-se ambas as composicionalidades (ou modularidades) vertical e horizontal, como segue ([Gorrieri 90]):

- Composicionalidade vertical: é uma metodologia de especificação hierárquica a qual permite adicionar estruturas a um sistema concorrente, em diferentes níveis de abstração. Seria a generalização da abordagem "top-down" de sistemas seqüenciais para sistemas concorrentes. Neste caso, um sistema (possivelmente complexo) pode ser definido em um nível mais alto,

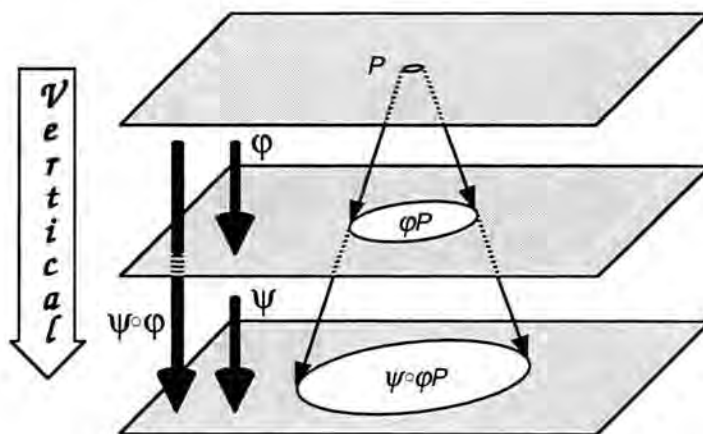


Figura 1.1 Composicionalidade vertical

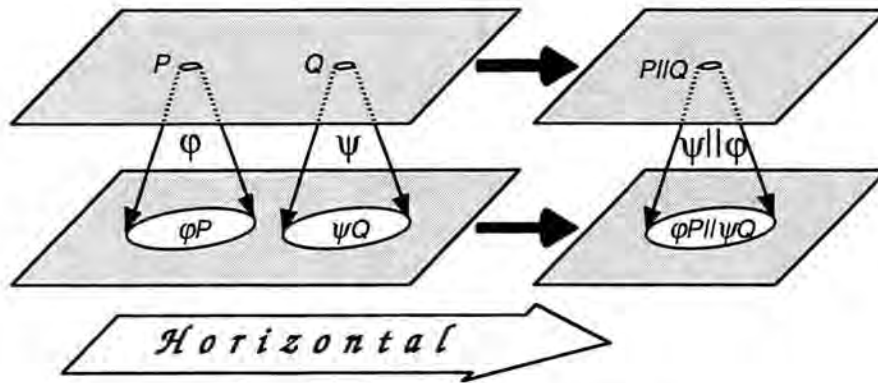


Figura 1.2 Composicionalidade horizontal

abstraído detalhes que são posteriormente especificados em níveis subsequentes, como entidades mais complexas. Adicionalmente, a abstração entre os diversos níveis deve ser composicional, como ilustrado na Figura 1.1.

- b) Composicionalidade horizontal: um sistema complexo é uma entidade estruturada a qual pode ser melhor entendido e desenvolvido se for possível considerar as suas diversas partes componentes separadamente. Assim, o comportamento conjunto da reificação das partes deve ser equivalente à reificação do sistema como um todo, como ilustrado na Figura 1.2, onde o símbolo \parallel representa uma composição de sistemas.

Deve-se destacar que, do que se tem conhecimento, modelo de concorrência algum satisfaz à composicionalidade diagonal, incluindo, por exemplo, traços ("traces") de Mazurkiewicz [Mazurkiewicz 88], estruturas de eventos ("event structures") [Winskel 87b], sistemas de transições assíncronos ("asynchronous transition systems") [Bednarczyk 88] e redes de Petri [Reisig 85].

1.2 Abordagem e Estruturação

Nos trabalhos referenciados no início desta introdução bem como em outros, houve um gradativo consenso de que um objeto é, basicamente, um processo juntamente com atributos dependentes de estados. Considerando-se os desenvolvimentos recentes relacionados às redes com destaque para [Winskel 87], [Meseguer & Montanari 90] e [Sassone *et al* 93], conclui-se que modelos baseados em redes seriam bons candidatos para o domínio semântico a ser proposto. Entretanto, a maioria destes modelos como redes de Petri no sentido de [Reisig 85] e sistemas de transições etiquetadas ("labeled transition systems") em [Milner 89] não possuem mecanismos de composição e de abstração em suas definições originais. A ausência destes mecanismos é uma forte justificativa para o uso de teoria das categorias, considerando-se que:

- em [Winskel 87] é proposto que as construções categorias de produto e coproduto sejam vistas como as operações de composição paralela e de escolha, respectivamente;
- em [Meseguer & Montanari 90] é proposto que um morfismo especial de redes represente a implementação de uma rede em computações de uma outra;

Adicionalmente, a teoria das categorias provê técnicas para comparar a expressividade das diferentes categorias de modelos (classes de modelos categorialmente estruturados) através de adjunções (geralmente reflexões e co-reflexões) expressando a relação entre as suas semânticas, como em [Sassone *et al* 93] onde é proposta uma classificação formal para modelos de concorrência.

Assim, para atingir os objetivos propostos, um novo modelo semântico categorial denominado *autômato não-sequencial* é introduzido. Resumidamente, a abordagem proposta e sua estruturação é como segue.

Capítulos 2 a 4 - Autômatos Seqüenciais

Inicialmente são introduzidos diversos conceitos relacionados com grafos, grafos reflexivos e categorias, com destaque para a noção de nodos distinguidos, inspirados em [Menezes 95], onde são apresentadas formas gerais de tratar nodos com significados especiais. Adicionalmente, são provados resultados gerais para que as categorias de grafos e de categorias resultantes sejam completas e cocompletas.

Um autômato seqüencial é, basicamente, um sistema de transições etiquetado. É definido como um grafo reflexivo com nodos distinguidos e arcos etiquetados. Os nodos distinguidos são interpretados como estados iniciais e finais e os arcos etiquetados como transições. A categoria resultante é completa e cocompleta, sendo o produto interpretado como a composição paralela.

As operações de sincronização e de encapsulação de autômatos seqüenciais são categorialmente explicadas usando as técnicas de fibração e de cofibração, respectivamente, inspiradas em [Winskel 87] e baseadas em [Menezes & Costa 93] e [Menezes *et al* 95].

A sincronização determina o comportamento conjunto de autômatos seqüenciais. É uma operação funtorial e consiste, basicamente, de uma restrição da composição paralela, induzida por um tabela de sincronizações ao nível das etiquetas dos autômatos componentes. Uma construção categorial da tabela de sincronizações também é introduzida, baseada em [Menezes *et al* 95b], generalizando as idéias propostas em [Menezes & Costa 93].

A encapsulação extrai uma visão de um autômato. É definida como uma operação funtorial e consiste, basicamente, de uma reetiquetagem das transições usando um símbolo especial para indicar as transições que não podem ser acessadas externamente e, portanto, que não podem ser sincronizadas. Note-se que, a encapsulação de um autômato introduz um não-determinismo interno.

Uma reificação de um autômato seqüencial é definida como um morfismo que implementa um autômato nas computações de outro, inspirado em [Meseguer & Montanari 90] e [Menezes & Costa 95]. As computações de um autômato são facilmente obtidas em um contexto categorial, considerando que um autômato é, basicamente, um grafo reflexivo que, quando enriquecido com uma operação de composição nas transições, resulta em uma categoria. Assim, cada composição de transições representa uma possível computação ou transação do autômato. A construção é, resumidamente, a seguinte a qual é ilustrada na Figura 1.3:

- considere um funtor que enriquece os autômatos seqüenciais com uma operação de composição nas transições, originando uma categoria;
- considere um funtor que esquece sobre a operação de composição de categorias;
- a composição dos dois funtores determina um endofuntor denominado fecho computacional o qual enriquece um autômato com todas as suas computações (seqüenciais) possíveis;
- um morfismo cujo autômato destino é enriquecido com o seu fecho computacional é um morfismo de reificação.

Destaque-se que, pela técnica de fibração usada, uma restrição do fecho computacional de um autômato pode induzir um morfismo de reificação. Na realidade, morfismos de reificação

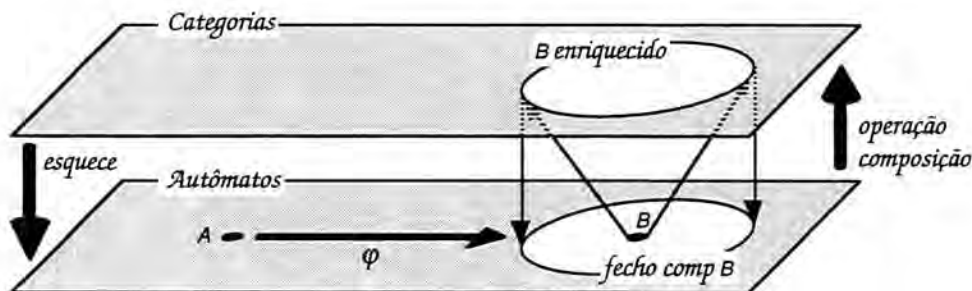


Figura 1.3 Morfismo de reificação de autômatos

induzidos por restrições seguidos de reetiquetagens são usados para dar semântica à linguagem Náutilus, introduzida adiante.

Para atingir a composicionalidade vertical, o problema é compor morfismos onde o objeto destino é enriquecido por um functor. A solução é inspirada nas categorias Kleisli. Por exemplo, para as reificações $\varphi: A_1 \rightarrow \text{fecho}A_2$ e $\psi: A_2 \rightarrow \text{fecho}A_3$ onde *fecho* denota o functor fecho computacional, a composição $\psi \circ \chi \varphi: A_1 \rightarrow \text{fecho}A_3$ é como na Figura 1.4. Note-se que todo o morfismo ψ é enriquecido com o seu fecho computacional. O achatamento é induzido pela adjunção e consiste, basicamente, na associação de computações de computações a computações.

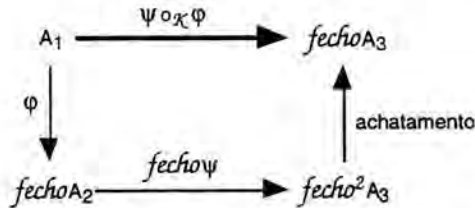


Figura 1.4 Composição de morfismos de reificação

Embora a composicionalidade vertical seja atingida, a construção não satisfaz à composicionalidade horizontal pois é fácil verificar que a composição paralela de reificações não é, em geral, uma reificação.

Capítulos 5 a 7 - Autômatos Não-Seqüenciais

Os diversos conceitos relacionados aos grafos (reflexivos) e categorias são generalizados para grafos (reflexivos) e categorias internos onde arcos, nodos e morfismos podem ser objetos e morfismos de uma categoria base qualquer (e, portanto, não são necessariamente conjuntos e funções totais). Adicionalmente, são provados os seguintes resultados gerais referentes aos limites e colimites (e estendidos para os nodos distinguidos) e, ao que se sabe, são originais:

- condições para que a completude e cocompletude de uma categoria de grafos internos seja herdada da categoria base, ou seja, condições para que, se a categoria base for completa ou cocompleta, a categoria dos grafos internos também o seja;
- condições para que a completude de uma categoria de categorias internas seja herdada da categoria base;

Um autômato não-seqüencial é, basicamente, um grafos reflexivo interno aos monóides comutativos com etiquetagem nos arcos. Portanto, é definido sobre um grafo cujos arcos e nodos são elementos de um monóide, inspirado em [Meseguer & Montanari 90]. Os arcos etiquetados e os nodos são denominados transições e estados, respectivamente, e o significado da estruturação monoidal é como segue:

- estado estruturado: é interpretado como uma coleção de estados locais representando recursos a serem consumidos ou produzidos, análogo aos "tokens" de uma rede de Petri;
- transição estruturada: é uma forma de explicitar que as transições componentes são independentes, ou seja, uma transição estruturada específica que as transições componentes são concorrentes entre si, analogamente à relação de independência dos sistemas de transições assíncronos ou dos traços de Mazurkiewicz.

Portanto, trata-se de um modelo do tipo não-intercalação ("noninterleaving"). Os Autômatos não-seqüenciais foram originalmente apresentados em [Menezes & Costa 95], [Menezes & Costa 95b] e [Menezes *et al* 95].

A categoria de autômatos não-seqüenciais resultante é completa e cocompleta com produtos isomorfos aos coprodutos. O produto (ou coproduto) categorial é interpretado como a composição

paralela. As construções referentes à sincronização, encapsulação e reificação dos autômatos seqüenciais são estendidas para os não-seqüenciais.

Neste contexto, a *composicionalidade horizontal é satisfeita*, ou seja, para reificações $\varphi: N_1 \rightarrow \text{fecho}M_1$ e $\psi: N_2 \rightarrow \text{fecho}M_2$, tem-se que:

$$\varphi N_1 \parallel \psi N_2 = \varphi \parallel \psi(N_1 \parallel N_2)$$

onde \parallel representa a composição paralela ou uma sincronização de autômatos não-seqüenciais e onde a reificação $\varphi \parallel \psi$ é induzida pelas reificações componentes. Na realidade, do que se tem conhecimento, os autômatos não seqüenciais constituem o primeiro modelo de concorrência a satisfazer a composicionalidade diagonal.

Adicionalmente, as operações de sincronização e encapsulação são estendidas para os morfismos de reificação, conforme proposto em [Menezes *et al* 95b]. Destaque-se que, para a abordagem proposta, a composicionalidade diagonal é fundamental para que a sincronização de reificações seja bem definida.

Capítulos 8 e 9 - Redes de Petri e o Relacionamento Entre os Autômatos e as Redes

Os grafos estruturados generalizam os grafos internos onde os arcos e os nodos podem pertencer a categorias distintas desde que sejam considerados funtores das categorias dos nodos e dos arcos para uma categoria base. Adicionalmente, são provados resultados gerais (e estendidos para os nodos distinguidos) para que as propriedades referentes aos limites e colimites sejam herdadas das categorias dos nodos e dos arcos. Estes resultados, ao que se sabe, são originais.

Assim, uma rede de Petri é uma grafo estruturado onde os nodos são elementos de um monóide comutativo e os arcos pertencem a um conjunto (apontado). Com o objetivo de facilitar a comparação com os autômatos, as categorias de redes de Petri são estendidas com etiquetagem nas transições e com estados distinguidos. As categorias de redes resultantes são completas e cocompletas, o que é um resultado expressivo quando comparado com [Winskel 87] e [Meseguer & Montanari 90] cujas categorias com marcação inicial possuem restrições para garantir a existência de coprodutos. Neste contexto, o produto de redes é interpretado como a composição paralela e, analogamente aos autômatos, a sincronização e a encapsulação são operações functoriais.

O relacionamento entre autômatos seqüenciais, não-seqüenciais e redes de Petri é realizado usando funtores e são baseadas em [Menezes & Costa 95] e [Menezes & Costa 95b], estendendo a abordagem em [Sassone *et al* 93]. Assim, tem-se que:

- dos passos que envolvem a passagem de um modelo para outro, pode-se inferir que os autômatos não-seqüenciais são mais concretos do que as redes de Petri e, como esperado, do que os autômatos seqüenciais. Comparativamente com as redes, os autômatos não-seqüenciais são capazes de expressar informações sobre alguns objetos e relacionamentos entre os objetos que são, de certa forma, perdidas quando da passagem para as redes. De fato, prova-se que a categoria das redes de Petri é isomorfa a uma subcategoria dos autômatos não-seqüenciais;
- compondo-se a adjunção das redes para os autômatos não-seqüenciais com a adjunção usada para a reificação de autômatos, define-se a reificação de redes de Petri. É fácil verificar que as redes *não* satisfazem à composicionalidade horizontal.

Portanto, entre os modelos relacionados, os autômatos não-seqüenciais constituem o modelo menos concreto que satisfaz à composicionalidade horizontal e, conseqüentemente, à composicionalidade diagonal. Este é um resultado importante, principalmente em relação às redes de Petri, considerado um dos modelos mais gerais de concorrência.

Capítulos 10 a 11 - Linguagem Náutilus e sua Semântica.

Uma linguagem de especificação baseada nos objetos, originalmente introduzida em [Menezes *et al* 95b], denominada Náutilus, é apresentada. Náutilus é baseada na linguagem orientada aos

objetos GNOME [Sernadas & Ramos 94] e [Ramos & Sernadas 95] que, por sua vez, é uma simplificação e revisão da linguagem OBLOG [SernadasC *et al* 92], [SernadasC *et al* 92b], [Sernadas *et al* 92] e [SernadasC *et al* 91]. Trata-se de uma linguagem textual, de alto nível, com facilidades de abstração, a qual objetiva a especificação de sistemas concorrentes e comunicantes. As principais diferenças em relação ao GNOME referem-se a simplificações como o fato de não suportar classes e herança e a extensões inspiradas no domínio semântico proposto como reificação e agregação. As principais características da linguagem Náutilus são as seguintes:

- objetos podem interagir através de chamadas;
- objetos podem ser agregados de forma a compor um objeto mais complexo;
- um objeto pode ser reificado em computações seqüenciais ou paralelas de outro objeto;
- pode-se extrair uma visão de um objeto;
- interações, agregações e reificações podem ser condicionadas a diversas alternativas, ou seja, podem ser dependentes de estados;
- as cláusulas de uma ação em um objeto podem ser compostas de forma seqüencial ou múltipla.

A principal diferença entre uma interação e uma agregação é que, no primeiro, o relacionamento entre os objetos é especificado dentro de cada objeto componente enquanto que, no segundo, o relacionamento é especificado à parte, em um objeto agregador. Uma composição múltipla é uma composição especial de cláusulas concorrentes a qual pode ser vista como uma generalização da noção de múltipla atribuição.

A semântica de um objeto em Náutilus, introduzida em [Menezes *et al* 95b], é dada por um morfismo de reificação de autômatos não-seqüenciais onde, a cada ação do objeto corresponde uma transição implementada como uma computação (indivisível, finita e determinista). A principal justificativa para o uso de um morfismo de reificação é o fato das cláusulas de uma ação de um objeto poderem ser compostas de forma seqüencial, a qual pode ser vista como uma reificação interna. Adicionalmente, a semântica de objetos resultantes de construções sobre outros objetos como interação, agregação, encapsulação ou reificação é a aplicação das correspondentes construções sobre autômatos não-seqüenciais reificados. Trata-se de uma semântica operacional, mas que pode ser considerada como uma semântica denotacional funtorial até ao nível de objeto. Resumidamente, a semântica proposta é como segue:

- objeto cuja especificação não depende de outros objetos: é um morfismo de reificação construído como uma implementação de um autômato não-seqüencial sobre outro, denominado base. O autômato base reflete todas as valorações possíveis sobre todos os atributos componentes do objeto;
- reificação de um objeto origem sobre um objeto destino: é a composição das semânticas dos objetos, ou seja, das reificações do origem sobre o destino e do destino sobre a sua base.
- interação ou agregação de objetos: é uma sincronização das semânticas dos objetos componentes, ou seja, dos correspondentes autômatos não-seqüenciais reificados;
- visão de um objeto: é uma encapsulação da semântica do objeto, ou seja, do correspondente autômato não-seqüencial reificado;

Como a interação e a agregação são relacionadas com a sincronização de reificações a qual, como foi dito acima, depende da composicionalidade horizontal para que seja bem definida, tem-se, como consequência, que as redes de Petri e os autômatos seqüenciais não são domínios semânticos adequados para a abordagem proposta.

Apêndices

Conceitos básicos de Teoria das Categorias e Semântica Formal são introduzidos em apêndices. Os conceitos introduzidos são baseados em bibliografias amplamente aceitas e objetivam permitir uma rápida referência aos conceitos, principais resultados e nomenclaturas (em português) usados ao longo da dissertação.

1.3 Sumário das Contribuições

Um sumário das principais contribuições é como segue:

- um novo domínio semântico é introduzido, denominado autômatos não-seqüenciais, do tipo não-intercalação, para sistemas reativos, concorrentes e comunicantes e que satisfaz à composicionalidade diagonal, ou seja, onde as reificações são composicionais e distribuem-se sobre a composição paralela. Do que se tem conhecimento, é o primeiro modelo de concorrência a satisfazer a composicionalidade diagonal;
- os autômatos não-seqüenciais são enquadrados em relação às redes de Petri e aos autômatos seqüenciais, estendendo a classificação formal proposta em [Sassone *et al* 93] e mostrando que os autômatos não-seqüenciais constituem o modelo menos concreto que satisfaz à composicionalidade diagonal;
- uma linguagem de especificação baseada nos objetos denominada Náutilus é introduzida, a qual é uma revisão da linguagem orientada aos objetos GNOME, simplificando alguns aspectos e introduzindo outros, inspirados no domínio semântico proposto, como a reificação e a agregação.

As seguintes contribuições também devem ser destacadas:

- apresentação dos autômatos seqüenciais e das redes de Petri equipados com operações de sincronização e encapsulação, bem como reificações verticalmente composicionais. As categorias de autômatos e redes são bicompletas, com destaque para a cocompletude das redes com marcações, principalmente quando comparadas com [Winskel 87] e [Meseguer & Montanari 90];
- uma construção categorial das tabelas de sincronizações para chamada, compartilhamento ou ambos os casos;
- condições gerais para a bicompletude dos grafos (reflexivos) internos e estruturados, bem como para a completude das categorias internas.

2 Grafos e Categorias com Elementos Distinguidos

Neste capítulo, são introduzidos os conceitos de diversos tipos de grafos e as correspondentes categorias. Excetuando-se os grafos com nodos distinguidos, todos os demais são os tradicionais. Os grafos com nodos distinguidos são inspirados em [Menezes 95] e são propostos como uma forma de tratar estruturas baseadas em grafos onde alguns nodos possuem significado especial. Os tipos de grafos introduzidos são os seguintes:

- a) grafo (direto): é constituído de nodos e arcos. A cada arco é associado um nodo origem e um nodo destino;
- b) grafo reflexivo: é um grafo onde cada nodo possui um arco identidade, ou seja, um arco distinguido com origem e destino no mesmo nodo;
- c) grafo reflexivo de um só nodo: é um grafo reflexivo cujo conjunto de nodos é um conjunto fixo de um único elemento;
- d) grafo com nodos distinguidos: são grafos (reflexivos ou não) que possuem alguns nodos especialmente identificados. A interpretação de um nodo distinguido depende da aplicação. No caso de autómatos sequenciais, são interpretados como estados iniciais ou finais.

Considerando que categorias são grafos reflexivos com uma operação de composição nos arcos, o conceito de grafos reflexivos com nodos distinguidos é facilmente estendido para categorias.

No texto que segue, são usados os conceitos e resultados referentes à categoria das setas ("comma category") e ao funtor diagonal, sendo ambos introduzidos em apêndice. Para a categoria das setas, as principais referências são [Casley 91] e [Asperti & Longo 91] e, para o funtor diagonal, [Mac Lane 71] e [Barr & Wells 90].

2.1 Grafo

Em um grafo, cada arco possui exatamente um nodo origem e um destino. Entretanto, um nodo pode ser origem ou destino de zero ou mais arcos. Tradicionalmente, um grafo é definido como uma quádrupla $\langle V, T, \partial_0, \partial_1 \rangle$ onde V é um conjunto de nodos, T é um conjunto de arcos e $\partial_0, \partial_1: T \rightarrow V$ são funções totais denominadas origem e destino, respectivamente, as quais associam a cada arco os correspondentes nodos de origem e destino. A seguir, é introduzida uma abordagem equivalente, a qual consiste em considerar um grafo como um elemento de uma categoria das setas. A categoria resultante é completa e cocompleta. No texto que segue, Set denota a categoria dos conjuntos e funções.

Definição 2.1 Categoria dos Grafos. Considere o funtor diagonal $\Delta: Set \rightarrow Set^2$. A categoria dos grafos, denotada por \mathcal{Gr} , é a categoria das setas $\Delta \downarrow \Delta$. \square

Portanto, um grafo é uma tripla $G = \langle T, \partial, V \rangle$ onde $\partial = \langle \partial_0: T \rightarrow V, \partial_1: T \rightarrow V \rangle$. A notação normalmente usada é a tradicional, ou seja, $G = \langle V, T, \partial_0, \partial_1 \rangle$.

Em um grafo, um arco cujos nodos de origem e destino coincidem é denominado ciclo (local) ou doarco e dois ou mais arcos com os mesmos nodos origem e destino são denominados paralelos. Um nodo que não é origem nem destino de qualquer arco é denominado nodo isolado. Um arco t tal que $\partial_0(t) = X, \partial_1(t) = Y$ é denotado por $t: X \rightarrow Y$. A representação gráfica de grafos adotada é análoga à normalmente usada para as redes de Petri (como em [Reisig 85]) onde os nodos são representados por círculos e os arcos por quadrados. Adicionalmente, nodos sem arcos associados (se existir) são omitidos, como ilustrado na Figura 2.1.

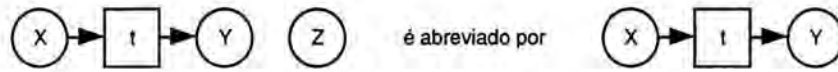


Figura 2.1 Nós isolados são omitidos na representação gráfica de grafos

Como esperado, um morfismo em \mathcal{Gr} preserva os nós origem e destino de cada arco. Assim, para os grafos $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$, um morfismo de grafos $h: G_1 \rightarrow G_2$ é um par de funções totais $\langle h_V: V_1 \rightarrow V_2, h_T: T_1 \rightarrow T_2 \rangle$ tal que o diagrama ilustrado na Figura 2.2 comuta.

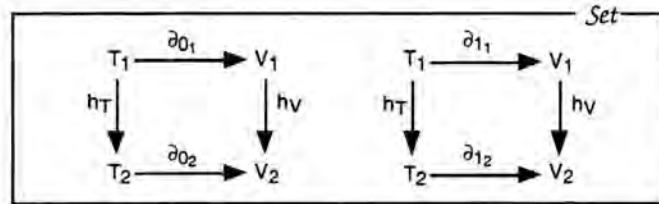


Figura 2.2 Diagrama comutativo para morfismos de grafos

Proposição 2.2 A categoria \mathcal{Gr} é completa e cocompleta.

Prova: Como \mathcal{Gr} é a categoria das setas $\Delta \downarrow \Delta$ e como Δ preserva limites e colimites (pois possui adjunto esquerdo e direito - veja, por exemplo, [Mac Lane 71, p.85]), \mathcal{Gr} é bicompleta. \square

Os grafos resultantes do produto e do coproduto de $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$ são como segue:

$$G_1 \times G_2 = \langle V_1 \times V_2, T_1 \times T_2, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12} \rangle$$

$$G_1 + G_2 = \langle V_1 + V_2, T_1 + T_2, \partial_{01} + \partial_{02}, \partial_{11} + \partial_{12} \rangle$$

onde $\partial_{01} \times \partial_{02}$, $\partial_{11} \times \partial_{12}$, $\partial_{01} + \partial_{02}$, $\partial_{11} + \partial_{12}$ são os morfismos univocamente induzidos pelo produto e coproduto em Set . Sempre que possível, o nome da categoria é omitido nos símbolos de produto e coproduto. Por exemplo, $G_1 \times G_2$ e $V_1 + V_2$ acima representam $G_1 \times_{\mathcal{Gr}} G_2$ e $V_1 +_{Set} V_2$, respectivamente. A seguinte notação é usada para pares de nós e arcos, a qual é conveniente no contexto dos autômatos:

- $X \oplus Y$ para o nó (X, Y) ;
- $t_1 | t_2$ para o arco (t_1, t_2) .

Em \mathcal{Gr} , a representação gráfica de um objeto resultante de um coproduto é um grafo distribuído representando a justaposição, lado a lado, dos grafos componentes e a representação de um objeto resultante de um produto é um grafo cujos arcos e nós são todos os pares de arcos e nós dos grafos componentes.

Exemplo 2.3 Considere a Figura 2.3. Os grafos $G_1 + G_2$ e $G_1 \times G_2$ representam os objetos resultante

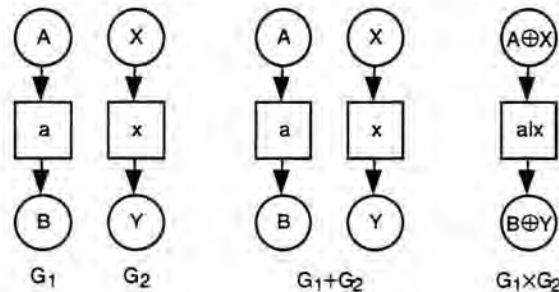


Figura 2.3 Coproduto e Produto de grafos

do coproduto e do produto de G_1 e G_2 em \mathcal{Gr} , respectivamente. □

2.2 Grafo Reflexivo

A categoria dos grafos reflexivos é completa e cocompleta. A categoria dos grafos reflexivos de um só nodo é uma subcategoria plena dos grafos reflexivos e é isomorfa à categoria dos conjuntos apontados e, portanto, também é completa e cocompleta. No texto que segue, para um objeto V , $id_V: V \rightarrow V$ denota o morfismo identidade em V .

Grafo Reflexivo

Definição 2.4 Categoria dos Grafos Reflexivos.

- a) Um grafo reflexivo G é uma quintupla $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ onde $\langle V, T, \partial_0, \partial_1 \rangle$ é um grafo e $\iota: V \rightarrow T$ é uma função total denominada (arco) identidade, a qual associa a cada nodo o correspondente endoarco identidade e é tal que o diagrama ilustrado na Figura 2.4 comuta.
- b) Sejam $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11}, \iota_1 \rangle$, $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12}, \iota_2 \rangle$ grafos reflexivos. Um morfismo de grafos reflexivos $h: G_1 \rightarrow G_2$ é um morfismo de grafos $\langle h_V: V_1 \rightarrow V_2, h_T: T_1 \rightarrow T_2 \rangle$ tal que o diagrama ilustrado na Figura 2.5 comuta.
- c) Grafos reflexivos e os correspondentes morfismos constituem a categoria \mathcal{RGr} . □

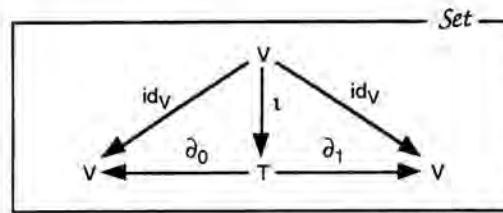


Figura 2.4 Diagrama comutativo para a função identidade

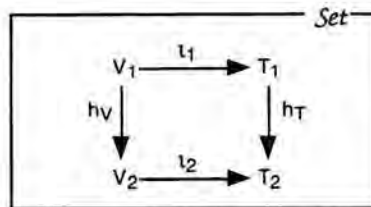


Figura 2.5 Diagrama comutativo adicional para morfismos de grafos reflexivos

Para um nodo X o correspondente arco identidade é denotado por $\iota_X: X \rightarrow X$ ou $X: X \rightarrow X$. Na representação gráfica de grafos reflexivos, os arcos identidade são, em geral, omitidos, como ilustrado na Figura 2.6.

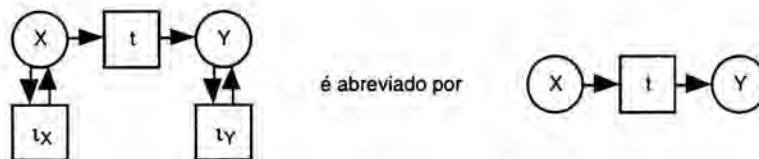


Figura 2.6 Arcos identidade são omitidos na representação gráfica de grafos reflexivos

Exemplo 2.5 Seja $G = \langle \{A, B, C, D\}, \{a, b, c, d, \iota_A, \iota_B, \iota_C, \iota_D\}, \partial_0, \partial_1, \iota \rangle$ um grafo reflexivo onde ∂_0, ∂_1 são tais que $a: A \rightarrow B, b: B \rightarrow C, c: C \rightarrow B, d: B \rightarrow D$. A representação gráfica de G é ilustrada na Figura 2.7. \square

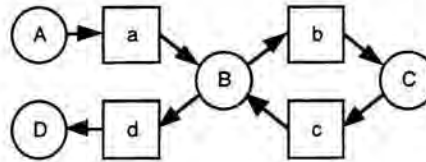


Figura 2.7 Representação gráfica de um grafo reflexivo

Proposição 2.6 A categoria \mathcal{RGr} é completa e cocompleta.

Prova: \mathcal{RGr} é a categoria das precategorias como definida em [Bednarczyk 88, p. 22] (não confundir com a categoria das precategorias como definida em [Mac Lane 71, p. 49]), o qual prova que é uma categoria bicompleta. \square

Os grafos resultantes do produto e do coproduto de $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11}, \iota_1 \rangle$ e $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12}, \iota_2 \rangle$ são como segue:

$$G_1 \times G_2 = \langle V_1 \times V_2, T_1 \times T_2, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12}, \iota_1 \times \iota_2 \rangle$$

$$G_1 + G_2 = \langle V_1 + V_2, T_1 + T_2, \partial_{01} + \partial_{02}, \partial_{11} + \partial_{12}, \iota_1 + \iota_2 \rangle$$

onde $\partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12}, \iota_1 \times \iota_2, \partial_{01} + \partial_{02}, \partial_{11} + \partial_{12}, \iota_1 + \iota_2$ são morfismos univocamente induzidos pelo produto e coproduto em Set . Por simplicidade, quando o contexto não deixar dúvidas, os pares de arcos da forma $t | \iota_X$ ou $\iota_X | t$ são denotados simplesmente por t .

Em \mathcal{RGr} , a representação gráfica de um objeto resultante do coproduto é um grafo distribuído representando a justaposição, lado a lado, dos grafos componentes e o produto resulta em um grafo com todas as combinações entre os arcos e os nodos dos grafos componentes.

Exemplo 2.7 Considere a Figura 2.8. Os grafos $G_1 + G_2$ e $G_1 \times G_2$ representam os objetos resultante do coproduto e do produto de G_1 e G_2 em \mathcal{RGr} , respectivamente (compare com o correspondente exemplo em \mathcal{Gr}). \square

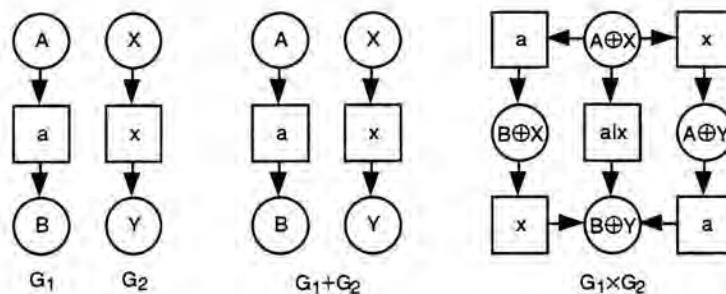


Figura 2.8 Coproduto e produto de grafos reflexivos

Grafo Reflexivo de um só Nódo

No texto que segue, Set^* denota a categoria dos conjuntos apontados e os correspondentes morfismos e, para uma categoria $C, !: X \rightarrow 1$ denota o único morfismo de um objeto X para um objeto terminal 1 . Lembre-se que um conjunto com um único elemento é objeto terminal em Set .

Definição 2.8 Categoria dos Grafos Reflexivos de um só Nódo.

- a) Um grafo reflexivo de um só nódo E é um grafo reflexivo $E = \langle 1, T, !, \iota, \iota \rangle$ tal que 1 é um conjunto fixo de um único nódo.

- b) Sejam $E_1 = \langle 1, T_1, !, !, \iota \rangle$ e $E_2 = \langle 1, T_2, !, !, \iota \rangle$ grafos reflexivos de um só nodo. Um morfismo de grafos reflexivos de um só nodo é um morfismo de grafos reflexivos $h = \langle !, h_T \rangle: E_1 \rightarrow E_2$.
- c) Grafos de um só nodo e os correspondentes morfismos constituem a categoria \mathcal{RGr}_1 . \square

Em um grafo reflexivo de um só nodo, o arco identidade do único nodo determinado pela função total $\iota: 1 \rightarrow T$ é denotado por τ .

Exemplo 2.9 A representação gráfica do grafo reflexivo de um só nodo $E = \langle 1, \{\tau, \text{inic}, \text{fim}, \text{flip}, \text{flop}\}, !, !, \iota \rangle$ é ilustrada na Figura 2.9. \square

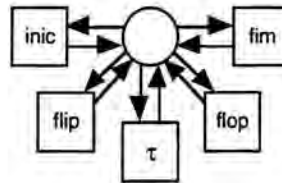


Figura 2.9 Representação gráfica de um grafo reflexivo de um só nodo

A categoria \mathcal{RGr}_1 a qual é uma subcategoria plena de \mathcal{RGr} e é isomorfa a Set^* , como demonstrado na proposição seguir.

Proposição 2.10 \mathcal{RGr}_1 é completa e cocompleta.

Prova: Considere:

- o funtor esquecimento $s: \mathcal{RGr}_1 \rightarrow Set^*$ tal que, para todo o \mathcal{RGr}_1 -objeto $E = \langle 1, T, !, !, \iota \rangle$, $sE = T$ onde arco identidade τ é associado ao elemento distinguido de T e, para todo o \mathcal{RGr}_1 -morfismo $h = \langle !: 1 \rightarrow 1, h_T: T_1 \rightarrow T_2 \rangle$, $sh = h_T$. Como h é um morfismo de grafos reflexivos, h_T preserva τ ;
- o funtor $g: Set^* \rightarrow \mathcal{RGr}_1$ tal que, para todo o conjunto apontado T , $gT = \langle 1, T, !, !, \iota \rangle$ onde o arco identidade do único nodo determinado pela função ι é o elemento distinguido de T e, para todo o Set^* -morfismo $h_T: T_1 \rightarrow T_2$, $gh_T = \langle !, h_T \rangle: \langle 1, T_1, !, !, \iota \rangle \rightarrow \langle 1, T_2, !, !, \iota \rangle$.

É fácil verificar que $g \circ s = id_{\mathcal{RGr}_1}$ e $s \circ g = id_{Set^*}$. Como Set^* é bicompleta, então \mathcal{RGr}_1 é bicompleta. \square

Considerando que as categorias \mathcal{RGr}_1 e Set^* são isomorfas, um grafo reflexivo de um só nodo $\langle 1, T, !, !, \iota \rangle$ pode ser denotado pelo seu correspondente conjunto apontado de arcos T , ou seja, $T = \langle 1, T, !, !, \iota \rangle$. O mesmo raciocínio é válido para um \mathcal{RGr}_1 -morfismo $\langle !, h_T \rangle$, ou seja, $h_T = \langle !, h_T \rangle$. Por simplicidade, quando o contexto não deixar dúvidas, a seguinte notação é adotada para pares de arcos:

$$\begin{aligned} t &\text{ para } \tau | t \text{ ou } t | \tau; \\ \tau &\text{ para } \tau | \tau. \end{aligned}$$

Exemplo 2.11 Sejam $T_1 = \{\tau, a, b\}$ e $T_2 = \{\tau, x\}$ grafos reflexivos de um só nodo. Então, $T_1 + T_2 = \{\tau, a, b, x\}$ e $T_1 \times T_2 = \{\tau, a, b, x, a|x, b|x\}$. \square

2.3 Grafo Com Nodos Distinguidos

Em um grafo, nodos distinguidos são nodos com um significado especial, cuja interpretação depende do contexto ou da aplicação. Exemplos típicos de nodos distinguidos são nodos iniciais, finais, coloridos, etc, em representações de sistemas na forma de grafos. Nodos distinguidos são definidos de forma análoga para grafos e grafos reflexivos. Assim, no texto que segue, \mathcal{G} representa as categorias \mathcal{Gr} ou \mathcal{RGr} indistintamente.

Definição 2.12 Categoria dos Grafos com Nodos Distinguidos. A categoria dos grafos com nodos distinguidos é a categoria das setas $d \downarrow \text{nodos}$, denotada por \mathcal{G}_d , onde:

- a) $d: \mathcal{D} \rightarrow Set$ é um funtor;
- b) *nodos*: $\mathcal{G} \rightarrow Set$ é o funtor esquecimento que associa a cada grafo G o correspondente conjunto de nodos V e a cada morfismo de grafos $h = \langle h_V, h_T \rangle$ o morfismo h_V . \square

Portanto, um grafo com nodos distinguidos M é uma tripla $M = \langle G, D, d \rangle$ (por conveniência, não é usada a seqüência usual dos componentes determinada pela categoria das setas) onde G é um grafo de \mathcal{G} , D é um objeto da categoria dos elementos distinguidos \mathcal{D} e $d: dD \rightarrow \text{nodos}G$ é uma função total que associa os elementos distinguidos aos nodos de G . Dependendo do significado desejado para os nodos distinguidos, a categoria \mathcal{D} e o funtor d podem ser definidos de diversas formas, como ilustrado no exemplo a seguir.

Exemplo 2.13 Uma aplicação típica dos grafos com nodos distinguidos é a representação de sistemas na forma de diagramas do tipo estado/transição, onde os nodos distinguidos assumem interpretações como estados iniciais, finais ou coloridos. A seguir são exemplificados a categoria \mathcal{D} e o funtor d para alguns casos:

- a) Conjunto de estados iniciais. Sejam $\mathcal{D} = \text{Set}$ e $d = id_{\text{Set}}$. Neste caso, a escolha de qual estado inicial a ser efetivamente considerado em tempo de execução pode ser um não-determinismo externo.
- b) Conjuntos de estados iniciais e finais. Sejam $\mathcal{D} = \text{Set}^2$ e $d = \perp\!\!\!\perp : \text{Set}^2 \rightarrow \text{Set}$ onde $\perp\!\!\!\perp$ é o funtor canonicamente induzido pelo coproduto categorial em Set . Assim, um par $\langle I, F \rangle$ de Set^2 representa os conjuntos dos nodos iniciais e finais e a função total $i+f: I+F \rightarrow \text{nodos}G$, ilustrada na Figura 2.10, associa os elementos distinguidos aos nodos de G . Note que, neste caso, um nodo pode ser simultaneamente inicial e final.
- c) Estados coloridos. Considere a categoria das colorações Col e a categoria das cores $\text{Cor} = \text{Set}^{\text{Col}}$. Sejam $\mathcal{D} = \text{Set}^{\text{Col}}$ e $d: \text{Set}^{\text{Col}} \rightarrow \text{Set}$ onde d é o funtor que reflete a combinação das inclusões das cores.

Note-se que, em qualquer caso, a instanciação $d: dD \rightarrow \text{nodos}G$ não necessariamente é mono. Por exemplo, dois ou mais elementos do conjunto de estados iniciais associados a um mesmo nodo é interpretado, simplesmente, como sendo um nodo inicial (o qual é a instância dos seus correspondentes estados iniciais). □

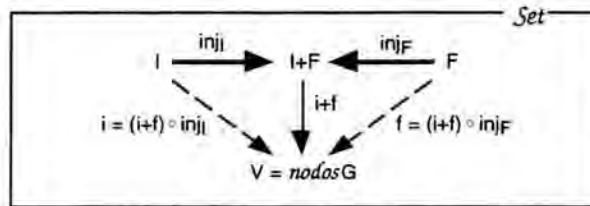


Figura 2.10 Instanciação dos nodos iniciais e finais

A representação de um grafo com nodos distinguidos $M = \langle G, D, d \rangle$ como um diagrama possui a forma de G , onde os nodos distinguidos são representados por círculos com um traço diferenciado. Por exemplo, na Figura 2.11, os nodos X e Y são distinguidos, como no Exemplo 2.13 item b), onde X e Y são elementos de I e F , respectivamente.

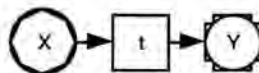


Figura 2.11 Representação gráfica de nodos distinguidos

Um morfismo de grafos com nodos distinguidos é um morfismo na categoria das setas $d \downarrow \text{nodos}$, ou seja, dados os grafos $M_1 = \langle G_1, D_1, d_1 \rangle$ e $M_2 = \langle G_2, D_2, d_2 \rangle$, um morfismo $h: M_1 \rightarrow M_2$ é uma tripla $h = \langle h_V, h_T, h_D \rangle$ onde $\langle h_V, h_T \rangle: G_1 \rightarrow G_2$ é um \mathcal{G} -morfismo e $h_D: D_1 \rightarrow D_2$ é um \mathcal{D} -morfismo, tal que o diagrama ilustrado na Figura 2.12 comuta.

Proposição 2.14 Considere o funtor $d: \mathcal{D} \rightarrow \text{Set}$ e a categoria \mathcal{G}_d . Então:

- a) Se \mathcal{D} é completa, então \mathcal{G}_d é completa.

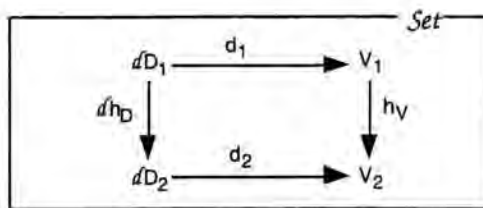


Figura 2.12 Diagrama comutativo adicional para morfismos de grafos com nodos distinguidos

b) Se \mathcal{D} é cocompleta e d preserva colimites, então \mathcal{G}_d é cocompleta.

Prova: Como \mathcal{G}_d é a categoria das setas $d \downarrow \text{nodos}$, é suficiente provar que o funtor $\text{nodos}: \mathcal{G} \rightarrow \text{Set}$ preserva limites. Considere o funtor $g: \text{Set} \rightarrow \mathcal{G}$ tal que, para todo conjunto V , gV é o grafo $\langle V, \{ \}, !, ! \rangle$ (no caso de \mathcal{G} representar a categoria $\mathcal{G}r$) ou $\langle V, V, \text{id}_V, \text{id}_V, \text{id}_V \rangle$ (no caso de \mathcal{G} representar $\mathcal{R}G_r$). É fácil verificar que g é adjunto esquerdo de nodos onde a unidade η é tal que, para todo o conjunto V , $\eta_V = V$. \square

Os grafos resultantes do produto e do coproduto de $M_1 = \langle G_1, D_1, d_1 \rangle$ e $M_2 = \langle G_2, D_2, d_2 \rangle$ em \mathcal{G}_d são como segue:

$$M_1 \times M_2 = \langle G_1 \times_{\mathcal{G}} G_2, D_1 \times_{\mathcal{D}} D_2, d_1 \times d_2 \rangle$$

$$M_1 + M_2 = \langle G_1 +_{\mathcal{G}} G_2, D_1 +_{\mathcal{D}} D_2, d_1 + d_2 \rangle$$

onde $d_1 \times d_2$ e $d_1 + d_2$ são os morfismos univocamente induzidos pelo produto e coproduto em Set , respectivamente.

No contexto dos autômatos seqüenciais, são considerados somente os grafos reflexivos cujos nodos distinguidos são pares de conjuntos os quais são interpretados como conjunto de estados iniciais e finais, como no Exemplo 2.13 item b). Por esta razão, a seguinte notação simplificada é adotada:

Notação 2.15 $\mathcal{DR}G_r$ denota a categoria dos grafos reflexivos cujos nodos distinguidos são elementos de Set^2 , isto é, $\mathcal{DR}G_r = \mathbb{1} \downarrow \text{nodos}$ onde $\mathbb{1}: \text{Set}^2 \rightarrow \text{Set}$ é o funtor canonicamente induzido pelo coproduto categorial em Set e $\text{nodos}: \mathcal{R}G_r \rightarrow \text{Set}$ é o funtor esquecimento que associa a cada grafo o seu correspondente conjunto de nodos. Neste caso, um grafo com nodos distinguidos $M = \langle G, \langle I, F \rangle, i + f \rangle$ onde $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ é denotado por $M = \langle G, I, F, i + f \rangle$ ou $M = \langle V, T, \partial_0, \partial_1, \iota, I, F, i + f \rangle$. \square

Exemplo 2.16 Considere a Figura 2.13. Os grafos $M_1 + M_2$ e $M_1 \times M_2$ representam os objetos resultante do coproduto e do produto de M_1 e M_2 em $\mathcal{DR}G_r$, respectivamente (compare com o correspondente exemplo em $\mathcal{R}G_r$). \square

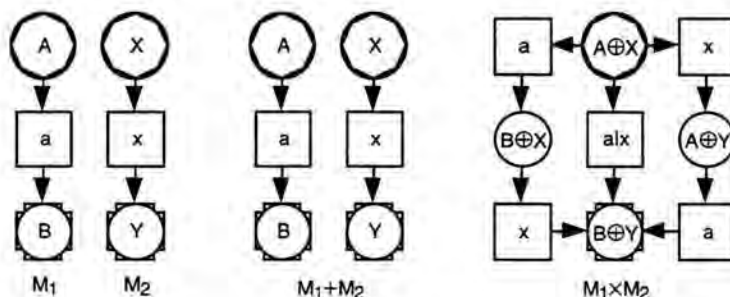


Figura 2.13 Coproduto e produto de grafos reflexivos com nodos distinguidos

2.4 Categoria com Objetos Distinguidos

A definição de objetos distinguidos para categorias é uma simples extensão da definição de nodos distinguidos para grafos.

Definição 2.17 Categoria das Categorias com Objetos Distinguidos. A categoria das categorias (pequenas) com objetos distinguidos é a categoria das setas $d \downarrow ob$, denotada por Cat_d , onde:

- $d: \mathcal{D} \rightarrow Set$ é um funtor;
- $ob: Cat \rightarrow Set$ é o funtor esquecimento que associa a cada categoria C o correspondente conjunto de objetos Ob_C e a cada funtor $h = \langle h_V, h_T \rangle$ a função total h_V . \square

Portanto uma categoria com objetos distinguidos C_d é uma tripla $C_d = \langle C, D, d \rangle$ onde C é uma categoria de Cat , D é um objeto de \mathcal{D} , representando os objetos distinguidos e $d: \mathcal{D} \rightarrow Ob_C$ é uma função total que instancia os objetos distinguidos em C .

Um morfismo de categorias com objetos distinguidos é um morfismo na categoria das setas $d \downarrow ob$, ou seja, dados as categorias $C_{d_1} = \langle C_1, D_1, d_1 \rangle$ e $C_{d_2} = \langle C_2, D_2, d_2 \rangle$, um morfismo $h: C_{d_1} \rightarrow C_{d_2}$ é uma tripla $h = \langle h_V, h_T, h_D \rangle$ onde $\langle h_V, h_T \rangle: C_1 \rightarrow C_2$ é um funtor e $h_D: D_1 \rightarrow D_2$ é um \mathcal{D} -morfismo, tal que o diagrama ilustrado na Figura 2.14 comuta.

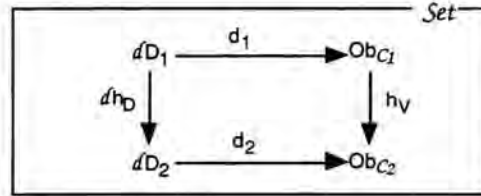


Figura 2.14 Diagrama comutativo adicional para morfismos de categorias com objetos distinguidos

Proposição 2.18 Considere o funtor $d: \mathcal{D} \rightarrow Set$. Então:

- Se \mathcal{D} é completa, então Cat_d é completa.
- Se \mathcal{D} é cocompleta e d preserva colimites, então Cat_d é cocompleta.

Prova: Como Cat_d é a categoria das setas $d \downarrow ob$, é suficiente provar que o funtor $ob: Cat \rightarrow Set$ preserva limites. Considere o funtor $c: Set \rightarrow Cat$ tal que para todo conjunto V , cV é a categoria $\langle V, V, id_V, id_V, id_V, \circ \rangle$ (categoria cujos únicos morfismos são os identidade). É fácil verificar que c é adjunto esquerdo de ob onde a unidade η é tal que, para todo o conjunto V , $\eta_V = V$. \square

No contexto dos autômatos seqüenciais, são consideradas somente as categorias cujos objetos distinguidos são pares de conjuntos. Assim, analogamente aos grafos reflexivos, a seguinte notação simplificada é adotada.

Notação 2.19 $DCat$ denota a categoria das categorias pequenas cujos nodos distinguidos são pares de conjuntos, ou seja, $DCat = \amalg \downarrow ob$ onde $\amalg: Set^2 \rightarrow Set$ é o funtor canonicamente induzido pelo coproduto categorial em Set e $ob: Cat \rightarrow Set$ é o funtor esquecimento que associa a cada categoria o seu correspondente conjunto de objetos. \square

3 Autômatos Seqüenciais

Um autômato seqüencial é uma máquina de estados onde cada transição representa uma ação atômica. Somente uma transição pode ser executada a cada momento, ou seja, não existe concorrência entre ações em um mesmo autômato. Este fato justifica o nome seqüencial. Uma transição só pode ser executada quando o autômato encontra-se no estado de origem da transição e, neste caso, o autômato assume o estado destino.

Um autômato seqüencial é visto como um grafo reflexivo com nodos distinguidos e arcos etiquetados. Os arcos etiquetados, os nodos e os nodos distinguidos representam as transições, os estados, e os estados iniciais ou finais do autômato, respectivamente.

A categoria dos autômatos seqüenciais é completa e cocompleta sendo o coproduto categorial interpretado como a escolha e o produto como a composição paralela refletindo simultaneamente a intercalação e a composição síncrona entre as transições dos autômatos componentes. As operações de sincronização e de encapsulação são functoriais e são construídas usando a técnica de fibração e cofibração, inspiradas em [Winskel 87] e baseadas em [Menezes & Costa 93] e generalizadas em [Menezes *et al* 95]. Ambos os funtores são induzidos a partir de morfismos ao nível das etiquetas.

Para uma dada tabela de sincronizações ao nível das etiquetas, a construção do autômato que reflete o comportamento conjunto desejado é realizada a partir da composição paralela, onde as transições que não satisfazem à tabela são apagadas. Adicionalmente, para o caso de sincronizações do tipo chamada ou compartilhamento (ou ambos) é apresentada uma construção categorial para a tabela de sincronizações, baseada em um colimite.

A operação de encapsulação extrai uma visão de um autômato, tornando visível ao exterior (do autômato) somente algumas transições, escondendo as demais. A encapsulação de um autômato é realizada reetiquetando as transições, onde uma transições etiquetada pelo símbolo especial τ é considerada escondida e, portanto, não pode ser especificada na tabela de sincronizações.

3.1 Autômato

Tradicionalmente, a etiquetação dos arcos de um grafo G é determinada por uma função total $lab: T \rightarrow L$ onde T é o conjunto de transições de G e L é o conjunto de etiquetas. Considerando que os grafos de um só nodo são isomorfos aos conjuntos apontados, a etiquetação de um grafo pode ser vista como um morfismo de grafos onde o grafo destino é de um só nodo. Portanto, um autômato pode ser definido como um objeto da categoria das setas determinada pelo functor identidade na categoria dos grafos reflexivos com nodos distinguidos e o functor inclusão da subcategoria dos grafos de um só nodo na categoria dos grafos reflexivos com nodos distinguidos.

Lembre-se que $DRGr$ denota a categoria dos grafos reflexivos cujos nodos distinguidos são pares de conjuntos, isto é, $DRGr = \Downarrow \text{nodos}$ onde $\Downarrow: Set^2 \rightarrow Set$ é o functor canonicamente induzido pelo coproduto categorial em Set e $\text{nodos}: RGr \rightarrow Set$ é o functor esquecimento que associa a cada grafo o seu correspondente conjunto de nodos.

Definição 3.1 Categoria dos Autômatos Seqüenciais. A categoria dos autômatos seqüenciais ou simplesmente categoria dos autômatos é a categoria das setas $id \downarrow inc$, denotada por \mathcal{Aut} , onde:

- a) $id: DRGr \rightarrow DRGr$ é o functor identidade;

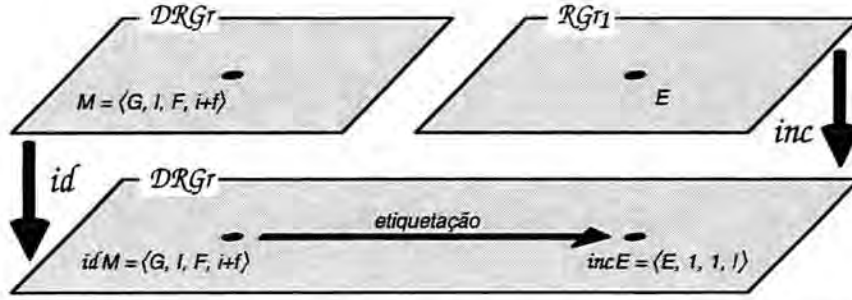


Figura 3.1 Autômato como um morfismo de grafos reflexivos com nodos distinguidos

b) $inc: \mathcal{RGr}_1 \rightarrow \mathcal{DRGr}$ é o funtor inclusão tal que, para cada \mathcal{RGr}_1 -objeto $E = \langle 1, T, !, !, \iota \rangle$, $incE = \langle E, 1, 1, !_{1+1} \rangle$ onde a função total $!_{1+1}: 1+1 \rightarrow 1$ é única e, para cada \mathcal{RGr}_1 -morfismo $h: E_1 \rightarrow E_2$, $inch = \langle h, !_1, !_1 \rangle: \langle E_1, 1, 1, !_{1+1} \rangle \rightarrow \langle E_2, 1, 1, !_{1+1} \rangle$ onde a função total $!_1: 1 \rightarrow 1$ é única. \square

Portanto, um autômato A é uma tripla $A = \langle M, E, etiq \rangle$, tal que (veja a Figura 3.1):

- $M = \langle G, l, F, i+f \rangle$ é um objeto de \mathcal{DRGr} onde o grafo reflexivo G representa a forma do autômato, l é o conjunto dos estados iniciais, F é o conjunto dos estados finais e a função total $i+f: l+F \rightarrow \text{nodos } G$ instancia os nodos iniciais e finais nos nodos da forma do autômato;
- E é um grafo reflexivo de um só nodo representando as etiquetas das transições do autômato;
- $etiq = \langle !_V, etiq_T, !_l, !_F \rangle: idM \rightarrow incE$ é um morfismo de grafos reflexivos com nodos distinguidos o qual associa uma etiqueta a cada arco (da forma) do autômato;

Um autômato $A = \langle M, E, etiq \rangle$ onde $M = \langle V, T, \partial_0, \partial_1, \iota, l, F, i+f \rangle$ pode ser denotado por $A: idM \rightarrow incE$ ou simplesmente $A: M \rightarrow E$ ou ainda por uma 10-upla $A = \langle V, T, \partial_0, \partial_1, \iota, l, F, i+f, E, etiq \rangle$. Os nodos (da forma) são denominados estados e os arcos etiquetados são denominados transições. A etiquetação é não extensional, no sentido em que podem existir dois ou mais arcos paralelos com uma mesma etiqueta. A etiquetação não-extensional é importante na semântica de reificação a linguagens, como será visto adiante. Um arco etiquetado por τ representa uma transição encapsulada (escondida), ou seja, uma transição que não pode ser referenciada externamente ao autômato. Como a forma de um autômato é um grafo reflexivo, cada estado possui uma transição identidade associada a qual é interpretada como uma transição do tipo sem operação ("no operation" ou "idle" - não há mudança de estado). Repare que toda transição identidade é etiquetada por τ . No caso do conjunto de estados iniciais do autômato possuir mais de um elemento, a escolha de qual estado inicial é efetivamente considerado no momento da execução do autômato é um não-determinismo externo. Um autômato pára quando nenhuma transição pode ser executada a partir do estado corrente. Diz-se que um autômato pára normalmente, se o autômato pára em um estado final. Um conjunto de estados iniciais (e não um único estado inicial) é uma solução já anteriormente adotada como, por exemplo, em [Jonsson 90].

A representação gráfica de um autômato $A: M \rightarrow E$ possui a forma de M , sendo as transições etiquetadas por E , de acordo com o morfismo de etiquetação. Uma transição t etiquetada por a é denotada por $a[t]$ ou simplesmente por a .

Considere um autômato $A = \langle M, E, etiq \rangle$ onde $M = \langle V, T, \partial_0, \partial_1, \iota, l, F, i+f \rangle$. Por simplicidade, se $i+f: l+F \rightarrow V$ é uma inclusão, então M é denotado simplesmente por $M = \langle V, T, \partial_0, \partial_1, \iota, l, F \rangle$. Adicionalmente, quando o conjunto de arcos é representado por extensão, os arcos identidade são omitidos. Por exemplo, o grafo $M = \langle \{A, B, C, D\}, \{a, b, c, d, \iota_A, \iota_B, \iota_C, \iota_D\}, \partial_0, \partial_1, \iota, \{A\}, \{D\}, inc_{\{A,D\}} \rangle$ é representado simplesmente por $M = \langle \{A, B, C, D\}, \{a, b, c, d\}, \partial_0, \partial_1, \iota, \{A\}, \{D\} \rangle$.

Exemplo 3.2 Considere os grafos reflexivo com nodos distinguidos $M = \langle \{A, B, C\}, \{a, b, c, d, e\}, \partial_0, \partial_1, \iota, \{A\}, \{D\} \rangle$ onde ∂_0, ∂_1 são tais que $a: A \rightarrow B, b: B \rightarrow C, c: C \rightarrow B, d: C \rightarrow B, e: C \rightarrow D$ e o grafo reflexivo de um só nodo $E = \{\tau, inic, flip, flop, fim\}$. A representação gráfica do autômato Flip-2Flop: $M \rightarrow E$ com a etiquetação determinada por $a \mapsto inic, b \mapsto flip, c \mapsto flop, d \mapsto flop, e \mapsto fim$ é ilustrada na Figura 3.2. \square

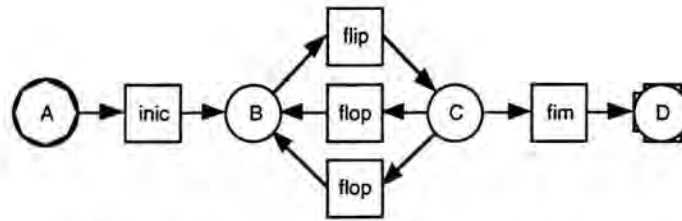


Figura 3.2 Representação gráfica de um autômato

Um morfismo de autômatos é um morfismo na categoria das seta $id \downarrow inc$, ou seja, dados os autômatos $A_1: M_1 \rightarrow E_1$ e $A_2: M_2 \rightarrow E_2$ onde $M_1 = \langle V_1, T_1, \partial_{10}, \partial_{11}, \iota_1, l_1, F_1, i_1 + f_1 \rangle$ e $M_2 = \langle V_2, T_2, \partial_{20}, \partial_{21}, \iota_2, l_2, F_2, i_2 + f_2 \rangle$, um morfismo $h: A_1 \rightarrow A_2$ é um par de morfismos de grafos $\langle h_M: M_1 \rightarrow M_2, h_E: E_1 \rightarrow E_2 \rangle$ tal que o diagrama ilustrado na Figura 3.3 comuta. Adicionalmente, $h: A_1 \rightarrow A_2$ também pode ser visto como uma quintupla $\langle h_V: V_1 \rightarrow V_2, h_T: T_1 \rightarrow T_2, h_l: l_1 \rightarrow l_2, h_F: F_1 \rightarrow F_2, h_E: E_1 \rightarrow E_2 \rangle$.

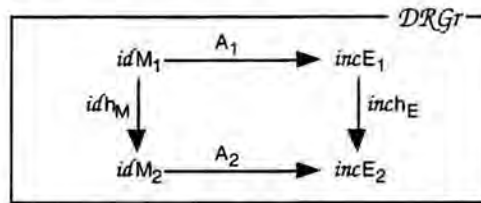


Figura 3.3 Diagrama comutativo para morfismos de autômatos

Proposição 3.3 A categoria Aut é completa e cocompleta.

Prova: A categoria $RGrl$ é bicompleta. Para verificar que $DRGr = G_{\Downarrow}$ é bicompleta, é suficiente verificar que $\Downarrow: Set^2 \rightarrow Set$ preserva colimites. De fato, \Downarrow é adjunto esquerdo do funtor diagonal $\Delta: Set \rightarrow Set^2$. Como $Aut = id \downarrow inc$ e como $id: DRGr \rightarrow DRGr$ trivialmente preserva colimites, para verificar que Aut é bicompleta, é suficiente verificar que $inc: RGrl \rightarrow DRGr$ preserva limites. Considere o funtor $\tau_{g_1}: DRGr \rightarrow RGrl$ tal que:

- para todo o $DRGr$ -objeto $M = \langle V, T, \partial_0, \partial_1, \iota, l, F, i + f \rangle$, $\tau_{g_1} M = \langle 1, T_\tau, !, !, \iota_\tau \rangle$ tal que T_τ e $\iota_\tau: 1 \rightarrow T_\tau$ são determinados pela soma amalgamada ilustrada na Figura 3.4 onde τ denota o elemento distinguido do conjunto apontado T_τ e ι_τ associa o único elemento de 1 a τ ;
- para todo o $DRGr$ -morfismo $h = \langle h_V, h_T, h_l, h_F \rangle$, $\tau_{g_1} h = \langle !_V, h_{T_\tau} \rangle$ onde h_{T_τ} é canonicamente induzida por h_T .

Logo, para todo o $DRGr$ -objeto $M = \langle V, T, \partial_0, \partial_1, \iota, l, F, i + f \rangle$, $inc \circ \tau_{g_1} M = \langle 1, T_\tau, !, !, \iota_\tau, 1, 1, ! \rangle$. É fácil verificar que τ_{g_1} é adjunto esquerdo de inc onde a unidade $\eta: id \rightarrow inc \circ \tau_{g_1}$ é tal que $\eta = \langle !_V, p, !, !, F \rangle$ sendo p determinado pela soma amalgamada da Figura 3.4. Portanto, Aut é bicompleta. \square

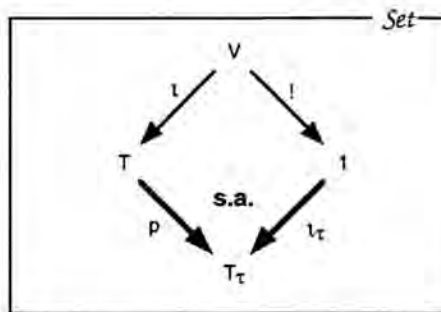


Figura 3.4 Soma amalgamada

Em \mathcal{Aut} , o coproduto representa a escolha entre autômatos e o produto a composição paralela refletindo simultaneamente a intercalação e a composição síncrona entre as transições dos autômatos componentes. Note-se que ambas as operações são comutativas e associativas. Os autômatos resultantes do produto e do coproduto de $A_1 = \langle M_1, E_1, \text{eti}q_1 \rangle$ e $A_2 = \langle M_2, E_2, \text{eti}q_2 \rangle$ são como segue:

$$A_1 \times A_2 = \langle M_1 \times \mathcal{DRGR} M_2, E_1 \times \mathcal{RG} E_2, \text{eti}q_1 \times \mathcal{DRGR} \text{eti}q_2 \rangle$$

$$A_1 + A_2 = \langle M_1 + \mathcal{DRGR} M_2, E_1 + \mathcal{RG} E_2, \text{eti}q_1 + \mathcal{DRGR} \text{eti}q_2 \rangle$$

onde os morfismos $\text{eti}q_1 \times \mathcal{DRGR} \text{eti}q_2$, $\text{eti}q_1 + \mathcal{DRGR} \text{eti}q_2$ são univocamente induzidos pelo produto e coproduto categoriais, respectivamente.

Exemplo 3.4 Considere a Figura 3.5. O autômato $A_1 \times A_2$ representa o objeto resultante do produto de A_1 e A_2 em \mathcal{Aut} . \square

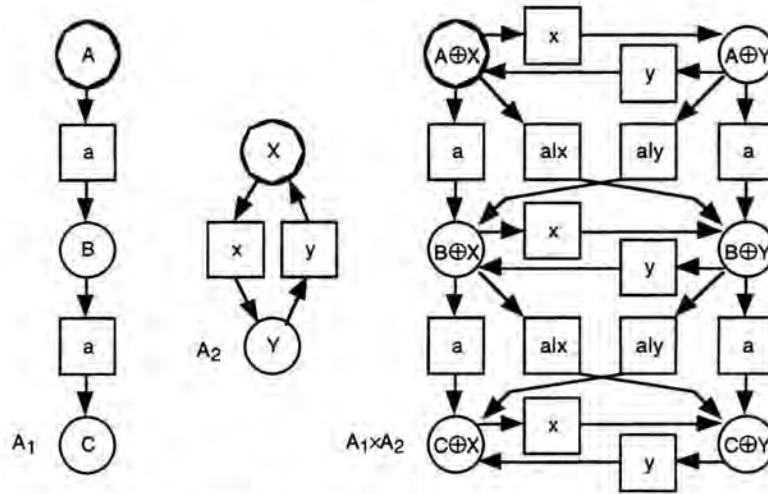


Figura 3.5 Produto de autômatos seqüenciais

Observação 3.5 Autômato Seqüencial Não-Reflexivo. Caso seja considerada a categoria dos autômatos seqüenciais *não-reflexivos*, o produto categorial reflete a operação de composição (totalmente) síncrona, ou seja, não reflete a intercalação. \square

3.2 Sincronização

A sincronização de autômatos é uma operação funtorial induzida como ilustrado na Figura 3.6 e é como segue (no texto a seguir, é omitido que $i \in I$, onde I é um conjunto):

- seja $\{A_i: M_i \rightarrow E_i\}$ um conjunto de autômatos a serem sincronizados;
- seja Tabela uma tabela de sincronizações (de etiquetas) que reflete a sincronização desejada entre os autômatos componentes;
- seja $\text{sinc}: \text{Tabela} \rightarrow \times E_i$ um morfismo de sincronização o qual instancia as etiquetas da tabela de sincronizações no produto das etiquetas dos autômatos componentes;
- seja $u: \mathcal{Aut} \rightarrow \mathcal{RG} E_i$ o functor esquecimento que associa a cada autômato o seu correspondente grafo de etiquetas. O functor u é uma fibração e as fibras $u^{-1} \text{Tabela}$, $u^{-1} \times E_i$ são subcategorias de \mathcal{Aut} ;
- o morfismo de sincronização sinc e a fibração u induzem um functor $\text{sinc}: u^{-1} \times E_i \rightarrow u^{-1} \text{Tabela}$. O functor sinc aplicado ao produto dos autômatos componentes, resulta no autômato que reflete a sincronização desejada.

Portanto, a sincronização é obtida a partir da especificação da sincronização ao nível das etiquetas dos autômatos componentes. Adicionalmente, no texto que segue, é apresentada uma

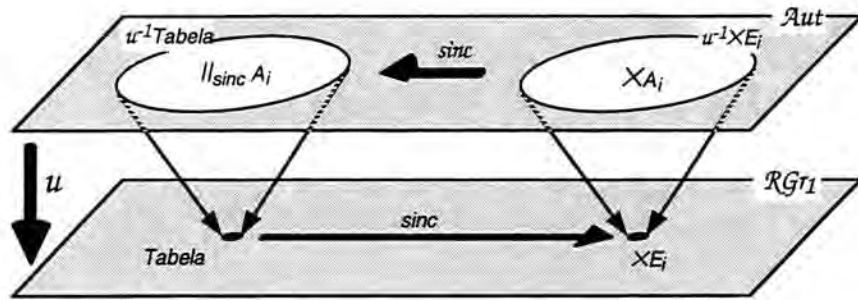


Figura 3.6 Indução do funtor de sincronização

construção categorial da tabela de sincronizações e do correspondente morfismo de sincronização para os casos de chamada e compartilhamento.

Os conceitos e propriedades usados referentes à fibração e ao funtor produto fibrado são introduzidos em apêndice. As principais referências são [Barr & Wells 90] para fibrações e [Johnstone 77] para o funtor produto fibrado.

Fibração

Proposição 3.6 O funtor esquecimento $u: \mathcal{A}ut \rightarrow \mathcal{R}Gr1$ que associa a cada autômato $A: M \rightarrow E$ o correspondente grafo reflexivo de um só nodo E , é uma fibração.

Prova: Seja $f: E_1 \rightarrow E_2$ um morfismo em $\mathcal{R}Gr1$ e $A_2 = \langle M_2, E_2, \text{eti}q_2 \rangle$ um autômato. Considere o produto fibrado representado na Figura 3.7. Então, $A_1 = \langle M_1, E_1, \text{eti}q_1 \rangle$ é um autômato por construção e $u = \langle u_M, f \rangle: A_1 \rightarrow A_2$ é morfismo cartesiano de f e A_2 .

Para verificar que u é morfismo cartesiano de f e A_2 , considere o autômato $A_3 = \langle M_3, E_3, \text{eti}q_3 \rangle$, o $\mathcal{A}ut$ -morfismo $v = \langle v_M, v_E \rangle: A_3 \rightarrow A_2$ e o $\mathcal{R}Gr1$ -morfismo $h: E_3 \rightarrow E_1$ tal que $uv = v_E = f \circ h$. Pela propriedade universal dos produtos fibrados, existe um morfismo univocamente induzido $w_M: M_3 \rightarrow M_1$ tal que $v_M = u_M \circ w_M$ e $\text{inch} \circ \text{eti}q_3 = \text{eti}q_1 \circ w_M$. De fato, $w = \langle w_M, h \rangle: A_3 \rightarrow A_1$ é o único $\mathcal{A}ut$ -morfismo tal que $v = u \circ w$ e $uw = h$. \square

Observação 3.7 Funtor Produto Fibrado Induzido. Na prova da proposição acima, o morfismo $f: E_1 \rightarrow E_2$ induz um funtor produto fibrado, como segue:

$$f^*: \text{id}_{\mathcal{R}Gr} \downarrow e_2 \rightarrow \text{id}_{\mathcal{R}Gr} \downarrow e_1$$

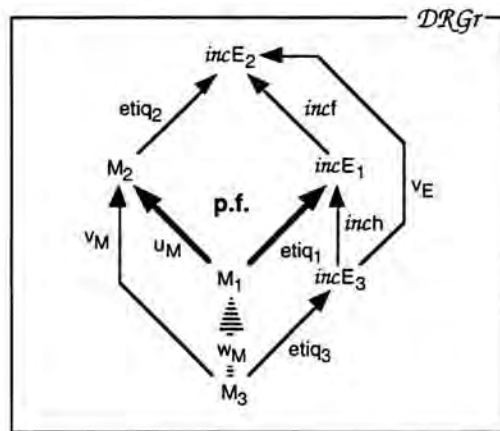


Figura 3.7 Produto fibrado e morfismo univocamente induzido

tal que, para k em $\{1, 2\}$, $e_k: E_k \rightarrow \mathcal{RGr}$ é o funtor inclusão onde E_k é a subcategoria determinada pelo objeto E_k e o morfismo identidade em E_k . Neste caso, $f^*eti_2 = eti_1$. \square

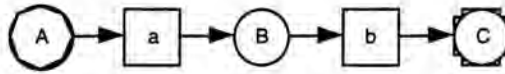
No texto que segue, considere o \mathcal{RGr} -morfismo $f: E_1 \rightarrow E_2$ da proposição anterior. Então, o produto fibrado usado na prova possui as seguintes propriedades:

- a) se $f: E_1 \rightarrow E_2$ é um isomorfismo, então A_1 é simplesmente uma reetiquetagem das transições de A_2 e, portanto, a forma do autômato não é alterada;
- b) se $f: E_1 \rightarrow E_2$ é um monomorfismo, então A_1 é uma restrição das transições de A_2 , ou seja, o conjunto das transições de A_1 é um subconjunto (reetiquetado) das transições de A_2 ;
- c) se f é tal que, para alguma etiqueta e de E_1 , $f(e) = \tau$, então uma endotransição distinta da identidade e etiquetada por e é introduzida em cada estado de A_1 ;
- d) se f é tal que para duas etiquetas distintas a e b de E_1 , $f(a) = f(b) = x$, então as transições de A_2 etiquetadas por x são substituídas em A_1 por duas transições paralelas etiquetadas por a e b com os mesmos estados origem e destino de x .

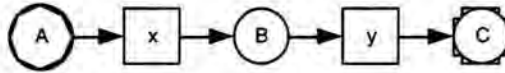
O exemplo a seguir ilustra cada um dos casos acima.

Exemplo 3.8 Considere a Figura 3.8. O autômato $A_2: M_2 \rightarrow E_2$ onde $E_2 = \{\tau, a, b\}$ é transformado no autômato A_1 pelo produto fibrado usado na prova da proposição anterior, de acordo com o

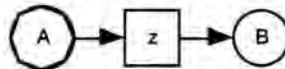
Autômato a ser transformado:



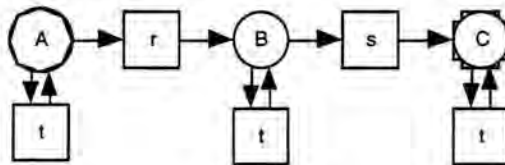
a) Reetiquetagem



b) Restrição



c) Adição de endotransições



d) Substituição de transição por transições paralelas

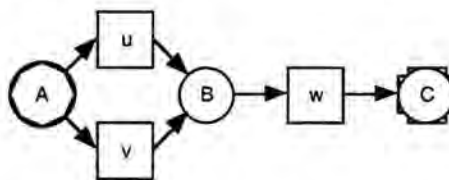


Figura 3.8 Autômatos original e os correspondentes transformados

morfismo f como especificado em cada um dos itens abaixo:

- a) Suponha que $f: E_1 \rightarrow E_2$ é um isomorfismo onde $E_1 = \{\tau, x, y\}$ e $x \mapsto a, x \mapsto b$;
- b) Suponha que $f: E_1 \rightarrow E_2$ é um monomorfismo onde $E_1 = \{\tau, z\}$ e $z \mapsto a$;
- c) Suponha o morfismo $f: E_1 \rightarrow E_2$ onde $E_1 = \{\tau, r, s, t\}$ e $t \mapsto \tau, r \mapsto a, s \mapsto b$;
- d) Suponha o morfismo $f: E_1 \rightarrow E_2$ onde $E_1 = \{\tau, u, v, w\}$ e $u \mapsto a, v \mapsto a, w \mapsto b$.

□

No caso de sincronização de autômatos, o morfismo f considerado para a elevação cartesiana é sempre um monomorfismo.

Funtor de Sincronização

O comportamento conjunto dos autômatos é determinado pelo operador funtorial de sincronização induzido pela fibração e pelo morfismo de sincronização, aplicado ao produto dos autômatos componentes.

Definição 3.9 Tabela de Sincronizações, Morfismo de Sincronização. Sejam $\{A_i: M_i \rightarrow E_i\}$ um conjunto de autômatos a serem sincronizados. Então:

- a) uma tabela de sincronizações de $\{A_i\}$, denotada por Tabela, é um \mathcal{RG}_{T1} -objeto que reflete a sincronização desejada entre os autômatos;
- b) um morfismo de sincronização $\text{sinc}: \text{Tabela} \rightarrow \times E_i$ é um \mathcal{RG}_{T1} -monomorfismo.

□

Definição 3.10 Funtor de Sincronização. Sejam $\{A_i: M_i \rightarrow E_i\}$ um conjunto de autômatos a serem sincronizados e $\text{sinc}: \text{Tabela} \rightarrow \times E_i$ um morfismo de sincronização. Considere a fibração $u: \mathcal{Aut} \rightarrow \mathcal{RG}_{T1}$, a construção dos correspondentes morfismos cartesianos como na prova da Proposição 3.6 e as fibras $u^{-1}\text{Tabela}$, $u^{-1}\times E_i$. Então $\text{sinc}: u^{-1}\times E_i \rightarrow u^{-1}\text{Tabela}$ é o funtor de sincronização induzido por u e sinc e o autômato resultante da sincronização de $\{A_i\}$, denotado por $\parallel_{\text{sinc}} A_i$, é $\text{sinc} \times A_i$.

□

Portanto, o autômato resultante de uma sincronização é uma restrição da composição paralela dos autômatos componentes.

Exemplo 3.11 Considere os autômatos produtor e consumidor com as etiquetas $E_1 = \{\tau, \text{envia}, \text{prod}\}$ e $E_2 = \{\tau, \text{rec}, \text{cons}\}$, respectivamente, bem como o correspondente autômato resultante do produto, ilustrados na Figura 3.9 (por simplicidade, as etiquetas envia, prod, rec, cons são abreviadas por e, p, r, c, respectivamente).

- a) Suponha a sincronização envia compartilha rec, representando uma comunicação sem "buffer",

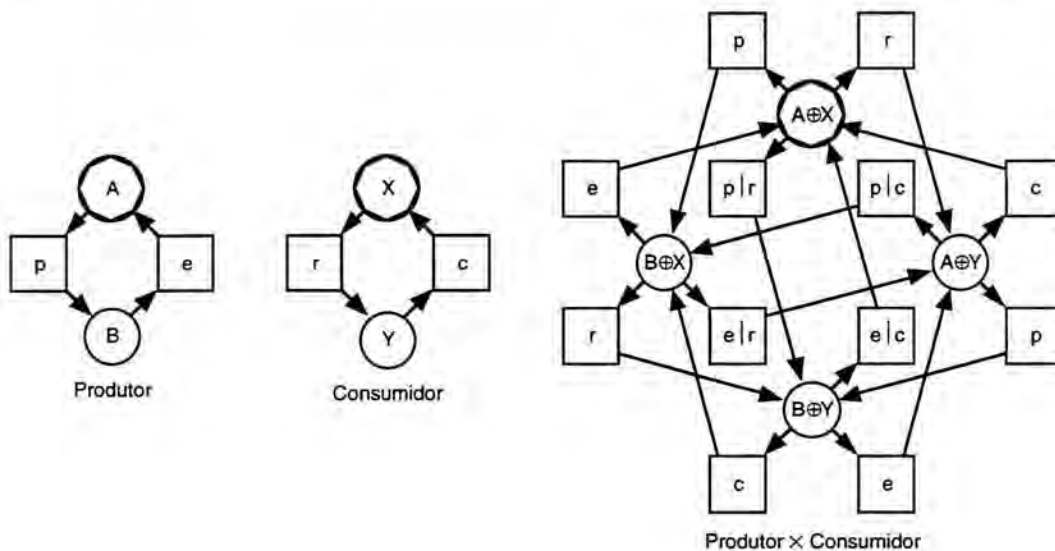


Figura 3.9 Autômatos produtor e consumidor e a correspondente composição paralela

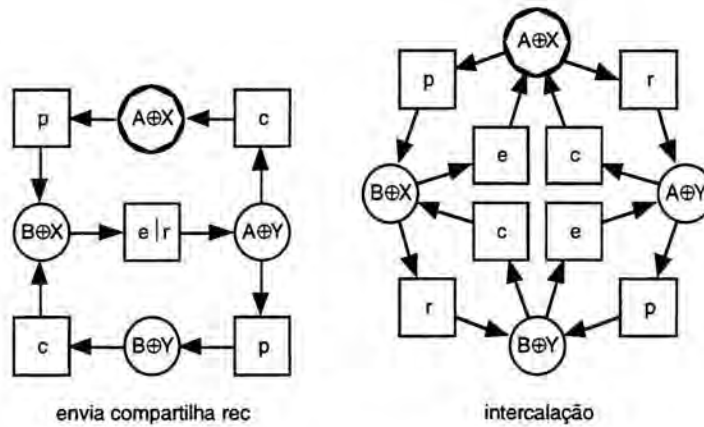


Figura 3.10 Sincronização dos autômatos produtor e consumidor

como em CSP [Hoare 85] ou CCS [Milner 80]. Então, a tabela de sincronizações é $Tabela = \{\tau, prod, cons, envia | rec\}$ e o autômato resultante é ilustrado na Figura 3.10 (esquerda).

- b) Suponha que não é desejada qualquer sincronização entre as transições dos autômatos componentes. Então, a tabela de sincronizações é $Tabela = \{\tau, prod, envia, rec, cons\}$ e o autômato resultante é como ilustrado na Figura 3.10 (direita) e representa a intercalação das transições dos autômatos componentes. □

Tabela de Sincronizações

A seguir, é apresentada uma construção categorial da tabela de sincronizações bem como de um morfismo de sincronização para sincronizações do tipo chamada, compartilhamento ou ambos os casos. A tabela de sincronizações é determinada por um colimite cujo diagrama possui uma forma espacial que lembra um diamante, como ilustrado na Figura 3.11 onde a seta central possui como origem um objeto denominado canal, e como destino a tabela de sincronizações.

Definição 3.12 Tabela de Sincronizações. Seja $\{A_i; M_i \rightarrow E_i\}$ um conjunto de autômatos a serem sincronizados. Então:

- a) seja Canal o menor subgrafo de $\times E_i$ que representa as etiquetas a serem sincronizadas;

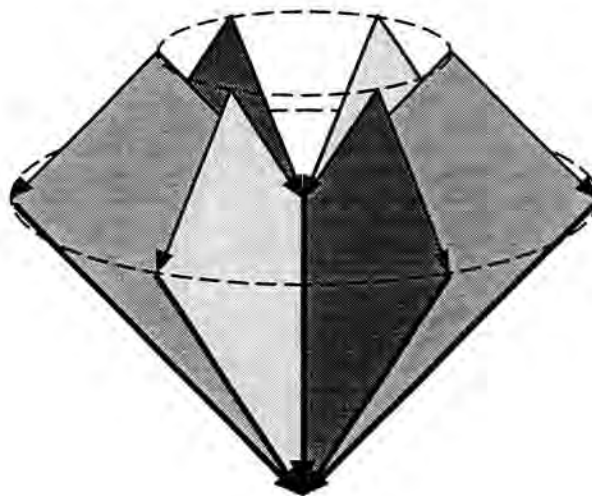


Figura 3.11 Tabela de sincronizações determinada por um colimite com a forma de um diamante

- b) para cada i , E_i' é o menor subgrafo de E_i que contém todas as etiquetas das transições de A_i as quais chamam alguma transição de outro autômato;
- c) para cada i , o morfismo chama $_i$: $E_i' \rightarrow \text{Canal}$ é tal que se a em E_i' chama as transições x_1, \dots, x_n , então chama $_i(a) = a | x_1 | \dots | x_n$.

Seja D o diagrama representado na Figura 3.12 onde, para cada i , sub $_i$: $E_i' \rightarrow E_i$ é um monomorfismo que caracteriza E_i' como subobjeto de E_i . A tabela de sincronizações denotada por Tabela é o objeto resultante do colimite do diagrama D . □

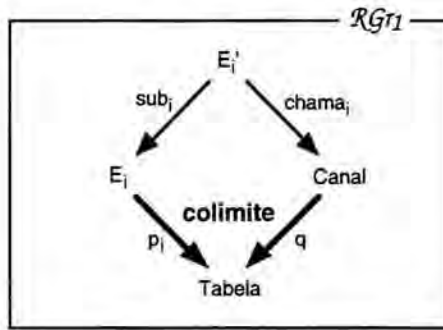


Figura 3.12 Tabela de sincronizações determinada por um colimite

A construção categorial acima não deve ser confundida com as construções baseadas em diagramas de sincronização como propostas em [Costa 91] e [Costa & Sernadas 92] (usando limites) e em [Sernadas *et al* 90] (usando colimites). A principal diferença é que estes consideram uma tabela de sincronizações previamente fornecida e objetivam determinar o objeto resultante da sincronização.

Da definição acima (item c), pode-se concluir que, para todo i , chama $_i$ é um monomorfismo. O diagrama D também é denominado diagrama de $\#l$ -picos onde $\#l$ é o cardinal de l . A construção da tabela de sincronizações proposta possui as seguintes propriedades:

- se $E_i' = E_i$, então todos os elementos de E_i estão sincronizados e, conseqüentemente, na tabela resultante, estão presentes somente uma vez e como componente de alguma upla;
- se a única etiqueta do canal for τ , então a tabela de sincronizações resulta no coproduto das etiquetas dos autômatos componentes. Neste caso, o autômato resultante da operação de sincronização, representa a intercalação pura das transições dos autômatos componentes.

Exemplo 3.13

a) Considere $E_1 = \{\tau, a, b, c\}$ e $E_2 = \{\tau, x, y\}$. Suponha que a chama x , b chama y e y chama b (ou seja, b compartilha y). Então, Canal = $\{\tau, a | x, b | y\}$ e a tabela de sincronizações Tabela é determinada

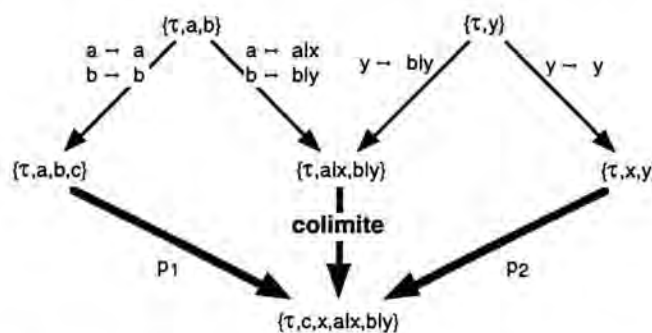


Figura 3.13 Exemplo de tabela de sincronizações para dois autômatos

como ilustrado na Figura 3.13.

- b) Considere $E_k = \{\tau, p_k, c_k\}$ para k com valores em $\{1, 2, 3\}$. Suponha o compartilhamento de c_1, c_2 e c_3 . Então, Canal = $\{\tau, c_1 | c_2 | c_3\}$ e Tabela = $\{\tau, p_1, p_2, p_3, c_1 | c_2 | c_3\}$.
- c) Considere o Exemplo 3.11. No caso em que envia compartilha rec, a construção da tabela de sincronizações possui Canal = $\{\tau, envia | rec\}$. No caso da intercalação, Canal = $\{\tau\}$. \square

Proposição 3.14 Considere o diagrama D cujo colimite determina o grafo Tabela juntamente com os morfismos $p_i: E_i \rightarrow$ Tabela. Então, para todo i , p_i e q possuem retração.

Prova: Os morfismos sub_i e $chama_i$ são mono e o colimite do diagrama D pode ser determinado por somas amalgamadas, como ilustrado na Figura 3.14 para o caso $\#I = 2$ (diagrama de 2-picos). Logo, todos os morfismos do diagrama das somas amalgamadas são mono (em um diagrama da forma $\bullet \leftarrow \bullet \rightarrow \bullet$, os morfismos da soma amalgamada também são mono). Considerando que a composição de monomorfismos é mono, então, para todo i , p_i e q também são mono. Portanto, para todo i , existem retrações para p_i e q tais que, para todo a em Tabela, se existe a_i em E_i tal que $p_i(a_i) = a$, então $p_i^R(a) = a_i$, caso contrário, $p_i^R(a) = \tau$ e, se existe c em Canal tal que $q(c) = a$, então $q^R(a) = c$, caso contrário, $q^R(a) = \tau$. \square

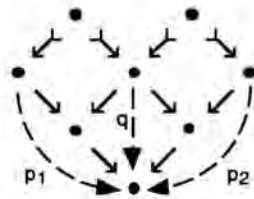


Figura 3.14 Determinação de um colimite de um diagrama de 2-picos usando somas amalgamadas

Na definição abaixo, lembre-se que \mathcal{RGr}_I é isomorfa a Set^* e que, em Set^* , $\times E_i$ é isomorfo a $(+_{Set} E_i') +_{Set} (\times_{Set} E_i') +_{Set} \{\tau\}$ onde E_i' é E_i sem o elemento distinguido e , portanto, existem monomorfismos $inj_i: E_i \rightarrow \times E_i$ canonicamente induzidos pelo coproduto categorial.

Definição 3.15 Morfismo de Sincronização. Seja $s: Canal \rightarrow \times E_i$ um monomorfismo que caracteriza Canal como subobjeto de $\times E_i$. Sejam $inj_i: E_i \rightarrow \times E_i$ monomorfismos canonicamente induzidos pelo coproduto categorial. O morfismo de sincronização determinado por um diagrama D é sinc: Tabela $\rightarrow \times E_i$ tal que $sinc(\tau) = \tau$ e, para todo $a \neq \tau$ em Tabela, se $q^R(a) \neq \tau$, então $sinc(a) = s \circ q^R(a)$, caso contrário existe um único k em I tal que $p_k^R(a) \neq \tau$ e $sinc(a) = inj_k \circ p_k^R(a)$. \square

É fácil verificar que o morfismo sinc definido acima é, de fato, um monomorfismo.

3.3 Encapsulação

Tradicionalmente, em teoria da concorrência, uma transição encapsulada é uma transição etiquetada por um símbolo especial, normalmente denotado por τ ([Winskel 87]). As transições encapsuladas, não podem ser usadas para comunicação com o exterior do autômato e, portanto, não podem ser sincronizadas. A encapsulação de transições de um autômato é uma operação funtorial induzida como ilustrado na Figura 3.15, a qual consiste de uma reetiquetagem especial, como segue:

- seja $A: M \rightarrow E_1$ um autômato a ser encapsulado;
- seja $caps: E_1 \rightarrow E_2$ um morfismo onde as etiquetas das transições a serem encapsuladas são associadas ao símbolo τ ;
- seja $u: Aut \rightarrow \mathcal{RGr}_I$ o mesmo funtor esquecimento usado na sincronização. O funtor u é uma cofibração (e portanto, é uma bifibração) e as fibras $u^{-1}E_1$ e $u^{-1}E_2$ são subcategorias de Aut ;
- o morfismo $caps$ e a cofibração u induzem o funtor $caps: u^{-1}E_1 \rightarrow u^{-1}E_2$. O funtor $caps$ aplicado ao autômato A resulta no autômato que reflete a encapsulação desejada.

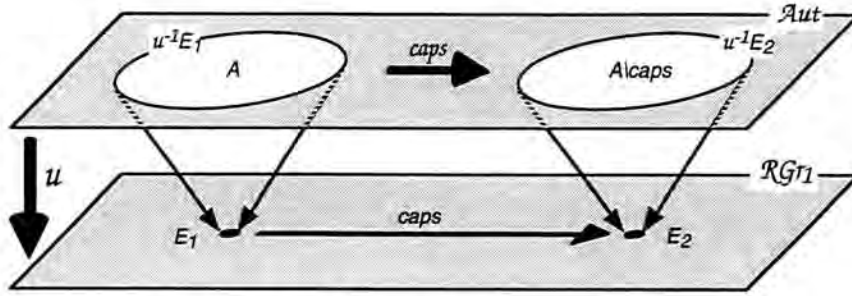


Figura 3.15 Indução do funtor de encapsulação

Portanto, a encapsulação de um autômato é obtida a partir da especificação de quais etiquetas devem ser encapsuladas.

Proposição 3.16 O funtor esquecimento $u: \mathcal{A}ut \rightarrow \mathcal{R}Gr_1$ que associa a cada autômato $A: M \rightarrow E$ o correspondente grafo reflexivo de um só nodo E , é uma cofibração.

Prova: Considere o morfismo $f: E_1 \rightarrow E_2$ em $\mathcal{R}Gr_1$ e o autômato $A_1 = \langle M_1, E_1, \text{eti}q_1 \rangle$. Sejam $A_2 = \langle M_1, E_2, f \circ \text{eti}q_1 \rangle$ um autômato e $u = \langle \text{id}_{M_1}, f \rangle: A_1 \rightarrow A_2$ um morfismo de autômatos. Então u é cocartesiano em relação ao f e A_1 .

Para verificar que u é cocartesiano, considere o autômato $A_3 = \langle M_3, E_3, \text{eti}q_3 \rangle$, o $\mathcal{A}ut$ -morfismo $v = \langle v_M, v_E \rangle: A_1 \rightarrow A_3$ e o $\mathcal{R}Gr_1$ -morfismo $h: E_2 \rightarrow E_3$ tal que $uv = v_E = h \circ f$. Então, $w = \langle v_M, h \rangle$ é o único morfismo tal que $v = w \circ u$. Para verificar que w é de fato um morfismo de autômatos, considere o diagrama ilustrado na Figura 3.16 (esquerda). Como $v_E = h \circ f$ e $A_2: M_1 \rightarrow E_2$, o diagrama pode ser reescrito como ilustrado na Figura 3.16 (direita). Portanto, como $\textcircled{1}$ comuta, w é um morfismo de autômatos. \square

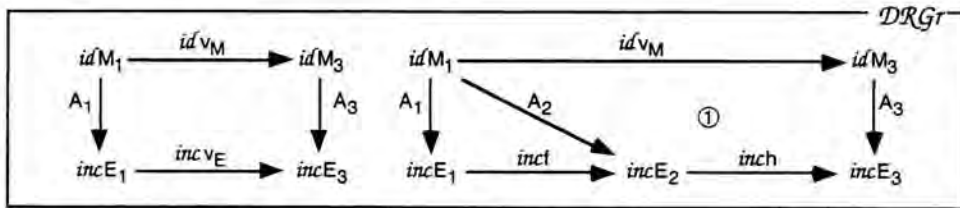


Figura 3.16 Diagramas comutativos

Observação 3.17 Bifibração e Adjunto Esquerdo do Funtor Produto Fibrado. A cofibração u usada na proposição acima, também é a fibração usada na sincronização de autômatos. Portanto, u é uma bifibração. Adicionalmente, o funtor produto fibrado $f^*: \text{id}_{\mathcal{R}Gr} \downarrow e_2 \rightarrow \text{id}_{\mathcal{R}Gr} \downarrow e_1$ (ver Observação 3.7) possui adjunto esquerdo, como segue:

$$f^*: \text{id}_{\mathcal{R}Gr} \downarrow e_1 \rightarrow \text{id}_{\mathcal{R}Gr} \downarrow e_2$$

o qual é a composição usada na prova de que u é uma cofibração. \square

Se o morfismo $f: E_1 \rightarrow E_2$ usado na prova da proposição anterior é bijetivo, então o morfismo cocartesiano referente a f é simplesmente uma reetiquetagem do autômato origem. Para encapsulação de transições, são considerados os morfismos de encapsulação, os quais são injetivos exceto para as etiquetas a serem encapsuladas, as quais são reetiquetadas com o símbolo especial τ .

Definição 3.18 Morfismo de Encapsulação. Seja E_1 as etiquetas do autômato considerado.

- a) Dado o subobjeto de etiquetas E_2 a serem exportadas. Seja $sub: E_2 \rightarrow E_1$ um monomorfismo que caracteriza E_2 como subobjeto de E_1 . Então $caps = sub^R: E_1 \rightarrow E_2$ onde sub^R é a retração de sub tal que, para todo a_1 em E_1 , se existe a_2 em E_2 tal que $sub(a_2) = a_1$, então $sub^R(a_1) = a_2$, caso contrário, $sub^R(a_1) = \tau$.
- b) Dado o subobjeto E de etiquetas a serem encapsuladas. Seja $sub: E \rightarrow E_1$ um monomorfismo que caracteriza E como subobjeto de E_1 . Então o morfismo de encapsulação $caps: E_1 \rightarrow E_2$ induzido por sub é determinado pela soma amalgamada ilustrada na Figura 3.17, onde o morfismo $!_\tau$ é único. □

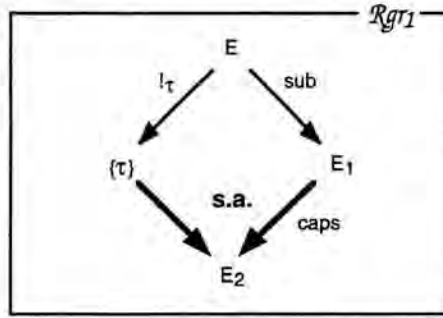


Figura 3.17 Morfismo de encapsulação determinado por uma soma amalgamada

Exemplo 3.19 Sejam $E = \{\tau, flop\}$, $E_1 = \{\tau, inic, flip, flop, fim\}$ e $sub: E \rightarrow E_1$ tal que $\tau \mapsto \tau$, $flop \mapsto flop$. Então, $E_2 = \{\tau, inic, flip, fim\}$ e $caps: E_1 \rightarrow E_2$ é tal que $\tau \mapsto \tau$, $inic \mapsto inic$, $flip \mapsto flip$, $flop \mapsto \tau$ e $fim \mapsto fim$. □

Definição 3.20 Funtor de Encapsulação. Considere o autômato $A: M \rightarrow E_1$, o morfismo de encapsulação $caps: E_1 \rightarrow E_2$, a cofibração $u: \mathcal{Aut} \rightarrow \mathcal{Rgr}_1$, a construção dos correspondentes morfismos cocartesianos como na prova da Proposição 3.16 e as fibras $u^{-1}E_1, u^{-1}E_2$. Então, $caps: u^{-1}E_1 \rightarrow u^{-1}E_2$ é o funtor de encapsulação induzido por u e $caps$ e o autômato resultante da encapsulação de A , denotado por $A \setminus caps$, é $capsA$. □

Portanto, para um morfismo de encapsulação $caps: E_1 \rightarrow E_2$ e um autômato $A = \langle M, E_1, etiq \rangle$, $capsA = \langle M, E_2, caps \circ etiq \rangle$

Exemplo 3.21 Considere a Figura 3.18. Suponha a encapsulação do autômato Flip-Flop: $M \rightarrow E_1$ de acordo com o morfismo $caps: E_1 \rightarrow E_2$ do exemplo anterior. Então, a transição etiquetada por $flop$ é encapsulada. □

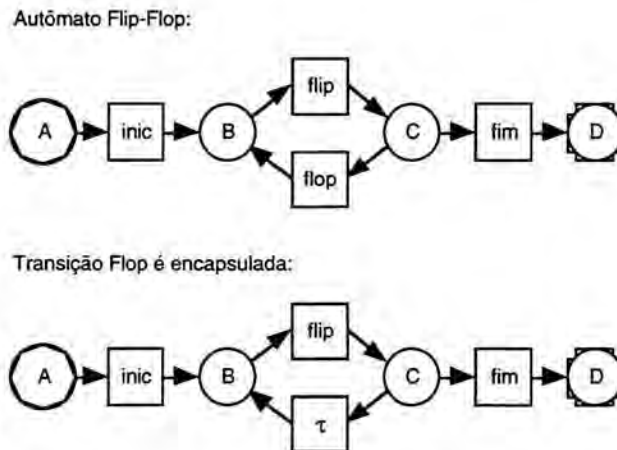


Figura 3.18 Encapsulação de um autômato

4 Reificação de Autômatos Sequenciais

Uma reificação implementa um autômato sobre as computações de um outro autômato. É definida como um morfismo especial de autômatos onde o objeto destino (mais concreto) é enriquecido com o seu fecho computacional (fecho transitivo nas transições) representando todas as computações (seqüenciais) possíveis que podem ser obtidas a partir das permutações das transições componentes, respeitando os estados de origem e destino. Um fecho computacional é facilmente obtido no contexto categorial pois, um grafo reflexivo estendido com uma operação de composição nos arcos determina uma categoria. Assim, a substituição dos grafos reflexivos por categorias na definição de autômatos seqüenciais, define uma nova estrutura denominada categoria computacional (seqüencial). Categorias computacionais e os correspondentes morfismo determinam a categoria denominada *Comp*. O funtor esquecimento $ca: Comp \rightarrow \mathcal{A}ut$ que esquece sobre a operação de composição, possui adjunto esquerdo $ac: \mathcal{A}ut \rightarrow Comp$ o qual gera livremente a composição de transições. Assim, o funtor fecho computacional é obtido pela composição dos funtores, ou seja, $fc = ca \circ ac: \mathcal{A}ut \rightarrow \mathcal{A}ut$.

Autômatos e reificações constituem a categoria *ReifAut*, cuja definição é induzida pela adjunção $\langle ac, ca \rangle: \mathcal{A}ut \rightarrow Comp$, baseada nas categorias Kleisli. Portanto, a composicionalidade vertical é satisfeita. Entretanto, esta construção não satisfaz à composicionalidade horizontal pois, em geral, o fecho computacional do produto de autômatos é diferente do produto dos fechos computacionais dos autômatos componentes e, portanto, a composição paralela de dois morfismos de reificação não é, em geral, um morfismo de reificação.

No contexto dos autômatos seqüenciais, é considerada a categoria das categorias pequenas cujos nodos distinguidos são pares de conjuntos, isto é, $DCat = \mathbb{I} \downarrow ob$ onde $\mathbb{I}: Set^2 \rightarrow Set$ é o funtor canonicamente induzido pelo coproduto categorial em *Set* e $ob: Cat \rightarrow Set$ é o funtor esquecimento que associa a cada categoria o seu correspondente conjunto de objetos. No texto que segue são usados conceitos e resultados referentes às mônadas e às categorias Kleisli, os quais são introduzidos em apêndice. As principais referências usadas são [Mac Lane 71], [Barr & Wells 90] e [Asperti & Longo 91]. Adicionalmente, a operação de composição de uma categoria também é representada pelo símbolo ";" e, neste caso, a composição de morfismos é denotada em ordem inversa. Por exemplo, $f;g$ representa $g \circ f$. Esta notação é usada para enfatizar que a composição categorial representa a composição seqüencial de computações.

4.1 Computações

A categoria das categorias computacionais é uma extensão da categoria dos autômatos onde os grafos reflexivos, grafos reflexivos de um só nodo e grafos reflexivos com nodos distinguidos são estendidos com a operação de composição nos arcos, resultando em categorias, categorias com um único objeto e categorias com objetos distinguidos, respectivamente. Ou seja, uma categoria computacional é, basicamente, um autômato seqüencial estendido com uma operação de composição nas transições. Note-se que, qualquer monóide pode ser visto como uma categoria com um único objeto onde a composição é a operação do monóide (veja, por exemplo, [Barr & Wells 90, p. 23]). De fato, a categoria *Mon* dos monóides é isomorfa à categoria das categorias (pequenas) com um único objeto, denotada por Cat_1 .

Definição 4.1 **Categoria das Categorias Computacionais.** A categoria das categorias computacionais, denotada por *Comp*, é a categoria das setas $id \downarrow inc$, onde:

- a) $id: DCat \rightarrow DCat$ é o funtor identidade;
- b) $inc: Cat_1 \rightarrow DCat$ é o funtor inclusão tal que, para cada Cat_1 -objeto \mathcal{E} , $inc\mathcal{E} = \langle \mathcal{E}, 1, 1, !_{1+1} \rangle$ onde a função total $!_{1+1}: 1+1 \rightarrow 1$ é única e, para cada Cat_1 -funtor $h: \mathcal{E}_1 \rightarrow \mathcal{E}_2$, $inc h = \langle h, !_1, !_1 \rangle: \langle \mathcal{E}_1, 1, 1, !_{1+1} \rangle \rightarrow \langle \mathcal{E}_2, 1, 1, !_{1+1} \rangle$ onde a função total $!_1: 1 \rightarrow 1$ é única. \square

Portanto, uma categoria computacional é uma tripla $\mathcal{A} = \langle \mathcal{M}, \mathcal{E}, etiq \rangle$, tal que \mathcal{M} é um $DCat$ -objeto, \mathcal{E} é um Cat_1 -objeto e $etiq: id_{\mathcal{M}} \rightarrow inc\mathcal{E}$ é um $DCat$ -morfismo.

As categorias Aut e $Comp$ são relacionadas por uma adjunção $\langle ac, ca \rangle: Aut \rightarrow Comp$ onde o funtor ac enriquece um dado autômato com uma operação de composição nas transições e o funtor ca esquece sobre a composição.

Definição 4.2 Funtor ca . O funtor esquecimento $ca: Comp \rightarrow Aut$ é tal que, para cada $Comp$ -objeto $\mathcal{A} = \langle \mathcal{M}, \mathcal{E}, etiq \rangle$ onde $\mathcal{M} = \langle C, l, F, i+f \rangle$, $C = \langle Ob_C, Mor_C, \partial_0, \partial_1, l, \circ \rangle$ e $\mathcal{E} = \langle 1, Mor_{\mathcal{E}}, !, !, l, \circ \rangle$, tem-se que

- para o $DCat$ -objeto \mathcal{M} , $ca\mathcal{M}$ é o $DRGr$ -objeto $M = \langle C, l, F, i+f \rangle$ onde $C = \langle Ob_C, Mor_C, \partial_0, \partial_1, l \rangle$;
- para o Cat_1 -objeto \mathcal{E} , $ca\mathcal{E}$ é o RGr_1 -objeto $E = \langle 1, Mor_{\mathcal{E}}, !, !, l \rangle$;
- para o $Comp$ -objeto \mathcal{A} , $ca\mathcal{A}$ é o Aut -objeto $A = \langle M, E, etiq \rangle$ onde $etiq: M \rightarrow E$ é o $DRGr$ -morfismo canonicamente induzido por $etiq: \mathcal{M} \rightarrow \mathcal{E}$.

E, para cada $Comp$ -morfismo $h = \langle h_{\mathcal{M}}, h_{\mathcal{E}} \rangle: \mathcal{A}_1 \rightarrow \mathcal{A}_2$ onde $h_{\mathcal{M}} = \langle h_V, h_T, h_l, h_F \rangle: \mathcal{M}_1 \rightarrow \mathcal{M}_2$ e $h_{\mathcal{E}} = \langle !, h_E \rangle: \mathcal{E}_1 \rightarrow \mathcal{E}_2$, tem-se que $ca h_{\mathcal{M}}$ é o $DRGr$ -morfismo $h_M = \langle h_V, h_T, h_l, h_F \rangle: M_1 \rightarrow M_2$, $ca h_{\mathcal{E}}$ é o RGr_1 -morfismo $h_E = \langle !, h_E \rangle: E_1 \rightarrow E_2$ e $ca h$ é o Aut -morfismo $h = \langle h_M, h_E \rangle: A_1 \rightarrow A_2$. \square

Definição 4.3 Funtor ac . O funtor $ac: Aut \rightarrow Comp$ é tal que, para cada Aut -objeto $A = \langle M, E, etiq \rangle$ onde $M = \langle G, l, F, i+f \rangle$, $G = \langle V, T, \partial_0, \partial_1, l \rangle$ e $E = \langle 1, L, !, !, l \rangle$ tem-se que:

- acM é o $DCat$ -objeto $\mathcal{M} = \langle G, l, F, i+f \rangle$ onde $G = \langle V, T^c, \partial_0^c, \partial_1^c, l^c, ; \rangle$ sendo $l^c: V \rightarrow T^c$ canonicamente induzido por $l: V \rightarrow T$ e sendo $T^c, \partial_0^c, \partial_1^c$ e a operação parcial de composição $_;: T^c \times T^c \rightarrow T^c$ definidos pelas seguintes regras de inferência:

$$\frac{t: A \rightarrow B \in T}{t: A \rightarrow B \in T^c}$$

$$\frac{t: A \rightarrow B \in T^c \quad u: B \rightarrow C \in T^c}{t;u: A \rightarrow C \in T^c}$$

sujeitas às seguintes regras equacionais:

$$\frac{t: A \rightarrow B \in T^c}{l_A; t = t \quad \& !_A = t}$$

$$\frac{t: A \rightarrow B \in T^c \quad u: B \rightarrow C \in T^c \quad v: C \rightarrow D \in T^c}{t;(u;v) = (t;u);v}$$

- acE é o Cat_1 -objeto $\mathcal{E} = \langle 1, L^c, !, !, !_1, ; \rangle$ onde L^c e a operação de composição são definidos como acima;
- acA é o $Comp$ -objeto $\mathcal{A} = \langle \mathcal{M}, \mathcal{E}, etiq \rangle$ onde $etiq = \langle !_V, etiq_T^c, !_l, !_F \rangle: \mathcal{M} \rightarrow \mathcal{E}$ é o $DCat$ -morfismo indutivamente definido a partir de $etiq = \langle !_V, etiq_T, !_l, !_F \rangle: M \rightarrow E$, tal que:
 - para t em T , $etiq_T^c(t) = etiq_T(t)$
 - para $t;u$ em T^c , $etiq_T^c(t;u) = etiq_T^c(t); etiq_T^c(u)$

E, para cada Aut -morfismo $h = \langle h_M, h_E \rangle: A_1 \rightarrow A_2$ onde $h_M = \langle h_V, h_T, h_l, h_F \rangle: M_1 \rightarrow M_2$ é um $DRGr$ -morfismo e $h_E = \langle !, h_E \rangle: E_1 \rightarrow E_2$ é um RGr_1 -morfismo, tem-se que:

- ach_M é o $DCat$ -morfismo $h_{\mathcal{M}} = \langle h_V, h_T^c, h_l, h_F \rangle: \mathcal{M}_1 \rightarrow \mathcal{M}_2$ onde h_T^c é definido como acima;
- ach_E é o Cat_1 -morfismo $h_{\mathcal{E}} = \langle !, h_E^c \rangle: \mathcal{E}_1 \rightarrow \mathcal{E}_2$ onde h_E^c é definido como acima;
- ach é o $Comp$ -morfismo $h = \langle h_{\mathcal{M}}, h_{\mathcal{E}} \rangle: \mathcal{A}_1 \rightarrow \mathcal{A}_2$. \square

Proposição 4.4 O funtor ac é adjunto esquerdo de ca .

Prova:

- a) Considere a transformação natural $\eta = \langle \eta_M, \eta_E \rangle: id_{Aut} \rightarrow ca \circ ac$ a qual é uma inclusão nos arcos de cada grafo componente. Então, para cada autômato $A = \langle M, E, etiq \rangle$, para cada categoria computacional $\mathcal{A} = \langle \mathcal{M}, \mathcal{E}, etiq \rangle$, para cada Aut -morfismo $f: A \rightarrow ca\mathcal{A}$, existe um único $Comp$ -morfismo $g: acA \rightarrow \mathcal{A}$ tal que $f = cag \circ \eta_A$. De fato, g é acf restrito ao objeto destino \mathcal{A} .
- b) De forma dual, $\varepsilon = \langle \varepsilon_M, \varepsilon_E \rangle: ac \circ ca \rightarrow id_{Comp}$ é a transformação natural que associa a cada transição composta $\langle l \rangle; \langle u \rangle$ a transição $\langle t; u \rangle$ e, a cada etiqueta composta $\langle a \rangle; \langle b \rangle$ a etiqueta $\langle a; b \rangle$. Portanto, $\langle ac, ca, \eta, \varepsilon \rangle: Aut \rightarrow Comp$ é uma adjunção. \square

O enriquecimento de um autômato com as transições que refletem todas as suas computações (seqüenciais) possíveis, é realizado por uma operação funtorial denominada fecho computacional, determinada pela composição dos funtores ac e ca definidos acima.

Definição 4.5 Fecho Computacional. O fecho computacional é a operação funtorial $fc: \mathcal{A}ut \rightarrow \mathcal{A}ut$ onde $fc = ca \circ ac$. \square

Exemplo 4.6 Na Figura 4.1, é ilustrada a representação gráfica de um autômato (esquerda) e o seu correspondente fecho computacional (direita), onde as transições incluídas pelo fecho estão representadas com um traço diferenciado. Considerando que as transições etiquetadas b e c constituem um ciclo (não atômico) no autômato original, o correspondente fecho computacional possui infinitas transições as quais são representadas usando o expoente i o qual assume valores em $\{1, 2, 3, \dots\}$. Por exemplo, $a;(b;c)^i$ representa as transições $a;b;c$, $a;b;c;b;c$, $a;b;c;b;c;b;c$...

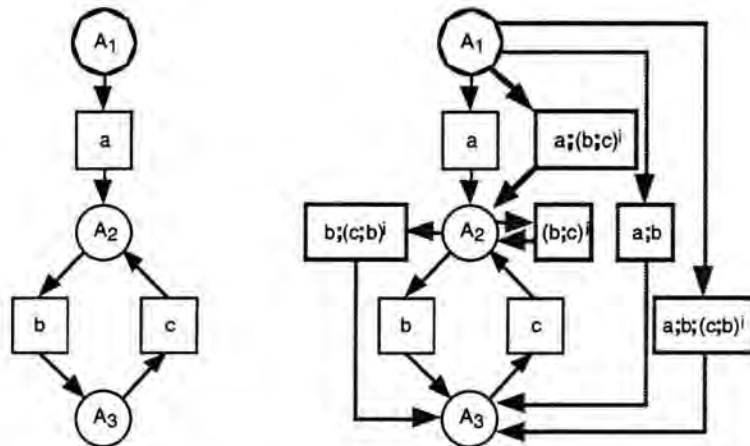


Figura 4.1 Autômato seqüencial e o seu correspondente fecho computacional

4.2 Composicionalidade Vertical

A adjunção $\langle ac, ca, \eta, \epsilon \rangle: \mathcal{A}ut \rightarrow \mathcal{C}omp$ determina uma mônada $T = \langle fc, \eta, \mu \rangle$ tal que fc é o endofuntor $ca \circ ac: \mathcal{A}ut \rightarrow \mathcal{A}ut$ e μ é a transformação natural $ca \epsilon ac: fc^2 \rightarrow fc$, onde $ca: ca \rightarrow ca$ e $ac: ac \rightarrow ac$ são transformações naturais identidade e $ca \epsilon ac$ é a composição horizontal de transformações naturais, como ilustrado na Figura 4.2. A mônada T é útil na compreensão das computações de um autômato. Assim, para cada autômato A :

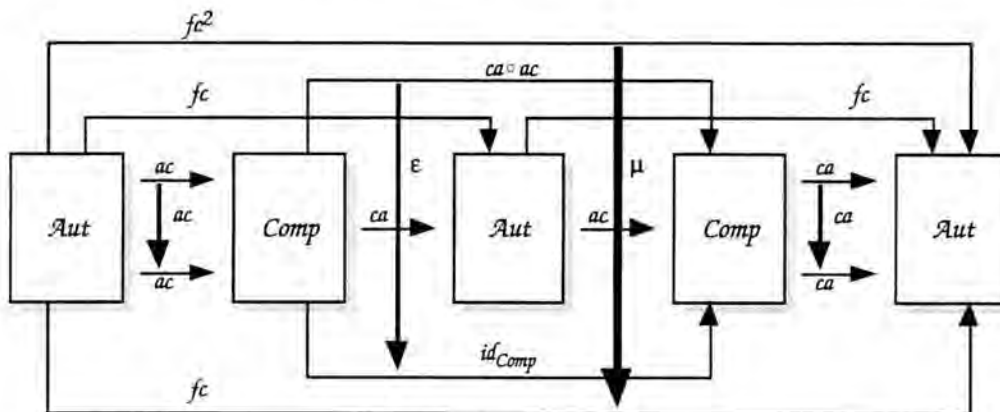


Figura 4.2 Composição horizontal de transformações naturais

- fcA resulta no fecho computacional de A , ou seja, no enriquecimento de A com as transições que refletem todas as suas computações possíveis;
- $\eta_A: id_{Aut}A \rightarrow fcA$ é a inclusão de A no seu correspondente fecho computacional;
- $\mu_A: fc^2A \rightarrow fcA$ é a operação denominada achatamento, a qual especifica como duas aplicações sucessivas do fecho computacional ao autômato A é instanciada em uma única aplicação do fecho computacional.

Na definição da reificação como um morfismo de autômatos onde o objeto destino é enriquecido com as suas computações, dois problemas necessitam ser resolvidos:

- Etiquetação. Transições distintas mas com mesmas etiquetas podem ser implementadas em computações distintas e com etiquetação também distintas. Ou seja, um morfismo de reificação não necessariamente deve preservar a etiquetação.
- Operação de composição. Dados duas reificações $\varphi: A \rightarrow fcB$ e $\psi: B \rightarrow fcC$ é necessário definir a reificação composta $\psi \circ \varphi: A \rightarrow fcC$. Note-se que a origem de ψ é diferente do destino de φ .

Uma solução natural para o problema da composição seria definir os autômatos e reificações como uma categoria Kleisli determinada pela mônada $T = \langle fc, \eta, \mu \rangle$. Entretanto, neste caso, as reificações obrigatoriamente preservariam as etiquetações. Assim, a solução proposta a seguir é baseada nas categorias Kleisli mas exclusivamente para a forma dos autômatos. Na categoria resultante, os morfismos representam reificações de autômatos e a operação de composição garante a composicionalidade vertical.

Por simplicidade, nos diagramas que seguem, os funtores $id: DRGr \rightarrow DRGr$ e $inc: RGr1 \rightarrow DRGr$ usados na definição da categoria Aut são omitidos. Por exemplo, em diagramas, $eti: M \rightarrow E$ denota $eti: idM \rightarrow incE$.

Definição 4.7 Categoria dos Autômatos e Reificações. Seja $T = \langle fc, \eta, \mu \rangle$ onde $\eta = \langle \eta_M, \eta_E \rangle$ e $\mu = \langle \mu_M, \mu_E \rangle$ a mônada determinada pela adjunção $\langle ac, ca, \eta, \epsilon \rangle: Aut \rightarrow Comp$. A categoria dos autômatos e reificações, denotada por $ReifAut$, é tal que (suponha os Aut -objetos $A_k = \langle M_k, E_k, eti_k \rangle$, para k com valores em $\{1, 2, 3\}$):

- os objetos são os mesmo de Aut ;
- os morfismos são construídos como segue:
 - $\varphi = \varphi_M: A_1 \rightarrow A_2$ é um $ReifAut$ -morfismo onde $\varphi_M: M_1 \rightarrow fcM_2$ é um $DRGr$ -morfismo;
 - para cada Aut -objeto A , $\varphi = \eta_M: A \rightarrow A$ é o $ReifAut$ -morfismo identidade de A ;
- sejam $\varphi: A_1 \rightarrow A_2$, $\psi: A_2 \rightarrow A_3$ $ReifAut$ -morfismos. A composição $\psi \circ \varphi$ é o morfismo $\psi_M \circ \chi \varphi_M: A_1 \rightarrow A_3$ onde $\psi_M \circ \chi \varphi_M$ é determinado como ilustrado na Figura 4.3. \square

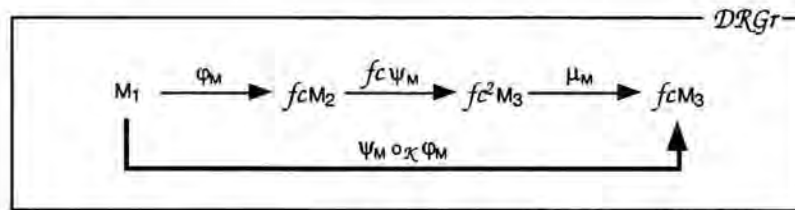


Figura 4.3 Composição de reificações é a composição na categoria Kleisli restrita à forma dos autômatos

Portanto, os morfismos e a composição de morfismos em $ReifAut$ são os morfismos e a composição na categoria Kleisli referente à forma dos autômatos, induzida pela mônada T . Com o objetivo de distinguir os $ReifAut$ -morfismos dos Aut -morfismo uma reificação φ do autômato A nas computações de B é denotada por $\varphi: A \rightarrow fcB$ ou $\varphi: A \Rightarrow B$.

Exemplo 4.8 A Figura 4.4 ilustra um morfismo de reificação. Repare que o morfismo não preserva as etiquetas das transições y do autômato origem. \square

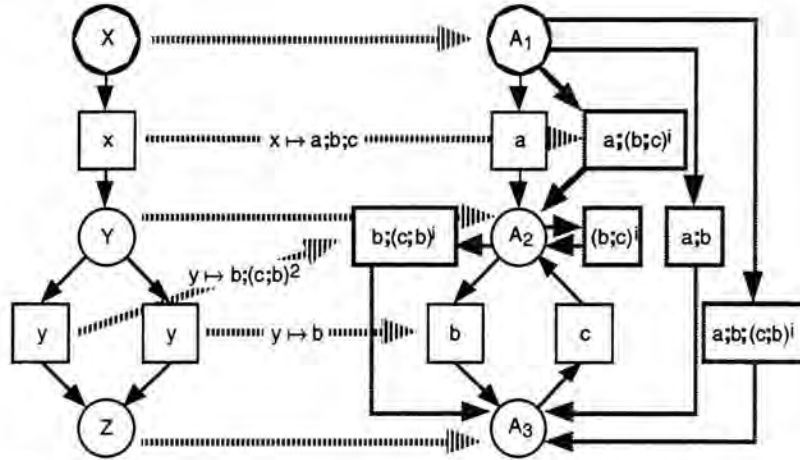


Figura 4.4 Reificação de autômatos não necessariamente preserva a etiquetação

Como $\mathcal{ReifAut}$ -morfismos são morfismos exclusivamente na forma dos autômatos, estes não são \mathcal{Aut} -morfismos. Entretanto, um morfismo na forma pode induzir um morfismo nas etiquetas determinando uma categoria de autômatos e reificações cujas reificações são \mathcal{Aut} -morfismos, como na categoria introduzida a seguir.

Definição 4.9 *Categoria dos Autômatos e Reificações com Etiquetação Induzida.* Seja $\mathcal{T} = \langle \mathcal{f}, \eta, \mu \rangle$ onde $\eta = \langle \eta_M, \eta_E \rangle$ e $\mu = \langle \mu_M, \mu_E \rangle$ a mônada determinada pela adjunção $\langle \mathcal{a}, \mathcal{c}, \eta, \varepsilon \rangle: \mathcal{Aut} \rightarrow \mathcal{Comp}$. A categoria dos autômatos e reificações com etiquetação induzida, denotada por $\mathcal{ReifAut}_{\mathcal{E}}$, é tal que (suponha os \mathcal{Aut} -objetos $A_k = \langle M_k, E_k, \text{eti}_k \rangle$, para k com valores em $\{1, 2, 3\}$):

- a) os objetos são os mesmo de \mathcal{Aut} ;
- b) os morfismos são construídos como segue:
 - b.1) seja $\varphi_M: M_1 \rightarrow \mathcal{f}cM_2$ um \mathcal{DRGr} -morfismo. Então $\varphi = \langle \varphi_M, \varphi_E \rangle: A_1 \rightarrow A_2$ é um $\mathcal{ReifAut}$ -morfismo onde φ_E é determinado pela soma amalgamada ilustrada na Figura 4.5;
 - b.2) para cada \mathcal{Aut} -objeto A , $\varphi = \langle \eta_M, \varphi_E \rangle: A \rightarrow A$ é o $\mathcal{ReifAut}$ -morfismo identidade de A onde φ_E é como definido no item anterior;
- c) sejam $\varphi: A_1 \rightarrow A_2, \psi: A_2 \rightarrow A_3$ $\mathcal{ReifAut}$ -morfismos. A composição $\psi \circ \varphi$ é o morfismo $\langle \psi_M \circ \varphi_M, \psi_E \circ \varphi_E \rangle: A_1 \rightarrow A_3$ onde $\psi_M \circ \varphi_M$ e $\psi_E \circ \varphi_E$ são determinados como ilustrado na Figura 4.6. \square

É fácil verificar que o objeto resultante da soma amalgamada na definição acima é, de fato, um grafo de etiquetas. Como é mostrado na proposição a seguir, $\mathcal{ReifAut}$ e $\mathcal{ReifAut}_{\mathcal{E}}$ são isomorfas. Portanto, todo morfismo de reificação pode ser visto como um morfismo especial de autômatos.

Proposição 4.10 As categorias $\mathcal{ReifAut}$ e $\mathcal{ReifAut}_{\mathcal{E}}$ são isomorfas.

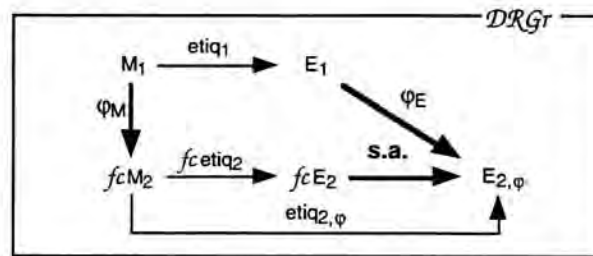


Figura 4.5 Reificação de autômatos com etiquetação induzida

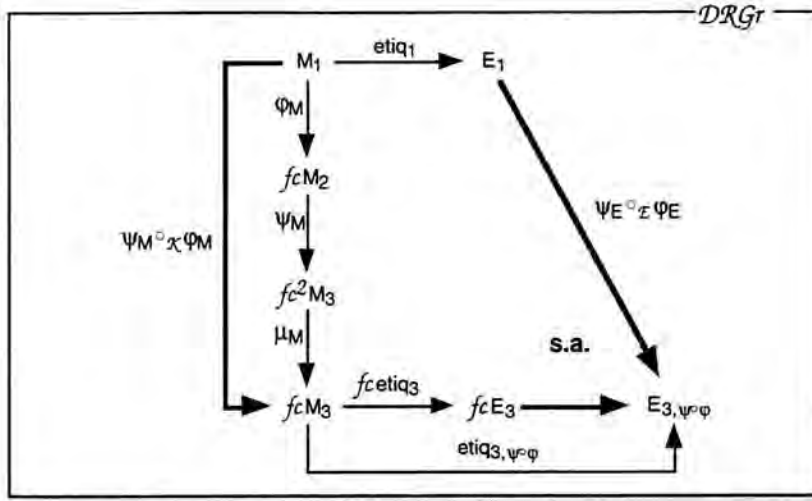


Figura 4.6 Composição de reificações de autômatos com etiquetação induzida

Prova: Considere:

- o funtor $re: ReifAut \rightarrow ReifAut_{\mathcal{E}}$ tal que, para todo o $ReifAut$ -objeto A , $reA = A$ e, para todo o $ReifAut$ -morfismo $\varphi = \langle \varphi_M, \varphi_E \rangle: A_1 \rightarrow A_2$, $re\varphi = \langle \varphi_M, \varphi_E \rangle$;
- o funtor $er: ReifAut_{\mathcal{E}} \rightarrow ReifAut$ tal que, para todo o $ReifAut_{\mathcal{E}}$ -objeto A , $erA = A$ e, para todo o $ReifAut_{\mathcal{E}}$ -morfismo $\varphi = \langle \varphi_M, \varphi_E \rangle: A_1 \rightarrow A_2$, $er\varphi = \varphi_M$.

É fácil verificar que $er \circ re = id_{ReifAut}$ e $re \circ er = id_{ReifAut_{\mathcal{E}}}$. □

Como $ReifAut$ e $ReifAut_{\mathcal{E}}$ são isomorfas, no texto que segue, $ReifAut$ denota qualquer das duas categorias.

4.3 Composicionalidade Horizontal

Embora a composicionalidade vertical seja satisfeita, a construção proposta não satisfaz à composicionalidade horizontal. A proposição a seguir mostra que o funtor fecho computacional não preserva limites.

Proposição 4.11 O endofuntor $fc: Aut \rightarrow Aut$ não preserva limites.

Prova: Para provar que fc não preserva limites, é suficiente mostrar um contra-exemplo. Considere os autômatos A_1, A_2 e o produto $A_1 \times A_2$ ilustrado na Figura 4.7. As representações

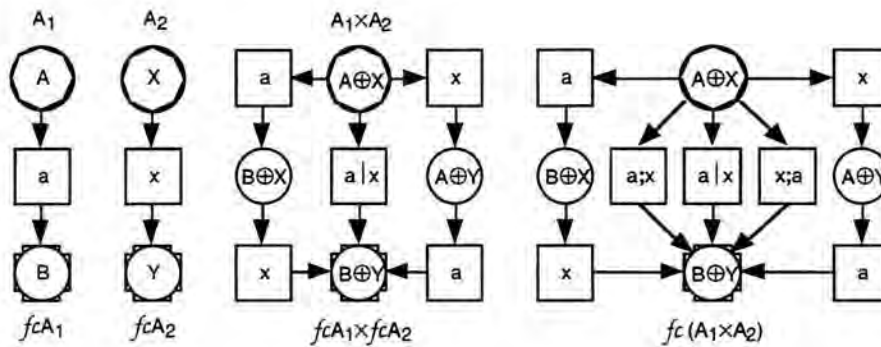


Figura 4.7 O funtor fecho computacional não preserva limites

gráficas referentes a A_1 , A_2 e $A_1 \times A_2$ também representam $fc A_1$, $fc A_2$ e $fc A_1 \times fc A_2$, respectivamente. Entretanto, $fc A_1 \times fc A_2$ não representa $fc(A_1 \times A_2)$. \square

Portanto, para as reificações $\varphi: A_1 \Rightarrow A_2$ e $\psi: B_1 \Rightarrow B_2$, o morfismo $\varphi \times \psi: A_1 \times B_1 \rightarrow fc A_2 \times fc B_2$ univocamente induzido pelo produto, como ilustrado na Figura 4.8, não é, necessariamente, um morfismo de $\mathcal{ReifAut}$, pois, em geral, $fc(A_2 \times B_2)$ e $fc A_2 \times fc B_2$ não são isomorfos.

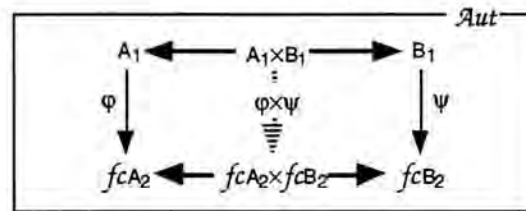


Figura 4.8 A composição paralela de reificações não é, em geral, uma reificação

5 Grafos e Categorias Internos

Um grafo $G = \langle V, T, \partial_0, \partial_1 \rangle$ pode ser considerado como um diagrama na categoria Set onde V, T são conjuntos e ∂_0, ∂_1 são funções totais. Adicionalmente, os morfismos de grafos são diagramas comutativos na categoria base Set . Portanto, pode-se afirmar que grafos são definidos internamente à Set e, analogamente, o mesmo pode ser afirmado para os grafos reflexivos e categorias pequenas. A generalização desta idéia, usando como base uma categoria arbitrária qualquer é conhecida como internalização (veja, por exemplo, [Corradini 90] e [Asperti & Longo 91]).

No texto que segue são usados conceitos e resultados referentes às categorias concretas, os quais são introduzidos em apêndice. A referência usada é [Adámek *et al* 90].

5.1 Grafo Interno

A categoria dos grafos internos herda, da categoria base, as propriedades referentes aos limites e colimites.

Definição 5.1 Categoria dos Grafos Internos. Considere a categoria C e o funtor diagonal $\Delta: C \rightarrow C^2$. A categoria dos grafos internos, denotada por $\mathcal{Gr}(C)$, é a categoria das setas $\Delta \downarrow \Delta$. \square

Portanto, um grafo interno à C pode ser visto como uma quádrupla $G = \langle V, T, \partial_0, \partial_1 \rangle$ onde V, T são C -objetos e $\partial_0, \partial_1: T \rightarrow V$ são C -morfismos. Para os grafos internos $G_1 = \langle V_1, T, \partial_{01}, \partial_{11} \rangle$ e $G_2 = \langle V_2, T, \partial_{02}, \partial_{12} \rangle$, um morfismo de grafos internos $h: G_1 \rightarrow G_2$ é um par de C -morfismos $h = \langle h_V: V_1 \rightarrow V_2, h_T: T_1 \rightarrow T_2 \rangle$ tal que o diagrama ilustrado na Figura 5.1 comuta.

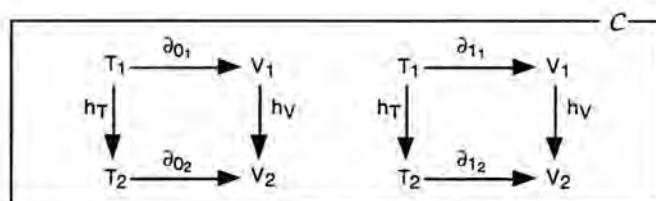


Figura 5.1 Diagramas comutativos para morfismos de grafos internos

Proposição 5.2 Considere a categoria C :

- se C é completa, então $\mathcal{Gr}(C)$ é completa;
- se C é cocompleta, então $\mathcal{Gr}(C)$ é cocompleta.

Prova: Como $\mathcal{Gr}(C)$ é a categoria das setas $\Delta \downarrow \Delta$ e como Δ preserva limites e colimites $\mathcal{Gr}(C)$ é bicompleta. \square

5.2 Grafo Reflexivo Interno

A categoria dos grafos reflexivos internos é uma extensão da categoria dos grafos internos, onde o morfismo identidade de cada objeto também é um morfismo da categoria base.

Analogamente aos grafos internos, as propriedades referentes aos limites e colimites são herdadas da categoria base.

Grafo Reflexivo Interno

Definição 5.3 Categoria dos Grafos Reflexivos Internos. Considere a categoria C

- Um grafo reflexivo interno é uma quintupla $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ onde $\langle V, T, \partial_0, \partial_1 \rangle$ é um grafo interno e $\iota: V \rightarrow T$ é um C -morfismo tal que o diagrama ilustrado na Figura 5.2 (esquerda) comuta.
- Um morfismo de grafos reflexivos internos é um morfismo de grafos internos tal que o diagrama ilustrado na Figura 5.2 (direita) comuta.
- Grafos reflexivos internos e os correspondentes morfismos constituem a categoria denotada por $\mathcal{RGr}(C)$. \square

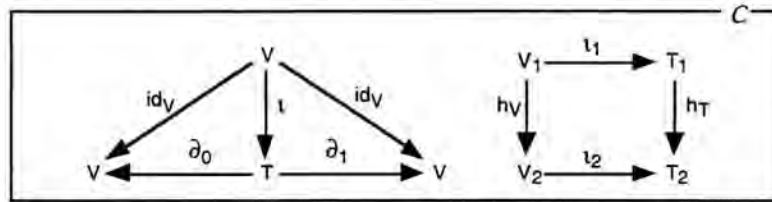


Figura 5.2 Diagramas comutativos para a categoria dos grafos reflexivos internos

Proposição 5.4 Considere a categoria C .

- se C é completa, então $\mathcal{RGr}(C)$ é completa;
- se C é cocompleta, então $\mathcal{RGr}(C)$ é cocompleta.

Prova: Seja $u: \mathcal{RGr}(C) \rightarrow \mathcal{Gr}(C)$ o funtor esquecimento tal que, para todo $\mathcal{RGr}(C)$ -objeto, $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$, $uG = \langle V, T, \partial_0, \partial_1 \rangle$ e, para todo $\mathcal{RGr}(C)$ -morfismo $h = \langle h_V, h_T \rangle: G_1 \rightarrow G_2$, $uh = \langle h_V, h_T \rangle: uG_1 \rightarrow uG_2$. Como u é um funtor fidedigno, $(\mathcal{RGr}(C), u)$ é categoria concreta sobre $\mathcal{Gr}(C)$. Adicionalmente, se C é bicompleta, então $\mathcal{Gr}(C)$ é bicompleta. Portanto, para provar que $\mathcal{RGr}(C)$ é bicompleta, é suficiente provar que, para cada diagrama de $\mathcal{RGr}(C)$, o limite (colimite) do diagrama em $\mathcal{Gr}(C)$ pode ser elevado como uma fonte inicial (sumidouro final) em $\mathcal{RGr}(C)$. Suponha I um conjunto arbitrário de índices, i com valores em I e k com valores em $\{0, 1\}$. Por simplicidade, no texto que segue, a expressão $i \in I$ é omitida.

- Elevação de produtos. Considere um conjunto indexado e arbitrário de $\mathcal{RGr}(C)$ -objetos $\{G_i = \langle V_i, T_i, \partial_{0i}, \partial_{1i}, \iota_i \rangle\}$ e o correspondente $\mathcal{Gr}(C)$ -produto $\times uG_i = \langle \times V_i, \times T_i, \times \partial_{0i}, \times \partial_{1i} \rangle$, juntamente com o conjunto indexado de morfismos $\{\pi_i: \times uG_i \rightarrow uG_i\}$. Então, $\times G_i = \langle \times V_i, \times T_i, \times \partial_{0i}, \times \partial_{1i}, \times \iota_i \rangle$, juntamente com o conjunto de morfismos $\{\pi_i: \times G_i \rightarrow G_i\}$, é fonte inicial de $\{G_i\}$ onde $\times \iota_i$ é univocamente induzido pelo produto categorial, como ilustrado na Figura 5.3.

Para provar que $\times G_i$ é um grafo reflexivo interno, é suficiente provar que o diagrama externo da Figura 5.3 comuta. De fato, $\times id_{V_i}$ é o único C -morfismo tal que $\textcircled{4}$ comuta. Adicionalmente, como $\textcircled{1}$, $\textcircled{2}$ e $\textcircled{3}$ comutam, $\pi_{V_i} \circ \times id_{V_i} = id_{V_i} \circ \pi_{V_i} = \partial_{k_i} \circ \iota_i \circ \pi_{V_i} = \partial_{k_i} \circ \pi_{T_i} \circ \times \iota_i = \pi_{V_i} \circ \times \partial_{k_i} \circ \times \iota_i$. Logo, pela unicidade de $\times id_{V_i}$ em $\textcircled{4}$, $\times id_{V_i} = \times \partial_{k_i} \circ \times \iota_i$.

Para verificar que $\times G_i$ (juntamente com $\{\pi_i\}$) é fonte inicial, considere a $\mathcal{RGr}(C)$ -fonte $\langle G, \{f_i: G \rightarrow G_i\} \rangle$. Como $\times uG_i$ é produto em $\mathcal{Gr}(C)$, então existe um único $\mathcal{Gr}(C)$ -morfismo $h: uG \rightarrow \times uG_i$ tal que, $f_i = \pi_{uG_i} \circ h$. A elevação de h é o próprio h .

- Elevação de igualadores. Considere os $\mathcal{RGr}(C)$ -objetos $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11}, \iota_1 \rangle$, $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12}, \iota_2 \rangle$, os $\mathcal{RGr}(C)$ -morfismos paralelos $f_1 = \langle v_1, t_1 \rangle$, $f_2 = \langle v_2, t_2 \rangle: G_1 \rightarrow G_2$ e o correspondente $\mathcal{Gr}(C)$ -igualador $\langle e_V, e_T \rangle: \langle V, T, \partial_0, \partial_1 \rangle \rightarrow uG_1$. Então $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ juntamente com o morfismo $e = \langle e_V, e_T \rangle: G \rightarrow G_1$ é fonte inicial em $\mathcal{RGr}(C)$, onde ι é univocamente induzido pelos

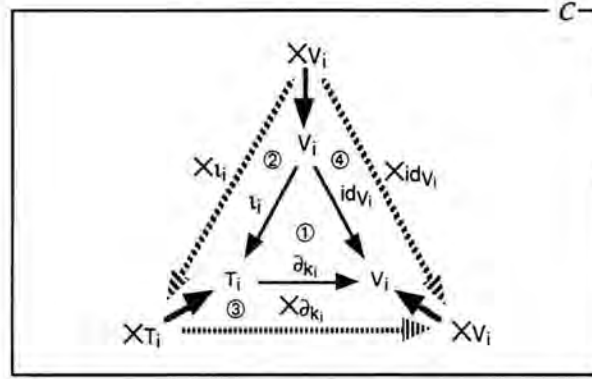


Figura 5.3 Morfismos univocamente induzidos pelo produto categorial

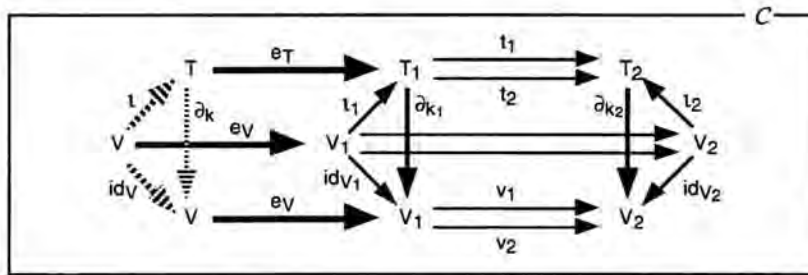


Figura 5.4 Morfismos univocamente induzidos pelos igualadores

C -igualadores, como ilustrado na Figura 5.4. A verificação de que G juntamente com e é uma fonte inicial, é análoga a realizada para produtos.

- c) Elevação de coprodutos. Considere um conjunto indexado e arbitrário de $\mathcal{RGr}(C)$ -objetos $\{G_i = \langle V_i, T_i, \partial_{0_i}, \partial_{1_i}, \iota_i \rangle\}$ e o correspondente $\mathcal{Gr}(C)$ -coproduto $+uG_i = \langle +V_i, +T_i, +\partial_{0_i}, +\partial_{1_i} \rangle$, juntamente com o conjunto indexado de morfismos $\{inj_i: uG_i \rightarrow +uG_i\}$. Então $+G_i = \langle +V_i, +T_i, +\partial_{0_i}, +\partial_{1_i}, +\iota_i \rangle$, juntamente com o conjunto de morfismos $\{inj_i: G_i \rightarrow +G_i\}$, é fonte inicial de $\{G_i\}$ onde $+\iota_i$ é univocamente induzido pelo coproduto categorial. Os detalhes da prova são análogos aos da prova para produtos arbitrários.
- d) Elevação de co-igualadores. Considere os $\mathcal{RGr}(C)$ -objetos $G_1 = \langle V_1, T_1, \partial_{0_1}, \partial_{1_1}, \iota_1 \rangle$, $G_2 = \langle V_2, T_2, \partial_{0_2}, \partial_{1_2}, \iota_2 \rangle$, os $\mathcal{RGr}(C)$ -morfismos paralelos $f_1 = \langle v_1, t_1 \rangle: G_1 \rightarrow G_2$ e o correspondente $\mathcal{Gr}(C)$ -igualador $\langle c_V, c_T \rangle: uG_1 \rightarrow \langle V, T, \partial_0, \partial_1 \rangle$. Então $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ juntamente com o morfismo $c = \langle c_V, c_T \rangle: G_1 \rightarrow G$ é fonte inicial em $\mathcal{RGr}(C)$, onde ι é univocamente induzido pelos C -co-igualadores. Os detalhes da prova são análogos aos da prova para igualadores. \square

Grafo Reflexivo Interno de um só Nodo

A categoria dos grafos reflexivos internos à C e com somente um nodo, denotada por $\mathcal{RGr}_1(C)$, é uma subcategoria plena de $\mathcal{RGr}(C)$. Se a categoria base C possui objeto zero, então $\mathcal{RGr}_1(C)$ é isomorfa a C .

Definição 5.5 Categoria dos Grafos Reflexivos Internos de um só Nodo. Considere a categoria C com objeto terminal.

- a) Um grafo reflexivo interno de um só nodo L é um grafo reflexivo interno $L = \langle 1, T, !, !, \iota \rangle$ tal que 1 é um objeto terminal fixo de C e $! : T \rightarrow 1$ é único.
- b) Considere os grafos internos $L_1 = \langle 1, T_1, !, !, \iota_1 \rangle$ e $L_2 = \langle 1, T_2, !, !, \iota_2 \rangle$. Um morfismo de grafos reflexivos internos de um só nodo é um morfismo de grafos reflexivos internos $h = \langle !, h_T \rangle : L_1 \rightarrow L_2$ onde $! : 1 \rightarrow 1$ é único.
- c) Grafos reflexivos internos de um só nodo e os correspondentes morfismos constituem a categoria $\mathcal{RGr}_1(C)$. □

Proposição 5.6 Considere a categoria C com objeto zero. Então, $\mathcal{RGr}_1(C)$ é isomorfa à C .

Prova: Como C possui objeto zero, então qualquer $\mathcal{RGr}_1(C)$ -grafo $L = \langle 1, T, !, !, \iota \rangle$, é tal que, 1 é simultaneamente objeto inicial e terminal e, portanto, $\iota = ! : 1 \rightarrow T$ é único. Considere o funtor esquecimento $c : \mathcal{RGr}_1(C) \rightarrow C$ tal que $c\langle 1, T, !, !, \iota \rangle = T$ e o funtor $g_1 : C \rightarrow \mathcal{RGr}_1(C)$ tal que $g_1 T = \langle 1, T, !, !, \iota \rangle$. É fácil verificar que $g_1 \circ c = id_{\mathcal{RGr}_1(C)}$ e $c \circ g_1 = id_C$. □

No caso da categoria C possuir objeto zero e considerando que $\mathcal{RGr}_1(C)$ e C são isomorfas, um grafo $\langle 1, T, !, !, \iota \rangle$ pode ser representado pelo correspondente C -objeto T , ou seja, $T = \langle 1, T, !, !, \iota \rangle$. O mesmo raciocínio é válido para um $\mathcal{RGr}_1(C)$ -morfismo $\langle !, h_T \rangle$, ou seja, $h_T = \langle !, h_T \rangle$.

5.3 Grafo Interno Com Nodos Distinguidos

A noção de internalização de grafos pode ser facilmente estendida para os nodos distinguidos, não só para a categoria de domínio dos nodos, como também para a categoria onde são instanciados no grafo. Nodos distinguidos são definidos de forma análoga para grafos internos e grafos reflexivos internos. Assim, no texto que segue, $\mathcal{G}(C)$ representa as categorias $\mathcal{Gr}(C)$ ou $\mathcal{RGr}(C)$ indistintamente.

Definição 5.7 Categoria dos Grafos Internos com Nodos Distinguidos. A categoria dos grafos internos com nodos distinguidos é a categoria das setas $d \downarrow \text{nodos}$, denotada por $\mathcal{G}_d(C)$, onde:

- a) $d : \mathcal{D} \rightarrow \mathcal{A}$ é um funtor onde \mathcal{D} é a categoria de domínio dos nodos distinguidos e \mathcal{A} é a categoria de instanciação dos nodos distinguidos no grafo;
- b) $\text{nodos} = c \circ v : \mathcal{G}(C) \rightarrow \mathcal{A}$ é um funtor tal que:
 - b.1) $v : \mathcal{G}(C) \rightarrow C$ é o funtor esquecimento que associa a cada grafo o correspondente C -objeto de nodos e a cada morfismo de grafos internos $h = \langle h_V, h_T \rangle$ o C -morfismo h_V ;
 - b.2) $c : C \rightarrow \mathcal{A}$ é um funtor que relaciona a categoria onde os grafos são considerados internamente com a categoria onde os nodos distinguidos são instanciados. □

Portanto, um grafo interno com nodos distinguidos M é uma tripla $M = \langle G, D, d \rangle$, onde G é um $\mathcal{G}(C)$ -grafo, D é um \mathcal{D} -objeto representando os nodos distinguidos e $d : dD \rightarrow \text{nodos}G$ é um \mathcal{A} -morfismo que instancia os nodos distinguidos em G .

Um morfismo de grafos com nodos distinguidos é um morfismo na categoria das setas $d \downarrow \text{nodos}$, ou seja, dados os grafos $M_1 = \langle G_1, D_1, d_1 \rangle$ e $M_2 = \langle G_2, D_2, d_2 \rangle$, um morfismo $h : M_1 \rightarrow M_2$ é uma tripla $h = \langle h_V, h_T, h_D \rangle$ onde $\langle h_V, h_T \rangle : G_1 \rightarrow G_2$ é um $\mathcal{G}(C)$ -morfismo e $h_D : D_1 \rightarrow D_2$ é um \mathcal{D} -morfismo, tal que o diagrama ilustrado na Figura 5.5 comuta.

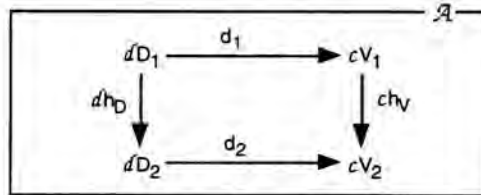


Figura 5.5 Diagrama comutativo para morfismos de grafos internos com nodos distinguidos

Proposição 5.8 Considere os funtores $d : \mathcal{D} \rightarrow \mathcal{A}$, $c : C \rightarrow \mathcal{A}$ e a categoria $\mathcal{G}_d(C)$:

- a) caso $G_d(C) = Gr_d(C)$: se C e \mathcal{D} são completas, c preserva limites e C possui objeto inicial, então $G_d(C)$ é completa;
- b) caso $G_d(C) = \mathcal{R}Gr_d(C)$: se C e \mathcal{D} são completas e c preserva limites, então $G_d(C)$ é completa;
- c) se C e \mathcal{D} são cocompletas e d preserva colimites, então $G_d(C)$ é cocompleta.

Prova: Como $G_d(C)$ é a categoria das setas $d \downarrow \text{nodos}$ onde $\text{nodos} = c \circ v$, é suficiente provar que o funtor $v: G(C) \rightarrow C$ preserva limites. Para $G_d(C) = Gr_d(C)$, considere o C -objeto inicial 0 e funtor $g: C \rightarrow Gr_d(C)$ tal que para o C -objeto V , gV é o grafo $\langle V, 0, !, ! \rangle$. Para $G_d(C) = \mathcal{R}Gr_d(C)$, considere funtor $g: C \rightarrow \mathcal{R}Gr_d(C)$ tal que para o C -objeto V , gV é o grafo $\langle V, V, id_V, id_V, id_V \rangle$. É fácil verificar que, nos dois casos, g é adjunto esquerdo de v . □

No contexto dos autômatos não-seqüenciais, são considerados somente os grafos reflexivos internos cujos nodos distinguidos são pares de objetos de C , interpretados como estados iniciais e finais. Por esta razão, a seguinte notação simplificada é adotada.

Notação 5.9 Categoria $\mathcal{DRGr}(C)$. Considere C uma categoria com todos os coprodutos binários. Então $\mathcal{DRGr}(C)$ denota a categoria dos grafos reflexivos internos cujos nodos distinguidos são objetos de C^2 , isto é, $\mathcal{DRGr}(C) = \Downarrow_C \downarrow \text{nodos}$ onde $\Downarrow_C: C^2 \rightarrow C$ é o funtor canonicamente induzido pelo coproduto categorial em C e $\text{nodos}: \mathcal{RGr}(C) \rightarrow C$ é o funtor esquecimento que associa a cada grafo o seu correspondente C -objeto de nodos. □

5.4 Categoria Interna

A definição das categorias internas é mais complexa comparativamente com a dos grafos internos, pois a operação de composição de uma categoria é uma função parcial. A composição é definida para $\{(f, g) \mid \partial_1(f) = \partial_0(g)\}$, o qual pode ser determinado pelo produto fibrado ilustrado na Figura 5.6. Seguindo esta idéia, a generalização do conceito de internalização para categorias requer uma categoria base finitamente completa.

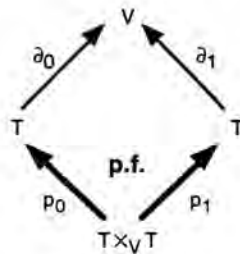


Figura 5.6 Domínio da operação de composição determinado por um produto fibrado

Definição 5.10 Categoria das Categorias Internas. Seja C uma categoria finitamente completa.

- a) Uma categoria interna à C é uma 6-upla $\mathcal{B} = \langle V, T, \partial_0, \partial_1, \iota, \circ \rangle$ onde $\langle V, T, \partial_0, \partial_1, \iota \rangle$ é um grafo reflexivo interno à C e $\circ: T \times_V T \rightarrow T$ é um morfismo de C tal que os diagramas ilustrados na Figura 5.7 comutam, onde os morfismos denotados por $\iota \times_V id_T$ e $\circ \times_V id_T$ são univocamente induzidos como ilustrado na Figura 5.8 ($id_T \times_V \iota$ é análogo ao $\iota \times_V id_T$ e $id_T \times_V \circ$ é isomorfo ao $\circ \times_V id_T$).
- b) Considere as categorias internas $\mathcal{B}_1 = \langle V_1, T_1, \partial_{01}, \partial_{11}, \iota_1, \circ_1 \rangle$ e $\mathcal{B}_2 = \langle V_2, T_2, \partial_{02}, \partial_{12}, \iota_2, \circ_2 \rangle$. Um funtor interno $f: \mathcal{B}_1 \rightarrow \mathcal{B}_2$ é um par de C -morfismos $\langle f_V: V_1 \rightarrow V_2, f_T: T_1 \rightarrow T_2 \rangle$ onde f é um morfismo de grafos reflexivos internos tal que o diagrama ilustrado na Figura 5.9 (esquerda) comuta sendo $f_T \times_{V_2} f_T$ é univocamente induzido pelos produtos fibrados, como ilustrado na Figura 5.9 (direita).
- c) As categorias internas à C e os correspondentes funtores internos constituem a categoria $Cat(C)$. □

As propriedades da categoria base C em relação aos limites são herdadas por $Cat(C)$.

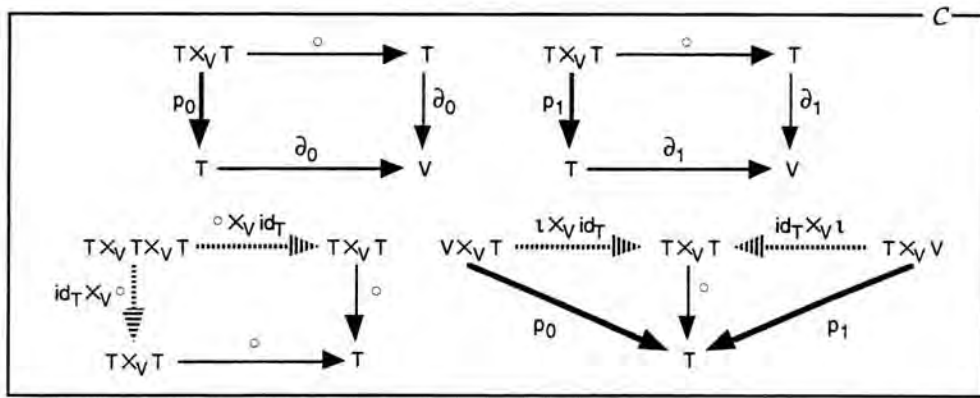


Figura 5.7 Domínio, codomínio, associatividade e identidade da composição

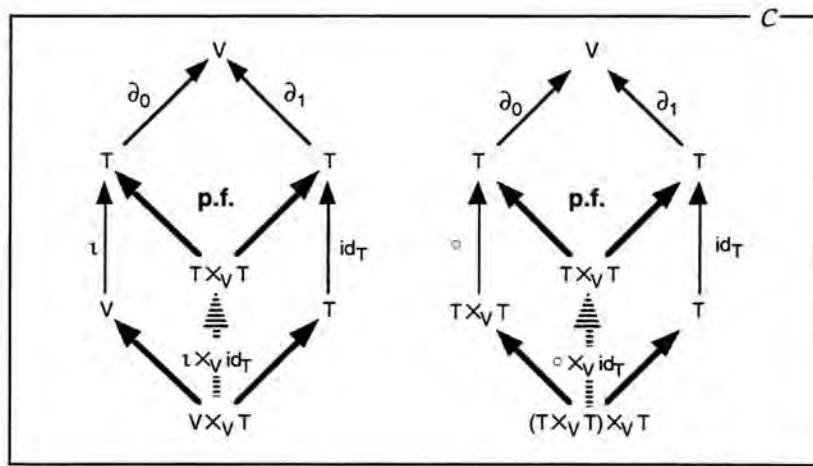


Figura 5.8 Morfismos univocamente induzidos

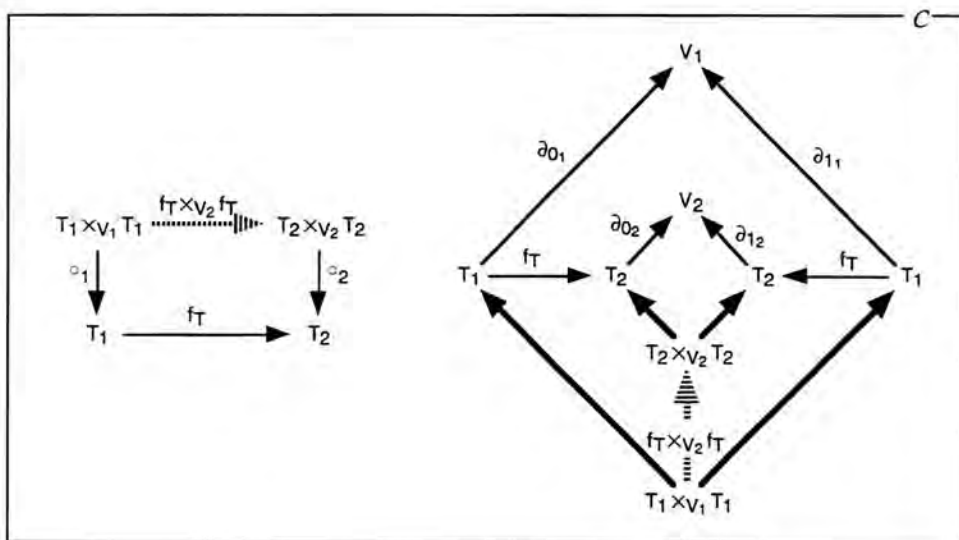


Figura 5.9 Diagramas comutativos para funtores internos

Proposição 5.11 Considere a categoria C . Se C é completa, então $Cat(C)$ é completa. Adicionalmente, se C possui todos os coprodutos os quais são isomorfos aos produtos, então $Cat(C)$ possui todos os coprodutos os quais são isomorfos aos produtos.

Prova: Considere o funtor esquecimento $u: Cat(C) \rightarrow \mathcal{RGr}(C)$ tal que, para todo $Cat(C)$ -objeto, $\mathcal{B} = \langle V, T, \partial_0, \partial_1, \iota, \circ \rangle$, $u\mathcal{B} = \langle V, T, \partial_0, \partial_1, \iota \rangle$ e, para todo $Cat(C)$ -morfismo $f = \langle f_V, f_T \rangle: \mathcal{B}_1 \rightarrow \mathcal{B}_2$, $uf = \langle f_V, f_T \rangle: u\mathcal{B}_1 \rightarrow u\mathcal{B}_2$. Como u é um funtor fidedigno, $\langle Cat(C), u \rangle$ é categoria concreta sobre $\mathcal{RGr}(C)$. Adicionalmente, se C é completa, então $\mathcal{RGr}(C)$ é completa. Portanto, para provar que $Cat(C)$ é completa, é suficiente provar que, para cada diagrama de $Cat(C)$, o limite do diagrama em $\mathcal{RGr}(C)$ pode ser elevado como uma fonte inicial em $Cat(C)$. Suponha I um conjunto arbitrário de índices, i com valores em I e k com valores em $\{0, 1\}$. Por simplicidade, no texto que segue, a expressão $i \in I$ é omitida.

a) Elevação de produtos. Considere um conjunto indexado e arbitrário de $Cat(C)$ -objetos $\{\mathcal{B}_i = \langle V_i, T_i, \partial_{0i}, \partial_{1i}, \iota_i, \circ_i \rangle\}$ e o correspondente $\mathcal{RGr}(C)$ -produto $\times u\mathcal{B}_i = \langle \times V_i, \times T_i, \times \partial_{0i}, \times \partial_{1i}, \times \iota_i \rangle$, juntamente com o conjunto indexado de morfismos $\{\pi_i: \times u\mathcal{B}_i \rightarrow u\mathcal{B}_i\}$. Então, $\times \mathcal{B}_i = \langle \times V_i, \times T_i, \times \partial_{0i}, \times \partial_{1i}, \times \iota_i, \times \circ_i \rangle$, juntamente com o conjunto de morfismos $\{\pi_i: \times \mathcal{B}_i \rightarrow \mathcal{B}_i\}$, é fonte inicial de $\{\mathcal{B}_i\}$ onde $\times \circ_i$ é univocamente induzido pelo produto categorial, como ilustrado na Figura 5.10 (esquerda).

Para verificar que $\times(T_i \times_{V_i} T_i)$ é produto fibrado, ou seja, $\times(T_i \times_{V_i} T_i) = \times T_i \times_{\times V_i} \times T_i$, considere o diagrama na Figura 5.10 (direita) onde $\times p_{0i}$ e $\times p_{1i}$ são univocamente induzidos pelo produto $\times T_i$ (por simplicidade, P_i e $\times P_i$ denotam $T_i \times_{V_i} T_i$ e $\times(T_i \times_{V_i} T_i)$, respectivamente). Sejam P um C -objeto e $p_0: P \rightarrow \times T_i, p_1: P \rightarrow \times T_i$ C -morfismos tais que $\times \partial_{0i} \circ p_0 = \times \partial_{1i} \circ p_1$. Então:

- para cada i , existe um único morfismo $f_i: P \rightarrow P_i$ tal que $p_{ki} \circ f_i = \pi_{T_i} \circ p_k$. Logo, existe um único morfismo $g: P \rightarrow \times P_i$ tal que, para cada i , $f_i = \pi_{P_i} \circ g$ e para cada k , existe um único morfismo $q_k: P \rightarrow \times T_i$ tal que, para cada i , $\pi_{T_i} \circ q_k = p_{ki} \circ f_i$. Adicionalmente, $\pi_{T_i} \circ q_k = p_{ki} \circ f_i = p_{ki} \circ \pi_{P_i} \circ g = \pi_{T_i} \circ \times p_{ki} \circ g$. Assim, pela unicidade de q_k , $q_k = \times p_{ki} \circ g$. Como $\pi_{T_i} \circ p_k = p_{ki} \circ f_i$, então, pela unicidade de q_k , $p_k = q_k$ e, portanto, $p_k = \times p_{ki} \circ g$;
- suponha $g': P \rightarrow \times P_i$ tal que $p_k = \times p_{ki} \circ g'$. Assim, $p_{ki} \circ f_i = \pi_{T_i} \circ p_k = \pi_{T_i} \circ \times p_{ki} \circ g' = p_{ki} \circ \pi_{P_i} \circ g'$. Logo, pela unicidade de f_i , $f_i = \pi_{P_i} \circ g'$ e, pela unicidade de g , $g = g'$.

Para verificar que $\times \mathcal{B}_i$ (juntamente com $\{\pi_i\}$) é fonte inicial, considere a $Cat(C)$ -fonte $\langle \mathcal{B}, \{b_i: \mathcal{B} \rightarrow \mathcal{B}_i\} \rangle$. Como $\times u\mathcal{B}_i$ é produto em $\mathcal{RGr}(C)$, então existe um único $\mathcal{RGr}(C)$ -morfismo $h: u\mathcal{B} \rightarrow \times u\mathcal{B}_i$ tal que $b_i = \pi_{u\mathcal{B}_i} \circ h$. A elevação de h é o próprio h .

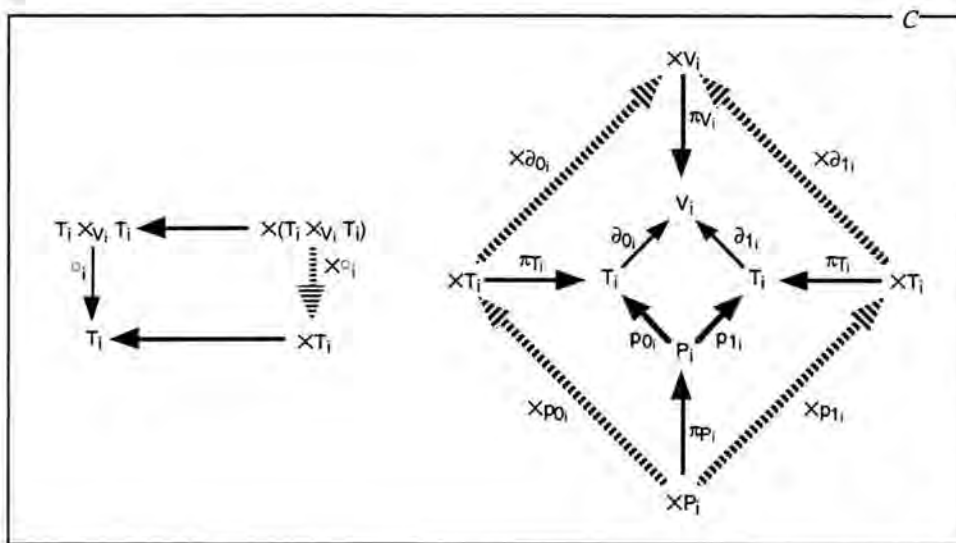


Figura 5.10 Morfismos univocamente induzidos

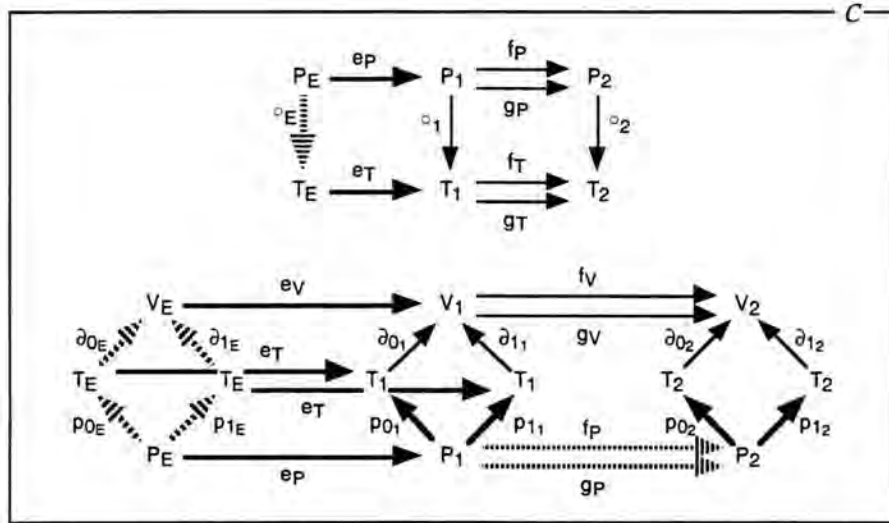


Figura 5.11 Morfismos univocamente induzidos

- b) Coprodutos. Para coprodutos isomorfos aos produtos, usando a prova do item anterior é fácil verificar que $Cat(C)$ possui todos os coprodutos e que são isomorfos aos produtos.
- c) Elevação de igualadores. Considere os $Cat(C)$ -objetos $\mathcal{B}_1 = \langle V_1, T_1, \partial_{01}, \partial_{11}, \iota_1, \circ_1 \rangle$, $\mathcal{B}_2 = \langle V_2, T_2, \partial_{02}, \partial_{12}, \iota_2, \circ_2 \rangle$ e os $Cat(C)$ -morfismos paralelos $f = \langle f_V, f_T \rangle, g = \langle g_V, g_T \rangle: \mathcal{B}_1 \rightarrow \mathcal{B}_2$ e o correspondente $RGr(C)$ -igualador $\langle e_V, e_T \rangle: \mathcal{E} \rightarrow \mathcal{B}_1$. Então $\mathcal{E} = \langle V_E, T_E, \partial_{0E}, \partial_{1E}, \iota_E \rangle \rightarrow \mathcal{B}_1$ é fonte inicial em $Cat(C)$, tal que \circ_E é definida pelos C -igualadores, como ilustrado na Figura 5.11 - superior (por simplicidade, P_1 e P_2 denotam $T_1 \times_{V_1} T_1$ e $T_2 \times_{V_2} T_2$, respectivamente) onde f_P e g_P são univocamente induzidos pelo produto fibrado P_2 , como ilustrado na Figura 5.11 - inferior (por simplicidade, os morfismos f_T e g_T são omitidos na figura). Os morfismos p_{0E} e p_{1E} são univocamente induzidos pelo igualador e_T .

Para verificar que P_E é um produto fibrado, ou seja, $P_E = T_E \times_{V_E} T_E$, considere o C -objeto P e os C -morfismos $p_0, p_1: P \rightarrow T_E$ tais que $\partial_{0E} \circ p_0 = \partial_{1E} \circ p_1$. Então:

- existe um único morfismo $h: P \rightarrow P_1$ tal que $p_{k1} \circ h = e_T \circ p_k$ e existem $u_f, u_g: P \rightarrow P_2$ morfismos únicos tais que $p_{k2} \circ u_f = f_T \circ p_{k1} \circ h$ e $p_{k2} \circ u_g = g_T \circ p_{k1} \circ h$, respectivamente. Adicionalmente, $p_{k2} \circ u_f = f_T \circ p_{k1} \circ h = f_T \circ e_T \circ p_k$ e $p_{k2} \circ u_g = g_T \circ p_{k1} \circ h = g_T \circ e_T \circ p_k$. Como e_T é igualador de f_T, g_T , então $p_{k2} \circ u_f = p_{k2} \circ u_g$. Pela unicidade de u_f e u_g , $u_f = u_g$. Seja $u = u_f = u_g$. Portanto, $p_{k2} \circ u = f_T \circ p_{k1} \circ h = p_{k2} \circ f_P \circ h$ e $p_{k2} \circ u = g_T \circ p_{k1} \circ h = p_{k2} \circ g_P \circ h$. Pela unicidade de u , $u = f_P \circ h$ e $u = g_P \circ h$. Logo, $f_P \circ h = g_P \circ h$. Como P_E juntamente com e_P é igualador, então existe um único morfismo $v: P \rightarrow P_E$ tal que $h = e_P \circ v$. Adicionalmente, $e_T \circ p_k = p_{k1} \circ e_P \circ v = e_T \circ p_{kE} \circ v$. Como e_T é igualador, p_k é o único morfismo tal que $e_T \circ p_k = e_T \circ p_{kE} \circ v$. Portanto, $p_k = p_{kE} \circ v$;
- Suponha $v': P \rightarrow P_E$ tal que $p_k = p_{kE} \circ v'$. Assim, $p_{k1} \circ h = e_T \circ p_k = e_T \circ p_{kE} \circ v' = p_{k1} \circ e_P \circ v'$. Então, pela unicidade de h , $h = e_P \circ v'$ e, pela unicidade de v , $v' = v$.

A verificação de que é uma fonte inicial é análoga a realizada para o produto arbitrário. \square

5.5 Categoria Interna com Objetos Distinguidos

A definição de objetos distinguidos para categorias internas é uma simples extensão da definição de nodos distinguidos para grafos internos.

Definição 5.12 Categoria das Categorias Internas com Objetos Distinguidos. Considere a categoria $Cat(C)$. A categoria das categorias internas com objetos distinguidos é a categoria das setas $d \downarrow ob$, denotada por $Cat_d(C)$, onde:

- a) $d: \mathcal{D} \rightarrow \mathcal{A}$ é um funtor;
- b) $ob = c \circ v: Cat(C) \rightarrow \mathcal{A}$ é um funtor tal que:
 - b.1) $v: Cat(C) \rightarrow C$ é o funtor esquecimento que associa a cada categoria $\mathcal{B} = \langle V, T, \partial_0, \partial_1, \iota, \circ \rangle$ o correspondente C -objeto V e a cada funtor $h = \langle h_V, h_T \rangle$ o C -morfismo h_V ;
 - b.2) $c: C \rightarrow \mathcal{A}$ é um funtor. □

Portanto, uma categoria interna com nodos distinguidos \mathcal{B}_d é uma tripla $\mathcal{B}_d = \langle \mathcal{B}, D, d \rangle$, onde \mathcal{B} é uma categoria de $Cat(C)$, D é um objeto de \mathcal{D} representando os objetos distinguidos e $d: \mathcal{D} \rightarrow ob \mathcal{B}$ é um \mathcal{A} -morfismo que instancia os objetos distinguidos em \mathcal{B} .

Um morfismo de categorias internas com objetos distinguidos é um morfismo na categoria das setas $d \downarrow ob$, ou seja, dados os $Cat_d(C)$ -objetos $\mathcal{B}_{d_1} = \langle \mathcal{B}_1, D_1, d_1 \rangle$ e $\mathcal{B}_{d_2} = \langle \mathcal{B}_2, D_2, d_2 \rangle$, um $Cat_d(C)$ -morfismo $h: \mathcal{B}_{d_1} \rightarrow \mathcal{B}_{d_2}$ é uma tripla $h = \langle h_V, h_T, h_D \rangle$ onde $\langle h_V, h_T \rangle: \mathcal{B}_1 \rightarrow \mathcal{B}_2$ é um funtor e $h_D: D_1 \rightarrow D_2$ é um \mathcal{D} -morfismo, tal que o diagrama ilustrado na Figura 5.12 comuta.

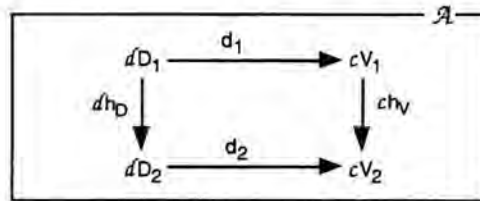


Figura 5.12 Diagrama comutativo para morfismos de categorias internas com objetos distinguidos

Proposição 5.13 Considere a categoria $Cat(C)$ e os funtores $d: \mathcal{D} \rightarrow \mathcal{A}$, $c: C \rightarrow \mathcal{A}$ tais que $Cat(C) = d \downarrow c \circ v$. Então:

- a) se $Cat(C)$ e \mathcal{D} são completas e $c: C \rightarrow \mathcal{A}$ preserva limites, então $Cat_d(C)$ é completa;
- b) se $Cat(C)$ e \mathcal{D} são cocompletas e d preserva colimites, então $Cat_d(C)$ é cocompleta.

Prova: Como $Cat_d(C)$ é a categoria das setas $d \downarrow ob$ e $ob = c \circ v$ é suficiente provar que o funtor $v: Cat(C) \rightarrow C$ preserva limites. Considere o funtor $ct: C \rightarrow Cat(C)$ tal que para C -objeto V , ctV é a categoria interna $\langle V, V, id_V, id_V, id_V, \circ \rangle$ (os únicos morfismos são os identidade). É fácil verificar que ct é adjunto esquerdo de v . □

No contexto dos autômatos não-sequenciais, são consideradas somente as categoria internas cujos nodos distinguidos são pares de objetos de C . Assim, analogamente aos grafos reflexivos internos, a seguinte notação simplificada é adotada.

Notação 5.14 Categoria $\mathcal{DCat}(C)$. Considere C uma categoria com todos os coprodutos binários. Então $\mathcal{DCat}(C)$ denota a categoria das categorias internas cujos nodos distinguidos são objetos de C^2 , isto é, $\mathcal{DCat}(C) = \Downarrow_C \downarrow ob$ onde $\Downarrow_C: C^2 \rightarrow C$ é o funtor canonicamente induzido pelo coproduto categorial em C e $ob: \mathcal{DCat}(C) \rightarrow C$ é o funtor esquecimento que associa a cada $Cat(C)$ -objeto $\mathcal{B} = \langle V, T, \partial_0, \partial_1, \iota, \circ \rangle$ o correspondente C -objeto V . □

6 Autômatos Não-Sequenciais

Um autômato não-sequencial é um autômato especial onde os estados e as transições possuem estrutura monoidal comutativa. Uma transição estruturada especifica uma relação de independência ou de concorrência entre as transições componentes como em [Bednarczyk 88] e [Mazurkiewicz 88]. Comparando com os sistemas de transições assíncronos (introduzidos em [Bednarczyk 88]), a relação de independência em um autômato não-sequencial é explícita na representação gráfica. Um estado estruturado pode ser visto como uma soma de recursos locais a serem consumidos ou produzidos pelas transições, analogamente à noção de "token" para redes de Petri. Comparando com as redes de Petri como em [Reisig 85], todos os estados atingíveis por um autômato são explicitados na representação gráfica enquanto que, em uma rede, as marcações atingíveis são implícitas.

Autômatos não-sequenciais e os correspondentes morfismos constituem uma categoria a qual é completa e cocompleta, com produtos isomorfos aos coprodutos (é uma decorrência dos produtos serem isomorfos aos coprodutos na categoria dos monóides comutativos). O produto (ou coproduto) é visto como a composição paralela. A sincronização e a encapsulação de autômatos não-sequenciais é uma extensão da sincronização e da encapsulação de autômatos sequenciais. Os autômatos não-sequenciais são introduzidos em [Menezes & Costa 95], [Menezes *et al* 95] e [Menezes & Costa 95b].

Por simplicidade e para evitar redundâncias, as partes cujas construções são análogas às usadas para os autômatos sequenciais são somente referenciadas como tal. Autômatos não-sequenciais são definidos sobre grafos monoidais comutativos, os quais são grafos reflexivos internos aos monóides comutativos. Assim, pode-se afirmar que, autômatos não-sequenciais são autômatos internos aos monóides comutativos. Os grafos monoidais comutativos são introduzidos a seguir.

6.1 Grafos Monoidais Comutativos

Grafos monoidais comutativos são grafos reflexivos internos à categoria dos monóides comutativos. Grafos monoidais comutativos de um só nodo são grafos monoidais onde o monóide referente aos nodos possui um único elemento (o neutro). A categoria dos monóides comutativos, denotada por \mathcal{CMon} , é completa e cocompleta, com produtos isomorfos aos coprodutos e onde um monóide com um único elemento (o neutro) é um objeto zero. A seguinte notação é usada para \mathcal{CMon} -objetos:

- um \mathcal{CMon} -objeto M é denotado como uma tripla $M = \langle M, \oplus, e \rangle$ onde M é o conjunto suporte, $_ \oplus _ : M \times M \rightarrow M$ é a operação e $e \in M$ é o elemento neutro;
- um monóide comutativo livremente gerado por um conjunto gerador S é denotado por $S^\oplus = \langle S^\oplus, \oplus, e \rangle$;
- um \mathcal{CMon} -objeto zero é denotado por $e = \langle \{e\}, \oplus, e \rangle$.

Grafo Monoidal Comutativo

A categoria dos grafos monoidais comutativos é completa e cocompleta, com produtos isomorfos aos coprodutos.



Figura 6.1 Representação gráfica de nodos e arcos estruturados

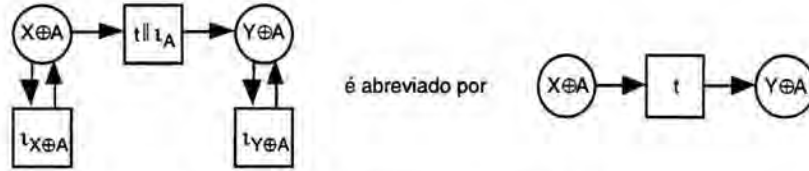


Figura 6.2 Simplificação da representação gráfica de grafos monoidais

Definição 6.1 Categoria dos Grafos Monoidais Comutativos. A categoria dos grafos (reflexivos) monoidais comutativos é a categoria interna $\mathcal{RGr}(CMon)$. \square

Portanto, um grafo monoidal comutativo G é uma quintupla $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ tal que $V = \langle V, \oplus, e \rangle$ é um monóide comutativo de nodos e $T = \langle T, \parallel, \tau \rangle$ é um monóide comutativo de arcos.

Como um nodo é um elemento de um monóide, pode ser representado como uma soma de nodos locais $n_1 A_1 \oplus \dots \oplus n_k A_k$ onde, para cada i em $\{1, \dots, k\}$, n_i representa a multiplicidade de A_i . Como o monóide é comutativo, a ordem dos termos na soma é irrelevante. A representação dos arcos como somas é análoga. Quando o contexto não deixar dúvidas, para um nodo A e para um arco $t: X \rightarrow Y$, o arco estruturado $t \parallel i_A: X \oplus A \rightarrow Y \oplus A$ é denotado simplesmente por $t: X \oplus A \rightarrow Y \oplus A$.

No caso de monóides livremente gerados, quando o conjunto gerador do monóide dos arcos é dado por extensão, os arcos identidade são omitidos. Por exemplo, o grafo $G = \langle \{A, B, C, D\}^\oplus, \{a, b, c, d, i_A, i_B, i_C, i_D\}^\parallel, \partial_0, \partial_1, \iota \rangle$ é denotado simplesmente por $G = \langle \{A, B, C, D\}^\oplus, \{a, b, c, d\}^\parallel, \partial_0, \partial_1, \iota \rangle$. Adicionalmente, um arco do conjunto suporte é denominado arco local ou gerador.

Em diagramas de grafos, o nodo $n_1 A_1 \oplus \dots \oplus n_k A_k$ e o arco $n_1 t_1 \parallel \dots \parallel n_k t_k$ são representados como ilustrado na Figura 6.1. Por simplicidade, em diagramas de grafos reflexivos monoidais, os arcos identidade são omitidos. As simplificações referentes à representação gráfica são ilustradas na Figura 6.2.

Exemplo 6.2 A representação gráfica do grafo monoidal $\langle \{X, Y\}^\oplus, \{x, y\}^\parallel, \partial_0, \partial_1, \iota \rangle$ (monóides livremente gerados) determinado pelos arcos locais $x: 2X \rightarrow Y$ e $y: 2X \rightarrow Y$ é ilustrado na Figura 6.3.

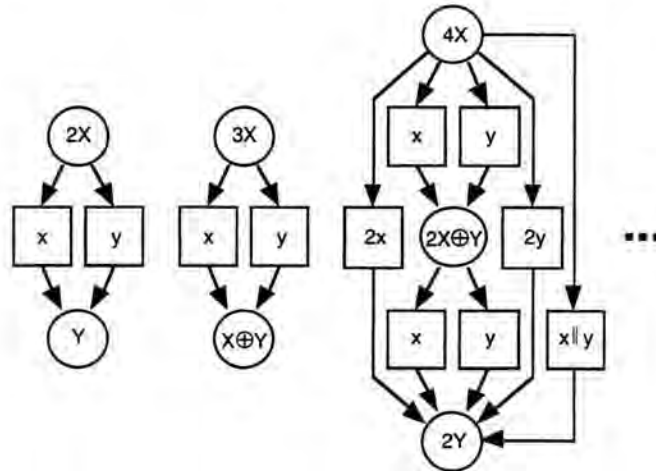


Figura 6.3 Representação gráfica de um grafo monoidal

Note-se que a representação gráfica é distribuída e infinita. □

Proposição 6.3 A categoria $\mathcal{RGr}(CMon)$ é completa e cocompleta. Adicionalmente, os produtos são isomorfos aos coprodutos.

Prova: Como $CMon$ é bicompleta, então $\mathcal{RGr}(CMon)$ é bicompleta. Como produtos e coprodutos são isomorfos em $CMon$, é fácil verificar que o mesmo ocorre em $\mathcal{RGr}(CMon)$. □

Exemplo 6.4 Considere os $\mathcal{RGr}(CMon)$ -grafos G_1 e G_2 com monóides livremente gerados nos nodos e arcos, onde:

- G_1 é determinado pelo arco local $a: A \rightarrow B$;
- G_2 é determinado pelos arcos locais $x: 2X \rightarrow Y$ e $y: 2X \rightarrow Y$.

Então a representação gráfica do grafo resultante do produto bem como do coproduto de G_1 e G_2 é como ilustrada na Figura 6.4. □

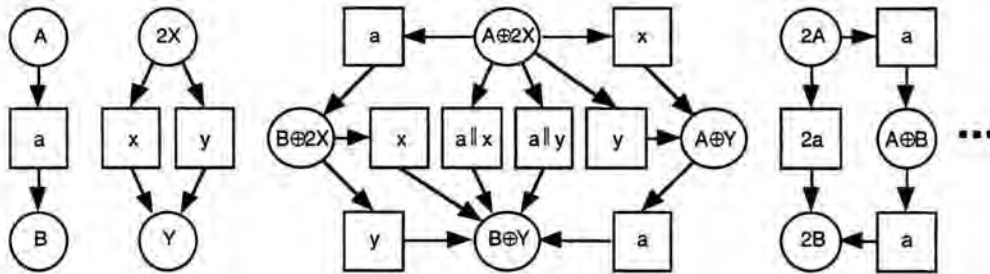


Figura 6.4 Produto ou coproduto de grafos monoidais

Grafo Monoidal Comutativo de um só Nódo

Grafos monoidais comutativos de um só nódo são grafos reflexivos de um só nódo internos à $CMon$. O único nódo é o elemento neutro do monóide de nodos. A categoria resultante é isomorfa à $CMon$ e, conseqüentemente, é completa e cocompleta, com produtos isomorfos aos coprodutos.

Definição 6.5 **Categoria dos Grafos Monoidais Comutativos de um só Nódo.** A categoria dos grafos (reflexivos) monoidais comutativos de um só nódo é a categoria interna $\mathcal{RGr}_1(CMon)$. □

Portanto, um grafo monoidal comutativo de um só nódo G é uma quintupla $E = \langle e, T, !, \parallel, \tau \rangle$ tal que e é um $CMon$ -objeto zero, $T = \langle T, \parallel, \tau \rangle$ é um monóide comutativo de arcos e os $CMon$ -morfismos $!: T \rightarrow e$ e $!_t: e \rightarrow T$ são únicos.

Exemplo 6.6 A representação gráfica do grafo monoidal de um só nódo $E = \langle e, \{u, v\}^!, !, \parallel, \tau \rangle$ é ilustrada na Figura 6.5. Note-se que possui infinitos arcos. □

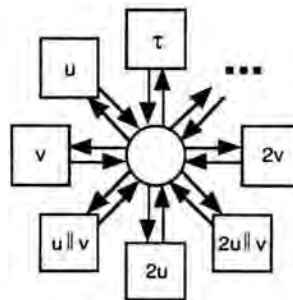


Figura 6.5 Representação gráfica de um grafo monoidal com um único nódo

Proposição 6.7 A categoria $\mathcal{RGr}_1(CMon)$ é isomorfa à categoria $CMon$.

Prova: Como $CMon$ possui objeto zero, $\mathcal{RGr}_1(CMon)$ é isomorfa à $CMon$. □

De fato, a categoria $\mathcal{RGr}_1(\mathcal{CMon})$ é subcategoria plena de $\mathcal{RGr}(\mathcal{CMon})$. Como $\mathcal{RGr}_1(\mathcal{CMon})$ e \mathcal{CMon} são isomorfas, então $\mathcal{RGr}_1(\mathcal{CMon})$ é completa e cocompleta sendo os produtos isomorfos aos coprodutos. Adicionalmente, um $\mathcal{RGr}_1(\mathcal{CMon})$ -grafo $\langle e, T, !, !, !_1 \rangle$ e um $\mathcal{RGr}_1(\mathcal{CMon})$ -morfismo $\langle !, h_T \rangle: \langle e, T_1, !, !, !_1 \rangle \rightarrow \langle e, T_2, !, !, !_1 \rangle$ podem ser representados simplesmente por T e $h_T: T_1 \rightarrow T_2$, respectivamente.

Grafo Monoidal Comutativo com Nodos Distinguidos

Grafos monoidais comutativos com nodos distinguidos são definidos como grafos reflexivos internos à \mathcal{CMon} e com nodos distinguidos também em \mathcal{CMon} . Lembre-se que, $\mathcal{DRGr}(\mathcal{CMon})$ denota a categoria interna com nodos distinguidos $\Downarrow_{\mathcal{CMon}} \text{ nodos}$ onde $\Downarrow_{\mathcal{CMon}}: \mathcal{CMon}^2 \rightarrow \mathcal{CMon}$ é o funtor canonicamente induzido pelo coproduto categorial em \mathcal{CMon} e $\text{ nodos}: \mathcal{RGr}(\mathcal{CMon}) \rightarrow \mathcal{CMon}$ é o funtor esquecimento que associa a cada grafo monoidal o correspondente monóide de nodos.

Definição 6.8 *Categoria dos Grafos Monoidais Comutativos com Nodos Distinguidos.* A categoria dos grafos (reflexivos) monoidais comutativos com nodos distinguidos é a categoria interna $\mathcal{DRGr}(\mathcal{CMon})$. □

Portanto, um grafo monoidal comutativo com nodos distinguidos M é uma tripla $M = \langle G, l, F, i+f \rangle$ onde $G = \langle V, T, \partial_0, \partial_1, ! \rangle$ é um $\mathcal{RGr}(\mathcal{CMon})$ -grafo, l e F são monóides comutativos de nodos distinguidos e $i+f: l +_{\mathcal{CMon}} F \rightarrow V$ é um \mathcal{CMon} -morfismo que instancia os nodos distinguidos em G . O grafo M também pode ser denotado como uma 8-upla $M = \langle V, T, \partial_0, \partial_1, l, l, F, i+f \rangle$.

Corolário 6.9 $\mathcal{DRGr}(\mathcal{CMon})$ é completa e cocompleta.

Prova: É consequência direta do funtor $\Downarrow_{\mathcal{CMon}}: \mathcal{CMon}^2 \rightarrow \mathcal{CMon}$ preservar colimites, pois é adjunto esquerdo do funtor diagonal $\Delta_{\mathcal{CMon}}: \mathcal{CMon} \rightarrow \mathcal{CMon}^2$. □

A representação gráfica dos nodos distinguidos em $\mathcal{DRGr}(\mathcal{CMon})$ é análoga a adotada para \mathcal{DRGr} .

Exemplo 6.10 Considere os $\mathcal{DRGr}(\mathcal{CMon})$ -grafos (monóides livremente gerados) como segue:

- M_1 é determinado pelo arco local $a: A \rightarrow B$ e por $l_1 = \{A\}^\oplus$ e $F_1 = \{B\}^\oplus$;
- M_2 é determinado pelos arcos locais $x: 2X \rightarrow Y, y: 2X \rightarrow Y$ e por $l_2 = \{X\}^\oplus$ e $F_2 = \{Y\}^\oplus$.

A Figura 6.6 ilustra o grafo resultante do produto (ou coproduto) de M_1 e M_2 (compare com o correspondente exemplo para $\mathcal{RGr}(\mathcal{CMon})$). □

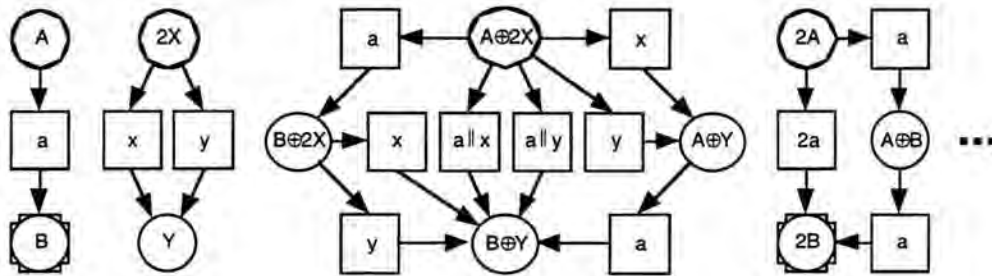


Figura 6.6 Produto ou coproduto de grafos monoidais com nodos distinguidos

6.2 Autômato Não-Sequencial

A definição de autômato não-sequencial é uma extensão da definição de autômato sequencial, onde os grafos são substituídos por grafos internos à categoria dos monóides comutativos.

Definição 6.11 *Categoria dos Autômatos Não-Sequenciais.* A categoria dos autômatos não-sequenciais é a categoria das setas $id \downarrow inc$, denotada por \mathcal{NAut} , onde:

- a) $id: \mathcal{DRGr}(\mathcal{CMon}) \rightarrow \mathcal{DRGr}(\mathcal{CMon})$ é o funtor identidade;

b) $inc: \mathcal{RGr}_1(CMon) \rightarrow \mathcal{DRGr}(CMon)$ é o funtor inclusão tal que, para cada $\mathcal{RGr}_1(CMon)$ -objeto $E = \langle e, T, !, !, !_V \rangle$, $incE = \langle E, e, !_e \rangle$ onde o morfismo $!_e: e \rightarrow e$ é único e, para cada $\mathcal{RGr}_1(CMon)$ -morfismo $h: E_1 \rightarrow E_2$, $inch = \langle h, !_e \rangle: \langle E_1, e, !_e \rangle \rightarrow \langle E_2, e, !_e \rangle$. \square

Portanto, um autômato não-sequencial N é uma tripla $N = \langle M, E, etiq \rangle$, tal que:

- $M = \langle G, l, i+f \rangle$ é um $\mathcal{DRGr}(CMon)$ -objeto onde o grafo reflexivo G representa a forma do autômato, l é o $CMon$ -objeto dos nodos iniciais, F é o $CMon$ -objeto dos nodos finais e o $CMon$ -morfismo $i+f: l+f \rightarrow nodosG$ instancia os nodos iniciais e finais nos nodos da forma do autômato;
- E é um grafo monoïdal de um só nodo representando as etiquetas das transições do autômato;
- $etiq = (!_V, etiq_T, !_l, !_F): idM \rightarrow incE$ é um morfismo de grafos monoïdais o qual associa uma etiqueta a cada arco (da forma) do autômato;

Um autômato não-sequencial $N = \langle M, E, etiq \rangle$ onde $M = \langle V, T, \partial_0, \partial_1, l, F, i+f \rangle$ pode ser denotado por $N: idM \rightarrow incE$ ou simplesmente por $N: M \rightarrow E$ ou ainda por uma 10-upla $N = \langle V, T, \partial_0, \partial_1, l, F, i+f, E, etiq \rangle$. A interpretação e representação dos componentes de um autômato não-sequencial são análogos aos do sequencial.

Exemplo 6.12 A representação gráfica do autômato não-sequencial $N = \langle \{A, B, C, X, Y\}^\oplus, \{a, b, x\}^{\parallel}, \partial_0, \partial_1, l, \{A, X\}^\oplus, \{C, Y\}^\oplus, \{u, v\}^{\parallel}, etiq \rangle$ (monóides livremente gerados) determinado pelas transições locais $a: A \rightarrow B$, $b: B \rightarrow C$, $x: 2X \rightarrow Y$ e pela etiquetagem local $a \mapsto u$, $b \mapsto u$, $x \mapsto v$ é ilustrado na Figura 6.7. Note-se que:

- o autômato possui infinitos estados distinguidos possíveis, pois estes são elementos de monóides livremente gerados;
- a escolha de qual estado inicial será efetivamente considerado no momento da execução do autômato é um não-determinismo externo. Por exemplo, para $A \oplus 2X$ somente a parte sombreada na Figura 6.7 necessita ser considerada;
- se uma parte conexa da representação gráfica de um autômato não-sequencial não possui estado inicial, então esta parte pode ser omitida. Por exemplo, para o autômato N , se $l = \{W\}^\oplus$ e $i: l \rightarrow \{A, B, C, X, Y\}^\oplus$ é induzido por $W \mapsto A \oplus 2X$, então somente as partes do diagrama referentes aos estados iniciais $A \oplus 2X, 2A \oplus 4X, \dots$ necessitam ser representadas. \square

Observação 6.13 Transições Estruturadas \times Quadrado de Independência. Considere a Figura 6.8. Suponha um autômato com as transições etiquetadas $a: A \rightarrow B$ e $x: X \rightarrow Y$. Então, $a \parallel x: A \oplus X \rightarrow B \oplus Y$, $a: A \oplus X \rightarrow B \oplus X$, $a: A \oplus Y \rightarrow B \oplus Y$, $x: A \oplus X \rightarrow A \oplus Y$ e $x: B \oplus X \rightarrow B \oplus Y$ também são transições etiquetadas do autômato, determinando o quadrado de independência associado à transição estruturada $a \parallel x$,

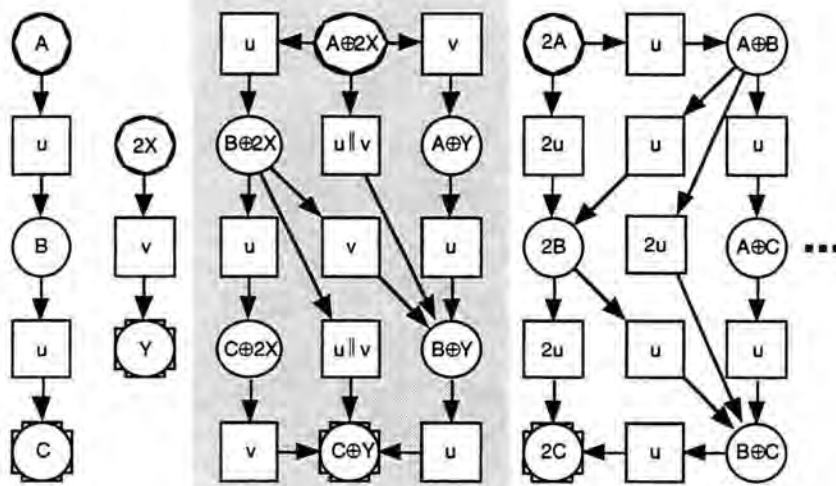


Figura 6.7 Representação gráfica de um autômato não-sequencial

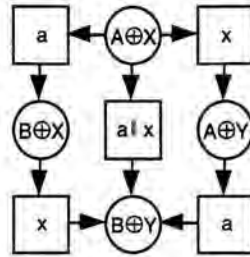


Figura 6.8 Quadrado de independência

ou seja:

- se duas transições podem ser executadas independentemente a partir de um mesmo estado origem, então podem ser executadas concorrentemente e, neste caso, atingem o mesmo estado destino;
- se duas transições independentes podem ser executadas uma imediatamente após a outra, então podem ser executadas em ordem invertida. \square

Observação 6.14 Autômato Não-Sequencial \times Monóide Comutativo Petri. O conceito de autômato não-sequencial não deve ser confundido com o de monóide comutativo Petri ("Petri commutative monoid") introduzido em [Meseguer & Montanari 90], estendido com etiquetas nas transições e com estados distinguidos. Enquanto que o significado de uma transição estruturada nos autômatos é o de explicitar que as transições componentes são independentes, em um monóide comutativo Petri, é o de introduzir uma nova transição na rede, correspondendo a uma execução síncrona de transições concorrentes. \square

Proposição 6.15 A categoria \mathcal{NAut} é completa e cocompleta. Adicionalmente, os produtos são isomorfos aos coprodutos.

Prova: As categorias $DRGr(CMon)$ e $RGr_1(CMon)$ são bicompletas. Como \mathcal{NAut} é a categoria das setas $id \downarrow inc$ e como $id: DRGr(CMon) \rightarrow DRGr(CMon)$ trivialmente preserva colimites, é suficiente verificar que $inc: RGr_1(CMon) \rightarrow DRGr(CMon)$ preserva limites. Considere o funtor $rg_1: DRGr(CMon) \rightarrow RGr_1(CMon)$ tal que:

- para todo o $DRGr(CMon)$ -objeto $M = \langle V, T, \partial_0, \partial_1, \iota, l, F, i + f \rangle$, $rg_1 M = \langle e, T, !, !, !, ! \rangle$;
- para todo o $DRGr(CMon)$ -morfismo $h = \langle h_V, h_T, h_l, h_F \rangle$, $rg_1 h = \langle !e, h_T \rangle$.

É fácil verificar que rg_1 é adjunto direito de inc onde a unidade $\eta: id \rightarrow inc \circ rg_1$ é tal que $\eta = \langle !V, id_T, !l, !F \rangle$ onde id_T é o $CMon$ -morfismo identidade. Portanto, \mathcal{NAut} é bicompleta. Como produtos e coprodutos são isomorfos em $DRGr(CMon)$ e em $RGr_1(CMon)$, o mesmo ocorre em \mathcal{NAut} . \square

O produto (ou coproduto) categorial em \mathcal{NAut} é interpretado como a composição paralela. O produto de dois autômatos não-sequenciais $N_1 = \langle M_1, E_1, etiq_1 \rangle$ e $N_2 = \langle M_2, E_2, etiq_2 \rangle$ é como segue:

$$N_1 \times N_2 = \langle M_1 \times_{DRGr(CMon)} M_2, E_1 \times_{RGr_1(CMon)} E_2, etiq_1 \times_{DRGr(CMon)} etiq_2 \rangle$$

onde o morfismo $etiq_1 \times_{DRGr(CMon)} etiq_2$ é univocamente induzido pelo produto em $DRGr(CMon)$.

Observação 6.16 Conjunto de Estados Iniciais e Finais. É perfeitamente possível definir uma categoria de autômatos não-sequenciais cujos estados iniciais e finais são conjuntos (e não monóides). Neste caso, os nodos iniciais e finais são instanciados em Set , ou seja, a categoria dos autômatos não-sequenciais é definida sobre a categoria dos grafos internos com nodos distinguidos $RGr_{\perp Set}(CMon) = \perp_{Set} \downarrow u$ onde \perp_{Set} é o funtor canonicamente induzido pelo coproduto em Set e $u: RGr(CMon) \rightarrow Set$ é o funtor esquecimento que associa a cada grafo reflexivo interno $G = \langle \langle V, \oplus, e \rangle, \langle T, \parallel, \tau \rangle, \partial_0, \partial_1, \iota \rangle$ o conjunto suporte V . É fácil verificar que u possui adjunto esquerdo e, portanto, preserva limites. Logo, $RGr_{\perp Set}(CMon)$ é completa e cocompleta e, conseqüentemente, a categoria dos autômatos não-sequenciais cujos estados iniciais e finais são conjuntos, também é completa e cocompleta. Entretanto, nesta categoria de autômatos, produto e coproduto não são isomorfos. A

principal consequência deste fato no estudo da reificação por fecho transitivo que segue, é que esta construção não satisfaz à composicionalidade horizontal. \square

6.3 Sincronização

A sincronização de autômatos não-sequenciais é uma extensão da sincronização para autômatos sequenciais. No texto que segue, é omitido que $i \in I$, para algum conjunto de índices I .

Proposição 6.17 O funtor esquecimento $u: \mathcal{N}Aut \rightarrow \mathcal{RGr}_1(CMon)$ que associa a cada autômato não-sequencial $N: M \rightarrow E$ o correspondente grafo monoideal de um só nodo E é uma fibração.

Prova: Seja $f: E_1 \rightarrow E_2$ um morfismo em $\mathcal{RGr}_1(CMon)$ e $N_2 = \langle M_2, E_2, \text{eti}q_2 \rangle$ um autômato não-sequencial. Considere o produto fibrado representado na Figura 6.9. Então, $N_1 = \langle M_1, E_1, \text{eti}q_1 \rangle$ é um autômato não-sequencial por construção e $u = \langle u_M, \hat{f} \rangle: N_1 \rightarrow N_2$ é morfismo cartesiano de f e N_2 . Os detalhes omitidos são análogos aos da prova referente aos autômatos sequenciais. \square

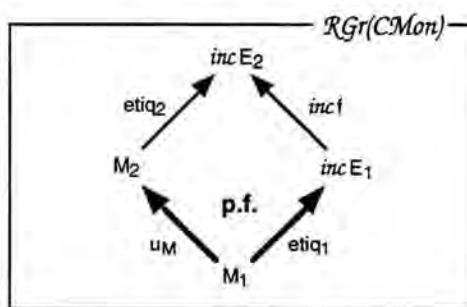


Figura 6.9 Produto fibrado

O produto fibrado usado na prova da proposição anterior, possui as seguintes propriedades, as quais são análogas às do autômato sequencial excetuando-se a última:

- se $f: E_1 \rightarrow E_2$ é um isomorfismo, então N_1 é uma reetiquetagem das transições de N_2 ;
- se $f: E_1 \rightarrow E_2$ é um monomorfismo, então N_1 é uma restrição das transições de N_2 ;
- se f é tal que, para alguma etiqueta a de E_1 , $f(a) = \tau$, então uma transição local etiquetada por a (e as correspondentes somas) é introduzida em cada estado de N_1 ;
- se f é tal que para duas etiquetas distintas a e b de E_1 , $f(a) = f(b) = x$, então as transições locais de N_2 etiquetadas por x (e as correspondentes somas) são substituídas em N_1 por duas transições locais paralelas etiquetadas por a e b (e as correspondentes somas) com os mesmos estados origem e destino de x ;
- se uma etiqueta a é associada a uma soma $x_1 \parallel \dots \parallel x_k$, então as transições de N_2 etiquetadas por x_1, \dots, x_k (e as correspondentes somas) são substituídas em N_1 por transições etiquetadas por a (e as correspondentes somas) com os mesmos estados de origem e destino de $x_1 \parallel \dots \parallel x_k$ (e as correspondentes somas). Este caso é ilustrado no exemplo adiante.

Definição 6.18 Tabela de Sincronizações, Morfismo de Sincronização. Seja $\{N_i: M_i \rightarrow E_i\}$ um conjunto de autômatos não-sequenciais a serem sincronizados. Então:

- uma tabela de sincronizações de $\{N_i\}$, denotada por Tabela, é um $\mathcal{RGr}_1(CMon)$ -objeto que reflete a sincronização desejada entre os autômatos ao nível das etiquetas;
- um morfismo de sincronização $\text{sinc}: \text{Tabela} \rightarrow \times E_i$ é um $\mathcal{RGr}_1(CMon)$ -monomorfismo. \square

Definição 6.19 Funtor de Sincronização. Sejam $\{N_i: M_i \rightarrow E_i\}$ um conjunto de autômatos não-sequenciais a serem sincronizados e $\text{sinc}: \text{Tabela} \rightarrow \times E_i$ um morfismo de sincronização. Considere a fibração $u: \mathcal{N}Aut \rightarrow \mathcal{RGr}_1(CMon)$, a construção dos correspondentes morfismos cartesianos como na prova da Proposição 6.17 e as fibras $u^{-1}\text{Tabela}$, $u^{-1}\times E_i$. Então:

- $\text{sinc}: u^{-1}\times E_i \rightarrow u^{-1}\text{Tabela}$ é o funtor de sincronização induzido por u e sinc ;

b) o autômato não-sequencial resultante da sincronização de $\{N_i\}$, denotado por $\parallel_{\text{sync}} N_i$ é $\text{sync} \times N_i$. \square

Portanto, o autômato resultante de uma sincronização é uma restrição da composição paralela dos autômatos componentes. A seguir, é apresentada uma construção categorial da tabela de sincronizações bem como do morfismo de sincronização univocamente induzido para sincronizações do tipo chamada, compartilhamento ou ambos os casos.

Definição 6.20 Construção Categorial da Tabela de Sincronizações. Seja $\{N_i: M_i \rightarrow E_i\}$ um conjunto de autômatos não-sequenciais a serem sincronizados. Então:

- a) seja Canal o menor subgrafo de $\times E_i$ que representa as etiquetas a serem sincronizadas;
- b) para cada i , E_i' é o menor subgrafo de E_i que contém todas as etiquetas das transições de N_i as quais chamam alguma transição de outro autômato;
- c) para cada i , o morfismo chama $_i$: $E_i' \rightarrow$ Canal é tal que se a em E_i' chama as transições x_1, \dots, x_n , então $\text{chama}_i(a) = a \mid x_1 \mid \dots \mid x_n$.

Seja D o diagrama representado na Figura 6.10 onde, para cada i , sub_i : $E_i' \rightarrow E_i$ é um monomorfismo que caracteriza E_i' como subobjeto de E_i . A tabela de sincronizações denotada por Tabela é o objeto resultante do colimite do diagrama D . \square

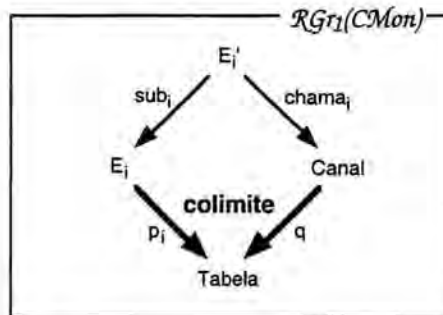


Figura 6.10 Tabela de sincronizações determinada por um colimite

Proposição 6.21 Considere o diagrama D cujo colimite determina o grafo Tabela juntamente com os morfismos p_i : $E_i \rightarrow$ Tabela. Então, para todo i , p_i e q possuem retração.

Prova: Os morfismos sub_i e chama_i são mono e o colimite do diagrama D pode ser determinado por somas amalgamadas. Logo, todos os morfismos do diagrama de somas amalgamadas são mono. Então, para todo i , p_i e q também são mono. Portanto, para todo i , existem retrações para p_i e q tais que, para todo a em Tabela, se existe a_i em E_i tal que $p_i(a_i) = a$, então $p_i^R(a) = a_i$, caso contrário, $p_i^R(a) = \tau$ e, se existe c em Canal tal que $q(c) = a$, então $q^R(a) = c$, caso contrário, $q^R(a) = \tau$. Os detalhes omitidos são análogos aos da correspondente prova para autômatos sequenciais. \square

Na definição abaixo, lembre-se que $\mathcal{RGr}_1(\mathcal{CMon})$ é isomorfa a \mathcal{CMon} e que, em \mathcal{CMon} , produtos são isomorfos aos coprodutos. Assim, sejam inj_i : $E_i \rightarrow \times E_i$ monomorfismos determinados pelo coproduto categorial.

Definição 6.22 Construção Categorial do Morfismo de Sincronização. Seja s : Canal $\rightarrow \times E_i$ um monomorfismo que caracteriza Canal como subobjeto de $\times E_i$. Sejam inj_i : $E_i \rightarrow \times E_i$ monomorfismos determinados pelo coproduto categorial. O morfismo de sincronização determinado por um diagrama D é sync : Tabela $\rightarrow \times E_i$ tal que $\text{sync}(\tau) = \tau$ e, para todo $a \neq \tau$ em Tabela, se $q^R(a) \neq \tau$, então $\text{sync}(a) = s \circ q^R(a)$, caso contrário existe um único k em I tal que $p_k^R(a) \neq \tau$ e $\text{sync}(a) = \text{inj}_k \circ p_k^R(a)$. \square

É fácil verificar que o morfismo sync definido acima é, de fato, um monomorfismo.

Exemplo 6.23 Considere os autômatos não-sequenciais produtor e consumidor com as transições etiquetadas determinadas por prod : $A \rightarrow B$, envia : $B \rightarrow A$ para o produtor e rec : $X \rightarrow Y$, cons : $Y \rightarrow X$

para o consumidor (monóides livremente gerados) e com os estados iniciais $\{A\}^\oplus$ e $\{X\}^\oplus$, respectivamente, análogos aos do correspondente exemplo para autômatos sequenciais. Então:

- a) Suponha que é desejada a sincronização envia compartilha rec. Então, o canal é $\text{Canal} = \{\text{envia} \mid \text{rec}\}^\dagger$, a tabela de sincronizações é $\text{Tabela} = \{\text{prod}, \text{cons}, \text{envia} \mid \text{rec}\}^\dagger$ e o autômato resultante é como representado na Figura 6.11 (por simplicidade, na representação gráfica, as etiquetas envia, prod, rec, cons são abreviadas para e, p, r, c, respectivamente). Repare que as transições envia e rec (e as correspondentes somas) são substituídas pela transição (e as correspondentes somas) envia \mid rec. Comparando com o correspondente exemplo para autômatos sequenciais, nota-se que a transição $\text{prod} \parallel \text{cons}$ não é apagada, pois é necessária para refletir a não-sequencialidade do autômato.
- b) Suponha que não é desejada qualquer sincronização entre as transições dos autômatos componentes. Então, o canal é um \mathcal{CMon} -objeto zero, a tabela de sincronizações é $\text{Tabela} = \{\tau, \text{prod}, \text{envia}, \text{rec}, \text{cons}\}^\dagger$ e o autômato resultante representa a composição paralela (transição alguma é apagada). \square

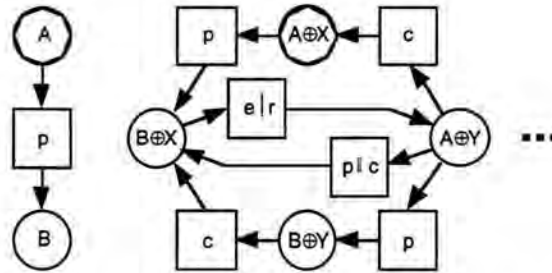


Figura 6.11 Sincronização dos autômatos produtor e consumidor

6.4 Encapsulação

A encapsulação de autômatos não-sequenciais é uma extensão da encapsulação para autômatos sequenciais.

Proposição 6.24 O funtor esquecimento $u: \mathcal{NAut} \rightarrow \mathcal{RGr}_1(\mathcal{CMon})$ que associa a cada autômato não-sequencial $N: M \rightarrow E$ o correspondente grafo reflexivo de um só nodo E , é uma cofibração.

Prova: Considere o $\mathcal{RGr}_1(\mathcal{CMon})$ -morfismo $f: E_1 \rightarrow E_2$ e o \mathcal{NAut} -objeto $N_1 = \langle M_1, E_1, \text{eti}_{q_1} \rangle$. Sejam $N_2 = \langle M_1, E_2, f \circ \text{eti}_{q_1} \rangle$ um \mathcal{NAut} -objeto e $u = (\text{id}_{M_1}, f): N_1 \rightarrow N_2$ um \mathcal{NAut} -morfismo. Então, u é cocartesiano em relação ao f e N_1 . Os detalhes omitidos são análogos aos da prova referente aos autômatos sequenciais. \square

Definição 6.25 Morfismo de Encapsulação. Seja E_1 as etiquetas do autômato considerado.

- a) Dado o subobjeto de etiquetas E_2 a serem exportadas. Seja $\text{sub}: E_2 \rightarrow E_1$ um monomorfismo que caracteriza E_2 como subobjeto de E_1 . Então $\text{caps} = \text{sub}^R: E_1 \rightarrow E_2$ onde sub^R é a retração de sub tal que, para todo a_1 em E_1 , se existe a_2 em E_2 tal que $\text{sub}(a_2) = a_1$, então $\text{sub}^R(a_1) = a_2$, caso contrário, $\text{sub}^R(a_1) = \tau$.
- b) Dado o subobjeto E de etiquetas a serem encapsuladas. Seja $\text{sub}: E \rightarrow E_1$ um monomorfismo que caracteriza E como subobjeto de E_1 . Então o morfismo de encapsulação $\text{caps}: E_1 \rightarrow E_2$ induzido por sub é determinado pela soma amalgamada ilustrada na Figura 6.12, onde $!$ é um $\mathcal{RGr}_1(\mathcal{CMon})$ -objeto terminal e o morfismo $!$ é único. \square

Definição 6.26 Funtor de Encapsulação. Considere o autômato $N: M \rightarrow E_1$, o morfismo de encapsulação $\text{caps}: E_1 \rightarrow E_2$, a cofibração $u: \mathcal{NAut} \rightarrow \mathcal{RGr}_1(\mathcal{CMon})$, a construção dos correspondentes morfismos cocartesianos como na prova da Proposição 6.24 e as fibras $u^{-1}E_1$, $u^{-1}E_2$. Então, $\text{caps}: u^{-1}E_1 \rightarrow u^{-1}E_2$ é o funtor de encapsulação induzido por u e caps e o autômato resultante da encapsulação de N , denotado por $N \backslash \text{caps}$, é $\text{caps}N$. \square

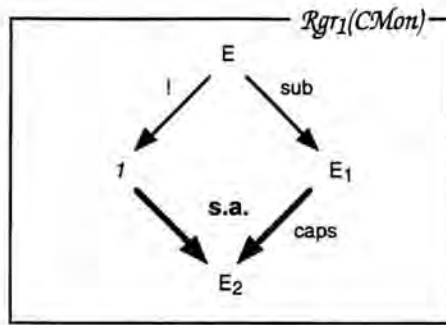


Figura 6.12 Morfismo de encapsulação determinado por uma soma amalgamada

Transições encapsuladas são omitidas em transições resultantes de somas. Por exemplo, uma transição etiquetada por $u \parallel \tau$ é denotada por u .

Exemplo 6.27 Considere o autômato não-sequencial introduzido no Exemplo 6.12. Suponha que é desejada a encapsulação da etiqueta u . Então, $E = \{u\}^\dagger$, o correspondente morfismo de encapsulação é determinado por $\tau \mapsto \tau, u \mapsto u, v \mapsto \tau$ e o autômato não-sequencial resultante da encapsulação desejada é ilustrado na Figura 6.13. \square

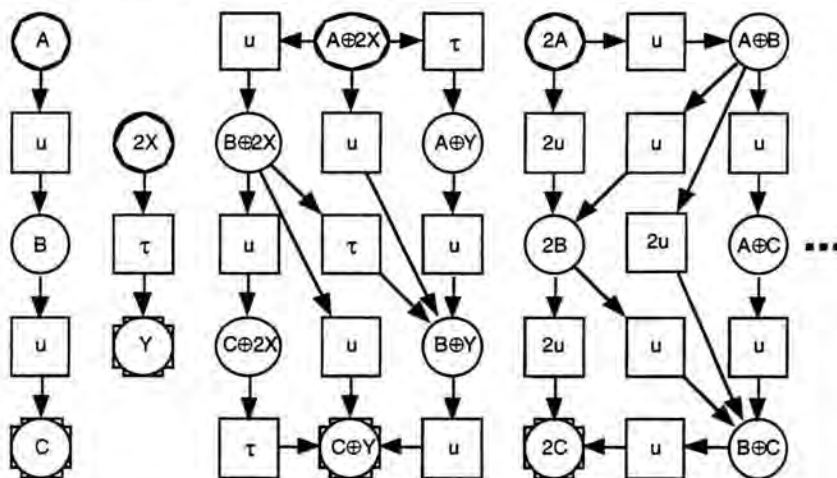


Figura 6.13 Encapsulação de um autômato não-sequencial

7 Reificação de Autômatos Não-Sequenciais

Analogamente aos autômatos sequenciais, a reificação de autômatos não-sequenciais é vista como um morfismo especial onde o objeto destino é enriquecido com o seu fecho computacional não-sequencial. A principal diferença é que o fecho computacional não-sequencial representa todas as computações possíveis que podem ser obtidas a partir das permutações sequenciais e *não-sequenciais* das transições componentes, respeitando os estados de origem e destino. A substituição dos grafos monoidais pelas correspondentes categorias monoidais na definição dos autômatos não-sequenciais define uma nova estrutura denominada categoria computacional não-sequencial. Portanto, uma categoria computacional não-sequencial é, basicamente, uma categoria interna aos monóides comutativos com objetos distinguidos e morfismos etiquetados. As categorias computacionais não-sequenciais e os correspondentes morfismos determinam uma categoria denominada \mathcal{NComp} . O funtor $cn: \mathcal{NComp} \rightarrow \mathcal{NAut}$ que esquece sobre a operação de composição e introduz alguns requisitos sobre a concorrência, possui adjunto esquerdo $nc: \mathcal{NAut} \rightarrow \mathcal{NComp}$ o qual gera livremente a composição nas transições. O fecho computacional não-sequencial é obtido pela composição dos dois funtores, ou seja, $nfc = cn \circ nc: \mathcal{NAut} \rightarrow \mathcal{NAut}$.

Autômatos não-sequenciais e reificações constituem a categoria $Reif\mathcal{NAut}$, definida de forma análoga à categoria $Reif\mathcal{Aut}$. Logo, a composicionalidade vertical é satisfeita. Adicionalmente, esta construção também *satisfaz à composicionalidade horizontal*, pois a composição paralela de dois morfismos de reificação resulta em um morfismo também de reificação. As operações de sincronização e encapsulação são estendidas para os autômatos não-sequenciais reificados.

A abordagem proposta nesta dissertação pode ser comparada como segue:

- a reificação proposta é inspirada nos morfismos de implementação introduzidos em [Meseguer & Montanari 90]. Note-se que, em [Meseguer & Montanari 90], não são tratadas questões como composicionalidades horizontal e vertical, sincronização, encapsulação, etiquetação e estados distinguidos;
- o fecho computacional de autômatos não-sequenciais é definido sobre categorias internas aos monóides e não sobre categorias monoidais (veja, por exemplo, [Mac Lane 71]) como em [Meseguer & Montanari 90]. O uso de categorias internas permite construções mais gerais e provas mais simples referentes aos limites e colimites. Desta forma, os monóides podem ser facilmente substituídos por outra estrutura (como, por exemplo, grupos), herdando, da nova estrutura, as propriedades referentes aos limites e colimites. Para a reificação de autômatos não-sequenciais usando categorias monoidais, veja [Menezes *et al* 95];
- o uso de categorias internas na reificação foi originalmente proposta em [Menezes & Costa 95] e [Menezes & Costa 95b]. A abordagem foi modificada e estendida em [Menezes *et al* 95b], possibilitando que morfismos de reificação não preservem etiquetação e introduzindo a sincronização e encapsulação para autômatos reificados (ou seja, morfismos de reificação). Esta dissertação estende a abordagem para estados distinguidos, conforme [Menezes 95].

Note-se que, no texto que segue, o termo categoria monoidal comutativa usado para denotar categoria interna aos monóides comutativos não deve ser confundido com o termo categoria monoidal como em [Mac Lane 71].

7.1 Categorias Internas à Categoria dos Monóides Comutativos

A seguir são introduzidas as categorias internas às categorias monoidais usadas no contexto dos autômatos não-sequenciais. Os resultados referentes aos limites e colimites são facilmente obtidos a partir dos resultados do capítulo referente aos grafos e categorias internas.

Definição 7.1 *Categoria das Categorias Monoidais Comutativas.* A categoria das categorias monoidais comutativas é a categoria interna $Cat(CMon)$. \square

Portanto, um $Cat(CMon)$ -objeto é uma 6-upla $C = \langle V, T, \partial_0, \partial_1, \iota, ; \rangle$ onde V, T são $CMon$ -objetos e $\partial_0, \partial_1, \iota$ são $CMon$ -morfismos.

Proposição 7.2 A categoria $Cat(CMon)$ é completa e possui todos os coprodutos os quais são isomorfos aos produtos.

Prova: É consequência direta de $CMon$ ser bicompleta e dos coprodutos serem isomorfos aos produtos. \square

Definição 7.3 *Categoria Monoidal Comutativa com um só Objeto.*

- Uma categoria monoidal comutativa com só objeto $\mathcal{E} = \langle e, T, !, !, !_1, \circ \rangle$ é um elemento da categoria $Cat(CMon)$ onde $e = \langle \{e\}, \otimes, e \rangle$ é um $CMon$ -objeto zero fixo e os $CMon$ -morfismos $!: T \rightarrow e, !_1: e \rightarrow T$ são únicos.
- Um morfismo de categorias monoidais comutativas de um só objeto é um $Cat(CMon)$ -morfismo $\langle !_e, h_T \rangle: \langle e, T_1, !, !, !_1 \rangle \rightarrow \langle e, T_2, !, !, !_1 \rangle$ onde o $CMon$ -morfismo $!_e: e \rightarrow e$, é único.
- As categorias monoidais com um só objeto e os correspondentes morfismos constituem a categoria $Cat_1(CMon)$. \square

A categoria $Cat_1(CMon)$ é subcategoria plena de $Cat(CMon)$.

Proposição 7.4 A categoria $Cat_1(CMon)$ é completa e possui todos os coprodutos os quais são isomorfos aos produtos.

Prova: Considere o funtor inclusão $inc: Cat_1(CMon) \rightarrow Cat(CMon)$ o qual é fidedigno. Portanto, $\langle Cat_1(CMon), inc \rangle$ é uma categoria concreta sobre $Cat(CMon)$. Suponha D um diagrama arbitrário em $Cat_1(CMon)$. Pela prova da completude para as categorias internas, pode-se inferir que o objeto resultante do limite de D em $Cat(CMon)$ é da forma $\langle e, T, !, !, !_1, \circ \rangle$. Como todo morfismo entre categorias monoidais comutativas de um só nodo em $Cat(CMon)$ também é um morfismo em $Cat_1(CMon)$, é consequência direta que o limite de D é fonte inicial em $Cat_1(CMon)$. Como coprodutos são isomorfos aos produtos em $Cat(CMon)$, o mesmo ocorre em $Cat_1(CMon)$. \square

Definição 7.5 *Categoria das Categorias Monoidais Comutativas com Objetos Distinguidos.* A categoria das categorias monoidais comutativas com objetos distinguidos é a categoria $DCat(CMon) = \amalg_{CMon} \downarrow ob$ onde $\amalg_{CMon}: CMon^2 \rightarrow CMon$ é o funtor canonicamente induzido pelo coproduto categorial em $CMon$ e $ob: Cat(CMon) \rightarrow CMon$ é o funtor esquecimento que associa a cada categoria monoidal o seu correspondente monóide de objetos. \square

Portanto, um $DCat(CMon)$ -objeto é uma quádrupla $\mathcal{M} = \langle C, I, F, i+f \rangle$ onde C é um $Cat(CMon)$ -objeto, I e F são $CMon$ -objetos e $i+f$ é um $CMon$ -morfismo.

Proposição 7.6 A categoria $DCat(CMon)$ é completa e possui todos os coprodutos os quais são isomorfos aos produtos.

Prova: É consequência direta das categorias $CMon$ e $Cat(CMon)$ serem completas e possuírem todos os coprodutos os quais são isomorfos aos produtos. \square

7.2 Computações Não-Sequenciais

A definição das categorias computacionais não-sequenciais estende a definição das categorias computacionais para o contexto das categorias monoidais comutativas. No texto que segue, considere $e = \langle \{e\}, \otimes, e \rangle$ um $CMon$ -objeto zero fixo. Note-se que o objeto resultante do coproduto $e+e$ e o morfismo univocamente induzido pelo coproduto $!_{e+e}: e+e \rightarrow e$ são isomorfos a e e $!_e: e \rightarrow e$,

respectivamente. Assim, por simplicidade, $e + e$ e $!_{e+e}$ são identificados com e e $!_e$, respectivamente.

Definição 7.7 *Categoria das Categorias Computacionais Não-Sequenciais.* A categoria das categorias computacionais não-sequenciais, denotada por \mathcal{NComp} , é a categoria das setas $id \downarrow inc$ onde:

- a) $id: \mathcal{DCat}(CMon) \rightarrow \mathcal{DCat}(CMon)$ é o funtor identidade;
- b) $inc: \mathcal{Cat}_1(CMon) \rightarrow \mathcal{DCat}(CMon)$ é o funtor inclusão tal que, para cada $\mathcal{Cat}_1(CMon)$ -objeto \mathcal{E} , $inc\mathcal{E} = \langle \mathcal{E}, e, e, !_e \rangle$ e, para cada $\mathcal{Cat}_1(CMon)$ -morfismo $h: \mathcal{E}_1 \rightarrow \mathcal{E}_2$, $inc h = \langle h, !_e, !_e \rangle: inc\mathcal{E}_1 \rightarrow inc\mathcal{E}_2$. \square

Portanto, uma categoria computacional não-sequencial é uma tripla $\mathcal{N} = \langle \mathcal{M}, \mathcal{E}, etiq \rangle$ onde \mathcal{M} é um $\mathcal{DCat}(CMon)$ -objeto, \mathcal{E} é um $\mathcal{Cat}_1(CMon)$ -objeto e $etiq: id\mathcal{M} \rightarrow inc\mathcal{E}$ é um $\mathcal{DCat}(CMon)$ -morfismo.

Proposição 7.8 A categoria \mathcal{NComp} é completa e possui todos os coprodutos os quais são isomorfos aos produtos.

Prova: As categorias $\mathcal{DCat}(CMon)$ e $\mathcal{Cat}_1(CMon)$ são completas. Como \mathcal{NComp} é a categoria das setas $id \downarrow inc$, é suficiente verificar que $inc: \mathcal{Cat}_1(CMon) \rightarrow \mathcal{DCat}(CMon)$ preserva limites. Considere o funtor $ct_1: \mathcal{DCat}(CMon) \rightarrow \mathcal{Cat}_1(CMon)$ tal que:

- para todo o $\mathcal{DCat}(CMon)$ -objeto $M = \langle V, T, \partial_0, \partial_1, !, ;, l, F, i+f \rangle$, $ct_1 M = \langle e, T, !, !, !_l, ; \rangle$;
- para todo o $\mathcal{DCat}(CMon)$ -morfismo $h = \langle h_V, h_T, h_l, h_F \rangle$, $ct_1 h = \langle !_e, h_T \rangle$.

É fácil verificar que ct_1 é adjunto esquerdo de inc onde a unidade $\eta: id \rightarrow inc \circ ct_1$ é tal que $\eta = \langle !_V, id_T, !_l, !_F \rangle$ onde id_T é o $CMon$ -morfismo identidade. Portanto, \mathcal{NComp} é completa. Como produtos e coprodutos são isomorfos em $\mathcal{DCat}(CMon)$ e em $\mathcal{Cat}_1(CMon)$, o mesmo ocorre em \mathcal{NComp} . \square

Definição 7.9 *Funtor cn .* O funtor esquecimento $cn: \mathcal{NComp} \rightarrow \mathcal{NAut}$ é tal que, para cada \mathcal{NComp} -objeto $\mathcal{N} = \langle \mathcal{M}, \mathcal{E}, etiq \rangle$ onde $\mathcal{M} = \langle C, l, F, i+f \rangle$, $C = \langle V, T, \partial_0, \partial_1, !, ; \rangle$ e $\mathcal{E} = \langle e, T, !, !, !_l, - \rangle$, tem-se que:

- para o $\mathcal{DCat}(CMon)$ -objeto \mathcal{M} , $cn\mathcal{M}$ é o $DRGr(CMon)$ -objeto $M = \langle C, l, F, i+f \rangle$ onde $C = \langle V, T^a, \partial_0^a, \partial_1^a, !^a \rangle$ tal que T^a é T sujeito à regra equacional abaixo e ∂_0^a , ∂_1^a e $!^a$ são os morfismos ∂_0 , ∂_1 e $!$ restritos à T^a ;

$$\frac{t: A \rightarrow B \in T^a \quad u: B \rightarrow C \in T^a \quad t': A' \rightarrow B' \in T^a \quad u': B' \rightarrow C' \in T^a}{(t;u) \parallel (t';u') = (t \parallel t'); (u \parallel u')}$$

- para o $\mathcal{Cat}_1(CMon)$ -objeto \mathcal{E} , $cn\mathcal{E}$ é o $RGr_1(CMon)$ -objeto $E = \langle e, T^a, !, !, !_l \rangle$ onde T^a é definido como acima;
- para o \mathcal{NComp} -objeto \mathcal{N} , $cn\mathcal{N}$ é o \mathcal{NAut} -objeto $N = \langle M, E, etiq \rangle$ onde $etiq: M \rightarrow E$ é o $DRGr(CMon)$ -morfismo canonicamente induzido por $etiq: \mathcal{M} \rightarrow \mathcal{E}$.

E, para cada \mathcal{NComp} -morfismo $h = \langle h_M, h_E \rangle: \mathcal{N}_1 \rightarrow \mathcal{N}_2$ onde $h_M = \langle h_V, h_T, h_l, h_F \rangle: \mathcal{M}_1 \rightarrow \mathcal{M}_2$ e $h_E = \langle !, h_E \rangle: \mathcal{E}_1 \rightarrow \mathcal{E}_2$, tem-se que $cn h_M$ é o $DRGr(CMon)$ -morfismo $h_M = \langle h_V, h_T, h_l, h_F \rangle: M_1 \rightarrow M_2$, $cn h_E$ é o $RGr_1(CMon)$ -morfismo $h_E = \langle !, h_E \rangle: E_1 \rightarrow E_2$ e $cn h$ é o \mathcal{NAut} -morfismo $h = \langle h_M, h_E \rangle: N_1 \rightarrow N_2$. \square

O funtor cn independentemente de esquecer sobre a operação de composição introduz o seguinte requisito referente à concorrência:

$$(t;u) \parallel (t';u') = (t \parallel t'); (u \parallel u')$$

Ou seja, a composição paralela de duas computações independentes $t;u$ e $t';u'$ possui o mesmo efeito que a composição sequencial das computações paralelas $t \parallel t'$ e $u \parallel u'$. Como ilustração, considere as transições etiquetadas $t: A \rightarrow B$ e $u: C \rightarrow D$. Então, para $t \parallel u: A \oplus C \rightarrow B \oplus D$, tem-se que (para um melhor entendimento, as transições etiquetadas por τ possuem o arco da forma explicitados):

$$\begin{aligned} t \parallel u &= (\tau[!_A]; t) \parallel (u; \tau[!_D]) = (\tau[!_A] \parallel u); (t \parallel \tau[!_D]) = u; t \\ t \parallel u &= u \parallel t = (\tau[!_C]; u) \parallel (t; \tau[!_B]) = (\tau[!_C] \parallel t); (u \parallel \tau[!_B]) = t; u \end{aligned}$$

Portanto, a execução concorrente de duas transições independentes é equivalente à execução sequencial, em qualquer ordem, das transições componentes. Como consequência, qualquer transição t resultante de somas $t = t_1 \parallel t_2 \parallel \dots \parallel t_n$ pode ser vista como uma composição sequencial, em

qualquer ordem, das transições locais componentes. Por exemplo (suponha $t_i: A_i \rightarrow B_i$, para i com valores em $\{1, \dots, n\}$):

$$t = t_1 \parallel t_2 \parallel \dots \parallel t_n = (t_1 \parallel \tau[\iota_{A_1}]); (t_2 \parallel \tau[\iota_{A_2}]); \dots; (t_n \parallel \tau[\iota_{A_n}]) = t_1; t_2; \dots; t_n$$

Definição 7.10 Funtor nc . O funtor $nc: \mathcal{NAut} \rightarrow \mathcal{NComp}$ é tal que, para cada \mathcal{NAut} -objeto $N = \langle M, E, \text{eti}_q \rangle$ onde $M = \langle G, l, F, i+f \rangle$, $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$ e $E = \langle e, L, !, \cdot, \iota \rangle$ tem-se que:

- ncM é o $\mathcal{DCat}(\mathcal{CMon})$ -objeto $\mathcal{M} = \langle G, l, F, i+f \rangle$ onde $G = \langle V, T^c, \partial_0^c, \partial_1^c, \iota^c, ; \rangle$ sendo $\iota^c: V \rightarrow T^c$ canonicamente induzido por $\iota: V \rightarrow T$ e sendo $T^c, \partial_0^c, \partial_1^c$ e a operação parcial $_;_:$ $T^c \times T^c \rightarrow T^c$ definidos pelas seguintes regras de inferência (τ é o elemento neutro de T):

$$\frac{t: A \rightarrow B \in T}{t: A \rightarrow B \in T^c} \quad \frac{t: A \rightarrow B \in T^c \quad u: B \rightarrow C \in T^c}{t;u: A \rightarrow C \in T^c} \quad \frac{t: A \rightarrow B \in T^c \quad u: C \rightarrow D \in T^c}{t \parallel u: A \oplus C \rightarrow B \oplus D \in T^c}$$

sujeitas às seguintes regras equacionais:

$$\frac{t \in T^c}{\tau; t = t \ \& \ t; \tau = t} \quad \frac{t: A \rightarrow B \in T^c}{\iota_A; t = t \ \& \ t; \iota_B = t} \quad \frac{t: A \rightarrow B \in T^c \quad u: B \rightarrow C \in T^c \quad v: C \rightarrow D \in T^c}{t; (u; v) = (t; u); v}$$

$$\frac{t \in T^c \quad u \in T^c}{t \parallel u = u \parallel t} \quad \frac{t \in T^c}{t \parallel \tau = t} \quad \frac{\iota_A \in T^c \quad \iota_B \in T^c}{\iota_A \parallel \iota_B = \iota_{A \oplus B}} \quad \frac{t \in T^c \quad u \in T^c \quad v \in T^c}{t \parallel (u \parallel v) = (t \parallel u) \parallel v}$$

- ncE é o $\mathcal{Cat}_1(\mathcal{CMon})$ -objeto $\mathcal{E} = \langle e, L^c, !, \cdot, \iota, ; \rangle$ onde L^c e a operação de composição são definidos como acima;
- ncN é o \mathcal{NComp} -objeto $\mathcal{N} = \langle \mathcal{M}, \mathcal{E}, \text{eti}_q \rangle$ onde $\text{eti}_q = \langle !v, \text{eti}_q^c, !l, !f \rangle: \mathcal{M} \rightarrow \mathcal{E}$ é o $\mathcal{DCat}(\mathcal{CMon})$ -morfismo indutivamente definido a partir de $\text{eti}_q = \langle !v, \text{eti}_q^c, !l, !f \rangle: \mathcal{M} \rightarrow \mathcal{E}$, tal que:

$$\begin{aligned} \text{para } t \text{ em } T, \text{ eti}_q^c(t) &= \text{eti}_q^c(t) \\ \text{para } t;u \text{ em } T^c, \text{ eti}_q^c(t;u) &= \text{eti}_q^c(t); \text{eti}_q^c(u) \\ \text{para } t \parallel u \text{ em } T^c, \text{ eti}_q^c(t \parallel u) &= \text{eti}_q^c(t) \parallel \text{eti}_q^c(u). \end{aligned}$$

E , para cada \mathcal{NAut} -morfismo $h = \langle h_M, h_E \rangle: N_1 \rightarrow N_2$ onde $h_M = \langle h_V, h_T, h_l, h_f \rangle: M_1 \rightarrow M_2$ é um $\mathcal{DRGr}(\mathcal{CMon})$ -morfismo e $h_E = \langle !, h_E \rangle: E_1 \rightarrow E_2$ é um $\mathcal{RGr}_1(\mathcal{CMon})$ -morfismo, tem-se que:

- nch_M é o $\mathcal{DCat}(\mathcal{CMon})$ -morfismo $h_M = \langle h_V, h_T^c, h_l, h_f \rangle: M_1 \rightarrow M_2$ onde h_T^c é definido como acima;
- nch_E é o $\mathcal{Cat}_1(\mathcal{CMon})$ -morfismo $h_E = \langle !, h_E^c \rangle: E_1 \rightarrow E_2$ onde h_E^c é definido como acima;
- nch é o \mathcal{NComp} -morfismo $h = \langle h_M, h_E \rangle: \mathcal{N}_1 \rightarrow \mathcal{N}_2$. \square

Proposição 7.11 O funtor nc é adjunto esquerdo do funtor cn .

Prova:

- A unidade da adjunção é a transformação natural $\eta: id_{\mathcal{NAut}} \rightarrow cn \circ nc$ a qual, para cada grafo componente, é a inclusão nas transições geradoras do fecho transitivo.
- De forma dual, a co-unidade da adjunção é a transformação natural $\varepsilon: nc \circ cn \rightarrow id_{\mathcal{NComp}}$ a qual, para cada grafo componente, associa os arcos compostos $\langle t \rangle; \langle u \rangle$ e $\langle t \rangle \parallel \langle u \rangle$ aos arcos $\langle t; u \rangle$ e $\langle t \parallel u \rangle$, respectivamente.

Portanto, $\langle nc, cn, \eta, \varepsilon \rangle: \mathcal{NAut} \rightarrow \mathcal{NComp}$ é uma adjunção. \square

Definição 7.12 Fecho Computacional. O fecho computacional (não-sequencial) é a operação functorial $nfc: \mathcal{NAut} \rightarrow \mathcal{NAut}$ tal que $nfc = cn \circ nc$. \square

Exemplo 7.13 Considere o autômato não-sequencial N_1 determinado pelas transições locais etiquetadas $a: A \rightarrow B$ e $b: B \rightarrow C$ (monóides livremente gerados). O correspondente fecho computacional não-sequencial é como ilustrado na Figura 7.1 onde as transições adicionadas estão representadas por um traço diferenciado. Note-se que as transições adicionadas e que são resultantes de somas são, de fato, classes de transições. Por exemplo, para a transição etiquetada $a; 2b: A \oplus B \rightarrow 2C$ tem-se que:

$$\begin{aligned} a; 2b &= a; (b \parallel b) = (\tau[\iota_B] \parallel a); (b \parallel b) = (\tau[\iota_B]; b) \parallel (a; b) = b \parallel (a; b) = \\ &= (b; \tau[\iota_C]) \parallel (\tau[\iota_A]; (a; b)) = (b \parallel \tau[\iota_A]); (\tau[\iota_C] \parallel (a; b)) = b; a; b = \dots \end{aligned} \quad \square$$

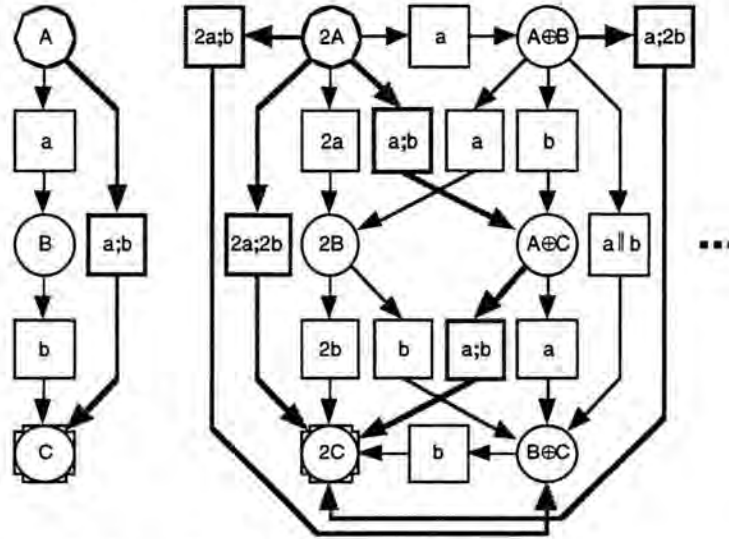


Figura 7.1 Autômato não-sequencial e o seu correspondente fecho computacional

7.3 Composicionalidade Vertical

Analogamente aos autômatos sequenciais, a adjunção $\langle nc, cn, \eta, \varepsilon \rangle: \mathcal{NAut} \rightarrow \mathcal{NComp}$ definida anteriormente induz uma mônada $T = \langle nfc, \eta, \mu \rangle$ em \mathcal{NAut} , onde $nfc = cn \circ nc: \mathcal{NAut} \rightarrow \mathcal{NAut}$ é um endofuntor e $\mu = cn \varepsilon nc: nfc^2 \rightarrow nfc$ é uma transformação natural. Usando a mônada T é introduzida uma categoria de autômatos não-sequenciais baseada nas categorias Kleisli, onde os morfismos são interpretados como reificações e a operação de composição garante a composicionalidade vertical.

Por simplicidade, nos diagramas que seguem, os funtores $id: DRGr(CMon) \rightarrow DRGr(CMon)$ e $inc: RGr_1(CMon) \rightarrow DRGr(CMon)$ usados na definição da categoria \mathcal{NAut} são omitidos. Por exemplo, em diagramas, $etiq: M \rightarrow E \rightarrow etiq$ denota $etiq; idM \rightarrow incE$.

Definição 7.14 *Categoria dos Autômatos Não-Sequenciais e Reificações.* Seja $T = \langle nfc, \eta, \mu \rangle$ onde $\eta = \langle \eta_M, \eta_E \rangle$ e $\mu = \langle \mu_M, \mu_E \rangle$ a mônada determinada pela adjunção $\langle nc, cn, \eta, \varepsilon \rangle: \mathcal{NAut} \rightarrow \mathcal{NComp}$. A categoria dos autômatos não-sequenciais e reificações, denotada por $Reif\mathcal{NAut}$, é tal que (suponha os \mathcal{NAut} -objetos $N_k = \langle M_k, E_k, etiq_k \rangle$, para k com valores em $\{1, 2, 3\}$):

- a) os objetos são os mesmo de \mathcal{NAut} ;
- b) os morfismos são construídos como segue:
 - b.1) $\varphi = \varphi_M: N_1 \rightarrow N_2$ é um $Reif\mathcal{NAut}$ -morfismo onde $\varphi_M: M_1 \rightarrow nfcM_2$ é um $DRGr(CMon)$ -morfismo;
 - b.2) para cada \mathcal{NAut} -objeto N , $\varphi = \eta_M: N \rightarrow N$ é o $Reif\mathcal{NAut}$ -morfismo identidade de N ;
- c) sejam $\varphi: N_1 \rightarrow N_2, \psi: N_2 \rightarrow N_3$ $Reif\mathcal{NAut}$ -morfismos. A composição $\psi \circ \varphi$ é o morfismo $\psi_M \circ_{\mathcal{X}} \varphi_M: N_1 \rightarrow N_3$ onde $\psi_M \circ_{\mathcal{X}} \varphi_M$ é determinado como ilustrado na Figura 7.2. \square

Com o objetivo de distinguir os $Reif\mathcal{NAut}$ -morfismos dos \mathcal{NAut} -morfismo uma reificação φ do autômato N_1 nas computações de N_2 é denotada por $\varphi: N_1 \rightarrow nfcN_2$ ou $\varphi: N_1 \Rightarrow N_2$.

Exemplo 7.15 Considere N_1 do exemplo anterior e o autômato não-sequencial N_2 determinado pelas transições locais etiquetadas $x: X \rightarrow Y$ e $y: Y \rightarrow X$ com monóides livremente gerados. Então $\varphi: N_1 \Rightarrow N_2$ determinado como abaixo é um morfismo de reificação:

$$\begin{array}{ll}
 A \rightarrow 2X & a \rightarrow x \parallel x \\
 B \rightarrow 2Y & b \rightarrow (y \parallel y); (x; y)^2; (x \parallel x) \\
 C \rightarrow 2Y &
 \end{array}$$

\square

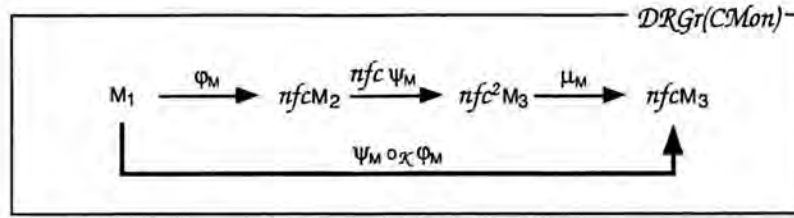


Figura 7.2 Composição de reificações é a composição na categoria Kleisli restrita à forma dos autômatos

Um morfismo na forma pode induzir um morfismo nas etiquetas determinando uma categoria de autômatos não-sequenciais e reificações cujas reificações são $\mathcal{N}Aut$ -morfismos, como na categoria introduzida a seguir.

Definição 7.16 *Categoria dos Autômatos Não-Sequenciais e Reificações com Etiquetação Induzida.* Seja $T = \langle nfc, \eta, \mu \rangle$ onde $\eta = \langle \eta_M, \eta_E \rangle$ e $\mu = \langle \mu_M, \mu_E \rangle$ a mônada determinada pela adjunção $\langle \etac, c\eta, \eta, \varepsilon \rangle: \mathcal{N}Aut \rightarrow \mathcal{N}Comp$. A categoria dos autômatos não-sequenciais e reificações com etiquetação induzida, denotada por $\mathcal{R}eif\mathcal{N}Aut_{\mathcal{E}}$, é tal que (suponha os $\mathcal{N}Aut$ -objetos $N_k = \langle M_k, E_k, etiq_k \rangle$, para k com valores em $\{1, 2, 3\}$):

- a) os objetos são os mesmo de $\mathcal{N}Aut$,

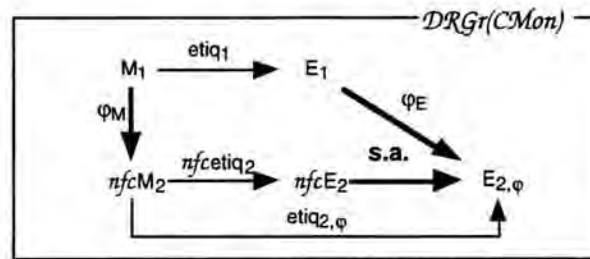


Figura 7.3 Reificação de autômatos não-sequenciais com etiquetação induzida

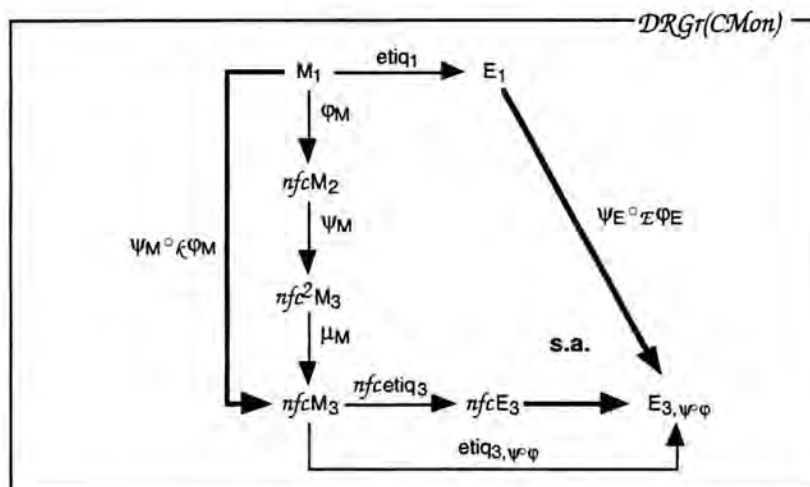


Figura 7.4 Composição de reificações de autômatos não-sequenciais com etiquetação induzida

b) os morfismos são construídos como segue:

b.1) seja $\varphi_M: M_1 \rightarrow nfcM_2$ um $\mathcal{DRGr}(CMon)$ -morfismo. Então $\varphi = \langle \varphi_M, \varphi_E \rangle: N_1 \rightarrow N_2$ é um $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ -morfismo onde φ_E é determinado pela soma amalgamada ilustrada na Figura 7.3;

b.2) para cada $\mathcal{N}\mathcal{Aut}$ -objeto N , $\varphi = \langle \eta_M: M \rightarrow nfcM, \varphi_E: E \rightarrow E_\eta \rangle: N \rightarrow N$ é o $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ -morfismo identidade de N onde φ_E é como definido no item anterior;

c) sejam $\varphi: N_1 \rightarrow N_2$, $\psi: N_2 \rightarrow N_3$ $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ -morfismos. A composição $\psi \circ \varphi$ é o morfismo $\langle \psi_M \circ \varphi_M, \psi_E \circ \varphi_E \rangle: N_1 \rightarrow N_3$ onde $\psi_M \circ \varphi_M$ e $\psi_E \circ \varphi_E$ são determinados como ilustrado na Figura 7.4. \square

É fácil verificar que o objeto resultante da soma amalgamada na definição acima é, de fato, um grafo de etiquetas. Como é mostrado na proposição a seguir, $\mathcal{Reif}\mathcal{N}\mathcal{Aut}$ e $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ são isomorfas. Portanto, todo morfismo de reificação pode ser visto como um morfismo especial de autômatos não-sequenciais.

Proposição 7.17 As categorias $\mathcal{Reif}\mathcal{N}\mathcal{Aut}$ e $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ são isomorfas.

Prova: Considere:

- o funtor $re: \mathcal{Reif}\mathcal{N}\mathcal{Aut} \rightarrow \mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ tal que, para todo o $\mathcal{Reif}\mathcal{N}\mathcal{Aut}$ -objeto N , $reN = N$ e, para todo o $\mathcal{Reif}\mathcal{N}\mathcal{Aut}$ -morfismo $\varphi = \langle \varphi_M: N_1 \rightarrow N_2, \varphi_E \rangle$,
- o funtor $er: \mathcal{Reif}\mathcal{N}\mathcal{Aut}_E \rightarrow \mathcal{Reif}\mathcal{N}\mathcal{Aut}$ tal que, para todo o $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ -objeto N , $erN = N$ e, para todo o $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ -morfismo $\varphi = \langle \varphi_M, \varphi_E \rangle: N_1 \rightarrow N_2$, $er\varphi = \varphi_M$.

É fácil verificar que $er \circ re = id_{\mathcal{Reif}\mathcal{N}\mathcal{Aut}}$ e $re \circ er = id_{\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E}$. \square

Como $\mathcal{Reif}\mathcal{N}\mathcal{Aut}$ e $\mathcal{Reif}\mathcal{N}\mathcal{Aut}_E$ são isomorfas, no texto que segue, $\mathcal{Reif}\mathcal{N}\mathcal{Aut}$ denota qualquer das duas categorias de reificações.

Observação 7.18 Morfismo de Reificação Induzido por um Morfismo nas Etiquetas. Dado um autômato não-sequencial N com as etiquetas E , considere o seu fecho computacional $nfcN$ e as correspondentes etiquetas $nfcE$. Seja $f_E: E' \rightarrow nfcE$ um morfismo de etiquetas. Então, f_E possui uma elevação cartesiana (definida na sincronização) $f_N: N' \rightarrow nfcN$ ao nível dos autômatos, o qual é um morfismo de reificação. Esta técnica é utilizada adiante na definição da semântica da linguagem Náutilus. Note-se que a mesma abordagem pode ser adotada para os autômatos sequenciais. \square

7.4 Composicionalidade Horizontal

A seguir, prova-se que os autômatos não-sequenciais satisfazem à composicionalidade horizontal, ou seja, que a reificação se distribui sobre a composição paralela dos autômatos componentes. Inicialmente, prova-se que o endofuntor fecho computacional não-sequencial preserva a composição paralela. Por simplicidade, no texto que segue, é omitido que $i \in I$, para um conjunto de índices I .

Proposição 7.19 O endofuntor $nfc: \mathcal{N}\mathcal{Aut} \rightarrow \mathcal{N}\mathcal{Aut}$ preserva produtos (ou coprodutos).

Prova: Seja $\{N_i\}$ uma família arbitrária de autômatos não-sequenciais. Como $nfc = cn \circ nc$ e como $\times N_i$ e $+ N_i$ são isomorfos, tem-se que, a menos de isomorfismo:

$$nfc \times N_i = cn \circ nc(+N_i)$$

Como nc preserva colimites (pois é adjunto esquerdo), então, a menos de isomorfismo:

$$nfc \times N_i = cn(+ncN_i)$$

Como cn preserva limites (pois é adjunto direito), tem-se que, a menos de isomorfismo:

$$nfc \times N_i = \times (cn \circ ncN_i)$$

Logo,

$$nfc \times N_i = \times (nfcN_i) \quad \square$$

Exemplo 7.20 Considere os autômatos não-sequenciais N_3 e N_4 (monóides livremente gerados) determinados pelas transições locais $a: A \rightarrow B$ para N_3 e $x: X \rightarrow Y$ para N_4 . O diagrama ilustrado

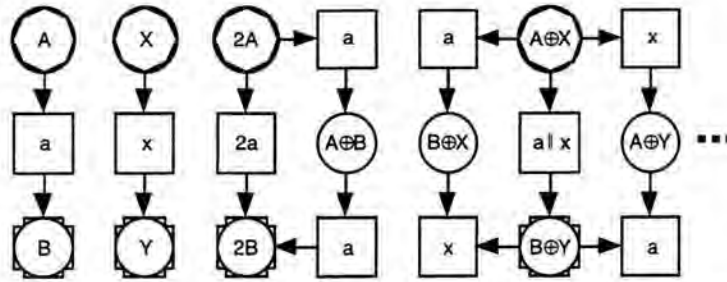


Figura 7.5 Fecho computacional não-sequencial preserva a composição paralela de autômatos

na Figura 7.5 representa igualmente $nfcN_3 \times nfcN_4$ e $nfc(N_3 \times N_4)$. Compare com o contra-exemplo apresentado na prova da não composicionalidade horizontal para autômatos sequenciais. \square

Proposição 7.21 Seja $\{\varphi_i: N_{1i} \Rightarrow N_{2i}\}$ uma família arbitrária de reificações de autômatos não-sequenciais. Então $\times \varphi_i: \times N_{1i} \Rightarrow \times N_{2i}$ é uma reificação.

Prova: No que segue, lembre-se que produtos e coprodutos são isomorfos em $DRGr(CMon)$. Suponha k com valores em $\{1, 2\}$ e $N_{ki} = \langle M_{ki}, E_{ki}, etiq_{ki} \rangle$. Considere o morfismo $\times \varphi_i: \times N_{1i} \rightarrow \times (nfcN_{2i})$ onde $\times \varphi_i = \langle \times \varphi_{M_i}, \times \varphi_{E_i} \rangle$ é univocamente induzido pelo produto categorial, como ilustrado na Figura 7.6 (lembre-se que nfc preserva produtos).

Para verificar que $\times \varphi_i$ é de fato uma reificação, é suficiente mostrar que o objeto $\times E_{2i,\varphi}$ juntamente com os morfismos q e $\times \varphi_{E_i}$ é soma amalgamada de $nfc(\times etiq_{2i}) \circ \times \varphi_{M_i}$ e $\times etiq_{1i}$. Então:

- Sejam $u: \times (nfcE_{2i}) \rightarrow S$ e $v: \times E_{1i} \rightarrow S$ morfismos tais que $u \circ nfc(\times etiq_{2i}) \circ \times \varphi_{M_i} = v \circ \times etiq_{1i}$. Como $+(nfcE_{2i})$ juntamente com $inj_{nfcE_{2i}}$ é coproduto, então existe um único morfismo $u': +(nfcE_{2i}) \rightarrow S$ tal que $u' \circ inj_{nfcE_{2i}} = u \circ inj_{nfcE_{2i}}$ e, portanto, $u = u'$. Analogamente, como $+E_{1i}$ juntamente com $inj_{E_{1i}}$ é coproduto, então existe um único morfismo $v': +E_{1i} \rightarrow S$ tal que $v' \circ inj_{E_{1i}} = v \circ inj_{E_{1i}}$ e, portanto, $v = v'$.
- Como $\textcircled{1}$ é soma amalgamada, para cada i , existe um único morfismo $h_i: E_{2i,\varphi} \rightarrow S$ tal que $h_i \circ p_i = u' \circ inj_{nfcE_{2i}}$ e $h_i \circ \varphi_{E_i} = v' \circ inj_{E_{1i}}$. Como $+E_{2i,\varphi}$ juntamente com os morfismos $inj_{E_{2i,\varphi}}$ é coproduto, então existe um único morfismo $h: +E_{2i,\varphi} \rightarrow S$ tal que, para cada i , $h_i = h \circ inj_{E_{2i,\varphi}}$.
- Como $h_i \circ \varphi_{E_i} = v' \circ inj_{E_{1i}}$ tem-se que, $h_i \circ \varphi_{E_i} = h \circ inj_{E_{2i,\varphi}} \circ \varphi_{E_i} = h \circ \times \varphi_{E_i} \circ inj_{E_{1i}} = v' \circ inj_{E_{1i}}$. Pela unicidade de v , $h \circ \times \varphi_{E_i} = v$. Analogamente, como $h_i \circ p_i = u' \circ inj_{nfcE_{2i}}$ tem-se que, $h_i \circ p_i = h \circ inj_{E_{2i,\varphi}} \circ p_i = h \circ q \circ inj_{nfcE_{2i}} = u' \circ inj_{nfcE_{2i}}$. Pela unicidade de u , $h \circ q = u$.
- Seja $h': +E_{2i,\varphi} \rightarrow S$ tal que $h' \circ \times \varphi_{E_i} = v$ e $h' \circ q = u$. Então, $v \circ inj_{E_{1i}} = h' \circ \times \varphi_{E_i} \circ inj_{E_{1i}} = h' \circ inj_{E_{2i,\varphi}} \circ \varphi_{E_i} = h_i \circ \varphi_{E_i}$. Pela unicidade de h_i , $h' \circ inj_{E_{2i,\varphi}} = h_i$ e, pela unicidade de h , $h = h'$. \square

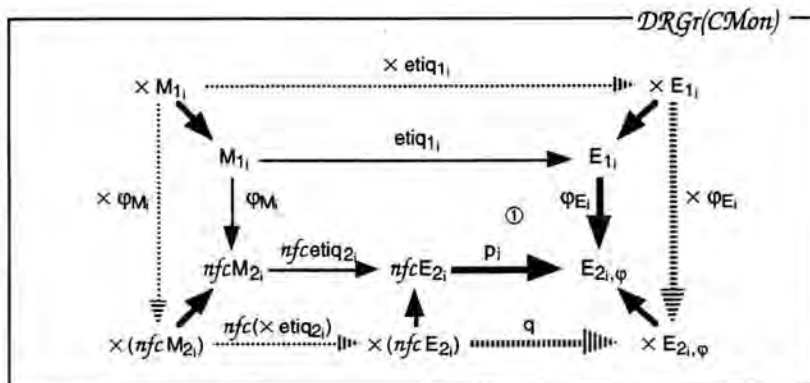


Figura 7.6 Morfismos univocamente induzidos

7.5 Sincronização

A sincronização de autômatos não-sequenciais reificados é um autômato que reflete a sincronização desejada juntamente com uma reificação induzida pelas reificações dos autômatos componentes. Na construção proposta, a composicionalidade horizontal é um resultado necessário. No texto que segue, suponha i com valores em I onde I é um conjunto de índices e k com valores em $\{1, 2\}$. Por simplicidade, é omitido que $i \in I$. Lembre-se que o endofuntor πfc preserva produtos e que todo morfismo de sincronização (ao nível das etiquetas) possui uma elevação cartesiana, ao nível dos autômatos.

Definição 7.22 Sincronização de Autômatos Não-Sequenciais Reificados. Considere a Figura 7.7. Sejam $\{\varphi_i: N_{1i} \Rightarrow N_{2i}\}$ uma família arbitrária de reificações onde $N_{ki} = \langle M_{ki}, E_{ki}, \text{eti}_{q_{ki}} \rangle$, $\times \varphi_i: \times N_{1i} \rightarrow \pi fc \times N_{2i}$ o correspondente morfismo de reificação induzido pelo produto categorial, $\text{sinc}_N: \text{Tabela} \rightarrow \times E_{1i}$ um morfismo de sincronização e $\text{sinc}_N: \times N_{1i} \rightarrow \times N_{1i}$ a correspondente elevação cartesiana. Então, a reificação do autômato sincronizado $\parallel N_{1i}$ é o morfismo $\parallel \varphi_i: \parallel N_{1i} \rightarrow \pi fc(\times N_{2i})$ e é tal que $\parallel \varphi_i = \times \varphi_i \circ \text{sinc}_N$ onde $\times \varphi_i$ é univocamente induzido pelo produto categorial. \square

Portanto, uma transição sincronizada é implementada na atomização da composição paralela das implementações das transições componentes no nível mais concreto.

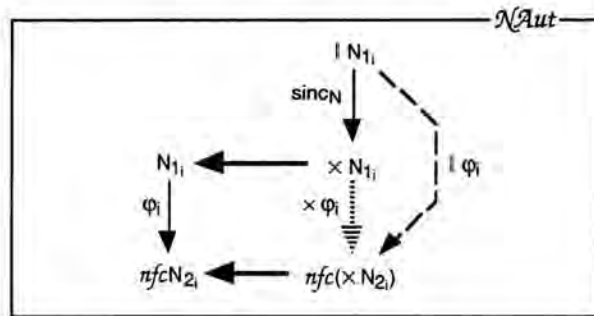


Figura 7.7 Sincronização de autômatos não-sequenciais reificados

7.6 Encapsulação

O efeito da encapsulação em um autômato não-sequencial reificado é somente na etiquetagem. No texto que segue, suponha k com valores em $\{1, 2\}$.

Definição 7.23 Encapsulação de Autômatos Não-Sequenciais Reificados. Considere a Figura 7.8. Seja $\varphi: N_1 \Rightarrow N_2$ uma reificação onde $N_k = \langle M_k, E_k, \text{eti}_{q_k} \rangle$ e $\varphi = \langle \varphi_M, \varphi_E \rangle$. Seja $\text{caps}: E_1 \rightarrow E_1'$ um morfismo de encapsulação e $\text{caps} N_1 = \langle M_1, E_1', \text{caps} \circ \text{eti}_{q_1} \rangle$ o autômato encapsulado. Então, o morfismo de reificação do autômato encapsulado é $\text{caps} \varphi = \varphi \setminus \text{caps} = \langle \varphi_M, \varphi_E \setminus \text{caps} \rangle$. \square

Como a encapsulação de um autômato reificado não depende da composicionalidade horizontal, uma definição análoga pode ser feita para os autômatos sequenciais.

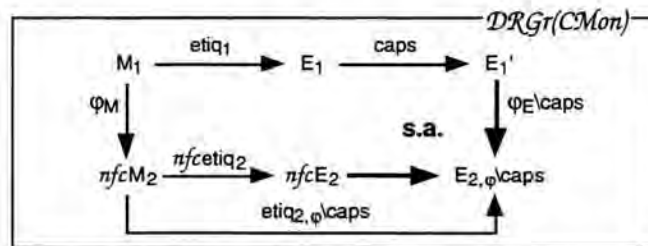


Figura 7.8 Encapsulação de autômatos não-sequenciais reificados

8 Redes de Petri

Redes de Petri, no texto que segue, são consideradas como o caso geral de redes do tipo lugar/transição. Inicialmente é introduzida a definição de rede de Petri vista como um grafo e a correspondente categoria, denotada por Petri , a qual é completa e cocompleta. Intuitivamente, o coproduto em Petri pode ser visto como a escolha entre as redes componentes e o produto como a composição totalmente síncrona, no sentido em que as transições da rede resultante são pares de transições das redes componentes. Entretanto, na prática, normalmente é desejável sincronizar algumas e não todas as transições componentes. Isto determina uma segunda categoria, denominada categoria das redes de Petri apontadas a qual é denotada por Petri^* , onde o conjunto de transições de cada rede possui um elemento distinguido representado por τ . Quando um morfismo de redes associa uma transição da rede origem à transição τ da rede destino, possui o efeito equivalente ao de esquecer ou apagar esta transição. A categoria Petri^* também é completa e cocompleta e, embora o coproduto possua a mesma interpretação que em Petri , o produto é interpretado como a composição paralela, da mesma forma que nas categorias dos autômatos.

Entretanto, para que o estudo do relacionamento entre redes de Petri e autômatos seja mais efetivo, é desejável modificar algumas características das redes, sem modificar seus princípios básicos. Assim, o conceito de rede de Petri é estendido como segue:

- rede de Petri marcada: estende a rede apontada com um conjunto de estados iniciais bem como um conjunto de estados finais. Analogamente aos autômatos, a escolha de qual estado inicial é efetivamente considerado no momento de execução da rede é um não-determinismo externo. Uma rede pára quando transição alguma pode ser executada a partir do estado corrente. Diz-se que a rede pára normalmente, se pára em um estado final;
- rede de Petri etiquetada: estende a rede marcada com uma etiquetagem nas transições.

A categoria resultante é completa e cocompleta. A abordagem proposta nesta dissertação pode ser comparada como segue:

- a cocompletude é um resultado expressivo principalmente quando comparado com [Winskel 87] e [Meseguer & Montanari 90], cujas categorias possuem restrições para garantir a existência de coprodutos. A principal diferença estrutural é que, na abordagem proposta, os estados iniciais (e distinguidos, em geral) constituem um conjunto (e não um único estado);
- as redes de Petri são vistas como grafos, inspirado em [Meseguer & Montanari 90]. São definidas como grafos estruturados, generalizando o conceito de grafos internos, onde os nodos e os arcos podem pertencer a categorias diferentes. Analogamente aos grafos internos usados na reificação, as categorias estruturadas permitem construções mais gerais e provas mais simples referentes aos limites e colimites;
- analogamente aos autômatos, as operações de sincronização e encapsulação são introduzidas para as redes de Petri marcadas e etiquetadas;
- a sincronização de redes usando a técnica de fibração, o uso de conjuntos de estados distinguidos e a definição de redes usando categorias estruturadas foram originalmente proposto em [Menezes & Costa 93], [Menezes 95] e [Menezes & Costa 95], respectivamente. Esta dissertação estende a abordagem para a encapsulação de redes, conforme [Menezes & Costa 95b].

No texto que segue, considere a categoria dos conjuntos apontados (conjuntos com um elemento distinguido), denotada por Set^* , a qual é completa e cocompleta.

8.1 Grafos Estruturados

Grafos estruturados generalizam o conceito de grafos internos onde os nodos e os arcos podem ser objetos de diferentes categorias dados funtores das categorias dos nodos e arcos para a categoria base. Os morfismos de origem e destino são morfismos na categoria base.

Grafo Estruturado

Definição 8.1 Categoria dos Grafos Estruturados. Considere a categoria base C e os funtores $v: \mathcal{V} \rightarrow C$ e $t: \mathcal{T} \rightarrow C$. A categoria dos grafos estruturados (internos à C) em relação aos funtores v e t , denotada por $\mathcal{G}r(v, t)$, é a categoria das setas $\Delta_{C^{\circ}} t \downarrow \Delta_{C^{\circ}} v$ onde $\Delta_C: C \rightarrow C^2$ é o funtor diagonal. \square

Portanto, um $\mathcal{G}r(v, t)$ -objeto G é uma quádrupla $G = \langle V, T, \partial_0, \partial_1 \rangle$ onde V é um \mathcal{V} -objeto, T é um \mathcal{T} -objeto e $\partial_0, \partial_1: tT \rightarrow vV$ são C -morfismos e um $\mathcal{G}r(v, t)$ -morfismo $h: G_1 \rightarrow G_2$ é uma dupla $h = \langle h_V, h_T \rangle$ onde $h_V: V_1 \rightarrow V_2$ é um \mathcal{V} -morfismo e $h_T: T_1 \rightarrow T_2$ é um \mathcal{T} -morfismo, tais que os diagramas ilustrados na Figura 8.1 comutam.

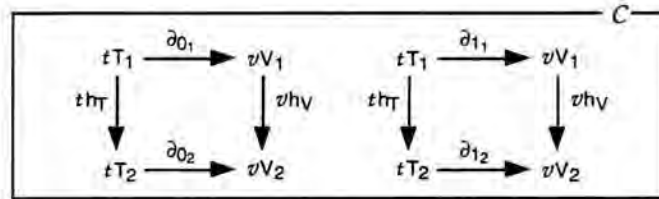


Figura 8.1 Diagramas comutativos para morfismos de grafos estruturados

Proposição 8.2

- Suponha que \mathcal{V}, \mathcal{T} são completas. Se $v: \mathcal{V} \rightarrow C$ preserva limites, então $\mathcal{G}r(v, t)$ é completa;
- Suponha que \mathcal{V}, \mathcal{T} são cocompletas. Se $t: \mathcal{T} \rightarrow C$ preserva colimites, então $\mathcal{G}r(v, t)$ é cocompleta.

Prova: É consequência direta de $\mathcal{G}r(v, t)$ ser definida como a categoria das setas $\Delta_{C^{\circ}} t \downarrow \Delta_{C^{\circ}} v$. \square

A categoria de grafos (não-estruturados) $\mathcal{G}r$ pode ser considerada como a categoria estruturada $\mathcal{G}r(id_{Set}, id_{Set})$. Como esperado, $\mathcal{G}r(id_{Set}, id_{Set})$ é bicompleta, pois $id_{Set}: Set \rightarrow Set$ trivialmente preserva limites e colimites. Analogamente, a categoria dos grafos internos $\mathcal{G}r(C)$ pode ser considerada como a categoria estruturada $\mathcal{G}r(id_C, id_C)$.

Grafo Estruturado com Nodos Distinguidos

A definição de grafo estruturado com nodos distinguidos é uma simples extensão da definição de grafo interno com nodos distinguidos.

Definição 8.3 Categoria dos Grafos Estruturados com Nodos Distinguidos. Considere a categoria base C e os funtores $v: \mathcal{V} \rightarrow C$ e $t: \mathcal{T} \rightarrow C$. A categoria dos grafos estruturados com nodos distinguidos é a categoria das setas $d \downarrow nodos$, denotada por $\mathcal{G}r_d(v, t)$, onde:

- $d: \mathcal{D} \rightarrow \mathcal{A}$ é um funtor onde \mathcal{D} é a categoria de domínio dos nodos distinguidos e \mathcal{A} é a categoria de instanciação dos nodos distinguidos no grafo;
- $nodos = c \circ n: \mathcal{G}r(v, t) \rightarrow \mathcal{A}$ é um funtor tal que:
 - $n: \mathcal{G}r(v, t) \rightarrow \mathcal{V}$ é o funtor esquecimento que associa a cada grafo o \mathcal{V} -objeto de nodos e a cada morfismo de grafos $h = \langle h_V, h_T \rangle$ o \mathcal{V} -morfismo h_V ;
 - $c: \mathcal{V} \rightarrow \mathcal{A}$ é um funtor que relaciona a categoria onde os grafos são considerados internamente e a categoria onde os nodos distinguidos são instanciados. \square

Portanto, um grafo estruturado com nodos distinguidos M é uma tripla $M = \langle G, D, d \rangle$ onde G é um $\mathcal{G}r(v, t)$ -grafo, D é um \mathcal{D} -objeto representando os nodos distinguidos e $d: dD \rightarrow nodosG$ é

\mathcal{A} -morfismo que instancia os nodos distinguidos em G . Um morfismo de grafos estruturados $h: M_1 \rightarrow M_2$ é uma tripla $h = \langle h_V, h_T, h_D \rangle$ onde $\langle h_V, h_T \rangle: G_1 \rightarrow G_2$ é um $\mathcal{G}r(v, t)$ -morfismo e $h_D: D_1 \rightarrow D_2$ é um \mathcal{D} -morfismo, tal que o diagrama ilustrado na Figura 8.2 comuta.

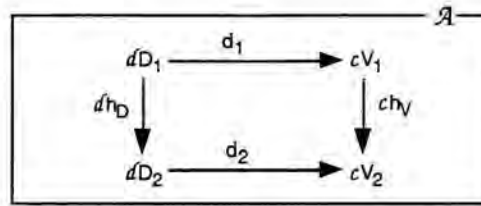


Figura 8.2 Diagrama comutativo para morfismos de grafos estruturados com nodos distinguidos

No contexto das redes de Petri, são considerados somente os grafos estruturados cujos nodos distinguidos são determinados por pares de conjuntos apontados, interpretados como estados iniciais e finais. Por esta razão, a seguinte notação simplificada é adotada.

Notação 8.4 Categoria $\mathcal{D}Gr^*(v, t)$. A categoria denotada por $\mathcal{D}Gr^*(v, t)$ é a categoria cujos nodos distinguidos são objetos de Set^{*2} , isto é, $\mathcal{D}Gr^*(v, t) = \coprod_{Set^*} \downarrow \text{nodos}$ onde $\coprod_{Set^*}: Set^{*2} \rightarrow Set^*$ é o funtor canonicamente induzido pelo coproduto categorial em Set^* e $\text{nodos}: Gr^*(v, t) \rightarrow Set^*$ é o funtor que associa os nodos de um grafo ao seu correspondente conjunto apontado. \square

8.2 Rede de Petri

Inicialmente é introduzida a definição de redes do tipo lugar/transição como em [Reisig 85] e, posteriormente, as redes de Petri consideradas como grafos inspiradas em [Meseguer & Montanari 90].

Definição 8.5 Rede Lugar/Transição. Uma rede (do tipo) lugar/transição é uma tripla $\langle S, T, F \rangle$ tal que S é um conjunto de lugares, T é um conjunto de transições e $F: (S \times T) + (T \times S) \rightarrow \mathbb{N}$ é a relação de dependência causal onde \mathbb{N} representa o conjunto dos números naturais. \square

A relação de dependência causal especifica quantas marcas ("tokens") são consumidas ou produzidas em cada lugar quando uma transição é executada. Por exemplo, $(A, a) \mapsto 3$ e $(a, B) \mapsto 5$ representadas na Figura 8.3, especificam que, quando a transição a é executada, 3 marcas são consumidas no lugar A e 5 marcas são produzidas no lugar B .

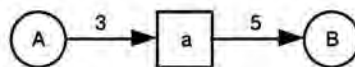


Figura 8.3 Representação gráfica de transições de redes

Considere a Figura 8.4. Uma transição pode ser:

- fonte: não consome marcas e, neste caso, a transição pode sempre ser executada;
- sumidouro: não produz marcas em lugar algum;
- isolada: é simultaneamente fonte e sumidouro.

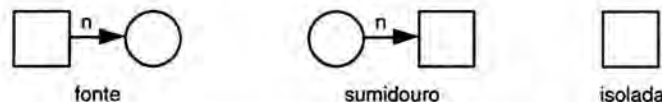


Figura 8.4 Tipos de transições

Por simplicidade, em diagramas de redes do tipo lugar/transição, o valor 1 é omitido quando uma transição consome ou produz uma marca, como ilustrado na Figura 8.5.

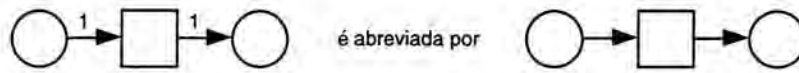


Figura 8.5 Simplificação na representação gráfica de redes

Exemplo 8.6 A rede Prod_Cons = (S, T, F) onde S = {A, B, C, D, E}, T = {p₁, p₂, c₁, c₂} e F é como na tabela abaixo, simula um processo produtor e um consumidor que se comunicam através de um "buffer". A representação gráfica da rede Prod_Cons é ilustrada na Figura 8.6.

(A, p ₁) → 1	(p ₁ , B) → 1	(p ₁ , C) → 2	(B, p ₂) → 1	(p ₂ , A) → 1
(C, c ₁) → 3	(c ₁ , D) → 1	(D, c ₂) → 1	(c ₂ , E) → 1	(E, c ₁) → 1

□

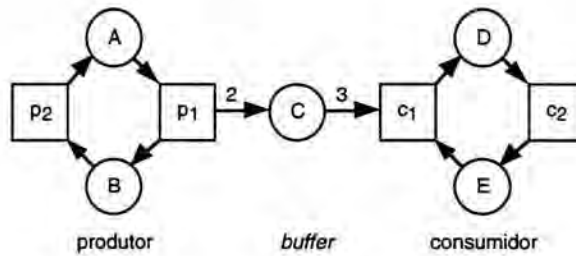


Figura 8.6 Representação gráfica de uma rede de Petri

Para definir uma rede de Petri como um grafo, considera-se que os nodos são elementos de um monóide comutativo. Neste caso, nodos e arcos são considerados como estados e transições de uma rede, respectivamente, onde:

- n marcas consumidas ou produzidas em um estado A é representado por nA;
- para i com valores em {1, ..., k}, a produção ou o consumo simultâneo de n_i marcas no estado A_i, é representado por n₁A₁ ⊕ n₂A₂ ⊕ ... ⊕ n_kA_k onde ⊕ é a operação aditiva do monóide. Como o monóide é comutativo, a ordem dos termos na soma é irrelevante.

Como ilustração, no exemplo anterior, as transições da rede de Petri vista como um grafo seriam as seguintes:

$$p_1: A \rightarrow B \oplus 2C \quad p_2: B \rightarrow A \quad c_1: 3C \oplus E \rightarrow D \quad c_2: D \rightarrow E$$

Portanto, uma rede de Petri pode ser definida como um grafo estruturado onde os nodos são objetos de *CMon* e os arcos são objetos de *Set*.

Definição 8.7 Categoria das Redes de Petri. A categoria das redes de Petri (do tipo lugar/transição), denotada por *Petri*, é a categoria dos grafos estruturados $Gr(v, id_{Set})$ onde $v: CMon \rightarrow Set$ é o funtor que esquece sobre a estrutura do monóide e $id_{Set}: Set \rightarrow Set$ é o funtor identidade. □

Portanto, uma rede de Petri pode ser vista como uma quádrupla $N = \langle V, T, \partial_0, \partial_1 \rangle$ onde $V = \langle V, \oplus, e \rangle$ é um monóide comutativo de estados, T é um conjunto de transições e $\partial_0, \partial_1: id_{Set} T \rightarrow vV$ são funções totais. Uma transição cujo estado de origem ou destino é o elemento neutro do monóide é uma fonte ou um sumidouro, respectivamente. Um morfismo de redes de Petri $h: N_1 \rightarrow N_2$ é uma dupla $h = \langle h_V, h_T \rangle$ onde $h_V: V_1 \rightarrow V_2$ é um *CMon*-morfismo e $h_T: T_1 \rightarrow T_2$ é um *Set*-morfismo tal que os estados de origem e destino das transições são preservados.

Proposição 8.8 A categoria *Petri* é completa e cocompleta.

Prova: Como *Petri* é a categoria dos grafos estruturados $Gr(v, id_{Set})$ é suficiente verificar que $v: CMon \rightarrow Set$ preserva limites. De fato, o funtor $f_{cmon}: Set \rightarrow CMon$ que, para cada conjunto, gera livremente o correspondente monóide comutativo é adjunto esquerdo de v . □

Em *Petri*, o coproduto representa a composição assíncrona e o produto a composição síncrona das redes componentes. Os objetos resultantes do produto e do coproduto das redes $N_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ e $N_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$ são como segue:

$$N_1 \times N_2 = \langle V_1 \times_{CMon} V_2, T_1 \times_{Set} T_2, \partial_{0_1} \times_{Set} \partial_{0_2}, \partial_{1_1} \times_{Set} \partial_{1_2} \rangle$$

$$N_1 + N_2 = \langle V_1 +_{CMon} V_2, T_1 +_{Set} T_2, \partial_{0_1} +_{Set} \partial_{0_2}, \partial_{1_1} +_{Set} \partial_{1_2} \rangle$$

onde os morfismos $\partial_{k_1} \times_{Set} \partial_{k_2}$, $\partial_{k_1} +_{Set} \partial_{k_2}$, para k com valores em $\{0, 1\}$, são univocamente induzidos pelo produto e coproduto em Set , respectivamente. Lembre-se que, em $CMon$, produtos e coprodutos são isomorfos. Um par de transições da forma (t_1, t_2) é denotado por $t_1 | t_2$.

Exemplo 8.9 Considere as Figuras 8.7 e 8.8.

- a) As redes $N_1 + N_2$ e $N_1 \times N_2$ representam os objetos resultantes do coproduto e do produto de N_1 e N_2 , respectivamente.
- b) A rede $N_3 \times N_4$ representa o objeto resultante do produto de N_3 e N_4 . □

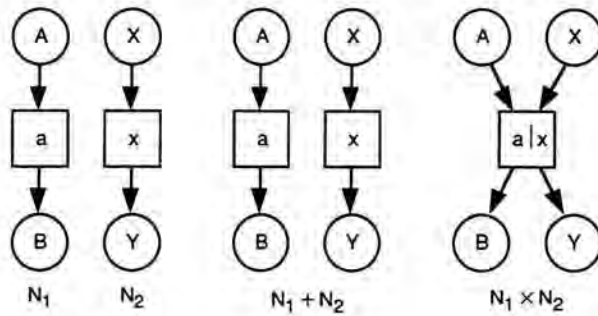


Figura 8.7 Coproduto e produto de redes de Petri

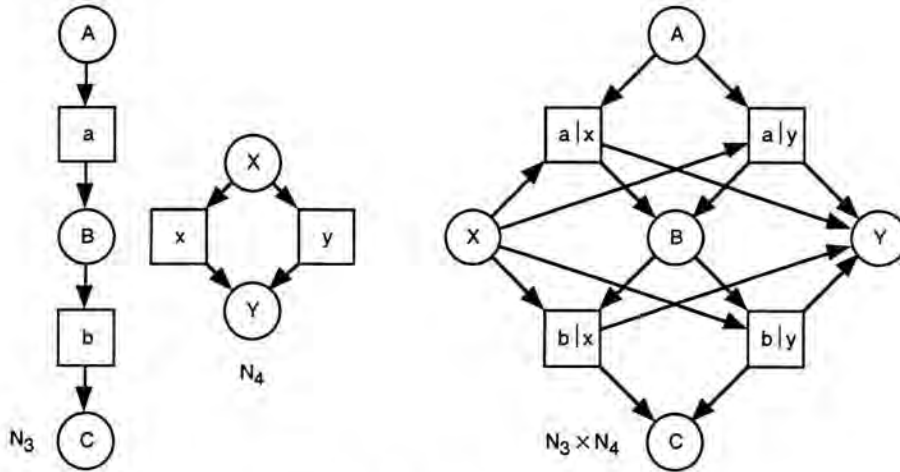


Figura 8.8 Produto de redes de Petri

8.3 Rede de Petri Apontada

A categoria das redes de Petri apontadas é tal que o conjunto das transições de cada rede possui um elemento distinguido e é definida como uma categoria de grafos estruturados internos à Set^* .

Um Set^* -objeto denotado na forma S_e explicita que e é o elemento distinguido. Um $CMon$ -objeto livremente gerado por um conjunto apontado S_e é denotado por $S_e^+ = (S_e^+, +, e)$.

Definição 8.10 Categoria das Redes de Petri Apontada. A categoria das redes de Petri apontadas, denotada por $Petri^*$, é a categoria dos grafos estruturados $Gr(v, id_{Set^*})$ tal que $v: CMon \rightarrow Set^*$ é o funtor que esquece sobre a estrutura do monóide onde o elemento neutro é associado ao elemento distinguido e $id_{Set^*}: Set^* \rightarrow Set^*$ é o funtor identidade. □

Portanto, uma rede de Petri apontada pode ser vista como uma quádrupla $N = \langle V, T_\tau, \partial_0, \partial_1 \rangle$ tal que $V = \langle V, \oplus, e \rangle$ é um monóide comutativo de estados, T_τ é um conjunto apontado de transições e $\partial_0, \partial_1: id_{Set^*} T_\tau \rightarrow vV$ são funções totais apontadas onde τ é associado ao e . O elemento distinguido de T_τ é denominado transição unidade ou sem operação ("no operation"). Note-se que τ é uma transição isolada. Por simplicidade, na representação gráfica de redes de Petri apontadas, a transição τ é omitida, como ilustrado na Figura 8.9.

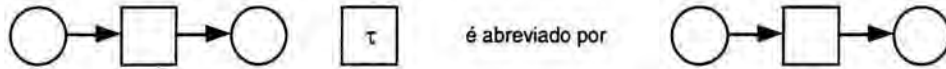


Figura 8.9 Simplificação na representação gráfica de redes apontadas

Um morfismo de redes de Petri apontadas $h: N_1 \rightarrow N_2$ é uma dupla $h = \langle h_V, h_T \rangle$ onde $h_V: V_1 \rightarrow V_2$ é um $CMon$ -morfismo e $h_T: T_{\tau_1} \rightarrow T_{\tau_2}$ é um Set^* -morfismo tal que os estados de origem e destino das transições são preservados. Portanto, para que uma transição seja "esquecida" em um morfismo de redes, é necessário que os seus estados de origem e destino sejam associados ao elemento neutro do monóide de estados da rede destino.

Proposição 8.11 A categoria $Petri^*$ é completa e cocompleta.

Prova: Como a $Petri^*$ é a categoria dos grafos estruturados $Gr(v, id_{Set^*})$ é suficiente verificar que $v: CMon \rightarrow Set^*$ preserva limites. De fato, o funtor $fcmon^*: Set^* \rightarrow CMon$ tal que, para cada conjunto apontado, gera livremente o correspondente monóide comutativo onde o elemento distinguido é associado ao elemento neutro do monóide, é adjunto esquerdo de v . \square

Como o coproduto em Set^* e em Set são isomorfos, o coproduto em $Petri^*$ e em $Petri$ são análogos e possuem a mesma interpretação. Entretanto, a interpretação do produto em $Petri^*$ é diferente e reflete a composição paralela, onde o conjunto das transições da rede resultante possui todas as combinações possíveis das transições das redes componentes (lembre-se que, para os conjuntos A e B , $A_e \times_{Set^*} B_e$ é isomorfo a $A +_{Set^*} (A \times_{Set^*} B) +_{Set^*} B$, onde A_e, B_e são A, B canonicamente estendidos como conjuntos apontados). Os objetos resultantes do produto e do coproduto das redes $N_1 = \langle V_1, T_{\tau_1}, \partial_{01}, \partial_{11} \rangle$ e $N_2 = \langle V_2, T_{\tau_2}, \partial_{02}, \partial_{12} \rangle$ são como segue:

$$N_1 \times N_2 = \langle V_1 \times_{CMon} V_2, T_{\tau_1} \times_{Set^*} T_{\tau_2}, \partial_{01} \times_{Set^*} \partial_{02}, \partial_{11} \times_{Set^*} \partial_{12} \rangle$$

$$N_1 + N_2 = \langle V_1 +_{CMon} V_2, T_{\tau_1} +_{Set^*} T_{\tau_2}, \partial_{01} +_{Set^*} \partial_{02}, \partial_{11} +_{Set^*} \partial_{12} \rangle$$

onde os morfismos $\partial_{k1} \times_{Set^*} \partial_{k2}, \partial_{k1} +_{Set^*} \partial_{k2}$, para k com valores em $\{0, 1\}$, são univocamente induzidos pelo produto e coproduto em Set^* , respectivamente. Os pares de transições da forma (t, τ) ou (τ, t) são denotados simplesmente por t .

Exemplo 8.12 Considere a Figura 8.10. A rede $N_1 \times N_2$ representa o objeto resultante do produto de N_1 e N_2 em $Petri^*$. \square

8.4 Rede de Petri Marcada

Uma rede de Petri marcada (ou com marcação inicial e final) é uma rede de Petri apontada juntamente com um dois conjuntos apontados de estados distinguidos e é definida como um grafo estruturado com nodos distinguidos. Caso seja necessário definir uma categoria de redes de Petri não-apontadas e marcadas o procedimento é análogo.

Definição 8.13 Categoria das Redes de Petri Marcadas. A categoria das redes de Petri marcadas, denotada por \mathcal{MPetri}^* , é a categoria dos grafos estruturados com nodos distinguidos $DGr^*(v, id_{Set^*})$ onde $v: CMon \rightarrow Set^*$ e $id_{Set^*}: Set^* \rightarrow Set^*$ são os mesmos funtores usados na definição da categoria $Petri^* = Gr(v, id_{Set^*})$. \square

Note-se que, na definição acima, o funtor $nodos: Petri^* \rightarrow Set^*$ é tal que associa a cada rede de Petri apontada $N = \langle \langle V_e, \oplus, e \rangle, T_\tau, \partial_0, \partial_1 \rangle$ o seu correspondente conjunto apontado V_e onde o elemento neutro é associado ao elemento distinguido. Uma rede de Petri marcada M é uma quádrupla $M = \langle N, l_e, F_e, i + f \rangle$ onde $N = \langle V, T_\tau, \partial_0, \partial_1 \rangle$ é uma rede de Petri apontada, l_e e F_e são os

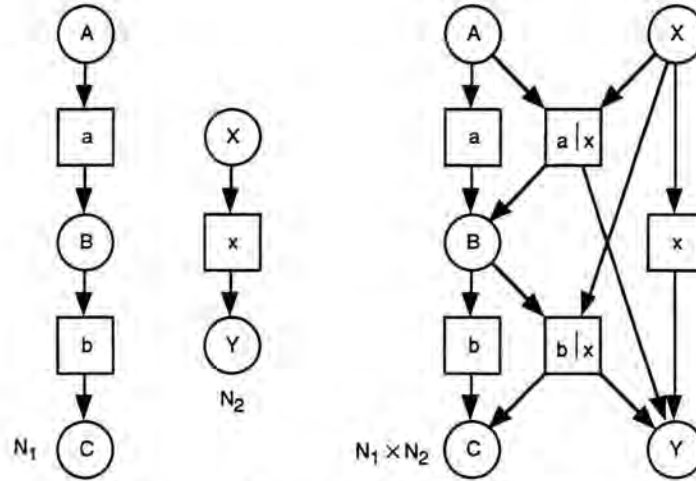


Figura 8.10 Produto de redes de Petri apontadas

conjunto apontado de estados (marcações) iniciais e finais, respectivamente, e $i+f: I_e + F_e \rightarrow \nu V$ é a função total apontada que instancia os estados iniciais e finais na rede. A rede M também pode ser vista como uma 7-upla $M = \langle V, T_\tau, \partial_0, \partial_1, I_e, F_e, i+f \rangle$. Por simplicidade, se $i+f$ é uma inclusão, então M é denotada simplesmente por $M = \langle V, T_\tau, \partial_0, \partial_1, I_e, F_e \rangle$. Um morfismo de redes de Petri marcadas $h: M_1 \rightarrow M_2$ é uma quádrupla $h = \langle h_V, h_T, h_I, h_F \rangle$ onde $h_V = \langle h_V, h_T \rangle: N_1 \rightarrow N_2$ é um *Petri**-morfismo e $h_I: I_{e_1} \rightarrow I_{e_2}$, $h_F: F_{e_1} \rightarrow F_{e_2}$ são *Set**-morfismos e é tal que os estados iniciais e finais são preservados

Proposição 8.14 A categoria \mathcal{MPetri}^* é completa e cocompleta.

Prova: Como \mathcal{MPetri}^* é a categoria das setas \Downarrow_{Set^*} nodos, é suficiente provar que o funtor *nodos*: $\mathcal{Petri}^* \rightarrow Set^*$ preserva limites. Considere o *Set**-objeto inicial $\{\tau\}$ e funtor $p: Set^* \rightarrow \mathcal{Petri}^*$ tal que para todo o conjunto apontado V_e , pV_e é o grafo $\langle V_e^+, \{\tau\}, !, ! \rangle$. É fácil verificar que p é adjunto esquerdo de *nodos*. \square

O produto e o coproduto em \mathcal{MPetri}^* possuem a mesma interpretação que em \mathcal{Petri}^* , ou seja, a composição paralela e a composição assíncrona, respectivamente. Os objetos resultantes do produto e do coproduto das redes $M_1 = \langle N_1, I_{e_1}, F_{e_1}, i_1+f_1 \rangle$ e $M_2 = \langle N_2, I_{e_2}, F_{e_2}, i_2+f_2 \rangle$ são como segue:

$$M_1 \times M_2 = \langle N_1 \times_{\mathcal{Petri}^*} N_2, I_{e_1} \times_{Set^*} I_{e_2}, F_{e_1} \times_{Set^*} F_{e_2}, (i_1+f_1) \times_{Set^*} (i_2+f_2) \rangle$$

$$M_1 + M_2 = \langle N_1 +_{\mathcal{Petri}^*} N_2, I_{e_1} +_{Set^*} I_{e_2}, F_{e_1} +_{Set^*} F_{e_2}, (i_1+f_1) +_{Set^*} (i_2+f_2) \rangle$$

onde os morfismos $(i_1+f_1) \times_{Set^*} (i_2+f_2)$ e $(i_1+f_1) +_{Set^*} (i_2+f_2)$ são univocamente induzidos pelo produto e coproduto categorial, respectivamente.

Em diagramas de redes de Petri marcadas, cada possível marcação inicial ou final é associada a um símbolo de marcação específico. No texto que segue, números com a forma ①, ②, ③, ... representam marcações iniciais e ①, ②, ③, ... representam marcações finais.

Exemplo 8.15 Considere a Figura 8.11. As redes $M_1 + M_2$ e $M_1 \times M_2$ representam os objetos resultantes do coproduto e produto de M_1 e M_2 em \mathcal{MPetri}^* , respectivamente. Nos diagramas, os conjuntos dos estados iniciais e finais são os seguintes:

$$I_{e_1} = \{e, A\}$$

$$I_{e_2} = \{e, X, X \oplus Y\}$$

$$F_{e_1} = \{e, B\}$$

$$F_{e_2} = \{e\}$$

$$I_{e_1} + I_{e_2} = \{e, A, X, X \oplus Y\}$$

$$I_{e_1} \times I_{e_2} = \{e, A, X, X \oplus Y, A \oplus X, A \oplus X \oplus Y\}$$

$$F_{e_1} + F_{e_2} = \{e, B\}$$

$$F_{e_1} \times F_{e_2} = \{e, B\}$$

Para cada marcação, o símbolo associado é como segue:

$$\textcircled{1} \quad A$$

$$\textcircled{2} \quad X$$

$$\textcircled{3} \quad X \oplus Y$$

$$\textcircled{4} \quad A \oplus X$$

$$\textcircled{5} \quad A \oplus X \oplus Y$$

$$\textcircled{6} \quad B$$

\square

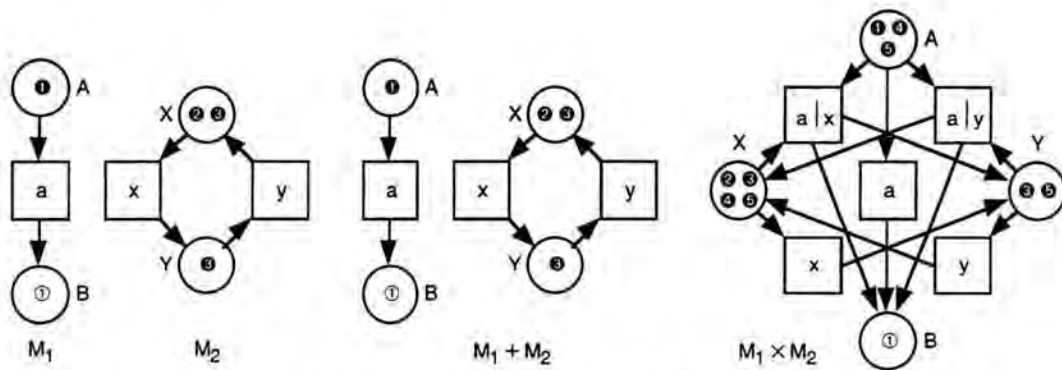


Figura 8.11 Coproduto e produto de redes de Petri marcadas

Observação 8.16 Conjunto de Estados Iniciais \times Um Único Estado Inicial. Uma importante vantagem em adotar um conjunto de estados iniciais (ou conjuntos de estados distinguidos em geral) comparativamente com um único estado inicial como em [Winskel 87] ou em [Meseguer & Montanari 90] é a existência dos coprodutos bem como dos colimites arbitrários. Com o objetivo de garantir a existência de coprodutos, Winskel restringiu suas categorias com (um) estado inicial às redes seguras ("safe nets" - ver, por exemplo, [Olderog 91, p. 30]). Meseguer e Montanari propõem uma solução menos restritiva: um estado inicial pode possuir, no máximo, uma marca em cada lugar. A categoria resultante possui todos os coprodutos. Entretanto, como ilustrado na Figura 8.12, o coproduto de redes reflete uma espécie de "escolha total", com aplicações muito restritas na definição de semânticas. Para mais detalhes sobre redes de Petri com conjuntos de marcações, ver [Menezes 95]. □

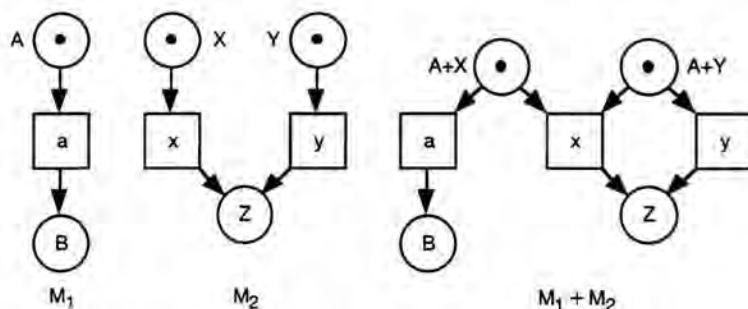


Figura 8.12 Coproduto de redes em Meseguer e Montanari

8.5 Rede de Petri Etiquetada

Uma rede de Petri etiquetada é uma rede de Petri marcada com etiquetação nas transições. A etiquetação é análoga à realizada para os autômatos, ou seja, é um morfismo de redes onde o objeto destino é uma rede de um só estado. Caso seja necessário definir etiquetação para as demais categorias de redes de Petri o procedimento é análogo.

No texto que segue, inicialmente é introduzida a categoria das redes de Petri com um só estado. Lembre-se que um conjunto de um único elemento é um objeto zero de Set^* .

Rede de Petri Apontada de um só Estado

Definição 8.17 Categoria das Redes de Petri Apontada de um só Estado. A categoria das redes de Petri apontadas de um só estado, denotada por $Petri_1^*$, é uma subcategoria da categoria $Petri^*$ onde o monóide comutativo dos estados é um $CMon$ -objeto zero fixo denotado por $e = \langle\{e\}, \oplus, e\rangle$. □

Portanto, uma rede de Petri apontada de um só estado pode ser vista como uma quádrupla $E = \langle e, T_\tau, !, ! \rangle$ onde $! : T_\tau \rightarrow \{e\}$ é único. Um Petri_1^* -morfismo $h = \langle !_e, h_T \rangle : \langle e, T_{\tau_1}, !, ! \rangle \rightarrow \langle e, T_{\tau_2}, !, ! \rangle$ é um morfismo de redes apontadas onde o CMon -morfismo $!_e : e \rightarrow e$ é único. A categoria Petri_1^* é subcategoria plena de Petri^* .

Proposição 8.18 As categorias Petri_1^* e Set^* são isomorfas.

Prova: Considere os funtores:

- a) $s : \text{Petri}_1^* \rightarrow \text{Set}^*$ tal que, para todo Petri_1^* -objeto $E = \langle e, T_\tau, !, ! \rangle$, $sE = T_\tau$ e, para todo Petri_1^* -morfismo $\langle !_e, h_T \rangle : E_1 \rightarrow E_2$, $s\langle !_e, h_T \rangle = h_T$;
- b) $p_1 : \text{Set}^* \rightarrow \text{Petri}_1^*$ tal que, para todo o conjunto apontado T_τ , $p_1 T_\tau = \langle e, T_\tau, !, ! \rangle$ e, para toda a função total de conjuntos apontados $h_T : T_{\tau_1} \rightarrow T_{\tau_2}$, $p_1 h_T = \langle !_e, h_T \rangle : \langle e, T_{\tau_1}, !, ! \rangle \rightarrow \langle e, T_{\tau_2}, !, ! \rangle$.

É fácil verificar que $p_1 \circ s = \text{id}_{\text{Petri}_1^*}$ e $s \circ p_1 = \text{id}_{\text{Set}^*}$. □

Como Petri_1^* e Set^* são isomorfas, então Petri_1^* é completa e cocompleta. Adicionalmente, um Petri_1^* -objeto $\langle e, T_\tau, !, ! \rangle$ e um Petri_1^* -morfismo $\langle !_e, h_T \rangle$ podem ser representados simplesmente por T_τ e h_T , respectivamente.

Rede de Petri Etiquetada

Definição 8.19 Categoria das Redes de Petri Etiquetadas. A categoria das redes de Petri etiquetadas, denotada por EMPetri^* , é a categoria das setas $\text{id} \downarrow \text{inc}$ onde:

- a) $\text{id} : \text{MPetri}^* \rightarrow \text{MPetri}^*$ é o funtor identidade;
- b) $\text{inc} : \text{Petri}_1^* \rightarrow \text{MPetri}^*$ é o funtor inclusão tal que:
 - b.1) para cada Petri_1^* -objeto $E = \langle e, T_\tau, !, ! \rangle$, $\text{inc}E = \langle E, 1, 1, !_{1+1} \rangle$ onde $!_{1+1} : 1 + 1 \rightarrow \{e\}$ é único;
 - b.1) para cada Petri_1^* -morfismo $h = \langle !_e, h_T \rangle : E_1 \rightarrow E_2$, $\text{inc}h = \langle !_e, h_T, !_1, !_1 \rangle : \text{inc}E_1 \rightarrow \text{inc}E_2$ onde $!_1 : 1 \rightarrow 1$ é único. □

Portanto, uma rede de Petri etiquetada P é uma tripla $P = \langle M, E, \text{eti}q \rangle$ tal que $M = \langle N, !_e, F_e, i + f \rangle$ é um MPetri^* -objeto onde a rede N representa a forma de M , E é um Petri_1^* -objeto representando as etiquetas de P e $\text{eti}q : \text{id}M \rightarrow \text{inc}E$ é um MPetri^* -morfismo que representa a etiquetagem das transições de P .

Analogamente aos autômatos, uma rede de Petri etiquetada $P = \langle M, E, \text{eti}q \rangle$ também é denotada por $P : \text{id}M \rightarrow \text{inc}E$ ou simplesmente por $P : M \rightarrow E$ ou ainda como uma 9-upla $P = \langle V, T_\tau, \partial_0, \partial_1, !_e, F_e, i + f, E, \text{eti}q \rangle$. Uma transição etiquetada por τ é dita encapsulada (escondida), ou seja, uma transição que não pode ser acessada externamente à rede. O diagrama de uma rede etiquetada $P : M \rightarrow E$ possui a forma de M , com as etiquetas de E , determinadas pelo morfismo de etiquetagem.

Exemplo 8.20 Considere a rede de Petri apontada $M = \langle \{A, B, C, X, Y\}^\oplus, \{\tau, a, b, x\}, \partial_0, \partial_1, \{e, A \otimes 2X\}, \{e, C \otimes Y\} \rangle$ (monóide livre) determinado pelas transições locais $a : A \rightarrow B$, $b : B \rightarrow C$, $x : 2X \rightarrow Y$, as etiquetas $\{\tau, u, v\}$ e a etiquetagem dada por $a \rightarrow u$, $b \rightarrow u$, $x \rightarrow v$. A representação gráfica da rede etiquetada P é como ilustrado na Figura 8.13 (compare com o correspondente exemplo para autômatos não-sequenciais):

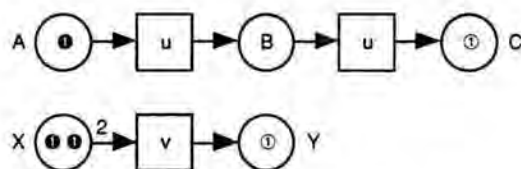


Figura 8.13 Rede de Petri etiquetada

Proposição 8.21 A categoria EMPetri^* é completa e cocompleta.

Prova: As categorias MPetri^* e Petri_1^* são bicompletas. Como EMPetri^* é a categoria das setas $\text{id} \downarrow \text{inc}$, então é suficiente provar que o funtor inc preserva limites. Considere o funtor $p_1 : \text{MPetri}^* \rightarrow \text{Petri}_1^*$ tal que:

- para todo o \mathcal{MPetri}^* -objeto $M = \langle V, T_\tau, \partial_0, \partial_1, l_e, F_e, i + f \rangle$, $p_1 M = \langle e, T_\tau, !, ! \rangle$;
- para todo o \mathcal{MPetri}^* -morfismo $h = \langle h_V, h_T, h_l, h_F \rangle: M_1 \rightarrow M_2$, $p_1 h = \langle l_e, h_T \rangle: \langle e, T_{\tau_1}, !, ! \rangle \rightarrow \langle e, T_{\tau_2}, !, ! \rangle$.

É fácil verificar que p_1 é adjunto esquerdo de inc . Portanto, $\mathcal{EMPetri}^*$ é bicompleta. \square

A interpretação do produto e do coproduto em $\mathcal{EMPetri}^*$ é análoga à interpretação em \mathcal{MPetri}^* . O produto e coproduto de duas redes de Petri etiquetadas $P_1 = \langle M_1, E_1, etiq_1 \rangle$ e $P_2 = \langle M_2, E_2, etiq_2 \rangle$ são como segue:

$$P_1 \times P_2: \langle M_1 \times_{\mathcal{MPetri}^*} M_2, E_1 \times_{\mathcal{Petri}_1^*} E_2, etiq_1 \times_{\mathcal{MPetri}^*} etiq_2 \rangle$$

$$P_1 + P_2: \langle M_1 +_{\mathcal{MPetri}^*} M_2, E_1 +_{\mathcal{Petri}_1^*} E_2, etiq_1 +_{\mathcal{MPetri}^*} etiq_2 \rangle$$

onde os morfismos $etiq_1 \times_{\mathcal{MPetri}^*} etiq_2$ e $etiq_1 +_{\mathcal{MPetri}^*} etiq_2$ são univocamente induzidos pelo produto e coproduto em \mathcal{MPetri}^* , respectivamente.

8.6 Sincronização e Encapsulação

A sincronização e encapsulação para redes de Petri são análogas as dos autômatos. Portanto, para evitar redundâncias, os conceitos e resultados são introduzidos com brevidade e somente para as redes etiquetadas. Para mais detalhes sobre sincronização em redes de Petri, ver [Menezes & Costa 93]. No texto que segue, é omitido que $i \in I$, para algum conjunto de índices I .

Bifibração

Proposição 8.22 O funtor esquecimento $u: \mathcal{EMPetri}^* \rightarrow \mathcal{Petri}_1^*$ que associa a cada rede $P: M \rightarrow E$ a correspondente rede de um só nodo E é uma bifibração.

Prova:

- Seja $f: E_1 \rightarrow E_2$ um \mathcal{Petri}_1^* -morfismo e $P_2 = \langle M_2, E_2, etiq_2 \rangle$ um $\mathcal{EMPetri}^*$ -objeto. Considere o produto fibrado representado na Figura 8.14. Então, $P_1 = \langle M_1, E_1, etiq_1 \rangle$ é um $\mathcal{EMPetri}^*$ -objeto e $u = \langle u_M, f \rangle: P_1 \rightarrow P_2$ é morfismo cartesiano de f e P_2 .
- Considere o \mathcal{Petri}_1^* -morfismo $f: E_1 \rightarrow E_2$ e o $\mathcal{EMPetri}^*$ -objeto $P_1 = \langle M_1, E_1, etiq_1 \rangle$. Sejam $P_2 = \langle M_1, E_2, f \circ etiq_1 \rangle$ um $\mathcal{EMPetri}^*$ -objeto e $u = \langle id_{M_1}, f \rangle: P_1 \rightarrow P_2$ um $\mathcal{EMPetri}^*$ -morfismo. Então, u é cocartesiano de f e P_1 . \square

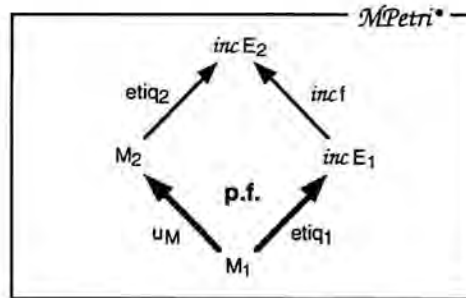


Figura 8.14 Produto fibrado

Sincronização

Definição 8.23 Tabela de Sincronizações, Morfismo de Sincronização, Funtor de Sincronização. Seja $\{P_i: M_i \rightarrow E_i\}$ um conjunto de $\mathcal{EMPetri}^*$ -objetos a serem sincronizados. Então:

- Uma tabela de sincronizações de $\{P_i\}$, denotada por Tabela, é um \mathcal{Petri}_1^* -objeto que reflete a sincronização desejada ao nível das etiquetas.
- Um morfismo de sincronização $sinc: Tabela \rightarrow \times E_i$ é um \mathcal{Petri}_1^* -monomorfismo.

- c) $\text{sinc}: \mathcal{U}^{-1} \times E_i \rightarrow \mathcal{U}^{-1} \text{Tabela}$ é o funtor de sincronização induzido pelo funtor \mathcal{U} , pela construção dos correspondentes morfismos cartesianos como na prova da Proposição 8.22 e pelo morfismo de sincronização sinc . O $\mathcal{EM} \text{Petri}^*$ -objeto resultante da sincronização de $\{P_i\}$, denotado por $\parallel_{\text{sinc}} P_i$ é $\text{sinc} \times P_i$. \square

Na definição abaixo, lembre-se que Petri_1^* é isomorfa a Set^* e que, em Set^* , $\times E_i$ é isomorfo a $(+_{\text{Set}} E_i) +_{\text{Set}} (\times_{\text{Set}} E_i) +_{\text{Set}} \{\tau\}$ onde E_i' é E_i sem o elemento distinguido e , portanto, existe um monomorfismo $\text{inj}_i: E_i \rightarrow \times E_i$ canonicamente induzido pelo coproduto categorial.

Definição 8.24 Construção Categorical: Tabela de Sincronizações, Morfismo de Sincronização. Seja $\{P_i: M_i \rightarrow E_i\}$ um conjunto de $\mathcal{EM} \text{Petri}^*$ -objetos a serem sincronizados. Então:

- a) Sejam:
- a.1) Canal a menor sub-rede de $\times E_i$ que representa as etiquetas a serem sincronizadas;
 - a.2) para cada i , E_i' a menor sub-rede de E_i que contém todas as etiquetas das transições de P_i as quais chamam alguma transição da outra rede;
 - a.3) para cada i , o morfismo chama $_i: E_i' \rightarrow$ Canal é tal que se a em E_i' chama as transições x_1, \dots, x_n , então $\text{chama}_i(a) = a \mid x_1 \mid \dots \mid x_n$;
 - a.4) D o diagrama representado na Figura 8.15 onde, para cada i , $\text{sub}_i: E_i' \rightarrow E_i$ é um monomorfismo que caracteriza E_i' como subobjeto de E_i .

Então, a tabela de sincronizações denotada por Tabela é o objeto resultante do colimite do diagrama D .

- b) Sejam $s: \text{Canal} \rightarrow \times E_i$ um monomorfismo que caracteriza Canal como sub-rede de $\times E_i$ e $\text{inj}_i: E_i \rightarrow \times E_i$ monomorfismos canonicamente induzidos pelo coproduto categorial. O morfismo de sincronização determinado por D é $\text{sinc}: \text{Tabela} \rightarrow \times E_i$ tal que $\text{sinc}(\tau) = \tau$ e, para todo $a \neq \tau$ em Tabela, se $q^R(a) \neq \tau$, então $\text{sinc}(a) = s \circ q^R(a)$, caso contrário existe um único k em I tal que $p_k^R(a) \neq \tau$ e $\text{sinc}(a) = \text{inj}_k \circ p_k^R(a)$. \square

Na definição acima, as provas de que, para cada i , p_i^R e q são retrações de p_i e q ,

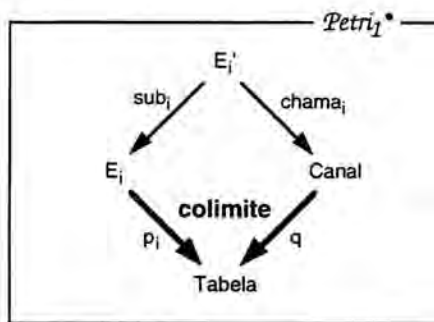


Figura 8.15 Tabela de sincronizações

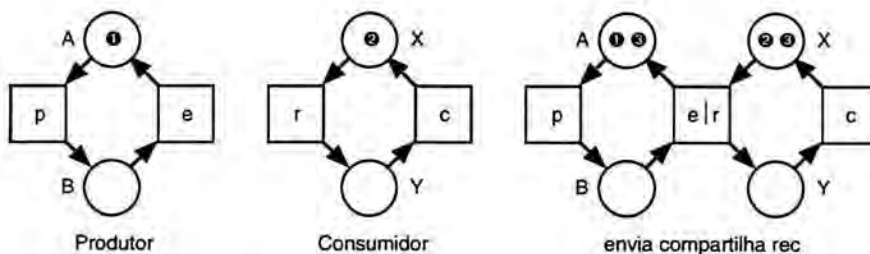


Figura 8.16 Sincronização de redes

respectivamente, é análoga à correspondente prova para os autômatos.

Exemplo 8.25 Considere a Figura 8.16. onde as etiquetas envia, prod, rec, cons são abreviadas para e, p, r, c, respectivamente. Suponha que é desejada a sincronização envia compartilha rec entre as redes Produtor e Consumidor. Então, o canal é $\text{Canal} = \{\tau, \text{envia} \mid \text{rec}\}$, a tabela de sincronização é $\text{Tabela} = \{\tau, \text{prod}, \text{cons}, \text{envia} \mid \text{rec}\}$ e a rede resultante é como representada na Figura 8.16 (direita). Compare com os correspondentes exemplos para autômatos. \square

Encapsulação

Definição 8.26 Morfismo de Encapsulação, Funtor de Encapsulação.

- a) Seja E_1 as etiquetas da rede considerada.
- a.1) Dado o subobjeto de etiquetas E_2 a serem exportadas. Seja $\text{sub}: E_2 \rightarrow E_1$ um monomorfismo que caracteriza E_2 como subobjeto de E_1 . Então $\text{caps} = \text{sub}^R: E_1 \rightarrow E_2$ onde sub^R é a retração de sub tal que, para todo a_1 em E_1 , se existe a_2 em E_2 tal que $\text{sub}(a_2) = a_1$, então $\text{sub}^R(a_1) = a_2$, caso contrário, $\text{sub}^R(a_1) = \tau$.
 - a.2) Dado o subobjeto E de etiquetas a serem encapsuladas. Seja $\text{sub}: E \rightarrow E_1$ um monomorfismo que caracteriza E como subobjeto de E_1 . Então o morfismo de encapsulação $\text{caps}: E_1 \rightarrow E_2$ induzido por sub é determinado pela soma amalgamada ilustrada na Figura 8.17, onde 1 é um Petri_1^* -objeto terminal e o morfismo $!$ é único.
- b) Considere o EMPetri^* -objeto $P: M \rightarrow E_1$, o morfismo de encapsulação $\text{caps}: E_1 \rightarrow E_2$, a cofibração $u: \text{EMPetri}^* \rightarrow \text{Petri}_1^*$, a construção dos correspondentes morfismos cocartesianos como na prova da Proposição 8.22 e as fibras $u^{-1}E_1, u^{-1}E_2$. Então, $\text{caps}: u^{-1}E_1 \rightarrow u^{-1}E_2$ é o funtor de encapsulação induzido por u e caps e a rede resultante da encapsulação de P , denotado por $P \setminus \text{caps}$, é $\text{caps}P$. \square

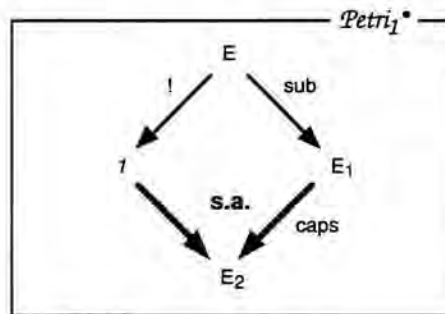


Figura 8.17 Morfismo de encapsulação determinado por uma soma amalgamada

9 Relacionamento entre Autômatos e Redes de Petri

O relacionamento entre autômatos seqüenciais, não-seqüenciais e redes de Petri etiquetadas é realizado usando funtores, inspirado em [Sassone *et al* 93]. Para comparar a expressividade entre os autômatos não-seqüenciais e as redes de Petri, é introduzida uma subcategoria dos autômatos não-seqüenciais, denotada por \mathcal{NAut}^* onde o monóide das transições e os correspondentes morfismos são livremente gerados por objetos e morfismos em Set^* , respectivamente. Os funtores entre as diversas categorias são ilustrados na Figura 9.1, sendo as adjunções representadas por setas sombreadas onde o sentido da seta é o mesmo do adjunto esquerdo. Note-se que \mathcal{NAut}^* é isomorfa a $EMPetri^*$ e, portanto, a categoria das redes de Petri é subcategoria dos autômatos não-seqüenciais. Assim, pode-se afirmar que os autômatos não-seqüenciais são mais concretos que as redes de Petri, no sentido em que \mathcal{NAut} é capaz de expressar objetos e morfismos os quais não são possíveis em $EMPetri^*$, sem que alguma informação seja perdida. De fato, nem todo \mathcal{NAut} -objeto é imagem de alguma rede e o functor de $EMPetri^*$ para \mathcal{NAut} não é pleno.

Compondo adjunções, define-se a reificação para redes de Petri etiquetadas, onde o fecho computacional não-seqüencial é obtido em \mathcal{NComp} . Entretanto, é fácil verificar que as redes de Petri não satisfazem à composicionalidade horizontal. Entre os modelos relacionados (bem como entre os modelos que se tem conhecimento), \mathcal{NAut} é a categoria menos concreta (ou a mais abstrata) que satisfaz à composicionalidade diagonal, considerando reificações como morfismos de

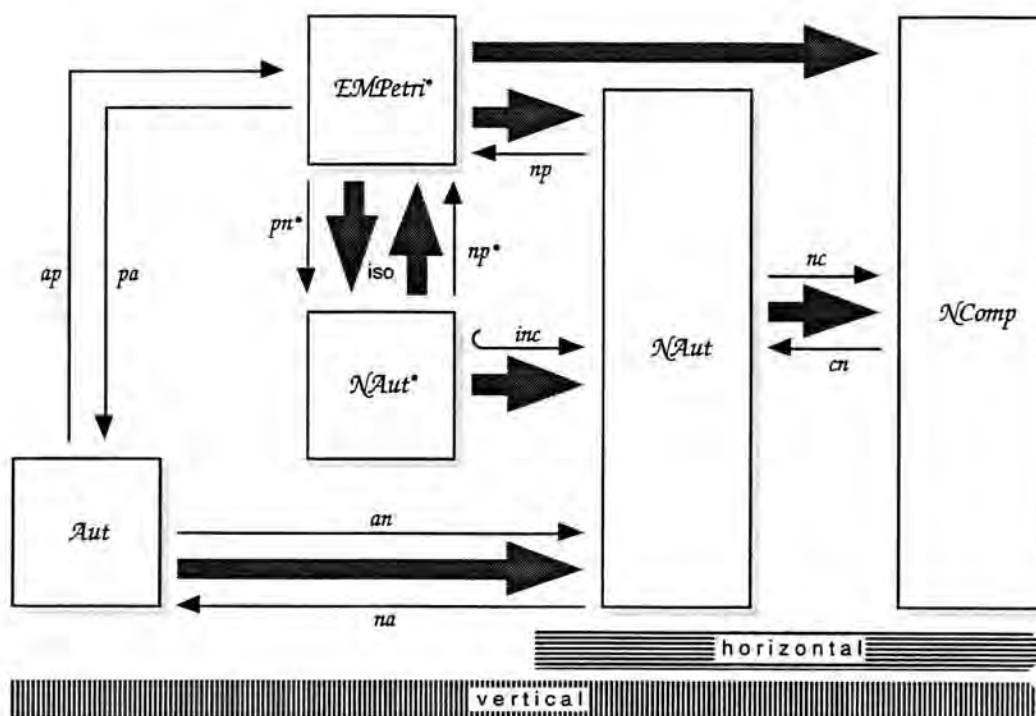


Figura 9.1 Relacionamento entre as diversas categorias

implementação em fechados computacionais. O relacionamento entre redes de Petri e autômatos não-seqüenciais é introduzido em [Menezes & Costa 95] e [Menezes & Costa 95b]. Esta dissertação estende o relacionamento para os autômatos seqüenciais.

As unidades e co-unidades das adjunções $\langle an, na \rangle: \mathcal{A}ut \rightarrow \mathcal{N}(\mathcal{A}ut)$ e $\langle inc \circ pn^*, np \rangle: \mathcal{EM}Petri^* \rightarrow \mathcal{N}(\mathcal{A}ut)$ não são isomorfismos naturais e, portanto, as adjunções não são reflexões nem co-reflexões. Os funtores ap e pa não constituem uma adjunção. A obtenção da adjunção respeitando a noção intuitiva de tradução entre os modelos, requer alterações básicas ou nos autômatos ou nas redes, descaracterizando as categorias apresentadas.

No texto que segue, suponha k com valores em $\{0, 1\}$ e i com valores em $\{1, 2\}$. Adicionalmente, a seguinte notação é adotada (algumas já foram introduzidas anteriormente):

- um $CMon$ -objeto M é denotado por $M = \langle M, \oplus, e \rangle$;
- para um conjunto S , S_e é o Set^* -objeto canonicamente gerado por S e $S^\oplus = \langle S^\oplus, \oplus, e \rangle$ e $S_{e^\oplus} = \langle S_{e^\oplus}, \oplus, e \rangle$ são os $CMon$ -objetos livremente gerados por S e S_e , respectivamente;
- para uma função total $f: A \rightarrow B$ o morfismo $f_e: A_e \rightarrow B_e$ denota o único Set^* -morfismo que estende f . Adicionalmente, $f^\oplus: A^\oplus \rightarrow B^\oplus$ e $f_{e^\oplus}: A_{e^\oplus} \rightarrow B_{e^\oplus}$ denotam os únicos $CMon$ -morfismos que estendem f e f_e , respectivamente;
- para um determinado estado X , a correspondente transição identidade é denotada por $X: X \rightarrow X$;
- uma transição t etiquetada por a é denotada por $a[t]$.

9.1 Autômatos Seqüenciais e Redes de Petri Etiquetadas

Uma justificativa para que os funtores ap e pa não constituam uma adjunção é o enriquecimento de estruturas nos dois sentidos, como segue: a estrutura monoidal dos estados na passagem para as redes e a reflexividade dos grafos na passagem para os autômatos. Uma solução seria definir as redes de Petri sobre os grafos reflexivos. Entretanto, a interpretação do produto na categoria resultante seria inadequada para a abordagem proposta.

Embora os funtores ap e pa não constituam uma adjunção, refletem adequadamente a interpretação do comportamento. O funtor ap é, basicamente, uma inclusão. De fato, qualquer autômato seqüencial pode ser facilmente simulado por uma rede. O funtor pa associa a cada rede o correspondente autômato que reflete a intercalação seqüencial das transições concorrentes. Portanto, o funtor $ap \circ pa: \mathcal{EM}Petri^* \rightarrow \mathcal{EM}Petri^*$ aplicado a uma rede resulta em uma segunda rede onde as transições concorrentes são substituídas por intercalações seqüenciais. Adicionalmente, a rede resultante é segura ("safe"), pois as marcações atingíveis a partir da marcação inicial possuem, no máximo, uma marca em cada lugar. Isto é consequência direta da definição de autômato seqüencial onde, a cada momento, no máximo uma transição pode ser executada.

O funtor $ap: \mathcal{A}ut \rightarrow \mathcal{EM}Petri^*$ definido a seguir é, resumidamente, como segue:

- estende o conjunto de estados com estrutura monoidal comutativa livremente gerada;
- estende o conjunto de transições com um elemento distinguido;
- esquece a função total t , embora mantenha as transições identidade;
- os morfismos origem, destino, instanciação dos estados distinguidos e etiquetação de uma rede são univocamente induzidos pelas correspondentes funções totais considerando o monóide dos estados e o conjunto apontado de transições;
- um morfismo de redes é univocamente induzidos considerando os monóides nos estados e o conjunto apontado nas transições.

Definição 9.1 Funtor ap . O funtor $ap: \mathcal{A}ut \rightarrow \mathcal{EM}Petri^*$ é tal que:

a) seja $A = \langle M, E, \text{eti} \rangle$ um $\mathcal{A}ut$ -objeto onde $M = \langle G, l, F, i + f \rangle$ e $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$. Então:

- a.1) o \mathcal{RGr} -objeto G é associado ao \mathcal{Petri}^* -objeto $G' = \langle V_e^\oplus, T_\tau, \partial_0', \partial_1' \rangle$ onde o Set^* -morfismo $\partial_k': T_\tau \rightarrow V_e^\oplus$ é a função total $\partial_k: T \rightarrow V$ canonicamente estendida em relação ao elemento distinguido e ao monóide gerado;

- a.2) o \mathcal{DRGr} -objeto M é associado ao \mathcal{MPetri}^* -objeto $M' = \langle G', l_e, F_e, i' + f' \rangle$ onde os \mathcal{Set}^* -morfismos $i': l_e \rightarrow V_e^\oplus$ e $f': F_e \rightarrow V_e^\oplus$ são as extensões canônicas de $i: l \rightarrow V$ e $f: F \rightarrow V$, respectivamente;
- a.3) o \mathcal{Aut} -objeto A é associado ao $\mathcal{EMPetri}^*$ -objeto $P = \langle M', E_\tau, \text{eti}' \rangle$ onde $\text{eti}' : T_\tau \rightarrow E_\tau$ é o único \mathcal{Set}^* -morfismo que estende $\text{eti} : T \rightarrow E$.
- b) Sejam $A_i = \langle M_i, E_i, \text{eti}_i \rangle$ \mathcal{Aut} -objetos onde $M_i = \langle G_i, l_i, F_i, i_i + f_i \rangle$ e $G_i = \langle V_i, T_i, \partial_{0i}, \partial_{1i}, l_i \rangle$ e $P_i \in \mathcal{EMPetri}^*$ -objetos construídos a partir dos correspondentes A_i como no item anterior. Seja $h = \langle h_M, h_E \rangle : A_1 \rightarrow A_2$ um \mathcal{Aut} -morfismo onde $h_M = \langle h_G, h_l, h_f \rangle : M_1 \rightarrow M_2$ e $h_G = \langle h_V, h_T \rangle : G_1 \rightarrow G_2$. Então:
- b.1) o \mathcal{RGr} -morfismo h_G é associado ao \mathcal{Petri}^* -morfismo $h_G' = \langle h_V', h_T' \rangle : G_1' \rightarrow G_2'$ onde $h_V' : V_{e_1}^\oplus \rightarrow V_{e_2}^\oplus$ é o único \mathcal{CMon} -morfismo que estende $h_V : V_1 \rightarrow V_2$ e $h_T' : T_{\tau_1} \rightarrow T_{\tau_2}$ é o único \mathcal{Set}^* -morfismo que estende $h_T : T_1 \rightarrow T_2$;
- b.2) o \mathcal{DRGr} -morfismo h_M é associado ao \mathcal{MPetri}^* -morfismo $h_M' = \langle h_G', h_l', h_f' \rangle : M_1' \rightarrow M_2'$ onde $h_l' : l_{e_1} \rightarrow l_{e_2}$ e $h_f' : F_{e_1} \rightarrow F_{e_2}$ são os únicos \mathcal{Set}^* -morfismos que estendem $h_l : l_1 \rightarrow l_2$ e $h_f : F_1 \rightarrow F_2$, respectivamente;
- b.3) o \mathcal{Aut} -morfismo h é associado ao $\mathcal{EMPetri}^*$ -morfismo $h' = \langle h_M', h_E' \rangle : P_1 \rightarrow P_2$ onde $h_E' : E_{\tau_1} \rightarrow E_{\tau_2}$ é o único \mathcal{Set}^* -morfismo que estende $h_E : E_1 \rightarrow E_2$. \square

Exemplo 9.2 A Figura 9.2 ilustra o funtor ap aplicado a um autômato seqüencial. Para facilitar o entendimento, as transições identidade do autômato *não* são omitidas na representação gráfica. \square

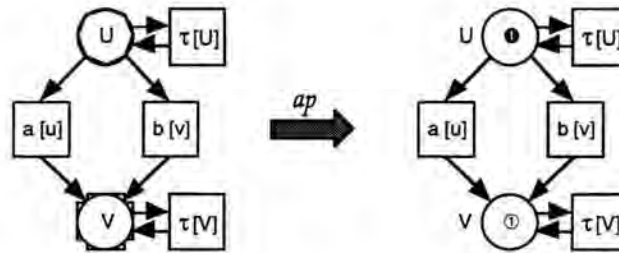


Figura 9.2 Um autômato seqüencial e a correspondente rede de Petri etiquetada

O funtor $pa : \mathcal{EMPetri}^* \rightarrow \mathcal{Aut}$ definido a seguir é, resumidamente, como segue:

- esquece a estrutura monoidal dos estados;
- o conjunto de transições do autômato é o objeto resultante do produto em \mathcal{Set} entre o conjunto das transições e o conjunto suporte do monóide de estados. As funções de origem, destino e etiquetação são definidas considerando o mesmo produto. A etiquetação considera somente a componente referente a transição da rede (ou seja, não considera a componente referente ao estado). Assim, por exemplo, para as transições etiquetadas de uma rede $a : A \rightarrow B$ e $x : X \rightarrow Y$ as transições etiquetadas $a : A \oplus X \rightarrow B \oplus X$, $a : A \oplus Y \rightarrow B \oplus Y$, $x : A \oplus X \rightarrow A \oplus Y$ e $x : B \oplus X \rightarrow B \oplus Y$ pertencem ao autômato correspondente. Note-se que estas transições formam uma espécie de "quadrado de independência" refletindo a intercalação das transições concorrentes para uma marcação $A \oplus X$ na rede;
- introduz a função total identidade.

Definição 9.3 **Functor pa .** O funtor $pa : \mathcal{EMPetri}^* \rightarrow \mathcal{Aut}$ é tal que:

- a) seja $P = \langle M, E_\tau, \text{eti} \rangle$ um $\mathcal{EMPetri}^*$ -objeto onde $M = \langle G, l_e, F_e, i + f \rangle$ e $G = \langle V_e, \oplus, e \rangle, T_\tau, \partial_0, \partial_1 \rangle$. Então:
- a.1) o \mathcal{Petri}^* -objeto G é associado ao \mathcal{RGr} -objeto $G' = \langle V_e, T_\tau \times_{\mathcal{Set}} V_e, \partial_0', \partial_1', l \rangle$ onde $t : V_e \rightarrow T_\tau \times_{\mathcal{Set}} V_e$ é tal que, para todo v em V_e , $l(v) = (\tau, v)$ e onde $\partial_k' = \oplus \circ (\partial_k \times \text{id}_{V_e}) : T_\tau \times_{\mathcal{Set}} V_e \rightarrow V_e$ é tal que $\partial_k \times \text{id}_{V_e}$ é univocamente induzida pelo produto em \mathcal{Set} , como ilustrado na Figura 9.3;
- a.2) o \mathcal{MPetri}^* -objeto M é associado ao \mathcal{DRGr} -objeto $M' = \langle G', l_e, F_e, i + f \rangle$;
- a.3) o $\mathcal{EMPetri}^*$ -objeto P é associado ao \mathcal{Aut} -objeto $A = \langle M', E_\tau, \text{eti}' \rangle$ onde $\text{eti}' : T_\tau \times_{\mathcal{Set}} V_e \rightarrow E_\tau$ é tal que, para todo (t, v) em $T_\tau \times_{\mathcal{Set}} V_e$, $\text{eti}'(t, v) = \text{eti}(t)$.

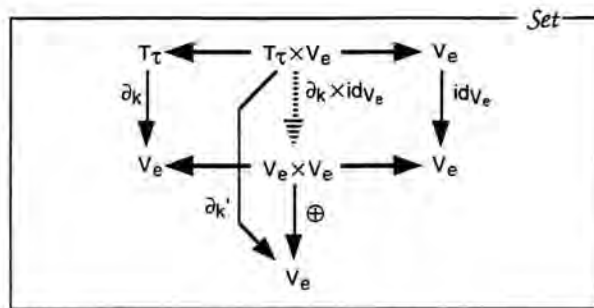


Figura 9.3 Funções totais de origem e destino

- b) Sejam $P_i = \langle M_i, E_{\tau_i}, etiq_i \rangle$ *EMPetri**-objetos tais que $M_i = \langle G_i, l_{e_i}, F_{e_i}, i_i + f_i \rangle$ e $G_i = \langle \langle V_{e_i}, \oplus, e \rangle, T_{\tau_i}, \partial_{0_i}, \partial_{1_i} \rangle$ e A_i os *Aut*-objetos construídos a partir dos correspondentes P_i como especificado no item anterior. Seja $h = \langle h_M, h_E \rangle: P_1 \rightarrow P_2$ um *EMPetri**-morfismo onde $h_M = \langle h_G, h_l, h_f \rangle: M_1 \rightarrow M_2$ e $h_G = \langle h_V, h_T \rangle: G_1 \rightarrow G_2$. Então:
- b.1) o *Petri**-morfismo h_G é associado ao *RGr*-morfismo $h_G' = \langle h_V', h_T' \rangle: G_1' \rightarrow G_2'$ onde $h_V': V_{e_1} \rightarrow V_{e_2}$ é a função total canonicamente induzida pelo *CMon*-morfismo $h_V: \langle V_{e_1}, \oplus, e \rangle \rightarrow \langle V_{e_2}, \oplus, e \rangle$ e onde $h_T' = h_T \times h_V: T_{\tau_1} \times V_{e_1} \rightarrow T_{\tau_2} \times V_{e_2}$ é o *Set*-morfismo univocamente induzido pelo produto categorial;
 - b.2) o *MPetri**-morfismo h_M é associado ao *DRGr*-morfismo $h_M' = \langle h_G', h_l, h_f \rangle: M_1' \rightarrow M_2'$;
 - b.3) o *EMPetri**-morfismo $h = \langle h_M, h_E \rangle: P_1 \rightarrow P_2$ é associado ao *Aut*-morfismo $h' = \langle h_M', h_E \rangle: A_1 \rightarrow A_2$. □

Exemplo 9.4 Considere as Figuras 9.4 a 9.6. Em cada figura, é apresentada a representação gráfica de uma rede de Petri etiquetada e o correspondente autômato seqüencial, de acordo com o functor *pa*. Somente na primeira figura as transições identidade do autômato resultante são representadas. Note-se que, na construção de um autômato correspondente a uma rede, os estados distinguidos não são importantes para determinar a forma resultante. Assim, nas figuras, a

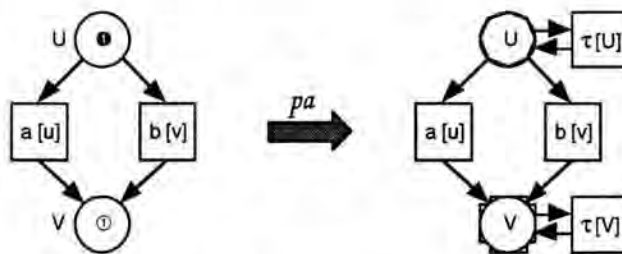


Figura 9.4 Escolha

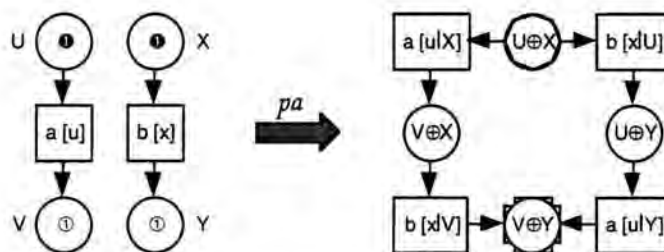


Figura 9.5 Concorrência

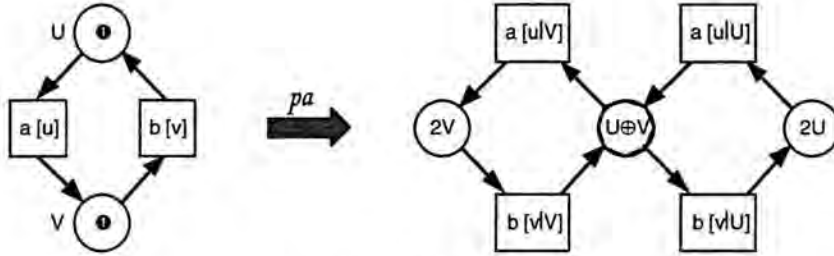


Figura 9.6 Ciclo

representação gráfica dos autômatos omite as transições que não são atingíveis a partir dos estados iniciais. Por exemplo, na Figura 9.4, é omitida a transição etiquetada $a[\tau|V]: U \oplus V \rightarrow 2V$ do autômato resultante. \square

9.2 Autômatos Sequenciais e Não-Sequenciais

Na adjunção $\langle an, na \rangle: \mathcal{Aut} \rightarrow \mathcal{NAut}$, o funtor an quando aplicado a um \mathcal{Aut} -objeto resulta em um \mathcal{NAut} -objeto gerado livremente a partir do autômato sequencial. O adjunto direito na simplesmente esquece sobre as estruturas monoidais e, portanto, as transições independentes do \mathcal{NAut} -objeto determinam, no correspondente \mathcal{Aut} -objeto, uma intercalação sequencial juntamente com uma sincronização simétrica das transições componentes.

Definição 9.5 Funtor an . O funtor $an: \mathcal{Aut} \rightarrow \mathcal{NAut}$ é tal que:

- a) seja $A = \langle M, E, \text{eti} \rangle$ um \mathcal{Aut} -objeto onde $M = \langle G, l, F, i + f \rangle$ e $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$. Então:
 - a.1) o \mathcal{RGr} -objeto G é associado ao $\mathcal{RGr}(\mathcal{CMon})$ -objeto $G' = \langle V^\oplus, T^l, \partial_0', \partial_1', \iota' \rangle$ onde $\partial_k': T^l \rightarrow V^\oplus$ e $\iota': V^\oplus \rightarrow T^l$ são os únicos \mathcal{CMon} -morfismos que estendem $\partial_k: T \rightarrow V$ e $\iota: V \rightarrow T$, respectivamente;
 - a.2) o \mathcal{DRGr} -objeto M é associado ao $\mathcal{DRGr}(\mathcal{CMon})$ -objeto $M' = \langle G', l^\oplus, F^\oplus, i' + f' \rangle$ onde $i': l^\oplus \rightarrow V^\oplus$ e $f': F^\oplus \rightarrow V^\oplus$ são os únicos \mathcal{CMon} -morfismos que estendem $i: l \rightarrow V$ e $f: F \rightarrow V$, respectivamente;
 - a.3) o \mathcal{Aut} -objeto A é associado ao \mathcal{NAut} -objeto $N = \langle M', E^l, \text{eti} \rangle$ onde $\text{eti}': T^l \rightarrow E^l$ é o único \mathcal{CMon} -morfismo que estende $\text{eti}: T \rightarrow E$.
- b) Sejam $A_i = \langle M_i, E_i, \text{eti}_i \rangle$ \mathcal{Aut} -objetos tais que $M_i = \langle G_i, l_i, F_i, i_i + f_i \rangle$ e $G_i = \langle V_i, T_i, \partial_{0i}, \partial_{1i}, \iota_i \rangle$ e sejam N_i os \mathcal{NAut} -objetos construídos a partir dos correspondentes A_i como no item anterior. Seja $h = \langle h_M, h_E \rangle: A_1 \rightarrow A_2$ um \mathcal{Aut} -morfismo tal que $h_M = \langle h_G, h_l, h_F \rangle: M_1 \rightarrow M_2$ e $h_G = \langle h_V, h_T \rangle: G_1 \rightarrow G_2$. Então:
 - b.1) o \mathcal{RGr} -morfismo h_G é associado ao $\mathcal{RGr}(\mathcal{CMon})$ -morfismo $h_G' = \langle h_V', h_T' \rangle: G_1' \rightarrow G_2'$ onde $h_V': V_1^\oplus \rightarrow V_2^\oplus$ e $h_T': T_1^l \rightarrow T_2^l$ são os únicos \mathcal{CMon} -morfismos que estendem $h_V: V_1 \rightarrow V_2$ e $h_T: T_1 \rightarrow T_2$, respectivamente;
 - b.2) o \mathcal{DRGr} -morfismo h_M é associado ao $\mathcal{DRGr}(\mathcal{CMon})$ -morfismo $h_M' = \langle h_G', h_l', h_F' \rangle: M_1' \rightarrow M_2'$ onde $h_l': l_1^\oplus \rightarrow l_2^\oplus$ e $h_F': F_1^\oplus \rightarrow F_2^\oplus$ são os únicos \mathcal{CMon} -morfismos que estendem $h_l: l_1 \rightarrow l_2$ e $h_F: F_1 \rightarrow F_2$, respectivamente;
 - b.3) o \mathcal{Aut} -morfismo h é associado ao \mathcal{NAut} -morfismo $h' = \langle h_M', h_E' \rangle: N_1 \rightarrow N_2$ onde $h_E': E_1^\oplus \rightarrow E_2^\oplus$ é o único \mathcal{CMon} -morfismo que estende $h_E: E_1 \rightarrow E_2$. \square

Definição 9.6 Funtor na . O funtor $na: \mathcal{NAut} \rightarrow \mathcal{Aut}$ é tal que:

- a) seja $N = \langle M, E, \text{eti} \rangle$ um \mathcal{NAut} -objeto onde $M = \langle G, l, F, i + f \rangle$ e $G = \langle V, T, \partial_0, \partial_1, \iota \rangle$. Então:
 - a.1) o $\mathcal{RGr}(\mathcal{CMon})$ -objeto G é associado ao \mathcal{RGr} -objeto $G' = \langle V, T, \partial_0', \partial_1', \iota' \rangle$ onde $\partial_k': T \rightarrow V$ e $\iota': V \rightarrow T$ são as funções totais canonicamente induzidas pelos \mathcal{CMon} -morfismos $\partial_k: T \rightarrow V$ e $\iota: V \rightarrow T$, respectivamente;
 - a.2) o $\mathcal{DRGr}(\mathcal{CMon})$ -objeto M é associado ao \mathcal{DRGr} -objeto $M' = \langle G', l, F, i' + f' \rangle$ onde i' e f' são as funções totais canonicamente induzidas pelos \mathcal{CMon} -morfismos i e f , respectivamente;

- a.3) o $\mathcal{N}Aut$ -objeto N é associado ao Aut -objeto $A = \langle M', E, \text{eti}' \rangle$ onde eti' é a função total canonicamente induzida pelo $CMon$ -morfismo eti .
- b) Sejam $N_i = \langle M_i, E_i, \text{eti}_i \rangle$ $\mathcal{N}Aut$ -objetos tais que $M_i = \langle G_i, l_i, F_i, i_i + f_i \rangle$ e $G_i = \langle V_i, T_i, \partial_{0i}, \partial_{1i}, t_i \rangle$ e sejam A_i os $\mathcal{N}Aut$ -objetos construídos a partir dos correspondentes N_i como no item anterior. Seja $h = \langle h_M, h_E \rangle: N_1 \rightarrow N_2$ um $\mathcal{N}Aut$ -morfismo tal que $h_M = \langle h_G, h_l, h_f \rangle: M_1 \rightarrow M_2$ e $h_G = \langle h_V, h_T \rangle: G_1 \rightarrow G_2$. Então:
- b.1) o $\mathcal{RGr}(CMon)$ -morfismo h_G é associado ao \mathcal{RGr} -morfismo $h_G' = \langle h_V', h_T' \rangle: G_1' \rightarrow G_2'$ onde $h_V': V_1 \rightarrow V_2$ e $h_T': T_1 \rightarrow T_2$ são as funções totais canonicamente induzidas pelos $CMon$ -morfismos $h_V: V_1 \rightarrow V_2$ e $h_T: T_1 \rightarrow T_2$, respectivamente.
- b.2) o $\mathcal{DRGr}(CMon)$ -morfismo h_M é associado ao \mathcal{DRGr} -morfismo $h_M' = \langle h_G', h_l', h_f' \rangle: M_1' \rightarrow M_2'$ onde $h_l': l_1 \rightarrow l_2$ e $h_f': F_1 \rightarrow F_2$ são as funções totais canonicamente induzidas pelo $CMon$ -morfismo $h_l: l_1 \rightarrow l_2$ e $h_f: F_1 \rightarrow F_2$, respectivamente;
- b.3) o $\mathcal{N}Aut$ -morfismo h é associado ao Aut -morfismo $h' = \langle h_M', h_E' \rangle: A_1 \rightarrow A_2$ onde $h_E': E_1 \rightarrow E_2$ é a função total canonicamente induzida pelo $CMon$ -morfismo $h_E: E_1 \rightarrow E_2$. \square

Exemplo 9.7 A Figura 9.7 ilustra os funtores $an: Aut \rightarrow \mathcal{N}Aut$ e $na: \mathcal{N}Aut \rightarrow Aut$ bem como a composição $na \circ an: Aut \rightarrow Aut$. Note-se que o autômato seqüencial resultante da aplicação do funtor na possui a mesma forma do não-seqüencial original, mas com interpretação diferente, pois a relação de independência é substituída por uma transição que representa a composição síncrona das transições componentes. \square

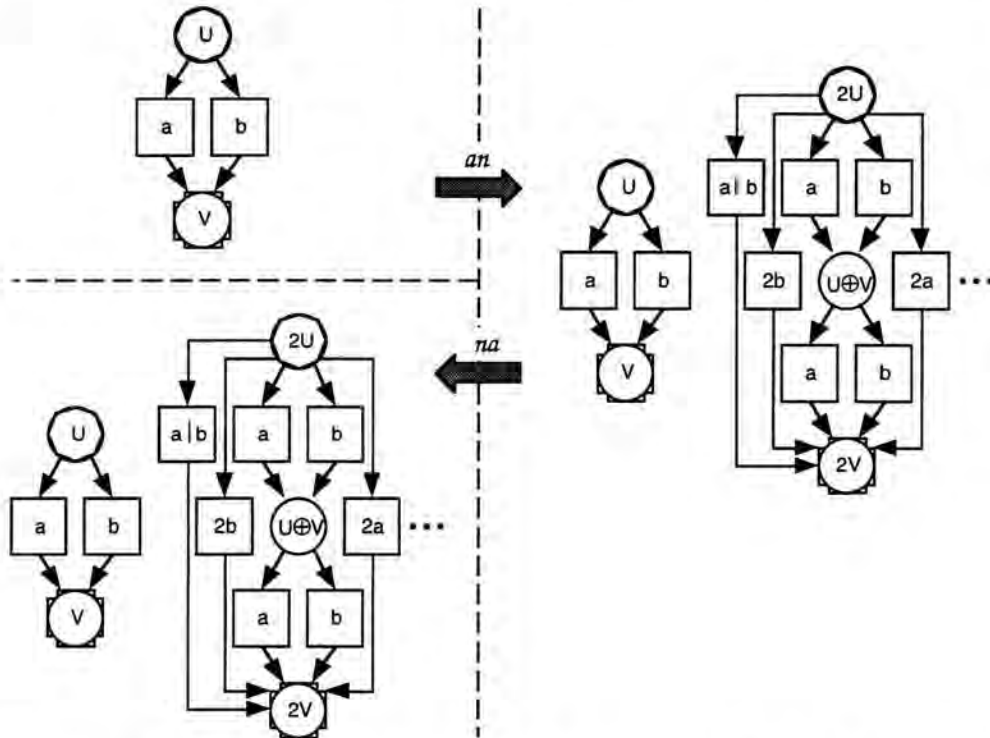


Figura 9.7 Um autômato seqüencial e as passagens para não-seqüencial e, novamente, para seqüencial

Proposição 9.8 O funtor $an: Aut \rightarrow \mathcal{N}Aut$ é adjunto esquerdo do funtor $na: \mathcal{N}Aut \rightarrow Aut$.

Prova:

- a) Considere o Aut -objeto $A = \langle V_A, T_A, \partial_{0A}, \partial_{1A}, l_A, l_A, F_A, i_A + f_A, E_A, \text{eti}_A \rangle$ e o $\mathcal{N}Aut$ -objeto $N = \langle V_N, T_N, \partial_{0N}, \partial_{1N}, l_N, l_N, F_N, i_N + f_N, E_N, \text{eti}_N \rangle$. Então:

$$anA = \langle V_A^\oplus, T_A^\oplus, \partial_{0A}^\oplus, \partial_{1A}^\oplus, l_A^\oplus, l_A^\oplus, F_A^\oplus, i_A^\oplus + f_A^\oplus, E_A^\oplus, \text{eti}_A^\oplus \rangle$$

$$na \circ anA = \langle V_A^\oplus, T_A^\oplus, \partial_{0A}^\oplus, \partial_{1A}^\oplus, l_A^\oplus, l_A^\oplus, F_A^\oplus, i_A^\oplus + f_A^\oplus, E_A^\oplus, \text{eti}_A^\oplus \rangle$$

$$naN = \langle V_N, T_N, \partial_{0N}, \partial_{1N}, \iota_N, I_N, F_N, i_N + f_N, E_N, \text{eti}_N \rangle$$

- b) a unidade da adjunção é a transformação natural $\eta_A = \langle \eta_V, \eta_T, \eta_I, \eta_F, \eta_E \rangle: A \rightarrow na \circ anA$ onde $\eta_V: V_A \rightarrow V_A^\oplus$, $\eta_T: T_A \rightarrow T_A^\oplus$, $\eta_I: I_A \rightarrow I_A^\oplus$, $\eta_F: F_A \rightarrow F_A^\oplus$ e $\eta_E: E_A \rightarrow E_A^\oplus$ são inclusões.
- c) Suponha $f = \langle f_V, f_T, f_I, f_F, f_E \rangle: A \rightarrow naN$. Então, $f^\oplus = \langle f_V^\oplus, f_T^\oplus, f_I^\oplus, f_F^\oplus, f_E^\oplus \rangle: anA \rightarrow N$ é o único $\mathcal{N}Aut$ -morfismo que estende f .

De fato, f^\oplus é o único morfismo tal que $f = anf^\oplus \circ \eta_A$. \square

Observação 9.9 Autômatos Não-Sequenciais são mais Concretos que os Sequenciais. A adjunção $\langle an, na \rangle: \mathcal{A}ut \rightarrow \mathcal{N}Aut$ caracteriza que os autômatos não-sequenciais são mais concretos que os sequenciais. Por "mais concreto em relação a uma adjunção", entende-se uma medida de expressividade que considera um ou mais dos itens que seguem:

- capacidade de expressar diferentes características dos diversos objetos;
- capacidade de expressar diferentes objetos;
- capacidade de expressar diferentes morfismos.

Assim, os autômatos não-sequenciais são mais concretos que os sequenciais pois (a) seus objetos podem expressar independência de transições, (b) qualquer $\mathcal{N}Aut$ -objeto com algum monóides não livre não é imagem de qualquer $\mathcal{A}ut$ -objeto e (c) morfismos entre monóides livres não são necessariamente gerados por morfismos nos correspondentes conjuntos geradores e, portanto, $\mathcal{N}Aut$ -morfismos nem sempre são imagem de $\mathcal{A}ut$ -morfismos, mesmo que os objetos origem e destino sejam. \square

9.3 Autômatos Não-Sequenciais e Redes de Petri Etiquetadas

A adjunção $\langle pn, np \rangle: \mathcal{E}MPetri^* \rightarrow \mathcal{N}Aut$ onde $pn = inc \circ pn^*$ estabelece uma comparação da expressividade entre as categorias de redes e de autômatos não-sequenciais, permitindo situar os autômatos não sequenciais em relação a outros modelos conhecidos. O funtor pn quando aplicado a uma rede resulta em um autômato não sequencial o qual explicita as concorrências implícitas da rede original. O adjunto direito np ao esquecer sobre a estrutura monoidal nas transições não só preserva a independência das transições do autômato origem, como explicita todas as composições síncronas possíveis entre as transições componentes. Assim, o funtor composto $np \circ pn$ resulta em uma espécie de fecho síncrono. Como esperado e contrariamente à composição $ap \circ pa$, o funtor $np \circ pn$ quando aplicado a uma rede não resulta necessariamente em uma rede segura.

A subcategoria dos autômatos não-sequenciais apontados, denotada por $\mathcal{N}Aut^*$ é introduzida para facilitar o entendimento da categoria $\mathcal{E}MPetri^*$. Como as categorias $\mathcal{N}Aut^*$ e $\mathcal{E}MPetri^*$ são isomorfas, obviamente os resultados e construções referentes a $\mathcal{E}MPetri^*$ como a bicompletude, sincronização, etc. podem ser considerados para $\mathcal{N}Aut^*$.

Autômatos Não-Sequenciais Apontados

Na definição da categoria $\mathcal{N}Aut^*$ a seguir, a restrição de que as transições são elementos de um monóide livre não é válida para as transições identidade, pois o monóide de estados pode não ser livre.

Definição 9.10 Categoria dos Autômatos Não-Sequenciais Apontados. A categoria dos autômatos não-sequenciais apontados, denotada por $\mathcal{N}Aut^*$ é uma subcategoria de $\mathcal{N}Aut$ onde:

- Considere a Figura 9.8. Um $\mathcal{N}Aut^*$ -objeto $N = \langle V, T^+, \partial_0^+, \partial_1^+, \iota, I_e^\oplus, F_e^\oplus, i^\oplus + f^\oplus, E_\tau^\oplus, \text{eti}^+ \rangle$ é um $\mathcal{N}Aut$ -objeto tal que os morfismos $\partial_k^+: T^+ \rightarrow V$, $\iota: V \rightarrow T^+$ são univocamente induzidos pelo coproduto $T_\tau^\oplus +_{\mathcal{C}Mon} V$ onde $\partial_k: T_\tau \rightarrow V$ são Set^* -morfismos, $\text{eti}^+: T^+ \rightarrow E_\tau^\oplus$ é univocamente induzido pelo produto $T_\tau^\oplus \times_{\mathcal{C}Mon} V$ (isomorfo ao coproduto $T_\tau^\oplus +_{\mathcal{C}Mon} V$) onde $\text{eti}: T_\tau \rightarrow E_\tau$ é um Set^* -morfismo e $i^\oplus: I_e^\oplus \rightarrow V$ e $f^\oplus: F_e^\oplus \rightarrow V$ são os $\mathcal{C}Mon$ -morfismos canonicamente induzidos pelos Set^* -morfismos $i: I_e \rightarrow V$ e $f: F_e \rightarrow V$, respectivamente.

- b) Considere a Figura 9.9. Um $\mathcal{N}Aut^*$ -morfismo $h = \langle h_V, h_{T^+}, h_l^\oplus, h_F^\oplus, h_E^\oplus \rangle: N_1 \rightarrow N_2$ é um $\mathcal{N}Aut$ -morfismo tal que $h_{T^+}: T_{\tau_1}^+ \rightarrow T_{\tau_2}^+$ é univocamente induzido pelo coproduto em $CMon$ onde $h_T: T_{\tau_1} \rightarrow T_{\tau_2}$ é um Set^* -morfismo e onde $h_E^\oplus: E_{\tau_1}^\oplus \rightarrow E_{\tau_2}^\oplus$, $h_l^\oplus: l_{e_1}^\oplus \rightarrow l_{e_2}^\oplus$ e $h_F^\oplus: F_{e_1}^\oplus \rightarrow F_{e_2}^\oplus$ são os únicos $CMon$ -morfismos que estendem os Set^* -morfismos $h_E: E_{\tau_1} \rightarrow E_{\tau_2}$, $h_l: l_{e_1} \rightarrow l_{e_2}$ e $h_F: F_{e_1} \rightarrow F_{e_2}$, respectivamente. \square

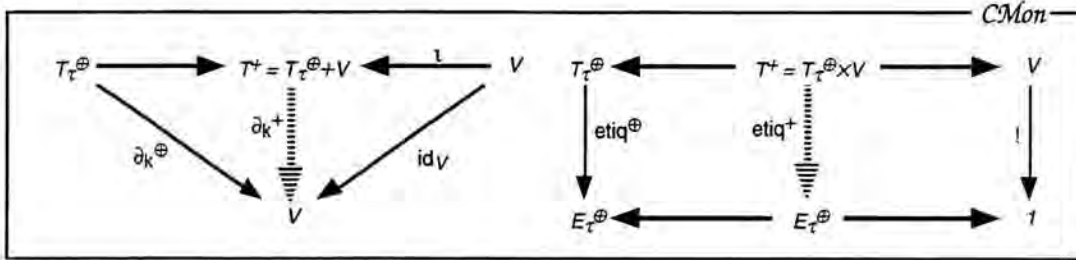


Figura 9.8 Diagramas comutativos para autômatos não-sequenciais apontados

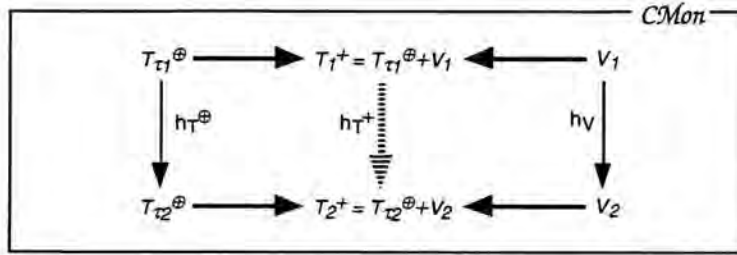


Figura 9.9 Diagrama comutativo para morfismos entre autômatos não-sequenciais apontados

Definição 9.11 Funtores pn^* e np^* . Considere a notação usada na definição da categoria dos autômatos não-sequenciais apontados. Então:

- a) o functor $pn^*: \mathcal{EMPetri}^* \rightarrow \mathcal{N}Aut^*$ é tal que:
- a.1) para cada $\mathcal{EMPetri}^*$ -objeto $P = \langle V, T_\tau, \partial_0, \partial_1, l_e, F_e, i+f, E_\tau, etiq \rangle$, $pn^*P = \langle V, T^+, \partial_0^+, \partial_1^+, l, l_e^\oplus, F_e^\oplus, i^\oplus+f^\oplus, E_\tau^\oplus, etiq^+ \rangle$;
 - a.2) para cada $\mathcal{EMPetri}^*$ -morfismos $h = \langle h_V, h_T, h_l, h_F, h_E \rangle$, $pn^*h = \langle h_V, h_{T^+}, h_l^\oplus, h_F^\oplus, h_E^\oplus \rangle$;
- b) o functor $np^*: \mathcal{N}Aut^* \rightarrow \mathcal{EMPetri}^*$ é tal que:
- b.1) para cada $\mathcal{N}Aut^*$ -objeto $N = \langle V, T^+, \partial_0^+, \partial_1^+, l, l_e^\oplus, F_e^\oplus, i^\oplus+f^\oplus, E_\tau^\oplus, etiq^+ \rangle$, $np^*N = \langle V, T_\tau, \partial_0, \partial_1, l_e, F_e, i+f, E_\tau, etiq \rangle$;
 - b.2) para cada $\mathcal{N}Aut^*$ -morfismo $h = \langle h_V, h_{T^+}, h_l^\oplus, h_F^\oplus, h_E^\oplus \rangle$, $np^*h = \langle h_V, h_T, h_l, h_F, h_E \rangle$. \square

Proposição 9.12 As categorias $\mathcal{EMPetri}^*$ e $\mathcal{N}Aut^*$ são isomorfas.

Prova: É fácil verificar que $np^* \circ pn^* = id_{\mathcal{EMPetri}^*}$ e $pn^* \circ np^* = id_{\mathcal{N}Aut^*}$. \square

Autômatos Não-Sequenciais

O relacionamento entre redes de Petri e autômatos não sequenciais é construído usando o fato de que $\mathcal{EMPetri}^*$ e $\mathcal{N}Aut^*$ são isomorfas.

Definição 9.13 Funtores pn e np .

- a) Considere o functor $pn^*: \mathcal{EMPetri}^* \rightarrow \mathcal{N}Aut^*$ definido anteriormente. Seja $inc: \mathcal{N}Aut^* \rightarrow \mathcal{N}Aut$ um functor que caracteriza $\mathcal{N}Aut^*$ como uma subcategoria de $\mathcal{N}Aut$. Então,

$$pn = inc \circ pn^*: \mathcal{EMPetri}^* \rightarrow \mathcal{N}Aut$$

- b) o functor $np: \mathcal{N}Aut \rightarrow \mathcal{EMPetri}^*$ é tal que:

- b.1) para cada $\mathcal{N}Aut$ -objeto $N = \langle V, T, \partial_0, \partial_1, l, f, E, \text{eti}_q \rangle$, $npN = \langle V, T, \partial_0', \partial_1', l, f, i+f, E, \text{eti}_q' \rangle$ onde $\partial_0', \partial_1', l, f$ e eti_q' são os Set^* -morfismos canonicamente induzidos pelos $CMon$ -morfismos $\partial_0', \partial_1', l, f$ e eti_q' , respectivamente;
- b.2) para cada $\mathcal{N}Aut$ -morfismo $h = \langle h_V, h_T, h_l, h_f, h_E \rangle$, $np h = \langle h_V, h_T', h_l', h_f', h_E' \rangle$ onde h_T', h_l', h_f' e h_E' são os Set^* -morfismos canonicamente induzidos pelos $CMon$ -morfismos h_T, h_l, h_f e h_E , respectivamente. \square

Proposição 9.14 O funtor $pn: \mathcal{EMPetri}^* \rightarrow \mathcal{N}Aut$ é adjunto esquerdo do funtor $np: \mathcal{N}Aut \rightarrow \mathcal{EMPetri}^*$.

Prova: Seja $\eta: id_{\mathcal{EMPetri}^*} \rightarrow np \circ pn$ a transformação natural tal que, para cada $\mathcal{EMPetri}^*$ -objeto $P = \langle V_P, T_{TP}, \partial_{0P}, \partial_{1P}, l_{EP}, F_{EP}, i_P + f_P, E_{TP}, \text{eti}_qP \rangle$, $\eta_P = \langle id_{V_P}, inc_{TP}, inc_{l_P}, inc_{F_P}, inc_{E_P} \rangle$ onde $inc_{TP}, inc_{l_P}, inc_{F_P}$ e inc_{E_P} são inclusões. Seja $N = \langle V_N, T_N, \partial_{0N}, \partial_{1N}, l_N, F_N, i_N + f_N, E_N, \text{eti}_qN \rangle$ um $\mathcal{N}Aut$ -objeto e $h = \langle h_V, h_T, h_l, h_f, h_E \rangle: P \rightarrow npN$ um $\mathcal{EMPetri}^*$ -morfismo. Então $g = \langle h_V, h_T', h_l', h_f', h_E' \rangle: pnP \rightarrow N$ é o único $\mathcal{N}Aut$ -morfismo tal que $h = np g \circ \eta_P$ onde $h_l': l_{EP} \rightarrow l_N$, $h_f': F_{EP} \rightarrow F_N$ e $h_E': E_{EP} \rightarrow E_N$ são as únicas extensões de $h_l: l_{EP} \rightarrow l_N$, $h_f: F_{EP} \rightarrow F_N$ e $h_E: E_{EP} \rightarrow E_N$, respectivamente, e onde h_T' é univocamente induzido pelo coproduto categorial em $CMon$ como ilustrado na Figura 9.10. \square

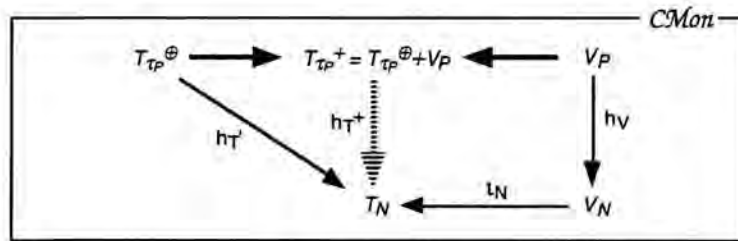


Figura 9.10 Morfismo univocamente induzido

Exemplo 9.15

- a) Considere as Figuras 9.11 a 9.13. Em cada figura, é apresentada a representação gráfica de uma rede de Petri etiquetada e o correspondente autômato não-sequencial, de acordo com o funtor pn . Somente na primeira figura as transições identidade do autômato resultante são representadas. Compare com os correspondentes casos referente ao funtor pa .
- b) A Figura 9.14 ilustra o funtor np aplicado a um autômato não-sequencial. Para um melhor entendimento, as transição identidade do autômato não são omitidas na representação gráfica. Note-se que a rede resultante possui infinitas transições e que os conjuntos apontados de estados iniciais e finais são $\{\tau, U, 2U, 3U, \dots\}$ e $\{\tau, V, 2V, 3V, \dots\}$, respectivamente.

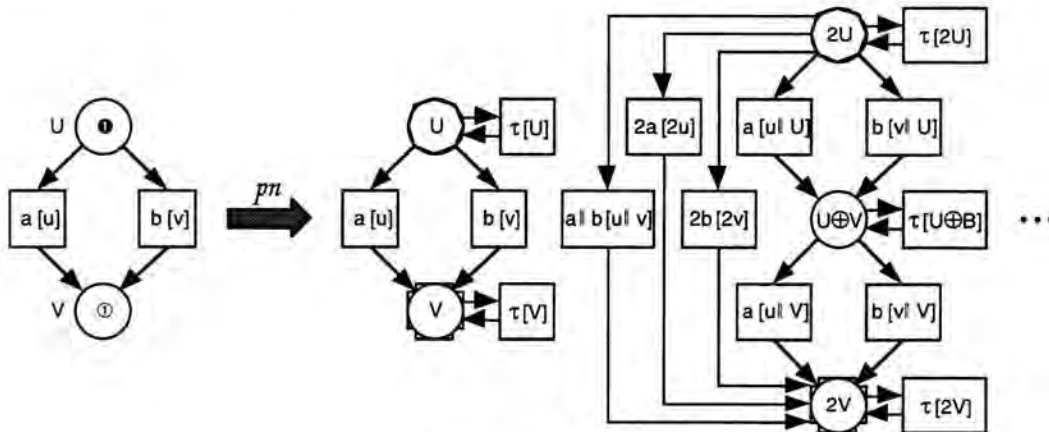


Figura 9.11 Escolha

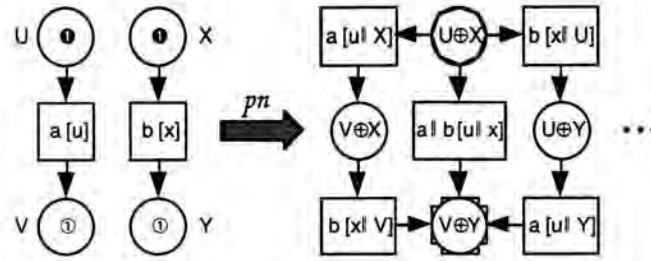


Figura 9.12 Concorrência

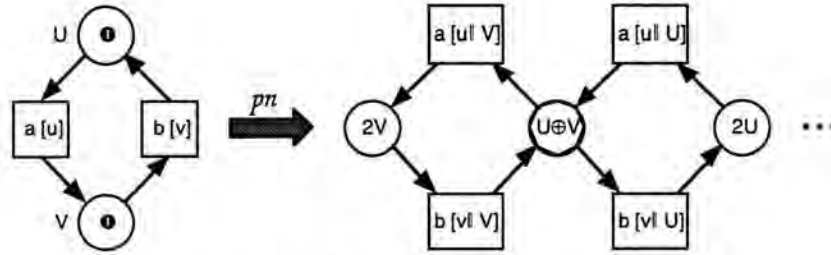


Figura 9.13 Ciclo

c) Note-se que a Figuras 9.11 juntamente com a Figuras 9.14 ilustram o funtor composto $np \circ pn$ aplicado a uma rede. \square

Observação 9.16 Autômatos Não-Sequenciais são mais Concretos que as Redes de Petri. Os autômatos não-sequenciais são mais concretos que as redes de Petri pois:

- a) qualquer $\mathcal{N}Aut$ -objeto com algum monóides não livre nas transições não é imagem de $EMPetri^*$ -objeto algum. Um exemplo seria considerar um autômato não-sequencial onde as transições são elementos de um grupo comutativo (todo grupo é um monóide) onde, para cada transição a existe uma transição \underline{a} tal que $a \parallel \underline{a} = \tau$, significando que não são independentes, refletindo algum critério de sincronização;
- b) morfismos entre monóides livres de transições não são necessariamente gerados por morfismos nos correspondentes conjuntos geradores e, portanto, $\mathcal{N}Aut$ -morfismos nem sempre são imagem de $EMPetri^*$ -morfismos, mesmo que os objetos origem e destino sejam. \square

9.4 Reificação de Redes de Petri Etiquetadas

A reificação de redes de Petri é obtida a partir da composição das adjunções $\langle pn, np \rangle: EMPetri^* \rightarrow \mathcal{N}Aut$ e $\langle nc, cn \rangle: \mathcal{N}Aut \rightarrow \mathcal{N}Comp$. Lembre-se que a composição de adjunções é uma adjunção (ver apêndice).

Composicionalidade Vertical

A categoria das redes de Petri e reificações é construída de forma análoga à dos autômatos. Por simplicidade, nos diagramas que seguem, os funtores $id: MPetri^* \rightarrow MPetri^*$ e $inc: Petri_1^* \rightarrow MPetri^*$ usados na definição da categoria $EMPetri^*$ são omitidos. Por exemplo, em diagramas, $etiq: M \rightarrow E$ denota $etiq: idM \rightarrow incE$.

Definição 9.17 Fecho Computacional. Considere as adjunções $\langle pn, np, \eta_P, \varepsilon_P \rangle: EMPetri^* \rightarrow \mathcal{N}Aut$ e $\langle nc, cn, \eta_N, \varepsilon_N \rangle: \mathcal{N}Aut \rightarrow \mathcal{N}Comp$. Então $\langle pc, cp, \eta, \varepsilon \rangle: EMPetri^* \rightarrow \mathcal{N}Comp$ denota a adjunção resultante da composição das adjunções onde $pc = nc \circ pn$, $cp = np \circ cn$, $\eta = \eta_P \circ pn \eta_N np$ e $\varepsilon = nc \varepsilon_P cn \circ \varepsilon_N$. O fecho computacional (não-sequencial) é a operação funtorial $pfc: EMPetri^* \rightarrow EMPetri^*$ tal que $pfc = cp \circ pc$. \square

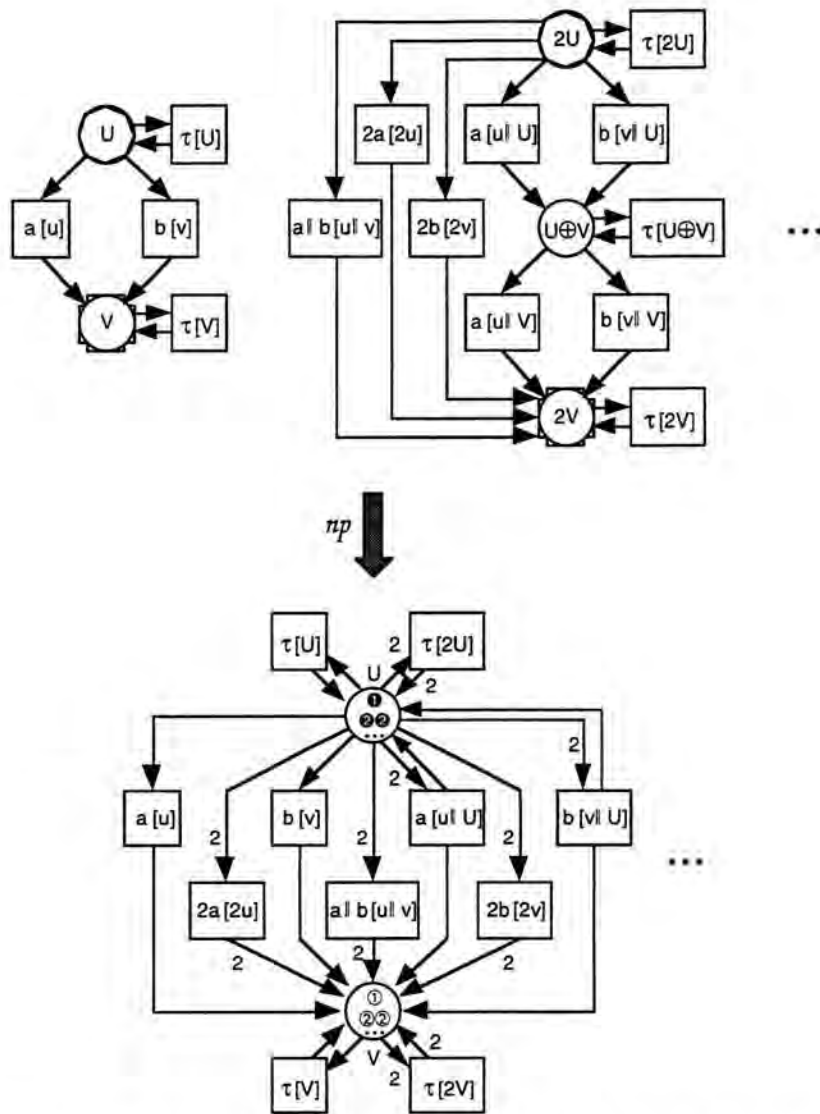


Figura 9.14 Um autômato não-sequencial e a correspondente rede de Petri etiquetada

Exemplo 9.18 Considere uma rede com as transições etiquetadas $a[u]: A \rightarrow B$, $b[v]: B \rightarrow C$ e $x[t]: X \rightarrow Y$. Então, as seguintes transições (entre outras) pertencem ao fecho computacional da rede:

- $\tau[A]: A \rightarrow A$
- $a;b[u;v]: A \rightarrow C$
- $a||x[u|v]: A \oplus X \rightarrow B \oplus Y$
- $(a;b)||x[(u;v)||t]: A \oplus X \rightarrow C \oplus Y$

Note-se que algumas transições são, de fato, classes de transições. Por exemplo para a transição $(a;b)||x[(u;v)||t]$ tem-se que (por simplicidade, excetuando-se para as transições identidade, as transições são identificadas somente pela sua etiqueta):

$$(a;b)||x = (a;b)||x;(\tau[Y]) = (a||x);(b||\tau[Y]) = (a||x);b$$

$$(a;b)||x = ((a;b);(\tau[C]))||(\tau[X];x) = ((a;b)||\tau[X]);(\tau[C]||x) = a;b;x$$

Definição 9.19 Reificação. Seja $T = \langle pfc, \eta, \mu \rangle$ onde $\eta = \langle \eta_M, \eta_E \rangle$ e $\mu = \langle \mu_M, \mu_E \rangle$ a mônada determinada pela adjunção $\langle pc, cp, \eta, \varepsilon \rangle: \mathcal{EMPetri}^* \rightarrow \mathcal{NComp}$. A categoria das redes de Petri etiquetadas e reificações, denotada por $\mathcal{ReifEMPetri}^*$, é tal que (suponha os $\mathcal{EMPetri}^*$ -objetos $P_k = \langle M_k, E_k, \text{etiq}_k \rangle$, para k com valores em $\{1, 2, 3\}$):

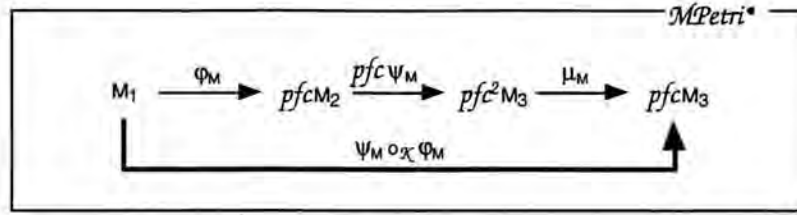


Figura 9.15 Composição de reificações é a composição na categoria Kleisli sem considerar as etiquetas

- a) os objetos são os mesmo de $EMPetri^*$;
- b) os morfismos são construídos como segue:
 - b.1) $\varphi = \varphi_M: P_1 \rightarrow P_2$ é um $ReifEMPetri^*$ -morfismo onde $\varphi_M: M_1 \rightarrow pfcM_2$ é um $MPetri^*$ -morfismo;
 - b.2) para cada $MPetri^*$ -objeto P , $\varphi = \eta_M: P \rightarrow P$ é o $ReifEMPetri^*$ -morfismo identidade de P ;
- c) sejam $\varphi: P_1 \rightarrow P_2, \psi: P_2 \rightarrow P_3$ $ReifEMPetri^*$ -morfismos. A composição $\psi \circ \varphi$ é o morfismo $\psi_M \circ_{\chi} \varphi_M: P_1 \rightarrow P_3$ onde $\psi_M \circ_{\chi} \varphi_M$ é determinado como ilustrado na Figura 9.15. \square

Um morfismo de redes sem considerar as etiquetas pode induzir um morfismo nas etiquetas determinando uma categoria de redes etiquetadas e reificações cujas reificações são $EMPetri^*$ -morfismos. A construção é análoga a realizada para os autômatos e é fácil verificar que a categoria resultante é isomorfa à $ReifEMPetri^*$.

Composicionalidade Horizontal

As redes de Petri não satisfazem à composicionalidade horizontal.

Proposição 9.20 O endofuntor $pfc: EMPetri^* \rightarrow EMPetri^*$ não preserva limites.

Prova: Considere as redes P_1 e P_2 ilustradas na Figura 9.16. Os diagramas das redes $pfc(P_1 \times P_2)$ e

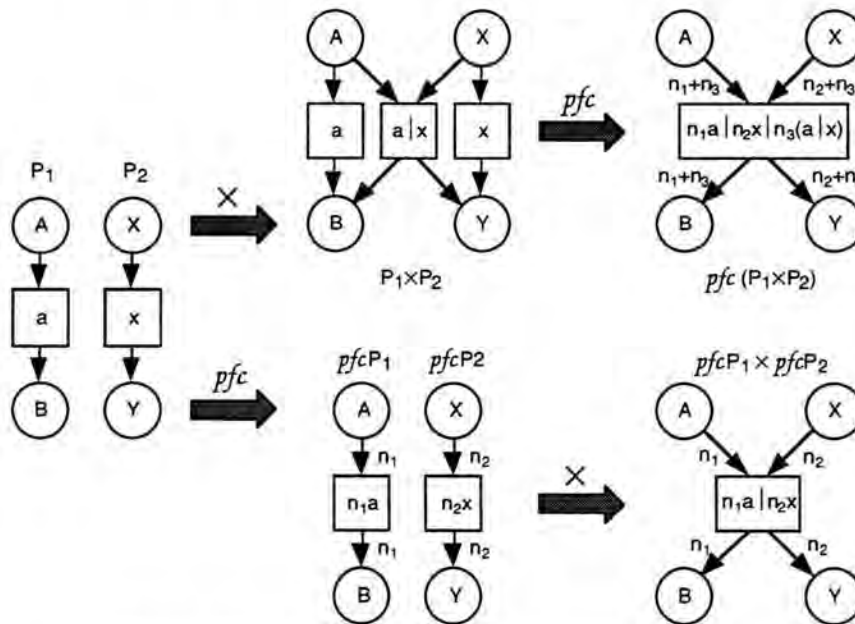


Figura 9.16 O functor fecho computacional não preserva produtos

$pfcp_1 \times pfcp_2$ possuem infinitas transições as quais são representadas genericamente, para n_1, n_2, n_3 com valores em $\{0, 1, 2, \dots\}$. Note-se que $pfcp(P_1 \times P_2)$ não é isomorfo a $pfcp_1 \times pfcp_2$. Por exemplo, a transição $2a \mid 3x \mid 4(a \mid x): 6A \oplus 7X \rightarrow 6B \oplus 7Y$ em $pfcp(P_1 \times P_2)$ não possui correspondência em $pfcp_1 \times pfcp_2$. \square

10 Linguagem Náutilus

A linguagem Náutilus, originalmente introduzida em [Menezes *et al* 95b], é baseada nos objetos e foi inspirada na linguagem GNOME [Sernadas & Ramos 94] que, por sua vez, é uma simplificação e revisão da linguagem orientada aos objetos OBLOG [SernadasC *et al* 92], [SernadasC *et al* 92b], [Sernadas *et al* 92], [SernadasC *et al* 91]. Trata-se de uma linguagem textual, de alto nível e com facilidades de abstração. Para evitar que detalhes não relevantes desviem a atenção dos principais objetivos propostos por este trabalho, a linguagem foi definida de forma relativamente simples e compacta. Assim, Náutilus pode ser considerada como uma parte representativa de uma linguagem mais complexa. Comparativamente com a linguagem GNOME, as seguintes características gerais de Náutilus devem ser destacadas:

- não suporta classe, herança, criação e deleção dinâmica de objetos, evento ("event broadcasting"), região (definida sobre classes, possibilita especificar outras classes, tipos, etc, os quais são encapsulados) e biblioteca parametrizada;
- como em GNOME, existem ações externas (públicas) e internas (privadas). As ações internas não podem ser referenciadas externamente e podem ocorrer sempre que ativas. Entretanto, em relação às ações externas, enquanto que em GNOME estas podem ocorrer somente quando chamadas, em Náutilus, dependendo do relacionamento especificado, estas podem ocorrer sempre que ativas;
- em Náutilus, as cláusulas componentes de uma ação podem ser especificadas de forma seqüencial ou múltipla. Uma composição múltipla é uma composição especial de cláusulas concorrentes a qual pode ser vista como uma generalização da noção de múltipla atribuição (onde as cláusulas de valoração são avaliadas antes da atribuição dos valores resultantes aos correspondentes atributos);
- além das interações no estilo GNOME, em Náutilus é possível especificar agregações. A principal diferença é que, em uma interação, o relacionamento entre os objetos componente é especificado internamente, na forma de chamadas, enquanto que, em uma agregação, o relacionamento é especificado à parte, em um objeto agregador. A seguinte analogia com a eletrônica pode auxiliar no entendimento da idéia geral de uma agregação: diversos *componentes (objetos)* podem ser agregados em uma única unidade funcional através de uma *placa (objeto agregador)* onde as *portas (ações exportadas)* são interligadas por *trilhas (composições)*, respeitando as características dos *sinais (entradas/saídas)*;
- em Náutilus, objetos podem ser reificados, ou seja, um objeto pode ser implementado sobre transações de outro. Um exemplo típico de reificação seria um compilador onde uma ação de uma linguagem de alto nível é implementada como uma transação de uma outra linguagem de um nível mais baixo;
- em Náutilus é possível extrair uma visão de um objeto restringindo (encapsulando) ou eventualmente renomeando as ações que podem ser acessadas externamente;
- a noção de alternativa, presente em GNOME para interações, é estendida em Náutilus para reificações e agregações. A noção de alternativa, objetiva garantir que uma mesma ação possa corresponder a diversos métodos, de acordo com diferentes condições de ativação. Ou seja, em Náutilus, interações, agregações e reificações podem ser dependentes de estado.

Como curiosidade, o nome Náutilus, originalmente sugerido por A. Sernadas, foi escolhido por ser uma palavra com sonoridade agradável, de fácil memorização, comum a diversas línguas e que lembra *Naut*, a categoria dos autômatos não-sequenciais.

A linguagem Náutilus é introduzida gradativamente. Inicialmente, é apresentada a gramática e, posteriormente, são introduzidos diversos exemplos. A cada momento, as principais características da linguagem são discutidas.

10.1 Gramática

A linguagem Náutilus é apresentada em uma notação estilo BNF (Backus-Naur Form - veja, por exemplo, [Hopcroft & Ullman 79, p. 78]), onde a seguinte convenção é adotada:

- **normal**: símbolos não-terminais;
- **carregado**: símbolos terminais (palavras reservadas da linguagem);
- **itálico**: símbolos não terminais cuja estrutura não é detalhada;
- para um símbolo x , $x \dots x$ especifica que x ocorre pelo menos uma vez.

Definição 10.1 Gramática da Linguagem Náutilus.

```

unidade =
  spec id_spec
  [objeto...objeto]
  end spec

objeto =
  object id_objeto
  complemento
  end id_objeto

complemento =
  compl_simples |
  compl_visão |
  compl_agregação |
  compl_reificação |

compl_simples =
  [import]
  [export]
  [categoria]
  [corpo]

compl_visão =
  view of obj_inter
  [export_der]
  [categoria]
  [corpo_visão]

compl_agregação =
  aggregation of obj_inter
  obj_inter...obj_inter
  [export_der]
  [categoria]
  [corpo Agr]

compl_reificação =
  over obj_inter
  [export_der]
  [categoria]
  [corpo_reif]

obj_inter =
  id_objeto | [interação]

interação =
  interaction
  id_objeto
  id_objeto...id_objeto
  end interaction

import =
  import
  [id_ação_compl...id_ação_compl]

id_ação_compl =
  id_ação ent_sai [of id_objeto]

ent_sai =
  [entrada...entrada]
  [saída...saída]

entrada =
  in id_ent: tipo

saída =
  out id_sai: tipo

tipo =
  boolean | natural | posint |
  integer | string | enumeração

export =
  export [ação_exp...ação_exp]

ação_exp =
  id_ação ent_sai

categoria =
  category [cat_ação...cat_ação]

```

```

cat_ação =
  cat id_ação...id_ação
cat =
  birth | death |
  [birth | death] request

export_der =
  export
  [ação_exp_der...ação_exp_der]
ação_exp_der =
  id_ação ent_sai_der
ent_sai_der =
  [entrada_der...entrada_der]
  [saída_der...saída_der]
entrada_der =
  der in id_ent: tipo
  by id_ação.id_ent [of id_objeto]
saída_der =
  der out id_sai: tipo
  by id_ação.id_sai [of id_objeto]

corpo =
  body
  [atributo...atributo]
  [ação...ação]
atributo =
  slot id_atrib: tipo
ação =
  act id_ação método

método =
  método_trans |
  método_alt
método_trans =
  [ativação...ativação]
  [chamada...chamada]
  [agregação_arg...agregação_arg]
  [transação]
  [resposta...resposta]

ativação =
  enb condição

chamada =
  call id_ação [of id_objeto]
  [argumento...argumento]
argumento =
  arg id_ação.id_ent = expressão

agregação_arg =
  agg [agr_arg...agr_arg]
agr_arg =
  id_ação.id_sai = expressão

```

```

resposta =
  ret id_ação.id_sai = expressão

transação =
  seqüencial |
  múltipla |
  [valoração]
seqüencial =
  seq
  transação...transação
  end seq
múltipla =
  cps
  transação...transação
  end cps
valoração =
  val id_atrib << expressão

método_alt =
  alternativa...alternativa
alternativa =
  alt [id_alt] método_trans

corpo_visão =
  body
  [ação_visão...ação_visão]
ação_visão =
  act id_ação as ação_obj
ação_obj =
  id_ação_obj | inter_ação_obj
id_ação_obj =
  id_ação [of id_objeto]
inter_ação_obj =
  int
  id_ação_obj...id_ação_obj
  end int

corpo_agr =
  body [ação_agr...ação_agr]
ação_agr =
  act id_ação
  composed by ação_obj...ação_obj
  [equ...equ]
equ =
  match ent_obj sai_obj |
  match sai_obj ent_obj...ent_obj
ent_obj =
  id_ação.id_ent [of id_objeto]
sai_obj =
  id_ação.id_sai [of id_objeto]

```

```

corpo_reif =
  body [ação_reif...ação_reif]
ação_reif =
  act id_ação método_reif

método_reif =
  transação_reif |
  método_alt_reif
transação_reif =
  seqüencial_reif |
  múltipla_reif |
  ação_obj

```

```

seqüencial_reif =
  seq
  transação_reif...transação_reif
end seq
múltipla_reif =
  cps
  transação_reif...transação_reif
end cps
método_alt_reif =
  alternativa_reif...alternativa_reif
alternativa_reif =
  alt [id_alt] transação_reif

```

□

Os seguintes comentários são importantes para um entendimento geral da linguagem Náutilus:

a) Sobre uma unidade e os seus objetos componentes:

- uma unidade especifica uma comunidade de objetos;
- objetos podem ser construídos sobre outros objetos através de reificações, agregações e visões. Adicionalmente, objetos podem interagir com outros objetos através de chamadas de ações;
- reificações, agregações, visões e interações são construções composicionais e podem ser especificadas em qualquer seqüência;
- objetos que interagem com outros objetos constituem, na realidade, um único objeto com especificação distribuída. Neste caso, a referência ao objeto resultante, é a referência aos objetos componentes (compreendidos entre os símbolos terminais **interaction** e **end interaction**);
- um objeto não pode interagir consigo próprio, direta ou indiretamente;
- uma reificação implementa um objeto sobre transações de um outro objeto denominado base de reificação. Transações são ações complexas, especificadas como composições de ações seqüenciais ou múltiplas do objeto base;
- uma agregação especifica o comportamento conjunto de objetos sendo possível definir o relacionamento entre as ações dos objetos componentes bem como renomear as ações do objeto resultante;
- um objeto não pode agregar consigo próprio, direta ou indiretamente;
- em uma visão, as ações exportadas no objeto original e não exportadas no resultante, são consideradas encapsuladas;
- um objeto usado em uma construção de reificação, agregação ou visão não mais pode ser usado em qualquer outra construção e não pode ser ativado;
- a ativação de uma unidade é a ativação de todos os objetos que podem ser ativados e que possuem ações **birth**;
- a ativação de um objeto é sempre por uma ação da categoria **birth**;
- um objeto pode ativar outro objeto chamando uma correspondente ação **birth**;

b) Sobre as ações (lembre-se que uma ação pode ser composta por transações):

- as categorias que podem ser associadas a uma ação são as seguintes:
 - **request**: só é executada quando chamada/agregada e se estiver ativa. Uma ação desta categoria só se justifica se for referenciada em uma interação ou agregação;
 - **birth**: pode ser executada, no máximo, uma vez e determina a ativação do objeto;
 - **death**: pode ser executada uma única vez e determina a desativação do objeto;
 - uma ação pode ser simultaneamente **birth/request** e **death/request**;
- uma ação está ativa se:
 - todas as condições de ativação são satisfeitas;

- todas as chamadas podem ser efetuadas;
- uma ação com alternativas está ativa se pelo menos uma alternativa está ativa;
- uma ação pode ser especificada como:
 - **export**: somente as ações exportadas podem ser referenciadas por outros objetos;
 - **import**: ação exportada de outro objeto a qual pode ser chamada por este;
 - encapsulada: não exportada. Uma ação originalmente exportada pode ser encapsulada quando da extração de uma visão do objeto;
- a execução de uma ação é:
 - indivisível: ou executa todas as transações componentes, ou a ação não é executada;
 - finita: a execução termina em um tempo finito;
 - determinista: para os mesmos valores de atributos e entradas, sempre atinge os mesmos resultados. No caso de uma ação com alternativas, cada alternativa é determinista. Pode-se considerar que, ações com alternativas são, na realidade, diversas ações (uma para cada alternativa) com a mesma identificação;
- entre as ações ativas, a escolha de qual é executada é um não-determinismo interno. Em uma ação com alternativas, somente uma entre as alternativas ativas é escolhida para ser executada e, neste caso, a escolha também é um não-determinismo interno;
- a execução de uma ação pode ser como segue:
 - por solicitação: a ação é da categoria **request** e é chamada ou é usada em uma agregação. Neste caso, a ação só pode ocorrer quando ocorrer a ação chamadora/agregada;
 - espontânea: a ação não é da categoria **request** e não é chamada/agregada;
 - mista: a ação não é da categoria **request** e é chamada/agregada. Neste caso, a ação pode ocorrer independentemente da ocorrência da ação chamadora/agregada;
- a seguinte tabela resume as combinações para a execução de uma ação em Náutilus comparativamente com GNOME:

	Espontânea	Por Solicitação	Mista
Exportada	Náutilus	Náutilus GNOME	Náutilus
Encapsulada	Náutilus GNOME	não	não

- como objetos que interagem constituem um objeto com especificação distribuída, a identificação de uma ação resultante da composição de ações é a identificação das ações componentes, compreendidas entre os símbolos terminais **int** e **end int**;
- as avaliações das cláusulas em uma transação do tipo múltipla são executadas simultaneamente. Adicionalmente, a parte direita das expressões é avaliada antes de eventuais atribuições;
- em uma ação sem alternativas ou em uma alternativa, só é possível especificar mais de uma atribuição a um mesmo atributo se estas forem seqüencialmente dependentes.
- quando de uma reificação, agregação ou interação, a informação de que uma ação de um objeto componente pertence à da categoria **request** não é herdada para a correspondente ação no objeto resultante;
- em um objeto resultante de uma interação, agregação ou reificação:
 - uma ação resultante é da categoria **birth** se for composta exclusivamente e por, pelo menos, uma ação **birth**. Logo, é suficiente ativar um dos objetos componentes para ativar o objeto resultante (e, neste caso, as ações dos demais objetos só podem ser referenciadas após a ativação dos mesmos). No caso de uma ação **birth** reificada, a primeira ação da correspondente transação deve ser **birth**;

- uma ação resultante é da categoria **death** se for composta exclusivamente e por, pelo menos, uma ação **death** e se, após a sua execução, nenhum objeto componente permanecer ativo. Note-se que é possível desativar alguns objetos componentes permanecendo o resultante ativo embora, neste caso, a ação não seja considerada **death**. Adicionalmente, no caso de uma ação reificada, a última ação da correspondente transação deve ser **death**.
- c) Sobre as entradas ou saídas:
- em uma interação, uma ação chamadora:
 - fornece os valores das entradas da ação chamada;
 - pode referenciar as saídas da ação chamada;
 - em uma agregação:
 - as entradas ou saídas de um objeto representam as informações a serem compartilhadas com outros objetos;
 - o compartilhamento entre os objetos é especificado através de **match's**, identificando entradas (saídas) de um objeto com saídas (entradas) de outro;
 - relativamente a uma resposta (**ret**) em uma ação sem alternativas ou em uma alternativa:
 - o valor é associado à saída *depois* da execução das diversas cláusulas da ação/alternativa;
 - não pode existir mais de uma resposta associada a uma mesma saída;
 - relativamente a um argumento de agregação (**agg**) em uma ação sem alternativas ou em uma alternativa:
 - o valor é associado à saída *antes* da execução das diversas cláusulas da ação/alternativa;
 - não pode existir mais de um argumento de agregação associado a uma mesma saída;
 - na especificação do objeto agregador, a expressão referente a um argumento de agregação não pode conter qualquer entrada identificada com alguma saída do objeto ao qual foi associado o argumento em questão.

10.2 Exemplos

A seguir é apresentada uma série de exemplos usando a linguagem Náutilus.

Exemplo 10.2 Especifica uma pessoa que compra um relógio e passa a consultar o tempo. Os objetos são Pessoa e Relógio. O objeto Pessoa, ao executar a ação Compra, determina o nascimento do objeto Relógio e, ao executar a ação Consulta, chama uma ação do Relógio.

```

spec Pessoa_Relógio

object Pessoa
import
  Novo of Relógio
  Obs out Tempo: natural of Relógio
category
  birth Nasc
body
  slot Tactual: natural
  slot Tanterior: natural
  slot Comprou: (Não, Sim)
act Nasc
  val Comprou << Não

act Compra
  enb Comprou = Não
  call Novo of Relógio
  cps
    val Tanterior << 0
    val Tactual << 0
    val Comprou << Sim
  end cps
act Consulta
  enb Comprou = Sim
  call Obs of Relógio
  cps
    val Tanterior << Tactual
    val Tactual << Obs.Tempo
  end cps
end Pessoa

```

```

object Relógio
export
  Novo
  Obs out Tempo: natural
category
  birth request Novo
  request Obs

```

```

body
  slot Tempo: natural
  act Novo
    val Tempo << 0
  act Tic
    val Tempo << Tempo + 1
  act Obs
    ret Obs.Tempo = Tempo
end Relógio
end spec

```

□

No exemplo acima, tem-se que:

- após o nascimento do Relógio, a ação Tic está sempre ativa e a ação Obs só está ativa quando chamada;
- como Consulta chama Obs, pode referenciar diretamente a saída Obs.Tempo;
- lembre-se que as avaliações das cláusulas em uma transação do tipo múltipla são executadas simultaneamente e que a parte direita das expressões é avaliada antes de eventuais atribuições. Assim, na ação Consulta, os valores resultantes dos atributos Tatuall e Tanterior é sempre o esperado.

Exemplo 10.3 O objeto Pilha é implementado sobre a agregação dos objetos Tabela e Apontador. Note-se que, a tabela possui um número de células limitado enquanto que o apontador pode variar livremente. Veja os comentários após o exemplo.

```

object Pilha over Apontador_Tabela
export
  Nasc
  Empilha der in Valor: integer
    by Atribui.Valor
  Desempilha der out Valor: integer
    by Consulta.Valor
category
  birth request Nasc
  request Empilha Desempilha
body
  act Nasc
    Novo
  act Empilha
    seq
      Mais_um
      Atribui
    end seq
  act Desempilha
    seq
      Consulta
      Menos_um
    end seq
end Pilha

object Apontador_Tabela
aggregation of
  Apontador
  Tabela

```

```

export
  Novo
  Atribui der in Valor: integer
    by Atribui.Valor of Tabela
  Consulta der out Valor: integer
    by Consulta.Valor of Tabela
  Mais_um
  Menos_um
category
  birth Novo
body
  act Novo composed by
    Novo of Apontador
    Novo of Tabela
  act Atribui composed by
    Posição of Apontador
    Atribui of Tabela
    match
      Posição.Pos of Apontador
      Atribui.Ind of Tabela
  act Consulta composed by
    Posição of Apontador
    Consulta of Tabela
    match
      Posição.Pos of Apontador
      Consulta.Ind of Tabela
  act Mais_um composed by
    Mais_um of Apontador
  act Menos_um composed by
    Menos_um of Apontador
end Apontador_Tabela

```

```

object Apontador
export
  Novo
  Mais_um
  Menos_um
  Posição out Pos: natural
category
  birth request Novo
  request Posição
body
  slot Pos: natural
  act Novo
    val Pos << 0
  act Mais_um
    val Pos << Pos + 1
  act Menos_um
    val Pos << Pos - 1
  act Posição
    ret Posição.Pos = Pos
end Apontador

```

```

object Tabela
export
  Novo
  Atribui
    in Valor: integer
    in Ind: natural
  Consulta
    out Valor: integer
    in Ind: natural
category
  birth request Novo
  request Atribui Consulta
body
  slot Tab: integer [1..100]
  act Novo
  act Atribui
    enb Atribui.Ind ≤ 100
    enb Atribui.Ind ≥ 1
    val Tab[Atribui.Ind] <<
      Atribui.Valor
  act Consulta
    enb Atribui.Ind ≤ 100
    enb Atribui.Ind ≥ 1
    ret Consulta.Valor =
      Tab[Consulta.Ind]
end Tabela

```

□

No exemplo acima, tem-se que:

- entre os objetos especificados, Pilha é o único que pode ser ativado;
- Valor das ações Atribui e Consulta do objeto Tabela são derivados para Apontador_Tabela e, posteriormente, para Pilha;
- como Pos é do tipo natural, a ação Menos_um só é possível para valores de Pos maior que zero;
- Tab possui um número limitado de células e Pos pode assumir qualquer valor natural. Entretanto, as ações Atribui e Consulta do objeto Apontador_Tabela ficam ativas somente para os valores para os quais foi possível realizar o match dos argumentos, ou seja, para Pos compreendido entre os valores 1 e 100. Para os demais valores, a ação Posição do objeto Apontador não é integrante do objeto agregado e, portanto, não existe a possibilidade de um valor inválido para o índice de acesso à tabela;
- as ações Empilha e Desempilha são possíveis somente para valores de Pos válidos para Atribui e Consulta, respectivamente.

Exemplo 10.4 Jantar dos filósofos. Cinco filósofos jantam em uma mesa redonda onde existem cinco garfos, um entre cada dois filósofos, como ilustrado na Figura 10.1. Cada filósofo necessita de dois garfos para comer, um situado à direita e outra à esquerda. Quando um filósofo não está comendo, está pensando ou caminhando. Em qualquer dos casos, deixa os garfos na mesa. O objeto Jantar é o resultado da agregação dos cinco filósofos com os cinco garfos e é tal que nunca ocorre um abraço mortal ("deadlock"). Antes de realizar a agregação, a ação Caminha dos filósofos é encapsulada.

```

spec Jantar_Filósofos

object Jantar
aggregation of
  F1 F2 F3 F4 F5
  G1 G2 G3 G4 G5

```

```

category
  birth Início
body
  act Início composed by
    all birth

```

```

act F1come composed by
  Come of F1
  Levanta of G1
  Levanta of G2
...
act F5come composed by
  Come of F5
  Levanta of G5
  Levanta of G1
act F1pensa composed by
  Pensa of F1
  Abaixa of G1
  Abaixa of G2
...
act F5pensa composed by
  Pensa of F5
  Abaixa of G5
  Abaixa of G1
end Jantar

object F1 view of Filósofol
export
  Nasce Come Pensa
category
  birth request Nasce
  request Come Pensa
end F1

object Filósofol
export
  Nasce Come Pensa Caminha
category
  birth Nasce
body
  slot Ação: (Comer, Pensar, Cam)
  act Nasce
    val Ação << Comer

act Come
  enb Ação = Comer
  val Ação << Pensar
act Pensa
  alt Co
    enb Ação = Pensar
    val Ação << Comer
  alt Ca
    enb Ação = Pensar
    val Ação << Cam
act Caminha
  enb Ação = Cam
  val Ação << Comer
end Filósofol

...
object F5
...
object G1
export
  Novo Levanta Abaixa
category
  birth request Novo
  request Levanta Abaixa
body
  slot Em_uso: boolean
  act Novo
    val Em_uso << false
  act Levanta
    enb Em_uso = false
    val Em_uso << true
  act Abaixa
    enb Em_uso = true
    val Em_uso << false
end G1

...
object G5
...
end spec

```

□

No exemplo acima, note-se:

- na visão, as ações Caminha não são exportadas e, portanto, não podem ser referenciadas externamente. Note-se que, em uma visão, as ações exportadas e não renomeadas, não necessitam ser especificadas no corpo do objeto;
- no objeto resultante da agregação, as ações encapsuladas Caminha de cada filósofo são as únicas que podem ser ativadas independentemente de qualquer outra ação.

Exemplo 10.5 A especificação de uma Fábrica é o resultado da agregação dos objetos Produtor e Consumidor. O objeto Produtor é implementado sobre a Linha_Produção o qual, por sua vez, é uma visão da interação dos objetos Produção e CQ (controle de qualidade). O objeto Consumidor é implementado sobre Linha_Consumo. O relacionamento entre os diversos objetos componentes é ilustrado na Figura 10.2.

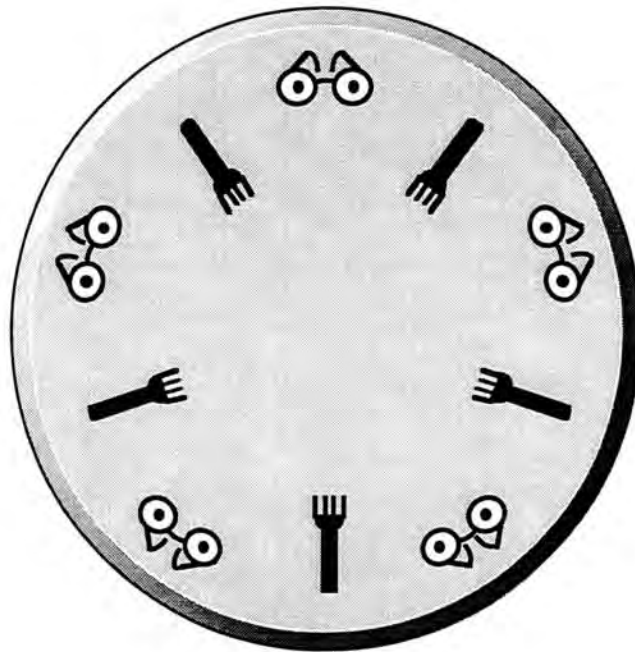


Figura 10.1 Jantar dos cinco filósofos

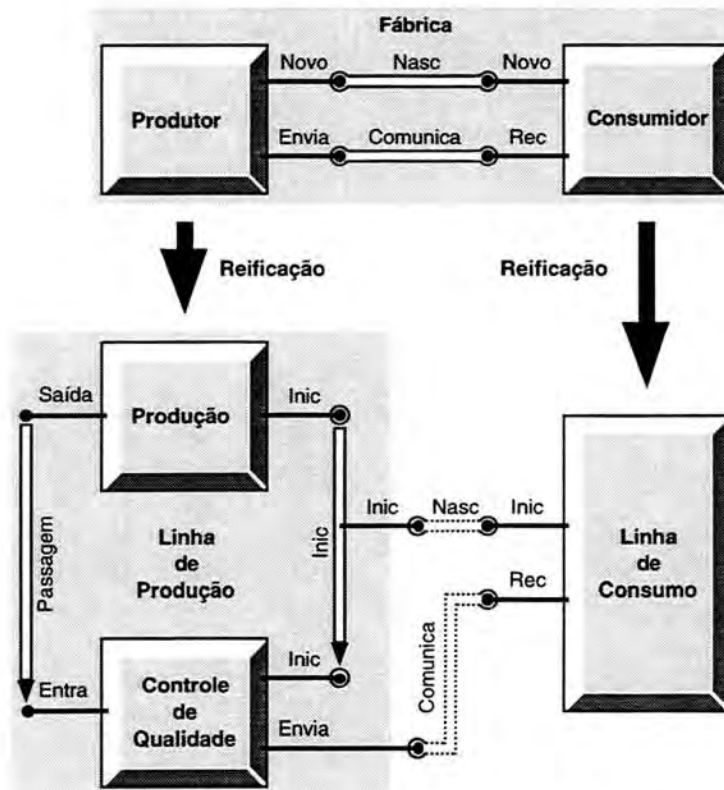


Figura 10.2 Relacionamento entre os objetos componentes de uma fábrica

```

spec Fábrica

object Fábrica_Raiz
aggregation of
  Produtor
  Consumidor
category
  birth Nasc
body
  act Nasc composed by
    Novo of Produtor
    Novo of Consumidor
  act Comunica composed by
    Envia of Produtor
    Recebe of Consumidor
  match
    Envia.Fator of Produtor
    Recebe.Fator of Consumidor
  match
    Envia.Msg of Produtor
    Recebe.Msg of Consumidor
end Fábrica_Raiz

object Produtor over Linha_Produção
export
  Novo
  Envia
  der in Fator: natural
  by Envia.Fator of Linha_Produção
  der out Msg: natural
  by Envia.Msg of Linha_Produção
category
  birth request Novo
  request Envia
body
  act Novo
  Inic
  act Prod_A
  seq
    Procl
    Montagem
    Passagem
    ProcCQ
  end seq
  act Prod_B
  seq
    cps
      Procl
      Proc2
    end cps
    Montagem
    Passagem
    ProcCQ
  end seq

act Envia
  Envia
end Produtor

object Linha_Produção view of
  interaction
  Produção
  CQ
  end interaction
export
  Inic Passagem
  Procl Proc2
  Montagem ProcCQ
  Envia
  der in Fator: natural
  by Envia.Fator of CQ
  der out Msg: natural
  by Envia.Msg of CQ
category
  birth Inic
body
  act Inic as
  int
    Inic of Produção
    Inic of CQ
  end int
  act Passagem as
  int
    Saída of Produção
    Entrada of CQ
  end int
end Linha_Produção

object Produção
import
  Inic of CQ
  Entrada in CodProd: natural of CQ
export
  Inic Procl Proc2
  Montagem Saída
category
  birth Inic
body
  slot P1: boolean
  slot P2: boolean
  slot NumProd: natural
  act Inic
  call Entrada of CQ
  cps
    val P1 << false
    val P2 << false
    val NumProd << 0
  end cps
  act Procl
  enb P1 = false
  val P1 << true

```

```

act Proc2
  enb P2 = false
  val P2 << true
act Montagem
  alt Mont_P1
    enb P1 = true and P2 = false
    val NumProd << NumProd + 1
    val P1 << false
  alt Mon_P2
    enb P2 = true and P1 = false
    val NumProd << NumProd + 1
    val P2 << false
  alt Mont_P1_P2
    enb P1 = true and P2 = true
    val NumProd << NumProd + 1
    val P1 << false
    val P2 << false
act Saída
  enb NumProd > 0
  call Entrada of CQ
    arg Entrada.CodProd = ...
  val NumProd << NumProd - 1
end Produção

object CQ
export
  Inic ProcCQ
  Entrada
    in CodProd: natural
  Envia
    in Fator: natural
    out Msg: natural
category
  birth Inic
body
  slot NumCQ: natural
  slot NumOK: natural
  act Inic
    cps
      val NumCQ << 0
      val NumOK << 0
    end cps
  act Entrada
    val NumCQ << NumCQ + 1
  act ProcCQ
    enb NumCQ > 0
    val NumOK << NumOK + 1
    val NumCQ << NumCQ - 1
  act Envia
    enb NumOK > 0
    val NumOK << NumOK - 1
    ret Envia.Msg = Fator * ...
end CQ

```

```

object Consumidor over Linha_Consumo
export
  Novo
  Recebe
    der out Fator: natural
    by Recebe.Fator
    der in Msg: natural
    by Recebe.Msg
category
  birth request Novo
  request Recebe
body
  act Novo
    Inic
  act Recebe
    seq
      Prepara_Rec
      Recebe
    end seq
  act Consume
    Consume
end Consumidor

object Linha_Consumo
export
  Inic
  Prepara_Rec
  Recebe
    out Fator: natural
    in Msg: natural
  Consume
Category
  birth Inic
body
  slot St: 1..3
  act Inic
    val St << 1
  act Prepara_Rec
    enb St = 1
    val St << 2
  act Recebe
    enb St = 2
    agg Recebe.Fator = ...
    val St << 3
  act Consume
    enb St = 3
    val St << 1
end Linha_Consumo
end spec

```

□

No exemplo acima, tem-se que:

- a referência a uma ação resultante de uma interação é a referência às ações componentes, compreendidas entre os símbolos terminais `int` e `end int`, como especificado no objeto `Linha_Produção`;
- em `Produtor`, as ações `Prod_A` e `Prod_B` referenciam a ação `Montagem` do objeto `Produção`. É fácil verificar que tratam-se de referências a diferentes alternativas de `Montagem`, ou seja, `Mont_P1` e `Mont_P1_P2`, respectivamente;
- em `Linha_Consumo`, na ação `Recebe`, a atribuição ao `Fator` é realizada antes das demais cláusulas. O mesmo ocorre nas construções subsequentes, incluindo a ação agregada `Comunica` em `Fábrica_Raiz`.

1 1 Semântica da Linguagem Náutilus

Uma semântica formal para a linguagem GNOME, baseada em lógica temporal, é definida em [Ramos 95], baseada em OSL - "Object Specification Language" [Sernadas *et al* 92b] e [Sernadas & SernadasC 94]. Um compilador é disponível para uma parte representativa da linguagem [Carmo & Penedo 94]. Além dos autômatos não-seqüenciais, outras semânticas alternativas estão correntemente em estudo para a linguagem GNOME e suas diversas variações, como estruturas de eventos ("event structure") [Caleiro 95], quantales [Resende 95], cálculo de objetos tipados [Ravara 95], álgebra de objetos [Aiguier *et al* 95] e, como objetivo em um futuro próximo, cálculo de situações. Os autômatos não-seqüenciais podem ser usados para dar semântica à linguagem GNOME, na sua forma original, embora alguns estudos adicionais são, provavelmente, necessários para tratar questões como classes e eventos.

No que se refere à linguagem Náutilus, a semântica de uma unidade é dada por um morfismo de reificação de autômatos não-seqüenciais onde, a cada ação corresponde uma transição e sua correspondente implementação (indivisível, finita e determinista). A principal justificativa para o uso de um morfismo de reificação é o fato das cláusulas de uma ação de um objeto poderem ser compostas de forma seqüencial, a qual pode ser vista como uma reificação interna. A apresentação da semântica que segue foi originalmente introduzida em [Menezes *et al* 95b] e é dividida nos seguintes casos:

- objeto simples (objeto cuja construção não depende de outros objetos): a semântica é um morfismo de reificação onde o autômato não-seqüencial destino, denominado base, é construído a partir de todas as alterações de valores possíveis sobre os atributos do objeto;
- reificação de um objeto Ob sobre o objeto Ob_1 : a semântica é a composição das semânticas dos objetos componentes, ou seja, o morfismo de reificação induzido por Ob (sobre Ob_1) composto com o morfismo de reificação correspondente ao Ob_1 ;
- interação ou agregação de objetos: em ambos os casos a semântica é uma sincronização das semânticas dos objetos componentes, ou seja, uma sincronização de morfismos de reificação;
- visão de um objeto: a semântica é uma encapsulação da semântica do objeto, ou seja, uma encapsulação de um morfismo de reificação;
- unidade em Náutilus composta por dois ou mais objetos terminais (objetos que não são usados na construção de outros objetos): a semântica é a composição paralela das semânticas dos objetos componentes, ou seja, a composição paralela de morfismos de reificação.

Embora considere-se a semântica proposta como operacional, também pode ser vista como denotacional até ao nível de objeto simples. Com algum esforço adicional, é possível estender a semântica denotacional a todos os níveis, analogamente à abordagem funtorial para sistemas de transições apresentada em [Winskel & Nielsen 95]. As semânticas operacional e denotacional, bem como alguns aspectos pertinentes à semântica formal, são introduzidos em apêndice.

Os detalhes da semântica de uma unidade em Náutilus são introduzidos gradativamente incluindo exemplos e discussões sobre os exemplos. No texto que segue, para um determinado atributo a de um objeto, \odot_a denota o valor inicial (de nascimento) do atributo. Assim, por exemplo, para um atributo a do tipo boolean, os seus valores possíveis são $\{\odot_a, F_a, T_a\}$. Para evitar possíveis confusões, na definição de uma reificação em Náutilus, os objetos de origem e destino são também denominados de partida e de chegada, respectivamente.

11.1 Objeto Simples

Definição 11.1 Objeto Simples. Um objeto simples em Náutilus é um objeto que não é o resultado de reificações, interações, agregações ou visões. \square

A semântica de um objeto simples em Náutilus é um morfismo de reificação de autômatos não-sequenciais, construído como segue:

- a) O objeto base é, essencialmente, um autômato não-sequencial livremente gerado, onde:
- o *CMon*-objeto de estados é gerado livremente a partir do conjunto de valores possíveis de todos os atributos do objeto;
 - a cada dois valores de um mesmo atributo, é definida uma transição. O *CMon*-objeto de transições é livremente gerado a partir do conjunto de todas as transições definidas.

Portanto, o fecho computacional não-sequencial do autômato base é capaz de simular qualquer computação sobre os atributos do objeto em Náutilus.

- b) O autômato origem é, basicamente, uma restrição reetiquetada do fecho computacional não-sequencial do base, onde:
- um morfismo de restrições definido ao nível das etiquetas é determinado pelas cláusulas das ações do objeto em Náutilus. Este morfismo possui uma elevação cartesiana ao nível dos autômatos (determinado pela técnica de fibração definida na sincronização), resultando em um morfismo de reificação;
 - um morfismo de reetiquetagens especificado ao nível das etiquetas é determinado pelos nomes das ações exportadas do objeto em Náutilus. Este morfismo induz um funtor de reetiquetagem de morfismos de reificação (determinado pela técnica de cofibração definida na encapsulação de autômatos reificados);
- c) A reetiquetagem do morfismo de reificação determina o morfismo de reificação correspondente à semântica desejada.

Na realidade, na construção do morfismo de restrições ao nível das etiquetas, uma ação depende de alguma entrada, saída ou argumento de agregação, é associada a uma família de etiquetas, indexada pelos possíveis valores correspondentes, como segue:

- ação que possui entradas ou que chama ações as quais possuem saídas: a indexação da família de etiquetas é documental, no sentido em que cada etiqueta identifica uma computação. Esta etiquetagem indexada é necessária, posteriormente, na semântica das interações ou agregações.
- ação que possui saídas, ou que chama ações as quais possuem entradas ou, ainda, que possui argumentos de agregação: diferentes etiquetas indexadas podem identificar uma mesma computação. Neste caso, a técnica de fibração substitui transições por transições paralelas.

No texto que segue, lembre-se que as categorias $\mathcal{RGr}_1(\mathit{CMon})$ e CMon são isomorfas. Adicionalmente, relativamente à construção de um conjunto de etiquetas, uma concatenação de conjuntos de símbolos, representada pela justaposição dos conjuntos, resulta no conjunto de símbolos concatenados ao nível dos componentes. Uma concatenação entre um símbolo e um conjunto, representada pela justaposição do símbolo com o conjunto, é uma concatenação de conjuntos onde o símbolo é visto como um conjunto unitário. Na definição abaixo, com o objetivo de tornar clara a identificação de uma transição (ou, eventualmente, da sua correspondente etiqueta), o seguinte critério é adotado: para um atributo a e para os valores A_1 e A_2 de a , uma transição que parte de A_1 e chega em A_2 é identificada (ou etiquetada) por $a(A_1, A_2)$.

Definição 11.2 Semântica de um Objeto Simples. Seja Ob um objeto simples em Náutilus. Seja l com valores em $l = \{1, \dots, n\}$, onde n é o número de atributos de Ob e, para cada atributo a_i onde $i \in l$, seja V_{a_i} o conjunto de todos os valores de a_i incluindo o correspondente valor inicial \odot_{a_i} (no que segue, a expressão $i \in l$ é omitida). Considere um atributo adicional (não especificado explicitamente no objeto e que objetiva indicar a ativação do mesmo) o qual possui um único valor,

denotado por \checkmark , além do valor de inicial \mathbb{O}_{\checkmark} . A semântica de Ob é um $\mathcal{R}eif\mathcal{N}Aut$ -morfismo $\text{Ob}: N_1 \rightarrow nfcN_2$ onde o $\mathcal{R}eif\mathcal{N}Aut$ -objeto base $N_2 = \langle V_2, T_2, \partial_{0_2}, \partial_{1_2}, i_2, l_2, F_2, i_2 + f_2, E_2, \text{eti}q_2 \rangle$ é tal que:

- $l_2 = \{\mathbb{O}_{\checkmark}\}^{\oplus}$ e $F_2 = \{\checkmark\}^{\oplus}$
- $V_2 = (+V_{a_i})^{\oplus} + {}_{CMon}\{\checkmark\}^{\oplus} + {}_{CMon}l_2 + {}_{CMon}F_2$
- i_2, f_2 são os morfismos canonicamente induzidos pelo coproduto categorial
- $T_2 = \{a(A_1, A_2) \mid a \text{ é atributo de } \text{Ob}, A_1, A_2 \in V_a \text{ e, se } A_2 = \mathbb{O}_a \text{ então } A_1 = \mathbb{O}_a\}^{\oplus} + {}_{CMon}\{\text{nasc}(\mathbb{O}_{\checkmark}, \mathbb{O}_{a_1} \oplus \dots \oplus \mathbb{O}_{a_n} \oplus \checkmark)\}^{\oplus} + {}_{CMon}\{\text{morte}(A_1 \oplus \dots \oplus A_n \oplus \checkmark, \checkmark) \mid A_i \in V_{a_i}\}^{\oplus}$
- $\partial_{0_2}, \partial_{1_2}$ são canonicamente induzidos por morfismos nos conjuntos geradores dos monóides onde, para cada $\alpha(\beta_1, \beta_2)$ em T_2 , $\partial_{0_2}(\alpha(\beta_1, \beta_2)) = \beta_1$ e $\partial_{1_2}(\alpha(\beta_1, \beta_2)) = \beta_2$
- $E_2 = T_2$ e $\text{eti}q_2 = \text{id}_{T_2}$

O $\mathcal{R}eif\mathcal{N}Aut$ -objeto de partida N_1 e o $\mathcal{R}eif\mathcal{N}Aut$ -morfismo $\text{Ob}: N_1 \rightarrow nfcN_2$ são obtido a partir do $nfcN_2$ através de uma restrição, seguida de uma substituição de transições (por transições paralelas), seguida de uma reetiquetagem do morfismo de reificação resultante. No que segue, para algum $\underline{x} = x_1, \dots, x_u$ e $\underline{X} = X_1, \dots, X_u$, $\underline{x} = \underline{X}$ denota $x_1 = X_1, \dots, x_u = X_u$.

Restrição. Em Ob , cada ação (no caso de uma ação sem alternativas) ou cada alternativa de cada ação (caso contrário) induz um conjunto de etiquetas as quais distinguem-se entre si dependendo dos valores dos atributos considerados. O conjunto de etiquetas correspondente a uma ação/alternativa é recursivamente definido como segue:

- seja α o conjunto cujo único elemento é a transação especificada na ação/alternativa em Ob
- se a ação é *birth*, então $\alpha = \text{nasc}(\mathbb{O}_{\checkmark}, \mathbb{O}_{a_1} \oplus \dots \oplus \mathbb{O}_{a_n} \oplus \checkmark); \alpha$
- se a ação é *death*, então $\alpha = \alpha; \{\text{morte}(A_1 \oplus \dots \oplus A_n \oplus \checkmark, \checkmark) \mid A_i \in V_{a_i}\}$
- se a ação possui entradas $\underline{x} = x_1, \dots, x_u$ ou possui chamadas que, por sua vez, possuem saídas $\underline{z} = z_1, \dots, z_v$, então $\alpha = \alpha\langle \underline{x} = \underline{X}, \underline{z} = \underline{Z} \rangle \mid X_1, \dots, X_u, Z_1, \dots, Z_v$ são valores de $x_1, \dots, x_u, z_1, \dots, z_v$, respectivamente)
- se as etiquetas em α possuem uma cláusula composta da forma *seq* $\alpha_1 \dots \alpha_p$ end *seq*, então a cláusula é substituída nas etiquetas por $(\alpha_1; \dots; \alpha_p)$
- se as etiquetas em α possuem uma cláusula composta da forma *cps* $\alpha_1 \dots \alpha_p$ end *cps*, então a cláusula é substituída nas etiquetas por $(\alpha_1 \mid \dots \mid \alpha_p)$
- se as etiquetas em α possuem uma cláusula de validação da forma *val* a << expressão, então, para cada etiqueta $\alpha_0\langle \underline{x} = \underline{X}, \underline{z} = \underline{Z} \rangle$ em α , a cláusula de validação é substituída pelo conjunto $\{a(A, \text{expressão}(\underline{B}, \underline{X}, \underline{Z})) \mid A \in V_a, \underline{B} = B_1, \dots, B_p \text{ são valores em } b_1, \dots, b_p, \text{ respectivamente}\}$ onde $\text{expressão}(\underline{B}, \underline{X}, \underline{Z})$ denota o valor resultante (válido) do cálculo da expressão para os valores de atributos \underline{B} , de entradas \underline{X} e de saídas de chamadas \underline{Z} .

O conjunto de etiquetas α deve ser sujeito às seguintes condições:

- uma etiqueta deve representar uma computação possível: assim, cada $\alpha_0\langle \underline{x} = \underline{X}, \underline{z} = \underline{Z} \rangle$ em α deve ter correspondência em alguma classe de etiquetas de $nfcN_2$ diferindo, eventualmente, no símbolo " \mid " (substituído por " \parallel " em $nfcN_2$) e desconsiderando o sufixo $\langle \underline{x} = \underline{X}, \underline{z} = \underline{Z} \rangle$. Caso não tenha correspondência, a etiqueta deve ser desprezada;
- se a ação/alternativa possui condição de ativação dependente de algum atributo, então considera-se somente as etiquetas que têm correspondência em alguma classe de etiquetas (e, conseqüentemente de transições) em $nfcN_2$ como em a) acima e cuja correspondente transição parte de um estado que satisfaz à condição de ativação.

E_1' é o monóide livremente gerado a partir do coproduto dos conjuntos de etiquetas correspondentes às ações/alternativas de Ob construídos em acima. O morfismo de etiquetas $\text{restr}_{\mathbb{E}}: E_1' \rightarrow nfcE_2$ é o canonicamente induzido pela correspondência em a). O morfismo $\text{restr}_{\mathbb{E}}$ possui uma elevação cartesiana $\text{restr}_{\mathbb{N}}: \text{restr}^{\circ} nfcN_2 \rightarrow nfcN_2$ ao nível dos autômatos onde restr é o funtor induzido pela técnica de fibração e $\text{restr}^{\circ} nfcN_2$ é a restrição de $nfcN_2$.

Substituição de transições por transições paralelas. O conjunto de etiquetas é recursivamente definido como segue:

- para cada ação/alternativa de Ob , seja β o conjunto de etiquetas α definido na restrição acima

- se a ação possui resultados para as saídas $\underline{y} = y_1, \dots, y_r$ tal que $\underline{y} = \underline{\text{exp}}_y$ onde $\underline{\text{exp}}_y = \text{exp}_{y_1}, \dots, \text{exp}_{y_r}$, então cada $\beta_0(\underline{x} = \underline{X}, \underline{z} = \underline{Z})$ em β é substituído pelo conjunto de etiquetas $\{\beta_0(\underline{x} = \underline{X}, \underline{y} = \underline{\text{exp}}_y(\underline{B}, \underline{X}, \underline{Z}), \underline{z} = \underline{Z})\}$ onde $\underline{\text{exp}}_y(\underline{B}, \underline{X}, \underline{Z})$ denotam os valores resultantes (válidos) do cálculo das correspondentes expressões para os valores $\underline{B}, \underline{X}, \underline{Z}$ e onde $\underline{B} = B_1, \dots, B_n$ é obtido a partir do estado de chegada $B_1 \oplus \dots \oplus B_n$ da transição correspondente a β_0 em nfcN_2 ;
- se a ação possui chamadas que, por sua vez possuem entradas $\underline{w} = w_1, \dots, w_s$ tal que $\underline{w} = \underline{\text{exp}}_w$ onde $\underline{\text{exp}}_w = \text{exp}_{w_1}, \dots, \text{exp}_{w_s}$, então cada $\{\beta_0(\underline{x} = \underline{X}, \underline{y} = \underline{\text{exp}}_y(\underline{B}, \underline{X}, \underline{Z}), \underline{z} = \underline{Z})\}$ em β é substituído pelo conjunto de etiquetas $\{\beta_0(\underline{x} = \underline{X}, \underline{y} = \underline{\text{exp}}_y(\underline{B}, \underline{X}, \underline{Z}), \underline{w} = \underline{\text{exp}}_w(\underline{A}, \underline{X}, \underline{Z}), \underline{z} = \underline{Z})\}$ onde $\underline{\text{exp}}_w(\underline{A}, \underline{X}, \underline{Z})$ denotam os valores resultantes (válidos) do cálculo das correspondentes expressões para os valores de atributos \underline{A} , de entradas \underline{X} e de saídas de chamadas \underline{Z} e onde $\underline{A} = A_1, \dots, A_n$ é obtido a partir do estado de partida $A_1 \oplus \dots \oplus A_n$ da transição correspondente a β_0 em nfcN_2 ;
- se a ação possui argumentos de agregação, o procedimento é análogo ao adotado para entradas de chamadas.

E_1'' é o monóide livremente gerado a partir do coproduto dos conjuntos de etiquetas correspondentes às ações/alternativas de Ob construídos acima. O morfismo de etiquetas $\text{subs}_E: E_1'' \rightarrow \text{nfcE}_1'$ é o canonicamente induzido como na restrição (note-se que subs_E não é um monomorfismo). O morfismo subs_E possui uma elevação cartesiana $\text{subs}_N: \text{subs} \circ \text{restr} \circ \text{nfcN}_2 \rightarrow \text{restr} \circ \text{nfcN}_2$ ao nível dos autômatos onde subs é o funtor induzido pela técnica de fibração e $\text{subs} \circ \text{restr} \circ \text{nfcN}_2$ substitui as transições em $\text{restr} \circ \text{nfcN}_2$ que dependem de saídas, entradas de chamadas e argumentos de agregação por famílias de transições paralelas indexadas pelos correspondentes valores.

Reetiquetação. E_1 é um CMon -objeto livremente gerado a partir do conjunto E_1 o qual é construído a partir dos nomes das ações exportadas de Ob . Uma ação (exportada) a qual possui entradas, saídas, chamadas ou argumentos de agregação induz uma família de etiquetas indexadas pelos correspondentes valores. O conjunto E_1 é o menor conjunto tal que, para cada ação exportada de nome ϕ , com entradas \underline{x} , saídas \underline{y} , entradas de chamadas ou argumentos de agregação \underline{w} e saídas de chamadas \underline{z} , $\phi(\underline{x} = \underline{X}, \underline{y} = \underline{\text{exp}}_y, \underline{w} = \underline{\text{exp}}_w, \underline{z} = \underline{Z})$ é elemento de E_1 , onde $\underline{X}, \underline{W}$ são valores de $\underline{x}, \underline{w}$, respectivamente e $\underline{\text{exp}}_y$ e $\underline{\text{exp}}_w$ são expressões válidas como resultados de \underline{y} e \underline{w} , respectivamente, como na substituição.

O morfismo de reetiquetação $\text{reetiq}: E_1'' \rightarrow E_1$ é canonicamente induzido pelo morfismo apontado nos geradores dos monóides (estendidos como conjuntos apontados), onde cada etiqueta $\beta_0(\underline{x} = \underline{X}, \underline{y} = \underline{\text{exp}}_y, \underline{w} = \underline{\text{exp}}_w, \underline{z} = \underline{Z})$ é associada ao elemento distinguido, se a ação é encapsulada (e, neste caso, \underline{x} e \underline{y} são seqüências vazias) ou ao correspondente nome da ação $\phi(\underline{x} = \underline{X}, \underline{y} = \underline{\text{exp}}_y, \underline{w} = \underline{\text{exp}}_w, \underline{z} = \underline{Z})$, se a ação é exportada. O morfismo reetiq induz um funtor de reetiquetação de autômatos reificados reetiq .

Assim, $\text{Ob} = \text{reetiq}(\text{subs}_N \circ \text{restr}_N: \text{subs} \circ \text{restr} \circ \text{nfcN}_2 \rightarrow \text{nfcN}_2) = (\text{subs}_N \circ \text{restr}_N) \setminus \text{reetiq}: N_1 \rightarrow \text{nfcN}_2$ onde $N_1 = \text{reetiq} \circ \text{subs} \circ \text{restr} \circ \text{nfcN}_2$. Entre os estados iniciais, é escolhido o referente ao valor \ominus . \square

No caso em que um objeto simples não possui saídas nem chamadas com entradas, a etapa referente à substituição não possui qualquer efeito e, conseqüentemente, pode ser desconsiderada, como no exemplo que segue:

Exemplo 11.3 Considere a especificação do objeto Obj em Náutilus abaixo (inicialmente, desconsidere a coluna à direita).

```

object Obj
export
  Inic
  Atu
  Fim

```

```

category
  birth Inic
  death Fim
body
  slot a: boolean
  slot b: boolean
  act Inic
    alt I1
      seq
        val a << false
        val b << false
      end seq
    alt I2
      cps
        val a << false
        val b << true
      end cps
  act Atu
    enb a = false
    cps
      val a << true
      val b << true
    end cps
  act Fim
    enb a = true and b = true
end Obj

```

$$\begin{array}{l}
t_0: \mathbb{O}_{\checkmark} \rightarrow \mathbb{O}_a \oplus \mathbb{O}_b \oplus \checkmark \\
t_1: \mathbb{O}_a \rightarrow F_a \\
t_2: \mathbb{O}_b \rightarrow F_b \\
t_0: \mathbb{O}_{\checkmark} \rightarrow \mathbb{O}_a \oplus \mathbb{O}_b \oplus \checkmark \\
t_1: \mathbb{O}_a \rightarrow F_a \\
t_3: \mathbb{O}_b \rightarrow T_b \\
t_4: F_a \rightarrow T_a \\
t_3: \mathbb{O}_b \rightarrow T_b, t_5: F_b \rightarrow T_b, t_6: T_b \rightarrow T_b \\
t_7: T_a \oplus T_b \rightarrow \hat{\top}
\end{array}$$

A semântica de Obj é um morfismo de reificação $\text{Obj}: N_1 \rightarrow \text{nfc}N_2$ onde N_2 é tal que:

- estados iniciais: $I_2 = \{\mathbb{O}_{\checkmark}\}^{\oplus}$
- estados finais: $F_2 = \{\hat{\top}\}^{\oplus}$
- estados: $V_2 = V_a^{\oplus} + \text{CMon}V_b^{\oplus} + \text{CMon}I_2 + \text{CMon}F_2$ onde $V_a = \{\mathbb{O}_a, F_a, T_a\}$ e $V_b = \{\mathbb{O}_b, F_b, T_b\}$;
- transições: um CMon -objeto T_2 determinado pelas seguintes transições locais onde, para k com valores em $\{1, 2\}$, A_k e B_k são valores dos atributos a e b , respectivamente:

$$\begin{array}{ll}
\text{nasc}(\mathbb{O}_{\checkmark}, \mathbb{O}_a \oplus \mathbb{O}_b \oplus \checkmark): \mathbb{O}_{\checkmark} \rightarrow \mathbb{O}_a \oplus \mathbb{O}_b \oplus \checkmark & \checkmark(\checkmark, \checkmark): \checkmark \rightarrow \checkmark \\
a(\mathbb{O}_a, \mathbb{O}_a): \mathbb{O}_a \rightarrow \mathbb{O}_a & b(\mathbb{O}_b, \mathbb{O}_b): \mathbb{O}_b \rightarrow \mathbb{O}_b \\
a(\mathbb{O}_a, F_a): \mathbb{O}_a \rightarrow F_a & b(\mathbb{O}_b, F_b): \mathbb{O}_b \rightarrow F_b \\
a(\mathbb{O}_a, T_a): \mathbb{O}_a \rightarrow T_a & b(\mathbb{O}_b, T_b): \mathbb{O}_b \rightarrow T_b \\
a(F_a, F_a): F_a \rightarrow F_a & b(F_b, F_b): F_b \rightarrow F_b \\
a(F_a, T_a): F_a \rightarrow T_a & b(F_b, T_b): F_b \rightarrow T_b \\
a(T_a, T_a): T_a \rightarrow T_a & b(T_b, T_b): T_b \rightarrow T_b \\
a(T_a, F_a): T_a \rightarrow F_a & b(T_b, F_b): T_b \rightarrow F_b \\
\text{morte}(A_1 \oplus B_1 \oplus \hat{\top}): A_1 \oplus B_1 \rightarrow \hat{\top} &
\end{array}$$

- para simplificar a representação gráfica deste exemplo, a seguinte etiquetação é adotada, a qual possui correspondência no objeto (veja a coluna à direita de Obj). Considere que as etiquetas das demais transições coincidem com as identificações das transições (como na definição acima). Deve-se destacar que o critério usado para a etiquetação não é importante para a construção proposta;

$$\begin{array}{l}
\text{nasc}(\mathbb{O}_{\checkmark}, \mathbb{O}_a \oplus \mathbb{O}_b \oplus \checkmark) \mapsto t_0 \\
a(\mathbb{O}_a, F_a) \mapsto t_1 \\
b(\mathbb{O}_b, F_b) \mapsto t_2 \\
b(\mathbb{O}_b, T_b) \mapsto t_3 \\
a(F_a, T_a) \mapsto t_4 \\
b(F_b, T_b) \mapsto t_5 \\
b(T_b, T_b) \mapsto t_6
\end{array}$$

$$\text{morte}(T_a \oplus T_b) \rightarrow t_7$$

Como Obj não possui saídas nem chamadas com entradas, a etapa referente à substituição não é necessária. Assim, o autômato não-sequencial N_1 é simplesmente uma restrição reetiquetada de $\text{nfc}N_2$, como segue:

- $\text{restr} \circ \text{nfc}N_2$ é o autômato resultante da restrição onde o funtor restr é induzido por um monomorfismo de etiquetas restr_E que, por sua vez, é induzido como abaixo a partir das cláusulas de cada ação. O morfismo restr_E possui uma elevação cartesiana $\text{restr}_N: \text{restr} \circ \text{nfc}N_2 \rightarrow \text{nfc}N_2$ ao nível dos autômatos;

$$\begin{aligned} t_0; t_1; t_2 &\rightarrow t_0; t_1; t_2 \\ t_0; (t_1 \mid t_3) &\rightarrow t_0; (t_1 \parallel t_3) \\ t_4 \mid t_3 &\rightarrow t_4 \parallel t_3 \\ t_4 \mid t_5 &\rightarrow t_4 \parallel t_5 \\ t_4 \mid t_6 &\rightarrow t_4 \parallel t_6 \\ t_7 &\rightarrow t_7 \end{aligned}$$

- $N_1 = \text{reetiq} \circ \text{restr} \circ \text{nfc}N_2$ é o autômato resultante da reetiquetagem onde o funtor reetiq é induzido por um morfismo de etiquetas reetiq que, por sua vez, é induzido como abaixo, a partir das identificações das ações exportadas:

$$\begin{aligned} t_0; t_1; t_2 &\rightarrow \text{Inic} \\ t_0; (t_1 \mid t_3) &\rightarrow \text{Inic} \\ t_4 \mid t_3 &\rightarrow \text{Atu} \\ t_4 \mid t_5 &\rightarrow \text{Atu} \\ t_4 \mid t_6 &\rightarrow \text{Atu} \\ t_7 &\rightarrow \text{Fim} \end{aligned}$$

Portanto, N_1 possui os mesmos estados, estados iniciais e estados finais de N_2 sendo as suas transições determinadas pelas seguintes transições locais etiquetadas:

$$\begin{aligned} \text{Inic}: \textcircled{\surd} &\rightarrow F_a \oplus F_b \oplus \surd \\ \text{Inic}: \textcircled{\surd} &\rightarrow F_a \oplus T_b \oplus \surd \\ \text{Atu}: F_a \oplus \textcircled{\surd} &\rightarrow T_a \oplus T_b \oplus \surd \\ \text{Atu}: F_a \oplus F_b \oplus \surd &\rightarrow T_a \oplus T_b \oplus \surd \\ \text{Atu}: F_a \oplus T_b \oplus \surd &\rightarrow T_a \oplus T_b \oplus \surd \\ \text{Fim}: T_a \oplus T_b \oplus \surd &\rightarrow \hat{\tau} \end{aligned}$$

O morfismo $\text{Obj}: N_1 \rightarrow \text{nfc}N_2$ é determinado como segue, onde somente as etiquetas estão representadas:

$$\begin{aligned} \text{Inic} &\rightarrow t_0; t_1; t_2 \\ \text{Inic} &\rightarrow t_0; (t_1 \parallel t_3) \\ \text{Atu} &\rightarrow t_4 \parallel t_3 \\ \text{Atu} &\rightarrow t_4 \parallel t_5 \\ \text{Atu} &\rightarrow t_4 \parallel t_6 \\ \text{Fim} &\rightarrow t_7 \end{aligned}$$

No momento da execução, é escolhido o estado inicial referente ao valor $\textcircled{\surd}$. A representação gráfica dos autômatos considera esta escolha de estado inicial. O morfismo de reificação $\text{Obj}: N_1 \rightarrow \text{nfc}N_2$ é ilustrado na Figura 11.1, onde a representação de $\text{nfc}N_2$ é parcial sendo o valor \surd omitido nas somas. Note-se que:

- a restrição de $\text{nfc}N_2$ que origina N_1 é representada com um traço diferenciado;
- as alternativas I1 e I2 da ação Inic determinam duas transições em N_1 , ambas com a mesma etiqueta Inic e correspondem às transições $t_0; t_1; t_2$ e $t_0; (t_1 \parallel t_3)$ de $\text{nfc}N_2$, respectivamente;
- a ação Atu determina duas transições em N_1 , com diferentes estados origem. Por simplicidade, a terceira implementação $t_4 \parallel t_3$ de Atu não é representada pois, em N_1 , não é atingível a partir do estado inicial. \square

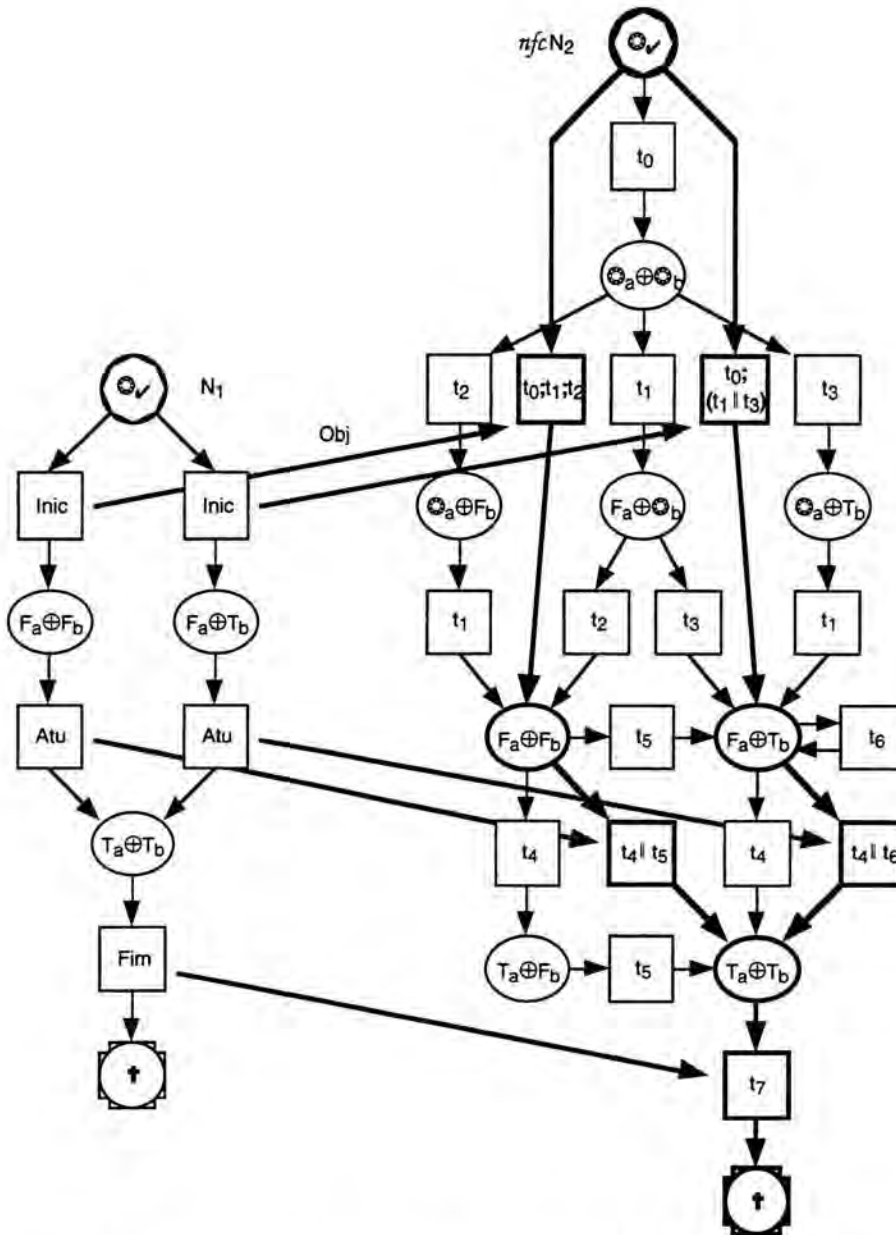


Figura 11.1 Semântica de um objeto em Náutilus como um morfismo de reificação

11.2 Reificação

A semântica de uma reificação é a composição das semânticas dos objetos de partida e de chegada, ou seja, o morfismo de reificação correspondente ao de partida (sobre o de chegada) composto com o morfismo de reificação correspondente ao de chegada. A semântica do objeto de partida é um morfismo de reificação construído de forma análoga ao de um objeto simples: é uma restrição reetiquetada do fecho computacional de um autômato. Note-se que, a etapa de substituição não é necessária, pois, contrariamente ao objeto simples, todas as informações referentes às entradas/saídas são herdadas do objeto de chegada.

Uma reificação pode ser dependente de estado, ou seja, uma ação do objeto de partida pode possuir implementações alternativas, especificadas como segue:

- explicitamente: as alternativas de implementação são especificadas no objeto de partida;
- implicitamente: uma ou mais ações componentes da computação do objeto de chegada que implementa a ação do objeto de partida possuem alternativas.

Em ambos os casos, na semântica do objeto de partida, cada alternativa determina uma transição no autômato origem (mas todas com a mesma etiqueta) e a sua correspondente implementação no autômato destino. Referente às implementações alternativas de uma ação, note-se que:

- no caso a), as correspondentes transições no autômato destino podem possuir diferentes etiquetas: o morfismo de reificação não necessariamente preserva a etiqueta da ação;
- no caso b), as correspondentes transições do autômato destino possuem a mesma etiquetas: o morfismo de reificação preserva a etiqueta da ação.

Definição 11.4 Semântica de uma Reificação. Seja Ob um objeto reificado sobre um objeto Ob_1 . Seja $Ob_1: N_1 \rightarrow nfcN_2$ a semântica de Ob_1 . A semântica de Ob reificado sobre Ob_1 é a composição das correspondentes semânticas (morfismos de reificação), ou seja, $Ob_1 \circ Ob$ tal que $Ob: M \rightarrow nfcN_1$ onde o *ReifN/Aut*-objeto origem M é uma restrição reetiquetada de $nfcN_1$:

Restrição. Em Ob , cada ação (no caso de uma ação sem alternativas) ou cada alternativa de cada ação (caso contrário) induz um conjunto de etiquetas, recursivamente definido como segue:

- seja α o conjunto composto pela transação especificada na ação/alternativa
- se as etiquetas em α possuem uma cláusula composta da forma $seq \alpha_1 \dots \alpha_p \text{ end } seq$, então a cláusula é substituída nas etiquetas por $(\alpha_1; \dots; \alpha_p)$
- se as etiquetas em α possuem uma cláusula composta da forma $cps \alpha_1 \dots \alpha_p \text{ end } cps$, então a cláusula é substituída nas etiquetas por $(\alpha_1 | \dots | \alpha_p)$
- se as etiquetas em α referenciam alguma etiqueta ϕ_0 importada de Ob_1 , então ϕ_0 é substituído pelo conjunto de etiquetas $\{\phi_0(\underline{x} = \underline{X}, \underline{y} = \underline{Y}, \underline{w} = \underline{W}, \underline{z} = \underline{Z}) \in E_1\}$

Cada etiqueta do conjunto resultante acima deve ter correspondência em alguma classe de etiquetas de $nfcN_1$, diferindo, eventualmente, no símbolo "|" (substituído por "||" em $nfcN_1$). Caso não tenha correspondência, a etiqueta deve ser desprezada. Adicionalmente, uma etiqueta α_0 em α composta por etiquetas $\phi_1(\underline{x}_1 = \underline{X}_1, \underline{y}_1 = \underline{Y}_1, \underline{w}_1 = \underline{W}_1, \underline{z}_1 = \underline{Z}_1), \dots, \phi_k(\underline{x}_k = \underline{X}_k, \underline{y}_k = \underline{Y}_k, \underline{w}_k = \underline{W}_k, \underline{z}_k = \underline{Z}_k)$ de E_1 , é substituída pela etiqueta $\alpha_0(\underline{x}_{1k} = \underline{X}_{1k}, \underline{y}_{1k} = \underline{Y}_{1k}, \underline{w}_{1k} = \underline{W}_{1k}, \underline{z}_{1k} = \underline{Z}_{1k})$ onde $\underline{x}_{1k} = \underline{x}_1, \dots, \underline{x}_k$ e assim sucessivamente.

E' é o monóide livremente gerado a partir do coproduto dos conjuntos de etiquetas correspondentes às ações/alternativas de Ob construídos acima. O morfismo de etiquetas $restr_E: E' \rightarrow nfcE_1$ é o canonicamente induzido pela correspondência acima. O morfismo $restr_E$ possui uma elevação cartesiana $restr_N: restr \circ nfcN_1 \rightarrow nfcN_1$ ao nível dos autômatos onde $restr$ é o funtor induzido pela técnica de fibração e $restr \circ nfcN_1$ é a restrição de $nfcN_1$.

Reetiquetação. E é o *CMon*-objeto livremente gerado a partir do conjunto E o qual é construído a partir dos nomes das ações exportadas de Ob . O conjunto E é o menor conjunto tal que, para cada ação exportada de nome ϕ , com entradas \underline{x} , saídas \underline{y} , entradas de chamadas ou argumentos de agregação \underline{w} e saídas de chamadas \underline{z} , $\phi(\underline{x} = \underline{X}, \underline{y} = \underline{Y}, \underline{w} = \underline{W}, \underline{z} = \underline{Z})$ é elemento de E , onde $\underline{X}, \underline{Y}, \underline{Z}, \underline{W}$ são valores (válidos) de $\underline{x}, \underline{y}, \underline{z}, \underline{w}$, respectivamente, como na restrição.

O morfismo de reetiquetação $reetiq: E' \rightarrow E$ é canonicamente induzido pelo morfismo apontado nos geradores dos monóides, estendidos como conjuntos apontados, onde cada etiqueta $\alpha_0(\underline{x} = \underline{X}, \underline{y} = \underline{Y}, \underline{w} = \underline{W}, \underline{z} = \underline{Z})$ é associada ao elemento distinguido, se a ação é encapsulada (e, neste caso, \underline{x} e \underline{y} são seqüências vazias) ou ao correspondente nome da ação $\phi(\underline{x} = \underline{X}, \underline{y} = \underline{Y}, \underline{w} = \underline{W}, \underline{z} = \underline{Z})$, se a ação é exportada. O morfismo $reetiq$ induz um funtor de reetiquetação de autômatos reificados *reetiq*.

Assim, $Ob = reetiq(Ob_1 \circ restr_N: restr \circ nfcN_1 \rightarrow nfcN_2) = (Ob_1 \circ restr_N) \backslash reetiq: M \rightarrow nfcN_2$ onde $M = reetiq \circ restr \circ nfcN_1$. Entre os estados iniciais, é escolhido o referente ao valor \ominus_{\checkmark} . \square

Exemplo 11.5 O objeto `Abstr` é implementado sobre o objeto `Concr`, onde:

- `Abstr` especifica implementações alternativas para uma mesma ação;
- `Concr` possui uma ação com alternativas a qual é referenciada na reificação de `Abstr`;

```
object Abstr over Concr
export
  Nasc X
category
  birth Nasc
  death Morte
body
  act Nasc
    alt N1
      N
    alt N2
      seq
        N
        A
        C
      end seq
    act X
      seq
        A
        B
      end seq
    act Morte
      M
end Abstr
```

```
object Concr
export
  A B C N M
category
  birth N
  death M
body
  slot state: 1..4
  act N
    val state << 1
  act A
    alt A1
      enb state = 1
      val state << 2
    alt A2
      enb state = 1
      val state << 2
    alt A3
      enb state = 1
      val state << 3
  act B
    enb state = 2
    val state << 4
  act C
    enb state = 3
    val state << 4
  act M
    enb state = 4
end Concr
```

O morfismo que implementa o objeto `Abstr` sobre `Concr` é ilustrado na Figura 11.2 (os parênteses nas etiquetas das transições objetivam auxiliar na identificação de quais alternativas originaram as transições). Deve-se destacar que:

- as transições etiquetadas por `X` em `Abstr` e `A` em `Concr` (para as alternativas `A1` e `A2`) ilustram a etiquetagem não-extensional;
- as duas transições etiquetadas por `X` são implementadas em computações distintas, mas ambas etiquetadas por `A;B`;
- as duas transições etiquetadas por `Nasc` são implementadas por computações distintas e com etiquetas também distintas;
- como a ação `Morte` não é exportada, a correspondente transição no autômato origem é encapsulada, ou seja, é etiquetada pelo símbolo especial τ .

A composição da reificação ilustrada na Figura 11.2 com a reificação de `Concr` sobre a sua base define a semântica de `Abstr` sobre `Concr`. \square

O exemplo a seguir ilustra a semântica de reificação de um objeto como a composição de reificações de autômatos não-sequenciais.

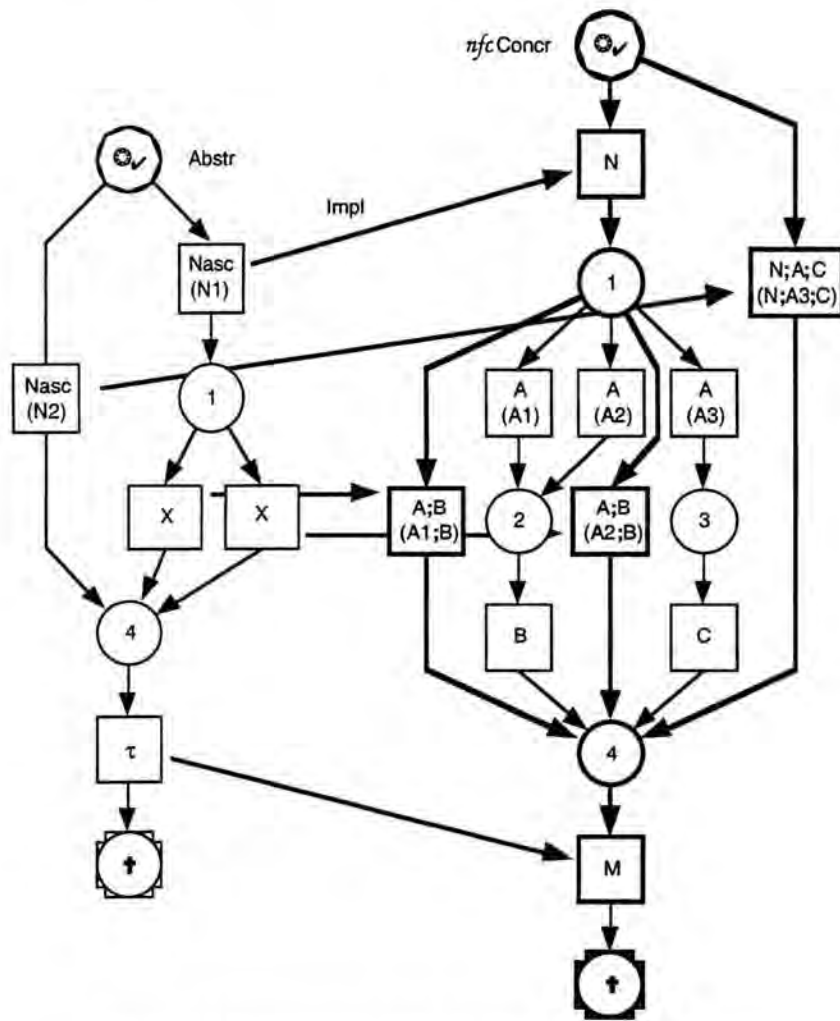


Figura 11.2 Representação gráfica da semântica de uma reificação

Exemplo 11.6 O objeto High_Obj é implementado sobre o objeto Obj do Exemplo 11.3:

```

object High_Obj over Obj
export
  All Part End
category
  birth All Part
  death All End
body
  act All
  seq
    Inic
    Atu
    Fim
  end seq
    
```

```

act Part
  seq
    Inic
    Atu
  end seq
act End
  Fim
end High_Obj
    
```

a) o morfismo que implementa o objeto High_Obj sobre o objeto Obj é como segue, onde somente as etiquetas das transições são explicitadas. Deve-se reparar que, como o autômato origem referente ao Obj possui duas transições distintas para as etiquetas Inic e Atu, existem duas transições compostas distintas etiquetadas por Inic;Atu:

All \rightarrow Inic;Atu;Fim
 All \rightarrow Inic;Atu;Fim
 Part \rightarrow Inic;Atu
 Part \rightarrow Inic;Atu
 End \rightarrow Fim

b) o morfismo que implementa o objeto `High_Obj` sobre a base de `Obj` é como segue, onde somente as etiquetas das transições são explicitadas:

All \rightarrow $t_0;t_1;t_2;(t_4 \parallel t_5);t_7$
 All \rightarrow $t_0;(t_1 \parallel t_3);(t_4 \parallel t_6);t_7$
 Part \rightarrow $t_0;t_1;t_2;(t_4 \parallel t_5)$
 Part \rightarrow $t_0;(t_1 \parallel t_3);(t_4 \parallel t_6)$
 End \rightarrow t_7

□

11.3 Visão

A semântica de uma visão de um objeto em Náutilus é, simplesmente, uma encapsulação do correspondente autômato não-sequencial reificado.

Definição 11.7 Semântica de uma Visão. Seja Ob_2 uma visão de um objeto Ob_1 . Seja $Ob_1: N_1 \rightarrow nfcM_1$ a semântica de Ob_1 onde E_1 é o *CMon*-objeto de etiquetas de N_1 . A semântica de Ob_2 é uma encapsulação de Ob_1 onde o morfismo de encapsulação $caps: E_1 \rightarrow E_2$ é o *CMon*-morfismo canonicamente induzido pelo morfismo apontado $caps: E_1 \rightarrow E_2$ sendo E_2 o *CMon*-objeto de etiquetas construído a partir das ações exportadas de Ob_2 . O morfismo $caps = sub^R$ onde $sub: E_2 \rightarrow E_1$ é definido a partir da especificação da visão, como segue:

- se Ob_2 exporta a ação ϕ_2 a qual é uma visão da ação ϕ_1 de Ob_1 , então:
 - para todo $\phi_1(\underline{u} = \underline{U})$ elemento de E_1 , $\phi_2(\underline{u} = \underline{U})$ é elemento de E_2
 - $sub(\phi_2(\underline{u} = \underline{U})) = \phi_1(\underline{u} = \underline{U})$
- adicionalmente, E_2 é o menor conjunto apontado que satisfaz acima.

□

Exemplo 11.8 Considere o objeto `Obj` do Exemplo 11.3. Suponha a encapsulação da ação `Atu` como especificado no seguinte trecho adicional em Náutilus:

```
object Janela_Obj view of Obj
export
  Inic
  Fim
end Janela_Obj
```

Então o morfismo de encapsulação ao nível das etiquetas é determinado como abaixo e o correspondente morfismo de implementação é ilustrado na Figura 11.3.

Inic \rightarrow Inic
 Atu \rightarrow τ
 Fim \rightarrow Fim

□

11.4 Interação ou Agregação

A semântica de uma interação ou agregação de objetos em Náutilus é dada por uma sincronização dos correspondentes autômatos não-sequenciais reificados. Uma agregação determina, adicionalmente, uma reetiquetagem do autômato reificado resultante da sincronização. Uma interação ou agregação de ações que dependem de entradas, saídas ou argumentos de agregação determina uma sincronização de famílias indexadas de transições onde as transições são identificadas sempre que os valores dos correspondentes índices coincidem.

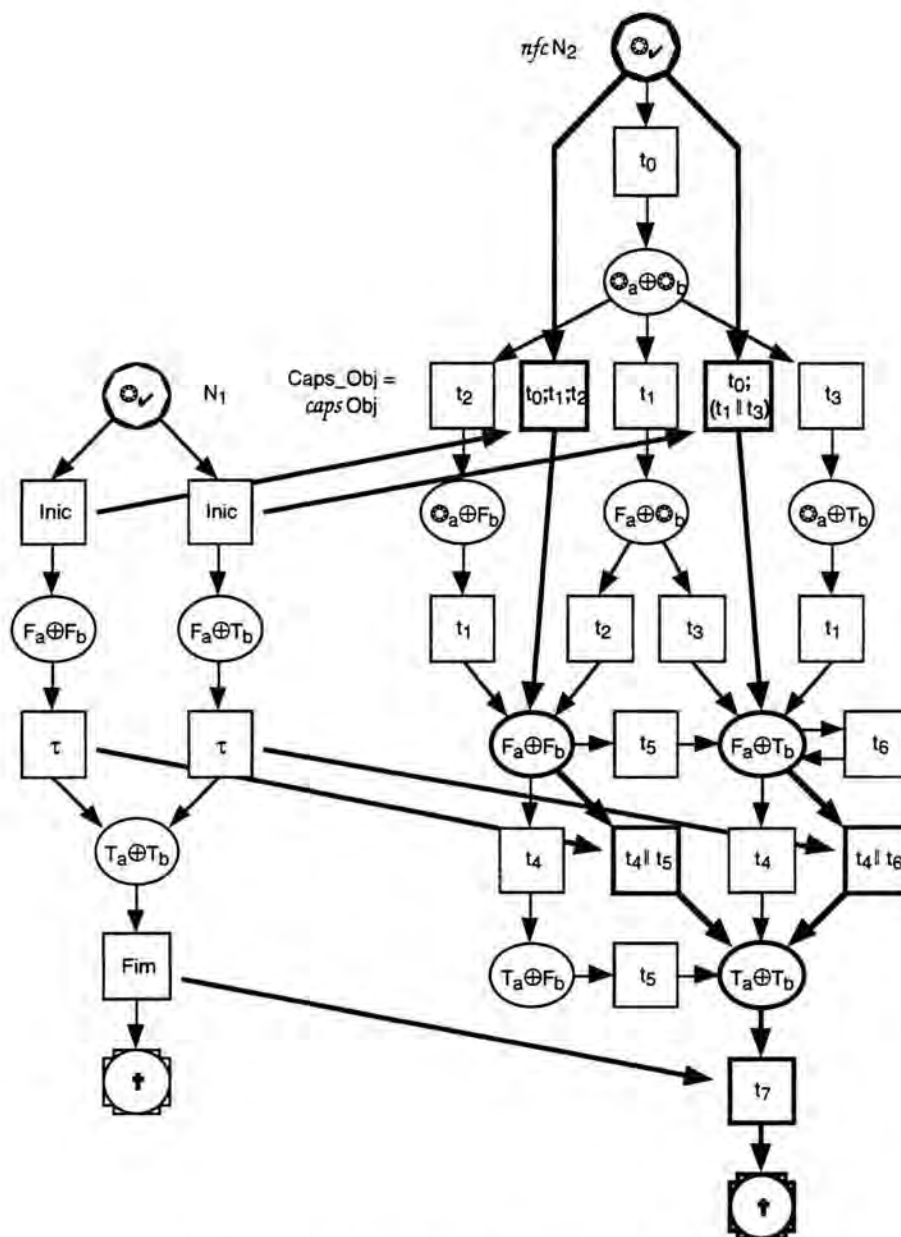


Figura 11.3 Representação gráfica da semântica de uma visão

Com o objetivo de simplificar o entendimento do que segue, um ponto de interrogação que precede a etiqueta de uma transição indica que a correspondente ação é da categoria request. No texto que segue, é omitido que $i \in I$ para algum conjunto $I = \{1, \dots, n\}$ onde $n > 1$ e, uma etiqueta com a forma $\phi(x = X, y = Y, w = W, z = Z)$ é abreviada para $\phi(u = U)$ onde $u = x, y, w, z$ e $U = X, Y, W, Z$, respectivamente.

Definição 11.9 Semântica de uma Interação. Seja $\{Ob_i\}$ um conjunto de objetos em Náutilus e seja $\{Ob_i; N_{1i} \rightarrow nfcN_{2i}\}$ o conjunto das correspondentes semânticas. A semântica da interação de $\{Ob_i\}$ é a sincronização das semânticas $\{Ob_i\}$, ou seja, $\| Ob_i; \| N_{1i} \rightarrow nfc \times N_{2i}$, onde a tabela de sincronizações é determinada a partir da especificação dos objetos componentes em Náutilus, como segue:

- seja *Canal* o *CMon*-objeto livremente gerado por *Canal* o qual é o menor conjunto apontado tal que para todo o objeto Ob_k em $\{Ob_i\}$ e para toda a ação ϕ de Ob_k , se ϕ chama as ações $\phi_{s_1}, \dots, \phi_{s_r}$ em $Ob_{s_1}, \dots, Ob_{s_r}$ e se para toda a etiqueta $\phi(\underline{u} = \underline{U})$ da semântica de Ob_k , existem as etiquetas $\phi_{s_1}(\underline{u} = \underline{U}), \dots, \phi_{s_r}(\underline{u} = \underline{U})$ nas semânticas de $Ob_{s_1}, \dots, Ob_{s_r}$, então a etiqueta $\phi(\underline{u} = \underline{U}) \mid \phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})$ é elemento de *Canal*;
- para cada Ob_k em $\{Ob_i\}$, seja E_{k_1}' o *CMon*-objeto livremente gerado por E_{k_1}' e seja $chama_k: E_{k_1}' \rightarrow Canal$ o *CMon*-morfismo canonicamente induzido pelo morfismo apontado $chama_k: E_{k_1}' \rightarrow Canal$ tais que, para toda a ação ϕ de Ob_k :
 - se ϕ chama as ações $\phi_{s_1}, \dots, \phi_{s_r}$ em $Ob_{s_1}, \dots, Ob_{s_r}$ e se para toda a etiqueta $\phi(\underline{u} = \underline{U})$ da semântica de Ob_k , $\phi(\underline{u} = \underline{U}) \mid \phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})$ é elemento de *Canal*, então $\phi(\underline{u} = \underline{U})$ é elemento de E_{k_1}' e $chama(\phi(\underline{u} = \underline{U})) = \phi(\underline{u} = \underline{U}) \mid \phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})$
 - caso contrário, se ϕ é da categoria *request*, então, para toda a etiqueta $\phi(\underline{u} = \underline{U})$ da semântica de Ob , $\phi(\underline{u} = \underline{U})$ é elemento de E_{k_1}' e $chama(\phi(\underline{u} = \underline{U})) = \tau$, onde τ é o elemento distinguido;
 - adicionalmente, E_{k_1}' é o menor conjunto apontado que satisfaz acima.

Entre os estados iniciais, é escolhido o referente ao valor $\oplus_1 \oplus \dots \oplus \oplus_n$ onde $\oplus_1, \dots, \oplus_n$ são os estados iniciais escolhidos para Ob_1, \dots, Ob_n , respectivamente. \square

A semântica de uma agregação é uma sincronização, construída de forma análoga a da interação, seguida de uma reetiquetagem. No objeto agregador, uma ação composta por uma única ação é tratada exclusivamente como uma reetiquetagem.

Definição 11.10 Semântica de uma Agregação. Seja $\{Ob_i\}$ um conjunto de objetos em Náutilus e seja $\{Ob_i: N_{1_i} \rightarrow nfcN_{2_i}\}$ as correspondentes semânticas. A semântica da agregação de $\{Ob_i\}$ é a sincronização reetiquetada das correspondentes semânticas (morfismos de reificação), ou seja, $reetiq(\parallel Ob_i: \parallel N_{1_i} \rightarrow nfc \times N_{2_i})$, onde:

Sincronização. A tabela de sincronizações é determinada a partir de:

- seja *Canal* o *CMon*-objeto livremente gerado por *Canal* o qual é o menor conjunto apontado tal que, para toda a ação ϕ do objeto agregador, para $r > 1$, se ϕ é composta pelas ações $\phi_{s_1}, \dots, \phi_{s_r}$ de $Ob_{s_1}, \dots, Ob_{s_r}$ e, para todo $\langle \underline{u} = \underline{U} \rangle$, se existem as etiquetas $\phi_{s_1}(\underline{u} = \underline{U}), \dots, \phi_{s_r}(\underline{u} = \underline{U})$ nas semânticas de $Ob_{s_1}, \dots, Ob_{s_r}$, então a etiqueta $\phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})$ é elemento de *Canal*;
- para cada Ob_k em $\{Ob_i\}$, seja E_{k_1}' o *CMon*-objeto livremente gerado por E_{k_1}' e seja $chama_k: E_{k_1}' \rightarrow Canal$ o *CMon*-morfismo canonicamente induzido pelo morfismo apontado $chama_k: E_{k_1}' \rightarrow Canal$ tais que, para toda a ação ϕ de Ob_k :
 - se ϕ é da categoria *request*, então, para toda a etiqueta $\phi(\underline{u} = \underline{U})$ da semântica de Ob_k , $\phi(\underline{u} = \underline{U})$ é elemento de E_{k_1}' e $chama(\phi(\underline{u} = \underline{U})) = \tau$, onde τ é o elemento distinguido;
 - adicionalmente, E_{k_1}' é o menor conjunto apontado que satisfaz acima.

Reetiquetagem. Suponha $\parallel E_i$ o *CMon*-objeto de etiquetas de $\parallel Ob_i$. Seja E' o *CMon*-objeto livremente gerado por E' e seja $reetiq: \parallel E_i \rightarrow E'$ o *CMon*-morfismo canonicamente induzido pelo morfismo apontado $reetiq: \parallel E_i \rightarrow E'$ tais que, para toda a ação ϕ do objeto agregador (suponha τ o elemento distinguido de E'):

- se ϕ é composta pelas ações $\phi_{s_1}, \dots, \phi_{s_r}$ de $Ob_{s_1}, \dots, Ob_{s_r}$, respectivamente, para $r > 1$, então, para todas as etiquetas $\phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})$ de *Canal*, se ϕ é exportada, então $\phi(\underline{u} = \underline{U})$ é elemento de E' e $reetiq(\phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})) = \phi(\underline{u} = \underline{U})$; caso contrário, $reetiq(\phi_{s_1}(\underline{u} = \underline{U}) \mid \dots \mid \phi_{s_r}(\underline{u} = \underline{U})) = \tau$
- se ϕ é composta pela ação ϕ_s de Ob_s , então, para todas as etiquetas $\phi_s(\underline{u} = \underline{U})$ na semântica de Ob_s , se ϕ é exportada e não é da categoria *request*, então $\phi(\underline{u} = \underline{U})$ é elemento de E' e $reetiq(\phi_s(\underline{u} = \underline{U})) = \phi(\underline{u} = \underline{U})$; caso contrário, $reetiq(\phi_s(\underline{u} = \underline{U})) = \tau$
- adicionalmente, E' é o menor conjunto apontado que satisfaz acima.

Entre os estados iniciais, é escolhido o referente ao valor $\oplus_1 \oplus \dots \oplus \oplus_n$ onde $\oplus_1, \dots, \oplus_n$ são os estados iniciais escolhidos para Ob_1, \dots, Ob_n , respectivamente. \square

Exemplo 11.11 Suponha a agregação de dois objetos, um Produtor e um Consumidor, os quais compartilham uma mensagem. O objeto Produtor é uma visão da interação de Prod com Part_Number. A especificação em Náutilus é como segue:

```

object Prod_Cons
aggregation of
  Produtor
  Consumidor
export
  Prod_Aleat
category
  birth Prod_Aleat
  death Fim
body
  act Prod_Aleat composed by
    Prod_Aleat of Produtor
  act Comunica composed by
    Envia of Produtor
    Recebe of Consumidor
  match
    Envia.Pmsg of Produtor
    Recebe.Cmsg of Consumidor
  act Fim composed by
    Fim of Produtor
    Fim of Consumidor
end Prod_Cons

object Produtor view of
  interaction
    Prod
    Part_Number
  end interaction
export
  Prod_Aleat Fim
  Envia der out Pmsg: natural
  by Send.Pmsg of Prod
category
  birth Prod_Aleat
  request Envia
  death request Fim
body
  act Prod_Aleat composed by
    int
      Produz of Prod
      Aleatório of Part_Number
    end int
  act Envia composed by
    Envia of Prod
  act Fim composed by
    int
      Fim of Prod
      Morte of Part_Number
    end int
end Produtor

```

```

object Prod
import
  Aleatório
  in Fator: natural
  out PN: natural
  of Part_Number
  Morte of Part_Number
export
  Produz Fim
  Envia out Pmsg: natural
category
  birth Produz
  death request Fim
body
  slot pr: 1..2
  slot num: natural
  act Produz
    call Aleatório of Part_Number
    arg Aleatório.Fator = 3
  cps
    val num << Aleatório.PN
    val pr << 1
  end cps
  act Envia
    enb pr = 1
    val pr << 2
    ret Envia.Pmsg = num
  act Fim
    enb pr = 2
    call Morte of Part_Number
end Prod

object Part_Number
export
  Aleatório
  in Fator: natural
  out PN: natural
  Morte
category
  birth request Aleatório
  death request Morte
body
  act Aleatório
    alt
      ret Aleatório.PN = Fator
    alt
      ret Aleatório.PN = 2 * fator
  act Morte
end Part_Number

```

```

object Consumidor
export
  Consome Fim
  Recebe in Cmsg: natural
category
  birth request Recebe
  death request Fim
body
  slot cs: 1..2
  slot inf: natural
    
```

```

act Recebe
  cps
    val inf << Recebe.Cmsg
    val cs << 1
  end cps
act Consome
  enb cs = 1
  val cs = 2
act Fim
  enb cs = 2
end Consumidor
    
```

Note-se que, como Produz chama Aleatório, os únicos valores possíveis do atributo num são 3 e 6. Conseqüentemente, os únicos valores das mensagens que necessitam ser considerados são somente os associados a estes valores, como ilustrado na Figura 11.4 (somente os autômatos origem estão representados e os identificadores estão abreviados).

A interação entre Produz e Part_Number é associada a uma sincronização onde a tabela de sincronizações é determinada como abaixo. Note-se que, na interação, a ação Envia ocorre independentemente, pois esta só é da categoria request quando da extração da visão de Produtor.

```

Produz | Aleatório
Envia
Fim | Morte
    
```

A extração da visão de Produtor é associada a uma reetiquetagem, induzida como segue:
 Produz | Aleatório → Prod_Aleat

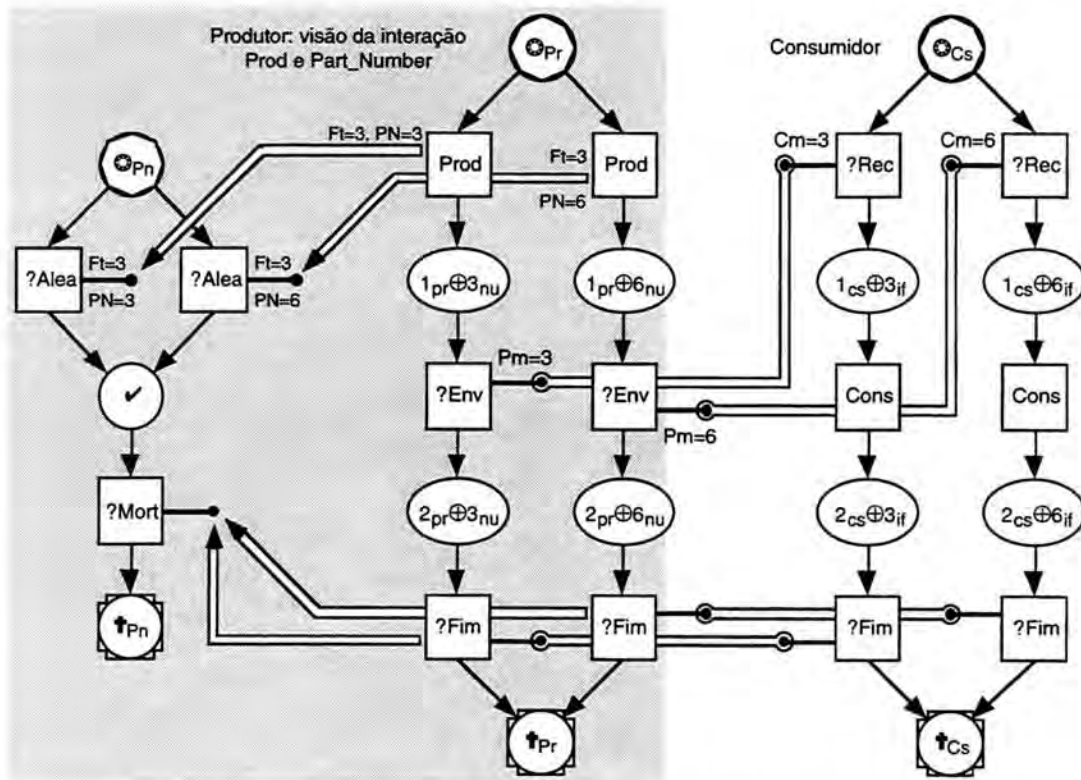


Figura 11.4 Relacionamento entre autômatos componentes de uma interação e de uma agregação

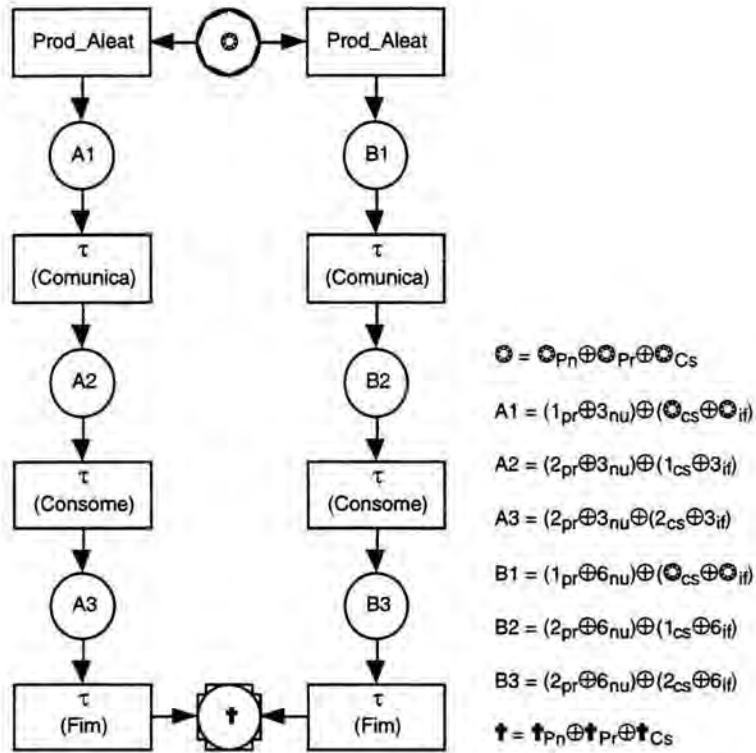


Figura 11.5 Autômato (origem) resultante de uma interação e de uma agregação

Envia \rightarrow Envia
 Fim | Morte \rightarrow Fim

A agregação entre Produtor e Consumidor é associada a uma sincronização onde a tabela de sincronizações é determinada como segue:

Prod_Aleat
 Envia | Recebe
 Consome
 Fim | Fim

Adicionalmente, a agregação é associada a uma reetiquetagem, induzida como segue:

Prod_Aleat \rightarrow Prod_Aleat
 Envia | Recebe \rightarrow τ
 Consome \rightarrow τ
 Fim | Fim \rightarrow τ

O autômato não-sequencial resultante da sincronização é ilustrado na Figura 11.5 onde as etiquetas das transições componentes que originaram as transições reetiquetadas estão entre parênteses. Novamente, somente o autômato origem é representado. □

11.5 Unidade

A semântica de uma unidade em Náutilus é a composição paralela das semânticas dos objetos terminais componentes. No texto que segue, é omitido que $i \in I$ para algum conjunto $I = \{1, \dots, n\}$.

Definição 11.12 Objeto Terminal. Um objeto terminal em Náutilus é um objeto que não é referenciado na especificação de reificações, interações, agregações ou visões de qualquer outro objeto. □

A semântica de um objeto terminal é a semântica da construção que resultou no objeto em questão, ou seja, dependendo do caso, é a semântica de um objeto simples, de uma reificação, de uma interação ou de uma agregação.

Definição 11.13 Semântica de uma Unidade. Seja $\{Ob_i\}$ o conjunto de objetos terminais de uma unidade em Náutilus e seja $\{Ob_i: N_{1i} \rightarrow nfc N_{2i}\}$ o conjunto das correspondentes semânticas. A semântica da unidade composta por $\{Ob_i\}$ é o produto categorial das semânticas $\{Ob_i\}$, ou seja, $\times Ob_i: \times N_{1i} \rightarrow nfc \times N_{2i}$. Entre os estados iniciais, é escolhido o referente ao valor $\otimes_1 \oplus \dots \oplus \otimes_n$ onde $\otimes_1, \dots, \otimes_n$ são os estados iniciais escolhidos para Ob_1, \dots, Ob_n , respectivamente. \square

12 Conclusão

Os principais objetivos propostos por esta dissertação, quais sejam, construir um domínio semântico que satisfaça à composicionalidade diagonal, enquadrar relativamente a outros domínios e aplicar à uma linguagem, são atingidos. As principais contribuições apresentadas são, resumidamente, as seguintes:

- 1) Um novo domínio semântico é introduzido, denominado autômatos não-seqüenciais, do tipo não-intercalação e baseado em [Meseguer & Montanari 90], para sistemas reativos, concorrentes e comunicantes e que satisfaz à composicionalidade diagonal, ou seja, onde as reificações são (verticalmente) posicionais refletindo uma metodologia de especificação hierárquica a qual permite adicionar estruturas a um sistema concorrente, em diferentes níveis de abstração e distribuem-se (horizontalmente) sobre a composição de sistemas, significando que o comportamento conjunto da reificação das partes é equivalente à reificação do sistema como um todo. Do que se tem conhecimento, é o primeiro modelo de concorrência a satisfazer a composicionalidade diagonal. Neste contexto, as operações de sincronização e de encapsulação de autômatos seqüenciais são functoriais e são definidas usando as técnicas de fibração e de cofibração, respectivamente.
- 2) Os autômatos não-seqüenciais são enquadrados em relação às redes de Petri e aos autômatos seqüenciais, estendendo a classificação formal proposta por [Sassone *et al* 93]. Dos passos que envolvem a passagem de um modelo para outro, pode-se inferir que os autômatos não-seqüenciais são mais concretos do que as redes de Petri e do que os autômatos seqüenciais. Comparativamente com as redes, os autômatos não-seqüenciais são capazes de expressar informações sobre alguns objetos e relacionamentos entre os objetos que são, de certa forma, perdidas quando da passagem para as redes. Entre os formalismos considerados, mostra-se que os autômatos não-seqüenciais constituem o modelo menos concreto que satisfaz à composicionalidade diagonal.
- 3) Uma linguagem baseada nos objetos, denominada Náutilus, é introduzida. Trata-se de uma revisão da linguagem GNOME, simplificando alguns aspectos como classes e herança e introduzindo outros, inspirados no domínio semântico proposto, como a reificação e a agregação. A semântica de um objeto em Náutilus é dada por um morfismo de reificação de autômatos não-seqüenciais onde, a cada ação do objeto corresponde uma computação. Assim, a semântica de uma reificação é uma composição (vertical) de reificações. A semântica de objetos resultantes de construções como interação, agregação ou encapsulação é a aplicação das correspondentes construções sobre autômatos não-seqüenciais reificados. Como a interação e a agregação são relacionadas com a sincronização de reificações de autômatos, a qual depende da composicionalidade horizontal para que seja bem definida, tem-se que as redes de Petri e os autômatos seqüenciais não são domínios semânticos adequados para a abordagem proposta.

Ao longo da dissertação, são apresentados diversos resultados adicionais aos objetivos propostos, com destaque para os seguintes:

- 4) São introduzidos os autômatos seqüenciais e as redes de Petri os quais também são equipados com as operações de sincronização e de encapsulação, bem como com reificações verticalmente posicionais, estendendo as abordagens em [Meseguer & Montanari 90] e [Winskel & Nielsen 95]. Note-se que, embora os morfismos de reificação sejam inspirados em [Meseguer &

Montanari 90], este não trata a composicionalidade horizontal e vertical bem como a sincronização e a encapsulação.

- 5) Os diversos domínios semânticos são construídos sobre estruturas baseadas em grafos e categorias. Inicialmente, são introduzidos os grafos (reflexivos) internos e as categorias internas onde os objetos e os morfismos podem pertencer a uma categoria base qualquer. A seguir, os grafos internos são generalizados como grafos estruturados onde os objetos e os morfismos podem pertencer a diferentes categorias, dados funtores destas categorias para uma categoria base. Por fim, são verificadas condições gerais para a bicompletude dos grafos (reflexivos) internos e estruturados, bem como para a completude das categorias internas, os quais, ao que se sabe, são originais. Destaque-se, ainda, a extensão das construções e dos resultados considerando nodos distinguidos. Neste contexto, tem-se que:

- autômatos seqüenciais são definidos sobre os grafos reflexivos;
- autômatos não-seqüenciais são definidos sobre os grafos reflexivos internos aos monóides comutativos;
- redes de Petri são definidos sobre os grafos estruturados onde os nodos são elementos de monóides comutativos e os arcos são elementos de conjuntos (apontados);
- computações não-seqüenciais são definidas sobre categorias internas aos monóides comutativos.

As categorias de autômatos e redes resultantes são bicompletas onde o produto categorial é interpretado como a composição paralela. Destaque-se que a cocompletude das categorias de redes de Petri marcadas com interpretações adequadas para os colimites, é um resultado expressivo quando comparado com as categorias propostas em [Winskel 87] e [Meseguer & Montanari 90], as quais possuem restrições para garantir a existência de coprodutos.

- 6) É introduzida uma construção categorial das tabelas de sincronizações para o tratamento de relacionamentos do tipo chamada, compartilhamento ou ambos os casos. A construção é baseada em um colimite, cujo diagrama possui uma forma espacial similar às arestas de um diamante e é definida para os autômatos seqüenciais, não-seqüenciais e as redes de Petri.

Em relação aos trabalhos futuros, destacam-se os seguintes:

- 1) Uma evolução natural desta dissertação, seria a reintrodução, na linguagem Náutilus, das simplificações adotadas em relação ao GNOME, incluindo classes, eventos, etc, e, neste contexto, desenvolver estudos adicionais no entorno dos autômatos não-seqüenciais.
- 2) Um trabalho que julga-se de grande interesse seria estudar o relacionamento dos autômatos não-seqüenciais com as lógicas, seguindo um caminho análogo ao proposto em [Fiadeiro & Costa 94]. Na seqüência deste trabalho, um dos principais objetivos seria construir lógicas de reificação.
- 3) Durante o desenvolvimento desta dissertação, foi explorado, durante um considerável período de tempo, uma forma alternativa de definir reificação, usando conceitos relacionados às gramáticas de grafos (com morfismos parciais), tendo sido atingido alguns resultados representativos como os apresentados em [Menezes 94]. Uma comparação entre as abordagens baseadas em gramáticas de grafos e a proposta nesta dissertação promete ser interessante e clarificante.

Referências

- [Abramsky 94] S. Abramsky, *Interaction Categories and Communication Sequential Processes*, A Classical Mind: Essays in Honor of C. A. R. Hoare (A. W. Roscoe, Ed.), pp. 1-16, Prentice Hall, 1994.
- [Adámek *et al* 90] J. Adámek, H. Herrlich, & G. Strecker, *Abstract and Concrete Categories*, Wiley, 1990.
- [Aiguier *et al* 95] Aiguier M. Bernot G. Ramos J. & Sernadas A., *An Algebraic Semantics for GNOME via a Translation to Étoile Specification*, submetido, Evry/França e IST/Portugal, 1995. Disponível via WWW em <http://www.cs.math.ist.utl.pt/cs/lcg/gnome.html>.
- [Arbib & Manes 75] M. A. Arbib & E. G. Manes, *Arrows, Structures and Functors - The Categorical Imperative*, Academic Press, 1975.
- [Asperti & Longo 91] A. Asperti & G. Longo, *Categories, Types and Structures - An Introduction to the Working Computer Science*, Foundations of Computing (M. Garey, A. Meyer, Eds.), MIT Press, 1991.
- [Barr & Wells 90] M. Barr & C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990.
- [Bednarczyk 88] M. A. Bednarczyk, *Categories of Asynchronous Systems*, tese de doutoramento, relatório técnico 1/88, University of Sussex, 1988.
- [Best & Fernández 88] E. Best & C. Fernández, *Nonsequential Processes - A Petri Net View*, EATCS Monographs on Theoretical Computer Science 13, Springer-Verlag, 1988.
- [Boudol & Castellani 88] G. Boudol & I. Castellani, *Concurrency and Atomicity*, Programming of Future Generation Computers II (K. Fuchi, L. Kott, Eds.), pp. 35-82, Elsevier Science Publishers B. V., North-Holland, 1988.
- [Brown *et al* 91] C. Brown, D. Gurr & V. de Paiva, *A Linear Specification Language for Petri Nets*, relatório técnico DAIMI PB-363, Aarhus University, 1991.
- [Caleiro 95] C. Caleiro, *Comunidades Distribuídas de Objetos*, tese de mestrado submetida, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1995.
- [Carmo & Penedo 94] P. Carmo & P. Penedo, *GNOME Compiler*, relatório técnico, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1994.
- [Casley 91] R. T. Casley, *On the Specification of Concurrent Systems*, tese de doutoramento, University of Stanford, 1991.
- [Corradini 90] A. Corradini, *An Algebraic Semantics for Transition Systems and Logic Programming*, tese de doutoramento, relatório técnico TD-8/90, Università di Pisa, 1990.
- [Costa 91] J. F. Costa, *Fundamentos Matemáticos da Concorrência*, tese de doutoramento, relatório técnico, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1991.
- [Costa & Sernadas 92] J. F. Costa & A. Sernadas, *Categorical Framework for Process Interconnection*, The INESC Journal of Research and Development 3(2), pp. 83-93, Lisboa, 1993.
- [Costa & Sernadas 95] J. F. Costa & A. Sernadas, *Progress Assumption in Concurrent Systems*, Formal Aspects of Computing 7(1), pp. 18-36, Springer-Verlag, 1995.

- [Costa *et al* 89] J. F. Costa, A. Sernadas & C. Sernadas, *OBLOG Users Manual*, relatório técnico INESC, Lisboa, 1989.
- [Costa *et al* 92] J. F. Costa, A. Sernadas, C. Sernadas & H. D. Ehrich, *Object Interaction*, Mathematical Foundations of Computer Science '92 (I. Havel, V. Koubek, Eds.), pp. 200-208, LNCS 629, Springer-Verlag, 1992.
- [Costa *et al* 93] J. F. Costa, A. Sernadas & C. Sernadas, *Data Encapsulation and Modularity: Tree Views of Inheritance*, Mathematical Foundation of Computer Science '93, (A. Borzyszkowski, S. Sokolowski, Eds.), pp. 382-391, LNCS 711, Springer-Verlag, 1993.
- [Costa *et al* 94] J. F. Costa, A. Sernadas & C. Sernadas, *Object Inheritance Beyond Subtyping*, Acta Informatica 31, pp. 5-26, Springer-Verlag, 1994.
- [Degano & Montanari 87] P. Degano & U. Montanari, *Concurrent Histories: A Basis for Observing Distributed Systems*, J. Comput. System Sci. 34, pp. 422-462, Nos. 2/3, 1987.
- [Degano *et al* 88] P. Degano, R. De Nicola & U. Montanari, *A Distributed Operational Semantics for CCS Based on Condition/Event Systems*, Acta Informatica 26, pp. 59-91, 1988.
- [Ehrich & Sernadas 90] H. D. Ehrich & A. Sernadas, *Algebraic Implementation of Objects over Objects*, Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness (J. W. de Bakker, W. -P. de Roever, G. Rozenberg, Eds.), pp. 239-266, Springer-Verlag, 1990.
- [Ehrich *et al* 91] H. D. Ehrich, J. Goguen & A. Sernadas, *A Categorical Theory of Objects as Observed Processes*, REX 90: Foundations of Object-Oriented Languages (J. W. de Bakker, W. -P. de Roever, G. Rozenberg, Eds.), pp. 203-228, LNCS 489, Springer-Verlag, 1991.
- [Fiadeiro & Costa 94] J. Fiadeiro & J. F. Costa, *Mirror, Mirror in My Hand... A Duality Between Specifications and Models of Process Behavior*, aguarda publicação em Mathematical Structures in Computer Science.
- [Floyd 67] R. W. Floyd, *Assigning Meaning to Programs*, Proceedings of Symposia in Applied Mathematics 19, pp. 19-32, 1967.
- [Glabbeek & Vaandrager 87] R. J. van Glabbeek & F. W. Vaandrager, *Petri Net Model for Algebraic Theories of Concurrency*, pp. 224-242, LNCS 259, Spring-Verlag, 1987.
- [Goguen 89] J. A. Goguen, *A Categorical Manifesto*, monografia técnica PRG-72, Programming Research Group, Oxford University Computing Laboratory, 1989.
- [Goguen 92] J. A. Goguen, *Sheaf Semantics for Concurrent Interacting Objects*, Mathematical Structures in Computer Science, pp. 159-191, Vol. 2, 1992.
- [Goldblatt 84] R. Goldblatt, *Topoi - The Categorical Analysis of Logic*, Studies of Logic and The Foundations of Mathematics 98, (J. Barwise *et al* Eds.), Elsevier Science Publishers, 1984.
- [Gorrieri 90] R. Gorrieri, *Refinement, Atomicity and Transactions for Process Description Language*, tese de doutoramento, Università di Pisa, 1990.
- [Hennessy 88] M. Hennessy, *Algebraic Theory of Processes*, MIT Press, 1988.
- [Hoare 69] C. A. R. Hoare, *An axiomatic Basis for Computer Programming*, Communication of the ACM, 12, pp. 576-580, 583, 1969.
- [Hoare 85] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice Hall, 1985.
- [Hopcroft & Ullman 79] J. E. Hopcroft & J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.
- [Johnstone 77] P. T. Johnstone, *Topos Theory*, Academic Press, 1977.
- [Jonsson 90] B. Jonsson, *A Hierarchy of Compositional Models of I/O-Automata*, Symposium on Mathematical Foundations of Computer Science, pp. 347-354, LNCS 452, Springer-Verlag, 1990.
- [Mac Lane 71] S. Mac Lane, *Categories for the Working Mathematician*, Springer-Verlag, 1971.
- [Mazurkiewicz 88] A. Mazurkiewicz, *Basic Notion of Trace Theory*, REX 88: Linear Time,

- Branching Time and Partial Orders in Logic and Models for Concurrency (J. W. de Bakker, W. -P. de Roever, G. Rozenberg, Eds.), pp. 285-363, LNCS 354, Springer-Verlag, 1988.
- [McCarthy 63] J. McCarthy, *A Basis for a Mathematical Theory of Computation*, Computer Programming and Formal Systems (P. Braffort, D. Hirschberg, Eds.), pp. 33-69, North-Holland, 1963.
- [Menezes 94] P. B. Menezes, *Compositional Reification of Petri Nets through Graph Transformation*, relatório técnico INESC/RT/74-94, preprint IST/DM/25-94, INESC/Instituto Superior Técnico, Lisboa, 1994. A ser submetido.
- [Menezes 95] P. B. Menezes, *Marked Petri Nets Within a Categorical Framework*, aguarda publicação na Revista de Informática Teórica e Aplicada, RITA, UFRGS, Brasil.
- [Menezes & Costa 93] P. B. Menezes & J. F. Costa, *Synchronization in Petri Nets*, aguarda publicação em Fundamenta Informaticae, Annales Societatis Mathematicae Polonae, IOS Press.
- [Menezes & Costa 95] P. B. Menezes & J. F. Costa, *Compositional Reification of Concurrent Systems*, Journal of the Brazilian Computer Society - Special Issue on Parallel Computation, Vol. 2, No. 1, pp. 50-67, 1995.
- [Menezes & Costa 95b] P. B. Menezes & J. F. Costa, *Systems for System Implementation*, em Proceedings of the 14th International Congress on Cybernetics, aguarda publicação no Journal of Cybernetics.
- [Menezes et al 95] P. B. Menezes, J. F. Costa & A. Sernadas, *Refinement Mapping for (Discrete Event) System Theory*, Proceedings of the Fifth International Conference on Computer Aided System Technology, EUROCAST 95, pp. 103-116, LNCS 1030, Springer-Verlag, 1996. Disponível como relatório técnico desde 1995.
- [Menezes et al 95b] P. B. Menezes, A. Sernadas & J. F. Costa, *Nonsequential Automata Semantics for a Concurrent Object-Based Language*, preprint IST/DM/21/95, 1995.
- [Meseguer 90] J. Meseguer, *Rewriting as a Unified Model of Concurrency*, SRI International SRI-CSL-90-02R, 1990.
- [Meseguer & Montanari 90] J. Meseguer & U. Montanari, *Petri Nets are Monoids*, Information and Computation 88, pp. 105-155, Academic Press, 1990.
- [Milner 80] R. Milner, *A Calculus for Communicating Systems*, LNCS 92, Springer-Verlag, 1980.
- [Milner 89] R. Milner, *Communication and Concurrency*, Prentice Hall, 1989.
- [Nielsen et al 94] M. Nielsen, V. Sassone & G. Winskel, *Relationship Between Models for Concurrency*, REX 93: A Decade of Concurrency - Reflections and Perspectives, (J. W. de Bakker, W. -P. de Roever, G. Rozenberg, Eds.), pp. 425-476, LNCS 803, Springer-Verlag, 1994.
- [Olderog 87] E. R. Olderog, *Operational Petri Net Semantics for CSP*, Advances in Petri Nets 1987 (G. Rozenberg, Ed.), pp. 196-223, Springer-Verlag, 1987.
- [Olderog 91] E. R. Olderog, *Nets Terms and Formulas*, Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press, 1991.
- [Plotkin 81] G. D. Plotkin, *A Structural Approach to Operational Semantics*, Lecture Notes, relatório técnico DAIMI FN-19, Computer Science Department, Aarhus University, 1981.
- [Ramos 95] J. Ramos, *Lógica de Certificação GNOME*, tese de mestrado, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1995.
- [Ramos & Sernadas 95] J. Ramos & A. Sernadas, *A Brief Introduction to GNOME*, relatório técnico, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1995. Disponível via WWW em <http://www.cs.math.ist.utl.pt/cs/lcg/gnome.html>.
- [Ratray 94] C. Ratray, *The Shape of Complex Systems*, EUROCAST 93: Computer Aided Systems Theory (F. Fichler, R. M. Díaz, Eds.), pp. 72-82, LNCS 763, Springer-Verlag, 1994.

- [Ravara 95] A. Ravara, *Semântica da Linguagem GNOME em TyCO*, tese de mestrado, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1995.
- [Reisig 85] W. Reisig, *Petri Nets: An Introduction*, EATCS Monographs on Theoretical Computer Science 4, Springer-Verlag, 1985.
- [Reisig 92] W. Reisig, *A Primer in Petri Net Design*, Springer-Verlag, 1992.
- [Resende 95] P. Resende, *Towards a Denotational Semantics of Imperative Objects*, preprint IST/DM/23-95, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1995. Disponível via WWW em <http://www.cs.math.ist.utl.pt/cs/lcg/publications.html>.
- [Rydeheard & Burstall 88] D. E. Rydeheard & R. M. Burstall, *Computational Category Theory*, Prentice Hall, 1988.
- [Sassone et al 93] V. Sassone, M. Nielsen & G. Winskel, *A Classification of Models for Concurrency*, CONCUR 93: Fourth International Conference of Concurrency (E. Best, Ed.), pp. 82-96, LNCS 715, Springer-Verlag, 1993.
- [Scott 70] *Outline of a Mathematical Theory of Computation*, Fourth Annual Princeton Conference on Information Science and Systems, pp. 169-176, Princeton University, 1970.
- [Sernadas & Ehrich 90] A. Sernadas & H. D. Ehrich, *What is an Object, After All*, Object-oriented Databases: Analysis, Design and Construction (R. Meersman, W. Kent, S. Khosla, Eds.), pp. 39-69, North-Holland, 1991.
- [Sernadas & Ramos 94] A. Sernadas & J. Ramos, *A Linguagem GNOME: Sintaxe, Semântica e Cálculo*, relatório técnico, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1994.
- [Sernadas & SernadasC 94] A. Sernadas & C. Sernadas, *Object Certification*, Fifth International Workshop on Deductive Approach to Information Systems (A. Olivé, Ed.), pp. 55-78, UP Catalunha, 1994. Versão completa submetido ao Knowledge and Data.
- [Sernadas et al 90] A. Sernadas, H. D. Ehrich & J. F. Costa, *From Processes to Objects*, The INESC Journal of Research and Development 1(1), pp. 7-27, 1990.
- [Sernadas et al 92] A. Sernadas, J. F. Costa & C. Sernadas, *Especificação de Objetos com Diagramas: Abordagem OBLOG*, relatório técnico, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1992. Prêmio Descartes 1992.
- [Sernadas et al 92b] A. Sernadas, C. Sernadas & J. F. Costa, *Object Specification Logic*, Journal of Logic and Computation, Vol. 5, No. 5, pp. 603-630, Oxford University Press, 1995. Disponível via WWW em <http://www.cs.math.ist.utl.pt/cs/lcg/gnome.html>. Disponível como relatório técnico desde 1992.
- [Sernadas et al 94] A. Sernadas, J. F. Costa & C. Sernadas, *An Institution of Object Behaviour*, Recent Trends in Data Type Specification (H. Ehrig, F. Orejas, Eds.), pp. 337-350. Springer-Verlag, 1994.
- [SernadasC et al 91] C. Sernadas, P. Resende, P. Gouveia & A. Sernadas, *In-the-Large Object-Oriented Design of Information Systems*, The Object-Oriented Approach in Information Systems (F. van Assche, B. Moulin, C. Rolland, Eds.), pp. 209-232, North-Holland, 1991.
- [SernadasC et al 92] C. Sernadas, P. Gouveia & A. Sernadas, *OBLOG: Object-Oriented, Logic-Based Conceptual Modeling*, relatório técnico, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 1992.
- [SernadasC et al 92b] C. Sernadas, P. Gouveia, J. Gouveia & P. Resende, *The Reification Dimension in Object-Oriented Database Design*, Specification of Data Base Systems (D. Harper, M. Norrie, Eds.), pp. 275-299, Springer-Verlag, 1992.
- [Stoy 77] J. E. Stoy, *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*, MIT Press, 1977.

- [Szabo 78] M. E. Szabo, *Algebra of Proofs*, Studies in Logic and the Foundations of Mathematics, Vol. 88, North-Holland, 1978.
- [Wegner 90] P. Wegner, *Concepts and Paradigms of Object-Oriented Programming*, OOPS Messenger, Vol. 1, No. 1, ACM Press, 1990.
- [Winskel 84] G. Winskel, *Categories of Models for Concurrency*, LNCS 197, Seminar on Concurrency (S. Brookes, Roscoe, G. Winskel, Eds.), pp. 246-267, Springer-Verlag, 1984.
- [Winskel 87] G. Winskel, *Petri Nets, Algebras, Morphisms and Compositionality*, Information and Computation 72, pp. 197-238, Academic Press, 1987.
- [Winskel 87b] G. Winskel, *Event Structures*, in Advances in Petri Nets (Brauer, Reissig & Rozenberg, Ed.), pp. 325-392, LNCS 255, Springer-Verlag, 1987.
- [Winskel 88] G. Winskel, *An Introduction to Event Structures*, REX 88: Linear Time, Branching Time and Partial Orders in Logic and Models for Concurrency (J. W. de Bakker, W. -P. de Roever, G. Rozenberg, Eds.), pp. 29-95, LNCS 354, Springer-Verlag, 1988.
- [Winskel & Nielsen 95] G. Winskel & M. Nielsen, *Models for Concurrency*, Handbook of Logic and Computer Science Vol. 4, Oxford University Press, 1995.

Apêndice: Semântica Formal

O texto que segue é baseado em [Stoy 77], excetuando-se onde destacado.

1 Sintaxe e Semântica

Historicamente, no estudo do entendimento das linguagens de programação, o problema sintático foi reconhecido antes do problema semântico e, com a aceitação generalizada da BNF (Backus-Naur Form - veja, por exemplo, [Hopcroft & Ullman 79, p. 78]), foi o primeiro a receber um tratamento adequado. Adicionalmente, os problemas sintáticos são de tratamento mais simples que os semânticos. Como consequência, foi dada uma grande ênfase à sintaxe, ao ponto de levar à idéia de que as questões das linguagens de programação resumiam-se às questões da sintaxe e da gramática. Atualmente, a teoria da sintaxe possui construções matemáticas bem definidas e universalmente reconhecidas como, por exemplo, as gramáticas de Chomsky.

Uma linguagem de programação (bem como qualquer modelo matemático) pode ser vista de duas formas:

- como uma entidade livre, ou seja, sem qualquer significado associado;
- como uma entidade juntamente com uma interpretação do seu significado.

A sintaxe trata das propriedades livres da linguagem como, por exemplo, a verificação gramatical de programas. A semântica objetiva dar uma interpretação para a linguagem como, por exemplo, um significado ou valor para um determinado programa. Conseqüentemente, a sintaxe basicamente manipula símbolos sem considerar os seus correspondentes significados. Note-se que, para resolver qualquer problema real, é necessário dar uma interpretação semântica aos símbolos como, por exemplo, "estes símbolos representam os inteiros".

Sintaticamente falando, não existe uma noção de programa "errado": neste caso, simplesmente não é um programa. Por outro lado, um programa sintaticamente válido ("correto"), pode não ser o programa que o programador esperava escrever. Assim, a questão de considerar um programa "correto" ou "errado" deve considerar se o mesmo modela adequadamente o comportamento desejado.

Nem sempre os limites entre a sintaxe e a semântica são claros. Um exemplo é a ocorrência de um nome em um programa o qual pode ser tratado de forma igualmente fácil como um problema sintático ou semântico. Entretanto, a distinção entre sintaxe e semântica em linguagens artificiais é, em geral, óbvia para a maioria dos problemas relevantes.

2 Semântica Formal e Linguagens de Programação

O cálculo e a álgebra de matrizes são exemplos típicos de notações formais para especificar operações especializadas. Tratam-se de notações desenvolvidas durante um período de tempo relativamente longo, sem pressões comerciais, onde as variações e modificações foram (e eventualmente são) gradativamente incorporadas. Conseqüentemente, são notações universalmente aceitas para as quais não existe a necessidade de dar uma semântica formal.

Em relação às linguagens de programação, uma grande quantidade e variedade de notações foram e são constantemente introduzidas em períodos de tempo relativamente curtos e, em geral, não são naturalmente aceitas. Assim, é comum que uma mesma notação tenha significados distintos em diferentes linguagens. Por exemplo, o significado da expressão $f(x)$ depende dos critérios de passagem de parâmetros de cada linguagem. Note-se que nem sempre uma notação reflete adequadamente a sua correspondente estrutura matemática podendo, inclusive, contradizer algumas noções intuitivas. Por exemplo, embora a expressão $x = x$ possua o valor intuitivo verdadeiro, alguma linguagens podem permitir expressões da forma $\text{random} = \text{random}$ cujo valor resultante não é, em geral, verdadeiro. Logo, quando diferentes convenções são adotadas, é necessário considerar um formalismo.

As notações matemática freqüentemente são tratadas com uma considerável informalidade, pois referem-se a uma comunicação homem-homem, onde a pessoa que recebe a informação é capaz de aceitar e interpretar adequadamente algumas inconsistências. Por outro lado, em uma comunicação homem-máquina, pelo menos até o momento, é necessário que os programas fornecidos satisfaçam a um conjunto bem definido de regras. Como a descrição precisa de regras não é trivial, é necessário que o formalismo seja desenvolvido cuidadosamente.

Três objetivos podem ser destacados para o uso de descrições formais:

- a) fornecer uma descrição suficientemente precisa da linguagem para que o implementador possa construir um compilador correto;
- b) fornecer uma descrição suficientemente precisa para permitir ao programador provar sentenças sobre o seu programa;
- c) sugerir meios ou formas para o projetista desenvolver linguagens melhores, mais elegantes ("limpas") e com descrições formais mais simples.

O primeiro objetivo pode ser considerado como a motivação inicial da descrição formal. Note-se que é comum uma linguagem sem uma descrição semântica formal ser definida em termos de uma implementação completa como, por exemplo, definida "por um determinado compilador". O segundo objetivo caracteriza-se como a principal ênfase atual onde é possível verificar, por exemplo, se uma implementação é correta (no sentido acima). Note-se que a idéia não é, necessariamente, associar uma prova a cada programa. O objetivo é construir as regras de forma suficientemente simples e universalmente aplicáveis para que sejam adequadamente seguidas mesmo em uma argumentação informal. O terceiro objetivo seria a principal tendência futura. A reificação na linguagem Náutilus é um exemplo desta tendência.

3 Métodos para Semântica Formal

Os métodos para descrever semanticamente as linguagens de programação podem ser classificados como segue:

- a) *Abordagem Operacional*. Define-se uma máquina abstrata, eventualmente denominada máquina de valoração, baseada em um estado, possivelmente com diversas componentes, em instruções primitivas e na especificação de como cada instrução modifica as componentes do estado. Uma máquina abstrata deve ser suficientemente simples para não permitir dúvidas sobre a execução de seu código. Uma descrição semântica de uma determinada linguagem de programação é a especificação de sua tradução em um código desta máquina. Assim, para determinar o efeito exato de um programa é suficiente analisar os passos do correspondente código.
- b) *Abordagem Denotacional*. Também é denominada abordagem matemática ou funcional. Definem-se funções, normalmente denominadas funções de valoração semântica, as quais associam construções sintáticas do programa a valores abstratos como números, valores lógicos, funções, etc. As funções de valoração são, em geral, definidas recursivamente: o valor

denotado por uma construção é especificado em termos dos valores *denotados* por suas subcomponentes sintáticas.

- c) *Abordagem Axiomática*. Associam-se regras às componentes da linguagem. As regras permitem afirmar o que será verdadeiro após a ocorrência de cada cláusula considerando o que era verdadeiro antes da ocorrência.

Originalmente, uma linguagem de alto nível era, basicamente, uma forma simplificada para escrever o código de máquina onde as computações eram analisadas ao nível do código de máquina. De certa forma, a abordagem operacional pode ser considerada como uma evolução natural desta noção de linguagem de alto nível. Seu desenvolvimento foi motivado, em grande parte, pelos sistemas reativos. Exemplos típicos são [Milner 89] e [Hoare 85]. Os seguintes aspectos podem ser destacados na abordagem operacional:

- podem existir diversas máquinas capazes de valorar ou até de definir uma determinada linguagem. Excetuando-se eventualmente por razões históricas, aparentemente não existe qualquer razão para considerar uma determinada máquina como canônica. Adicionalmente, a prova de que duas máquinas são equivalentes pode não ser trivial;
- uma descrição formal da semântica da máquina abstrata é necessária. Assim, de certa forma, a abordagem operacional transfere o problema da semântica para outro nível;
- uma definição operacional da semântica tende a indicar técnicas de implementação da linguagem, o que pode ser uma vantagem. Entretanto:
 - como destacado acima, a verificação de equivalência de implementações pode não ser trivial;
 - detalhes de implementação podem complicar ainda mais uma área que já não é simples por natureza;
- é desejável considerar a classe de todas as computações que podem ser realizadas por um determinado programa (ou classe de programas). Neste caso, a abordagem operacional aproxima-se da denotacional, no sentido em que considera-se a função induzida pelas computações do programa (ou classes de programas) e a correspondente implementação. Note-se que, seguindo este caminho, é necessário questionar se tal função existe e se é bem definida. Isto pode ser fácil para um determinado programa, mas a verificação da existência de uma função bem definida para todos os possíveis programas de uma linguagem é uma tarefa fundamental de qualquer teoria da semântica;
- não necessariamente existe uma correlação entre as subcomponentes sintáticas e seus operadores com o correspondente código da máquina abstrata e, portanto, não necessariamente existe uma correlação entre a forma do programador estruturar o seu programa e a correspondente semântica operacional.

Como afirmado em [Abramsky 94], a semântica denotacional é o mais influente e bem sedimentado paradigma para a semântica de computação sendo o que melhor se aproxima dos princípios da teoria matemática da computação no sentido de [McCarthy 63] ou [Scott 70]. Relativamente à abordagem denotacional, os seguintes aspectos destacam-se:

- em geral, a abordagem denotacional induz funções canônicas de valoração semântica dos programas da linguagem. Obviamente, para um determinado programa (ou classe de programas) podem existir mas de uma formulação da semântica denotacional, mas todas definem a mesma função. Portanto, para verificar se duas formulações são equivalentes, é suficiente verificar a identidade de duas funções;
- uma definição denotacional da semântica não necessariamente sugere técnicas para a implementação da linguagem. A verificação de que uma determinada implementação é fidedigna não é, em geral, uma tarefa simples;
- considerando-se que um valor denotado por uma construção é especificado em termos dos valores denotados por suas subcomponentes, tem-se que:
 - contrariamente à abordagem operacional, existe uma correlação entre a forma do programador estruturar o seu programa e a correspondente semântica denotacional;

- é fácil restringir a análise a uma determinada parte do programa a qual deseja-se examinar. Assim, o projeto de linguagens e o desenvolvimento de provas sobre programas tendem a ser facilitados;
- em [Abramsky 94], é destacado que, apesar da sua pretensa universalidade, a abordagem denotacional pende para a computação funcional, no sentido em que o comportamento de um programa é abstraído como a computação de uma função (incluindo, obviamente, a programação funcional). Entretanto, para paradigmas como sistemas distribuídos, em tempo real ou reativos, o sucesso da semântica denotacional é bem mais limitado. Como consequência, a sua capacidade de sugerir meios ou formas para o projetista de linguagens é parcialmente perdida.

A abordagem axiomática desenvolveu-se a partir de [Floyd 67]. A aplicação às linguagens de alto nível como em [Hoare 69] resultaram em importantes contribuições na prova da correção de programas. Em uma abordagem axiomática, tem-se que:

- as asserções são expressas em termos da linguagem de programação com pouca notação adicional como, por exemplo, quantificadores. Como consequência, é mais fácil para o próprio programador provar a correção de seus programas;
- nem sempre é claro em que condições os axiomas são aplicáveis como em situações de efeito colateral ("side effect");
- em geral, a semântica axiomática não define completamente uma linguagem bem como não é óbvio que existe uma linguagem que satisfaça aos axiomas. Por outro lado, o programador pode concentra-se nas propriedades que lhe interessam especialmente.

Apêndice: Teoria das Categorias

Este apêndice é baseado em [Barr & Wells 90], [Mac Lane 71], [Adámek *et al* 90] e [Asperti & Longo 91] e introduz os principais conceitos e resultados usados ao longo desta dissertação. O principal objetivo é permitir uma rápida referência bem como uma normalização das nomenclaturas usadas. Portanto, não são apresentadas introduções e explicações didáticas e, nas demonstrações das proposições, somente são indicadas as referências e as correspondentes páginas. No texto que segue, "se e somente se" é abreviado por "sse".

1 Construções Básicas

1.1 Categoria

Definição 1.1 Categoria. Uma categoria C é uma cinco-upla $C = (\text{Ob}_C, \text{Mor}_C, \partial_0, \partial_1, \iota, \circ)$ onde:

- Ob_C é uma coleção de objetos também denotada por C_0 ;
- Mor_C é uma coleção de morfismos, setas ou flechas, também denotada por C_1 ;
- $\partial_0, \partial_1: \text{Mor}_C \rightarrow \text{Ob}_C$ são operações denominadas domínio ou origem e codomínio, contradomínio ou destino, respectivamente. Um morfismo f tal que $\partial_0(f) = A$ e $\partial_1(f) = B$ é denotado por $f: A \rightarrow B$;
- $\iota: \text{Ob}_C \rightarrow \text{Mor}_C$ é uma operação denominada identidade tal que cada objeto A é associado a um morfismo $\iota_A: A \rightarrow A$;
- $\circ: (\text{Mor}_C)^2 \rightarrow \text{Mor}_C$ é uma operação parcial denominada composição tal que cada par de morfismos $\langle f: A \rightarrow B, g: B \rightarrow C \rangle$ é associado a um morfismo $g \circ f: A \rightarrow C$;

Adicionalmente, as operações identidade e composição devem satisfazer às seguintes propriedades:

- para todos o morfismo $f: A \rightarrow B$ em Mor_C , $f \circ \iota_A = \iota_B \circ f$;
- para todos os morfismos $f: A \rightarrow B, g: B \rightarrow C$ e $h: C \rightarrow D$ em Mor_C , $(h \circ g) \circ f = h \circ (g \circ f)$ □

É usual representar a operação de composição por ";". Neste caso, $g \circ f$ é denotado em ordem inversa, isto é, $f;g$. A coleção de todos os morfismos da categoria C com domínio em A e codomínio em B é denotada por $C[A, B]$ ou $\text{Hom}_C(A, B)$.

Definição 1.2 Categorias Pequena, Grande. Uma categoria C é pequena se Ob_C e Mor_C constituem conjuntos. Caso contrário, a categoria é grande. □

Definição 1.3 Subcategoria, Subcategorias Plena/Cheia, Ampla/Larga. Considere a categoria C .

- a) \mathcal{D} é uma subcategoria de C se
 - $\text{Ob}_{\mathcal{D}} \subseteq \text{Ob}_C$;
 - para todos os \mathcal{D} -objetos A e B , $\text{Hom}_{\mathcal{D}}(A, B) \subseteq \text{Hom}_C(A, B)$;
 - as operações origem, destino, identidade e composição em \mathcal{D} coincidem com as em C ;
- b) \mathcal{D} é uma subcategoria plena ou cheia de C se, para todos os \mathcal{D} -objetos A e B , $\text{Hom}_{\mathcal{D}}(A, B) = \text{Hom}_C(A, B)$;
- c) \mathcal{D} é uma subcategoria ampla ou larga de C se $\text{Ob}_{\mathcal{D}} = \text{Ob}_C$. □

Note-se que uma subcategoria plena pode ser completamente determinada pelos seus objetos.

1.2 Objetos Inicial, Terminal e Zero

Definição 1.4 **Objetos Inicial, Terminal, Zero.** Considere uma categoria C . Então:

Sejam 0 e 1 C -objetos.

- um C -objeto é inicial em C , denotado por 0_C , sse para todo o C -objeto A , existe um único morfismo em $C[0, A]$, denotado por $!_0: 0_C \rightarrow A$;
- um C -objeto é terminal em C , denotado por 1_C , sse para todo o C -objeto A existe um único morfismo em $C[A, 1]$ denotado por $!_1: A \rightarrow 1_C$;
- um C -objeto é zero em C sse for simultaneamente inicial e terminal em C . □

Proposição 1.5 Seja C uma categoria. Então:

- se 0_C e $0_{C'}$ são objetos iniciais, então são isomorfos;
- se 1_C e $1_{C'}$ são objetos terminais, então são isomorfos.

Prova: Veja [Barr & Wells 90, p. 38]. □

1.3 Morfismos Especiais

Definição 1.6 **Morfismos Paralelos.** Dois morfismos de uma determinada categoria são ditos paralelos sse possuem os mesmos domínio e codomínio. □

Definição 1.7 **Retração, Co-Retração, Inverso.** Considere uma categoria C . Sejam $f: A \rightarrow B$ e $g: B \rightarrow A$ C -morfismos tais que $g \circ f = 1_A$. Então:

- g é uma retração ou um morfismo inverso à esquerda de f ;
- f é uma co-retração, secção ou um morfismo inverso à direita de g .
- g é inverso de f sse g é simultaneamente uma retração e co-retração de f . □

Se um morfismo possui inverso, então este é único ([Mac Lane 71, p. 19]). A retração e o inverso de um morfismo f são denotados por f^R e f^L , respectivamente.

Definição 1.8 **Monomorfismo, Epimorfismo, Isomorfismo.** Considere uma categoria C . Um C -morfismo $h: A \rightarrow B$ é um:

- monomorfismo ou mono, também denotado por $h: A \rightarrow B$, sse para todos os C -morfismos $f, g: X \rightarrow A$, se $h \circ f = h \circ g$, então $f = g$;
- epimorfismo ou epi, também denotado por $h: A \rightarrow B$, sse para todos os C -morfismos $f, g: B \rightarrow X$, se $f \circ h = g \circ h$, então $f = g$;
- isomorfismo ou iso sse existe um C -morfismo $f: B \rightarrow A$ inverso de h , isto é, $f \circ h = 1_A$ (inverso à esquerda) e $h \circ f = 1_B$ (inverso à direita). □

Note-se que:

- toda retração é um epimorfismo ([Mac Lane 71, p. 19]);
- toda co-retração é um monomorfismo ([Mac Lane 71, p. 19]);
- em Set , todo monomorfismo é injetivo e todo epimorfismo é sobrejetivo;
- todo isomorfismo é mono e epi. Entretanto, o inverso não é necessariamente verdadeiro, o que contrasta com a intuição da teoria dos conjuntos.

Definição 1.9 **Relação.** Considere a categoria C . Sejam A, B e D C -objetos. Uma relação em A e B é um monomorfismo $rel: D \rightarrow A \times B$. □

1.4 Fatoração de Morfismo e Subobjeto

Definição 1.10 **Fatoração de morfismos.** Considere a categoria C . Sejam $f: X \rightarrow A$ e $g: Y \rightarrow A$ C -morfismos. Então, g fatora f (ou f é fatorado por g) sse existe um C -morfismo $h: X \rightarrow Y$ tal que o diagrama ilustrado na Figura 1.1 comuta. □

Se f e g são monomorfismos e g fatora f , então h é mono e único.

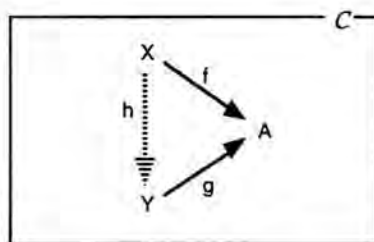


Figura 1.1 Diagrama comutativo para fatoração de morfismos

Definição 1.11 Subobjeto. Considere a categoria C . Sejam A um C -objeto e \sim uma relação de equivalência de monomorfismos tal que $f \sim g$ se e somente se f e g se fatoram mutuamente. Um subobjeto de A é uma classe de equivalência em \sim . \square

Intuitivamente, um subobjeto é determinado pela imagem de qualquer monomorfismo da classe de equivalência. Normalmente, um subobjeto é representado por um elemento representativo da classe em um diagrama da forma $A_0 \rightarrow A$. Neste caso, é normal se referir a A_0 como subobjeto de A .

2 Funtores

Definição 2.1 Funtor. Considere as categorias C e D . Um funtor (covariante) $f: C \rightarrow D$ é um par de operações $f = \langle f_O: \text{Ob}_C \rightarrow \text{Ob}_D, f_M: \text{Mor}_C \rightarrow \text{Mor}_D \rangle$ tal que:

- para todo o C -morfismo $f: A \rightarrow B$, $f_M(f): f_O(A) \rightarrow f_O(B)$ é um D -morfismo;
- para todo o C -objeto A , $f_M(1_A) = 1_{f_O(A)}$;
- para todos os C -morfismos $f: A \rightarrow B$ e $g: B \rightarrow C$, $f_M(g \circ f) = f_M(g) \circ f_M(f)$. \square

É usual omitir os parênteses e os subscritos de um funtor. Por exemplo, $f_M(g)$ é abreviado por fg . O funtor identidade em C é denotado por id_C .

Definição 2.2 Funtores Fidedigno/Fiel, Pleno/Cheio. Um funtor $f: C \rightarrow D$ é:

- a) fidedigno ou fiel sse para todos os C -morfismos paralelos $f, g: A \rightarrow B$ se $ff = fg$, então $f = g$;
- b) pleno ou cheio sse para todos os C -objetos A, B e para todo o D -morfismo $g: fA \rightarrow fB$ existe um C -morfismo $f: A \rightarrow B$ tal que $fg = g$. \square

Definição 2.3 Funtor Inclusão. Um funtor $inc: D \rightarrow C$ é inclusão se D é subcategoria de C e é tal que cada objeto e cada morfismo em D são mapeados em suas correspondentes identidades em C . \square

Note-se que todo funtor inclusão é fidedigno.

Definição 2.4 Isomorfismo Entre Categorias. As categorias C e D são isomorfas sse existem funtores $f: C \rightarrow D, g: D \rightarrow C$ tais que $g \circ f = id_C$ e $f \circ g = id_D$. \square

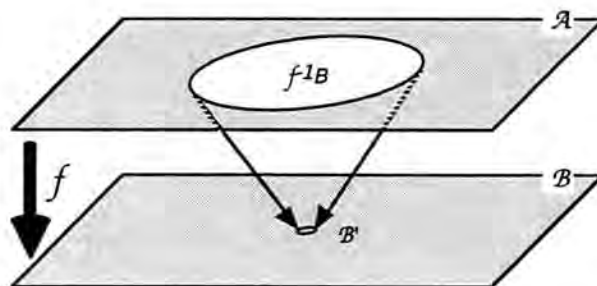


Figura 2.1 Fibra

Definição 2.5 Fibra. Considere o funtor $f: \mathcal{A} \rightarrow \mathcal{B}$. Sejam B um \mathcal{B} -objeto, 1_B o seu correspondente \mathcal{B} -morfismo identidade e \mathcal{B}' a subcategoria de \mathcal{B} constituída exclusivamente por B e 1_B . A fibra definida por f e B , denotada por $f^I B$, é a maior subcategoria de \mathcal{A} tal que $f(f^I B)$ é a subcategoria \mathcal{B}' , como ilustrado na Figura 2.1. \square

3 Transformações Naturais

Considere os funtores $f, g: \mathcal{C} \rightarrow \mathcal{D}$. Uma transformação natural de A em $\text{Ob}_{\mathcal{C}}$ pode ser vista como uma "translação" de fA para gA .

3.1 Transformação Natural

Definição 3.1 Transformação Natural. Considere os funtores $f, g: \mathcal{C} \rightarrow \mathcal{D}$. $\alpha: f \rightarrow g$ é uma transformação natural sse para todo o \mathcal{C} -objeto A , α_A é um morfismo em $\mathcal{D}[fA, gA]$ e para todo \mathcal{C} -morfismo $f: A \rightarrow B$ o diagrama ilustrado na Figura 3.1 comuta. \square

É usual denotar a transformação natural identidade em f por $f: f \rightarrow f$.

Definição 3.2 Isomorfismo Natural. Considere os funtores $f, g: \mathcal{C} \rightarrow \mathcal{D}$. A transformação natural $\alpha: f \rightarrow g$ é um isomorfismo natural se, para todo o \mathcal{C} -objeto A , α_A em $\mathcal{D}[fA, gA]$ é um isomorfismo. \square

Definição 3.3 Equivalência Entre Categorias. As categorias \mathcal{C} e \mathcal{D} são equivalentes sse existem funtores $f: \mathcal{C} \rightarrow \mathcal{D}$, $g: \mathcal{D} \rightarrow \mathcal{C}$ tais que $g \circ f$ e $f \circ g$ são naturalmente isomorfos a $id_{\mathcal{C}}$ e $id_{\mathcal{D}}$, respectivamente. \square

3.2 Composição de Transformações Naturais

Definição 3.4 Composição Vertical. Considere os funtores $f, g, h: \mathcal{C} \rightarrow \mathcal{D}$. Sejam $\alpha: f \rightarrow g$ e $\beta: g \rightarrow h$ transformações naturais. A composição vertical de transformações naturais é determinada pela composição de morfismos em \mathcal{D} , ou seja, para todo o \mathcal{C} -objeto A , $(\beta \circ \alpha)_A = \beta_A \circ \alpha_A$, como ilustrado na Figura 3.2. \square

Os funtores de \mathcal{C} para \mathcal{D} como objetos e as correspondentes transformações naturais constituem a categoria de funtores de \mathcal{C} para \mathcal{D} , denotado por $\text{Fun}(\mathcal{C}, \mathcal{D})$ ou $\mathcal{C}^{\mathcal{D}}$ onde a operação de composição é a composição vertical de transformações naturais e a transformação identidade é definida de maneira óbvia.

Definição 3.5 Composição Horizontal. Considere os funtores $f, g: \mathcal{B} \rightarrow \mathcal{C}$ e $h, k: \mathcal{C} \rightarrow \mathcal{D}$. Sejam $\alpha: f \rightarrow g$ e $\beta: h \rightarrow k$ transformações naturais. A composição horizontal $\beta \alpha: h \circ f \rightarrow k \circ g$ é tal que para todo o \mathcal{B} -objeto B , o diagrama ilustrado na Figura 3.3 comuta em \mathcal{D} . \square

Notação 3.6 Composição Vertical \times Horizontal. Em relação à composição de transformações naturais, é normal usar notação $\beta \circ \alpha$ para a composição vertical e $\beta \alpha$ para a composição horizontal. \square

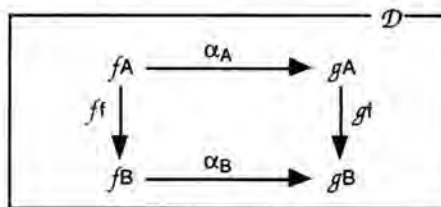


Figura 3.1 Diagrama comutativo para transformações naturais

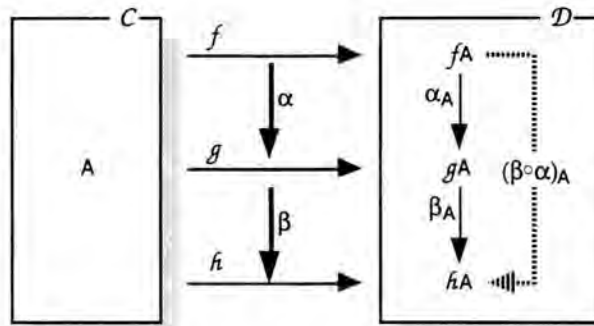


Figura 3.2 Composição vertical de transformações naturais

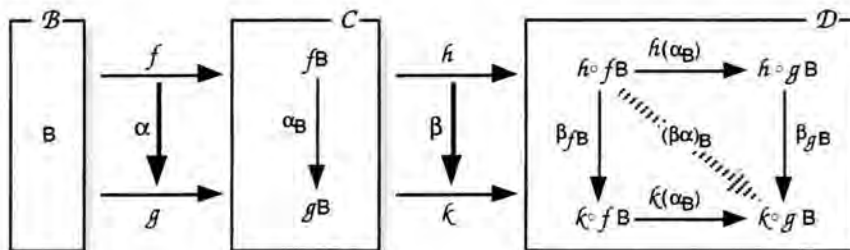


Figura 3.3 Composição horizontal de transformações naturais

4 Produtos e Coprodutos

4.1 Produto

Definição 4.1 Produto. Considere a categoria C . Sejam A e B C -objetos. Um produto (categorial) de A e B é um C -objeto denotado por $A \times B$ juntamente com dois C -morfismos $\pi_1: A \times B \rightarrow A$ e $\pi_2: A \times B \rightarrow B$ tais que para todo o C -objeto C e para todos os C -morfismos $f: C \rightarrow A$ e $g: C \rightarrow B$ existe um único C -morfismo $h: C \rightarrow A \times B$ que comuta o diagrama ilustrado na Figura 4.1. \square

Na definição acima, o objeto C juntamente com os morfismos f, g é denominado pré-produto dos objetos A, B .

Definição 4.2 Produtos Finito, Arbitrário. Considere a categoria C , uma coleção finita (arbitrária) I

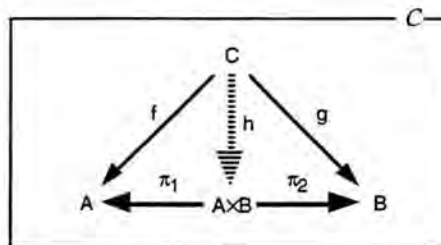


Figura 4.1 Diagrama comutativo para produtos binários

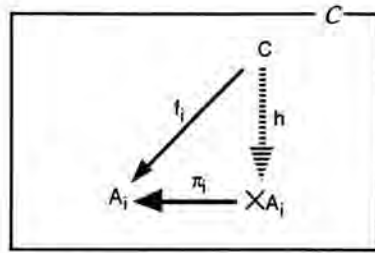


Figura 4.2 Diagrama comutativo para produtos finitos/arbitrários

e uma coleção indexada de objetos $\{A_i\}_{i \in I}$ (não necessariamente distintos). Um produto de $\{A_i\}_{i \in I}$ é um C -objeto denotado por $\prod_{i \in I} A_i$ ou $\prod_{i \in I} A_i$ (ou simplesmente $\prod A_i$ ou $\prod A_i$) juntamente com a coleção indexada de C -morfismos $\{\pi_i: \prod A_i \rightarrow A_i\}_{i \in I}$ tais que para todo C -objeto C e para toda a coleção indexada de C -morfismos $\{f_i: C \rightarrow A_i\}_{i \in I}$ existe um único morfismo $h: C \rightarrow \prod A_i$ que comuta o diagrama ilustrado na Figura 4.2. □

Proposição 4.3 Se um produto existe, então é único, a menos de isomorfismo.

Prova: Veja [Barr & Wells 90, p. 118]. □

Definição 4.4 Categoria Cartesiana. A categoria C é cartesiana sse possui todos os produtos finitos. □

Proposição 4.5 A categoria C é cartesiana sse possui objeto terminal e todos os produtos binários.

Prova: Veja [Barr & Wells 90, p. 131]. □

4.2 Coproduto

Coproduto é a noção dual de produto.

Definição 4.6 Coproduto. Considere a categoria C . Sejam A, B C -objetos. Um coproduto ou soma de A e B é um C -objeto denotado por $A+B$ juntamente com os C -morfismos $q_1: A \rightarrow A+B$ e $q_2: B \rightarrow A+B$ tais que para todo o C -objeto C e para todos os C -morfismos $f: A \rightarrow C$ e $g: B \rightarrow C$ existe somente um C -morfismo $h: A+B \rightarrow C$ que comuta o diagrama ilustrado na Figura 4.3. □

Na definição acima, o objeto C juntamente com os morfismos f, g é denominado pré-coproduto dos objetos A, B .

Definição 4.7 Coprodutos Finito, Arbitrário. Considere a categoria C . Sejam I uma coleção finita (arbitrária) e $\{A_i\}_{i \in I}$ uma coleção indexada de C -objetos (não necessariamente distintos). Um coproduto de $\{A_i\}_{i \in I}$ é um C -objeto denotado por $\coprod_{i \in I} A_i$ ou $\coprod_{i \in I} A_i$ (ou simplesmente $\coprod A_i$ ou $\coprod A_i$) juntamente com a coleção indexada de C -morfismos $\{q_i: A_i \rightarrow \coprod A_i\}_{i \in I}$ tais que para todo C -objeto C e para toda a coleção indexada de C -morfismos $\{f_i: A_i \rightarrow C\}_{i \in I}$ existe um único C -morfismo $h: \coprod A_i \rightarrow C$ que comuta o diagrama ilustrado na Figura 4.4. □

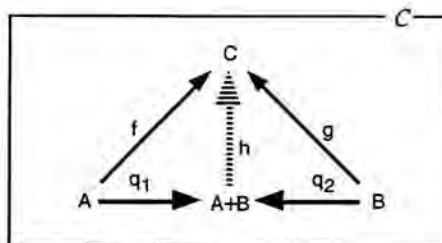


Figura 4.3 Diagrama comutativo para coprodutos binários

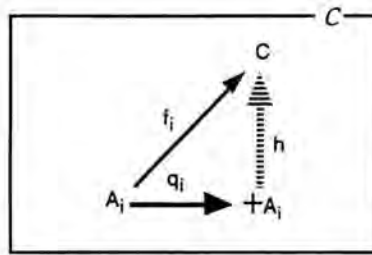


Figura 4.4 Diagrama comutativo para coprodutos finitos/arbitrários

Proposição 4.8 Se um coproduto existe, então é único, a menos de isomorfismo. Prova: Veja [Asperti & Longo 91, p. 18]. □

5 Limites e Colimites

5.1 Igualador e Co-Igualador

O igualador e o co-igualador dos morfismos paralelos $f, g: A \rightarrow B$ podem ser interpretados como segue:

- igualador: o "subobjeto maximal" de A para o qual f e g são iguais;
- co-igualador: o "menor objeto quociente" de B que força a igualdade de f e g .

Definição 5.1 Igualador. Considere a categoria C . Sejam $f, g: A \rightarrow B$ um par de C -morfismos paralelos. Um igualador de f e g é um C -objeto E juntamente com um C -morfismo $e: E \rightarrow A$ tais que $f \circ e = g \circ e$ e, para todo o C -morfismo $h: D \rightarrow A$ tal que $f \circ h = g \circ h$, existe um único C -morfismo $k: D \rightarrow E$ tal que $e \circ k = h$, como ilustrado na Figura 5.1. □

Na definição acima, o morfismo h é denominado pré-igualador dos morfismos f, g . Co-igualador é a noção dual de igualador.

Definição 5.2 Co-Igualador. Considere a categoria C . Sejam $f, g: A \rightarrow B$ um par de C -morfismos

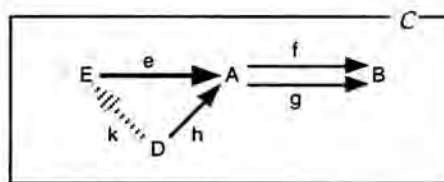


Figura 5.1 Diagrama comutativo para igualador

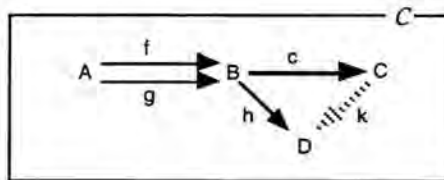


Figura 5.2 Diagrama comutativo para co-igualador

paralelos. Um igualador de f e g é um C -objeto C e um C -morfismo $c: B \rightarrow C$ tais que $c \circ f = c \circ g$ e para todo o C -morfismo $h: B \rightarrow D$ tal que $h \circ f = h \circ g$ existe um único C -morfismo $k: C \rightarrow D$ tal que $k \circ c = h$, ilustrado, como ilustrado na Figura 5.2. \square

Na definição acima, o morfismo h é denominado pré-co-igualador dos morfismos f e g .

Proposição 5.3

- a) Todo igualador é um monomorfismo.
- b) Todo co-igualador é um epimorfismo.

Prova: Veja [Asperti & Longo 91, p. 31] para a) e [Barr & Wells 90, p. 213] para b) \square

5.2 Produto Fibrado

Definição 5.4 Produto Fibrado. Considere a categoria C . Sejam $f: A \rightarrow C$ e $g: B \rightarrow C$ um par de C -morfismos com o mesmo destino. Um produto fibrado de f e g é um objeto $A \times_C B$ juntamente com um par de C -morfismos $p: A \times_C B \rightarrow A$ e $q: A \times_C B \rightarrow B$ tais que $f \circ p = g \circ q$ e, para todos os C -morfismos $u: P \rightarrow A$ e $v: P \rightarrow B$ tais que $f \circ u = g \circ v$, existe um único C -morfismo $k: P \rightarrow A \times_C B$ tal que $p \circ k = u$ e $q \circ k = v$, como ilustrado na Figura 5.3. \square

Proposição 5.5 Considere a categoria C e o diagrama comutativo ilustrado na Figura 5.4:

- a) se os diagramas ① e ② são produtos fibrados, então o diagrama externo também é um produto fibrado;
- b) se os diagramas externo e ② são produtos fibrados, então o diagrama ① também é um produto fibrado.

Prova: Veja [Asperti & Longo 91, p. 33]. \square

Proposição 5.6 Considere a categoria C , os C -morfismos $f: A \rightarrow C$, $g: B \rightarrow C$ e o produto fibrado ilustrado na Figura 5.5:

- a) se g é mono, então p também é mono;
- b) em *Set*, se g é epi, então p também é epi.

Prova: Veja [Asperti & Longo 91, p. 33] para a) e [Barr & Wells 90, p. 211] para b). \square

Proposição 5.7 Considere a categoria C e o diagrama comutativo ilustrado na Figura 5.6:

- a) se o diagrama externo é um produto fibrado, então o diagrama ① também é um produto fibrado;
- b) se o diagrama ① é um produto fibrado e h é um monomorfismo, então o diagrama externo também é um produto fibrado.

Prova: Veja [Adámek *et al* 90, p. 182]. \square

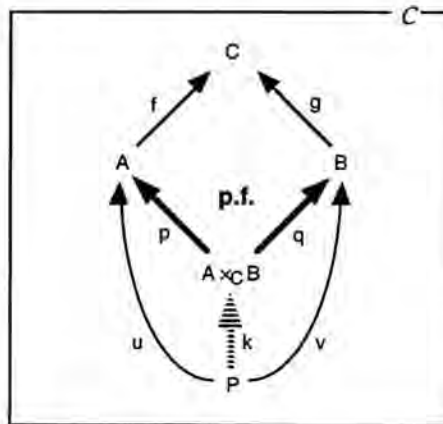


Figura 5.3 Diagrama comutativo para produto fibrado

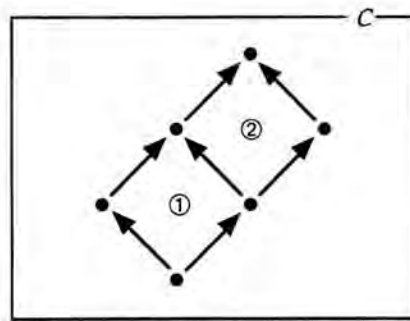


Figura 5.4 Diagrama comutativo

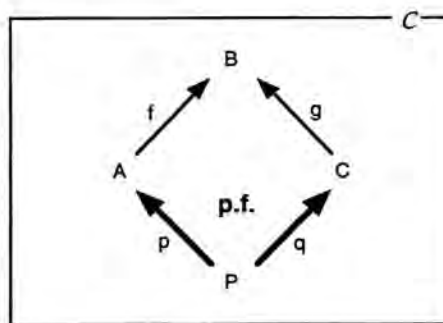


Figura 5.5 Produto fibrado

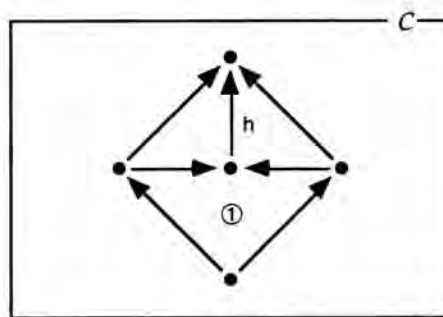


Figura 5.6 Diagrama comutativo

5.3 Soma Amalgamada

Soma amalgamada é a noção dual de produto fibrado. É uma forma de somar dois objetos identificando "partes comuns".

Definição 5.8 Soma Amalgamada. Considere a categoria C . Sejam $f: A \rightarrow B$ e $g: A \rightarrow C$ um par de C -morfismos com a mesma origem. Uma soma amalgamada de f e g é um objeto $B+_A C$ juntamente com um par de C -morfismos $p: B \rightarrow B+_A C$ e $q: C \rightarrow B+_A C$ tais que $p \circ f = q \circ g$ e, para todos os C -morfismos $u: B \rightarrow S$ e $v: C \rightarrow S$ tais que $u \circ f = v \circ g$, existe um único C -morfismo $k: B+_A C \rightarrow S$ tal que $k \circ p = u$ e $k \circ q = v$, como ilustrado na Figura 5.7. \square

Proposição 5.9 Considere a categoria C e o diagrama comutativo ilustrado na Figura 5.8. Se os diagramas ① e ② são somas amalgamadas, então o diagrama externo também é uma soma amalgamada.

Prova: Veja [Rydeheard & Burstall 88, p. 77]. □

Proposição 5.10 Considere a categoria Set e o diagrama comutativo ilustrado na Figura 5.9. Então o diagrama é simultaneamente soma amalgamada e produto fibrado.

Prova: Veja [Barr & Wells 90, p. 215]. □

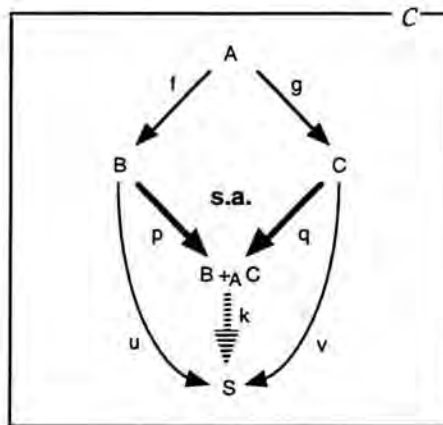


Figura 5.7 Diagrama comutativo para soma amalgamada

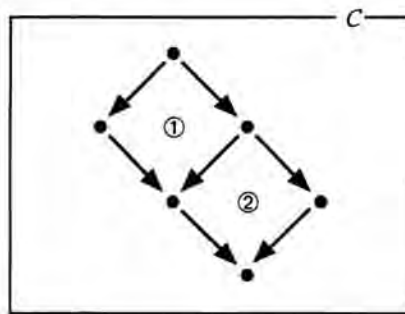


Figura 5.8 Diagrama comutativo

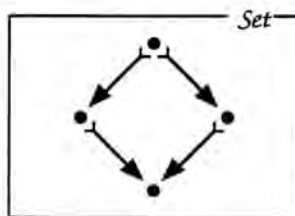


Figura 5.9 Diagrama comutativo

5.4 Limites e Colimites Finitos e Arbitrários

Um diagrama de uma categoria C é, basicamente, um grafo etiquetado nos nodos e arcos por C -objetos e C -morfismos, respectivamente, respeitando as operações de origem e de destino de C . Portanto, um diagrama pode ser visto como um morfismo de grafos, onde o objeto destino é uma categoria esquecendo as operações de identidade e composição.

Definição 5.11 Diagramas Finito, Arbitrário. Considere as categorias CAT e GRA (dos grafos largos e diretos). Sejam I um GRA -objeto denominado índice e $u: CAT \rightarrow GRA$ o funtor esquecimento que esquece sobre as operações de identidade e de composição. Então, um diagrama D de um CAT -objeto C com a forma I é um GRA -morfismo $D: I \rightarrow uC$. Se os nodos e os arcos de I são conjuntos finitos, então o diagrama D é finito. \square

Em uma representação textual de um diagrama $D: I \rightarrow uC$, um arco (nodo) é denotado por c_i ou simplesmente por c , quando o contexto não deixar dúvida de qual arco (nodo) é referido. Analogamente, a coleção dos nodos (arcos) do diagrama são denotados por $\{c_i\}_{i \in I}$. Em uma representação gráfica de um diagrama, é normal explicitar a forma de I mas com as etiquetas em C , de acordo com o morfismo D . Da definição acima, pode-se inferir que:

- em um diagrama, mais de um nodo ou arco podem possuir uma mesma etiqueta;
- diagramas com diferentes formas podem possuir as mesmas etiquetas nos nodos ou arcos.

Definição 5.12 Cone, Cocone. Considere a categoria C . Seja $D: I \rightarrow uC$ um diagrama de C . Então:

- um cone em D é um C -objeto P , juntamente com uma família de C -morfismo $\{p_i: P \rightarrow D_i\}_{i \in I}$ tal que, para todos os D -nodos D_u, D_v e, para todo o D -arco $d: D_u \rightarrow D_v$, $d \circ p_u = p_v$, como ilustrado na Figura 5.10;
- um cocone é definido de forma dual. \square

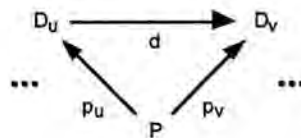


Figura 5.10 Cone

Dado uma categoria C e um diagrama D , os cones em D formam uma categoria onde um morfismo $h: \langle P, \{p_i: P \rightarrow D_i\}_{i \in I} \rangle \rightarrow \langle Q, \{q_i: Q \rightarrow D_i\}_{i \in I} \rangle$ é um C -morfismo $h: P \rightarrow Q$ tal que, para todo $i \in I$, o diagrama ilustrado na Figura 5.11 comuta.

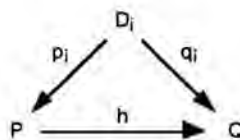


Figura 5.11 Diagrama comutativo para morfismos de cones

Definição 5.13 Limite. Considere a categoria C . Seja $D: I \rightarrow uC$ um diagrama de C . Então:

- um limite de D é um cone $\langle P, \{p_i: P \rightarrow D_i\}_{i \in I} \rangle$ tal que, para qualquer outro cone $\langle Q, \{q_i: Q \rightarrow D_i\}_{i \in I} \rangle$, existe um único morfismo $h: Q \rightarrow P$ como ilustrado na Figura 5.12;
- um colimite é definido de forma dual. \square

Definição 5.14 Categorias Completa, Cocompleta, Bicompleta. Considere a categoria C

- se todo o diagrama em C possui um limite então C é completa;
- se todo o diagrama em C possui um colimite então C é cocompleta;
- C é bicompleta sse é completa e cocompleta. \square

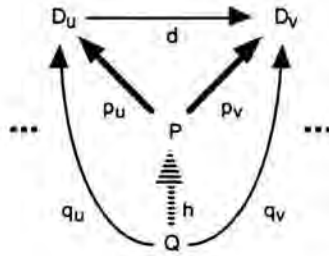


Figura 5.12 Limite

Proposição 5.15 Se a categoria C possui objeto terminal (inicial), produto (coproduto) para todo par de objetos e igualador (co-igualador) para todo par de morfismos paralelos, então C possui todos os limites (colimites) finitos.

Prova: Veja [Barr & Wells 90, p.206]. □

Proposição 5.16 Se a categoria C possui objeto terminal (inicial) e todos os produtos fibrados (somadas amalgamadas), então C possui todos os limites (colimites finitos).

Prova: Veja [Barr & Wells 90, p.206]. □

Proposição 5.17 Se a categoria C possui produtos (coprodutos) arbitrários e igualador (co-igualador) para todo o par de morfismos paralelos, então C possui todos limites (colimites).

Prova: Veja [Asperti & Longo 91, p. 131]. □

6 Adjunções

Uma adjunção pode ser vista como uma medida da expressividade de uma categoria em relação à outra.

6.1 Adjunção

Definição 6.1 Adjunção. Considere os funtores $\ell: \mathcal{A} \rightarrow \mathcal{B}$ e $r: \mathcal{B} \rightarrow \mathcal{A}$ e a Figura 6.1. O functor ℓ é adjunto esquerdo de r ou r é adjunto direito de ℓ se existe uma transformação natural $\eta: id_{\mathcal{A}} \rightarrow r \circ \ell$ denominada unidade tal que para todo morfismo $f: A \rightarrow rB$ em \mathcal{A} , existe um único morfismo $g: \ell A \rightarrow B$ tal que $f = rg \circ \eta_A$. Ou, de forma dual, se existe uma transformação natural $\varepsilon: \ell \circ r \rightarrow id_{\mathcal{B}}$ denominada co-unidade tal que para todo morfismo $v: \ell X \rightarrow Y$ em \mathcal{A} , existe um único morfismo $u: X \rightarrow rY$ tal que $v = \varepsilon_Y \circ \ell u$. □

Normalmente, uma adjunção é denotada por $\langle \ell, r, \eta, \varepsilon \rangle: \mathcal{A} \rightarrow \mathcal{B}$ ou nas seguintes formas abreviadas: $\langle \ell, r, \eta \rangle$, $\langle \ell, r, \varepsilon \rangle$, ou $\langle \ell, r \rangle$. Considere a adjunção $\langle \ell, r, \eta \rangle: \mathcal{A} \rightarrow \mathcal{B}$, um objeto B em $Ob_{\mathcal{B}}$ e $id_{rB}: rB \rightarrow rB$ um morfismo em \mathcal{A} . O único morfismo $g: \ell rB \rightarrow B$ em \mathcal{B} tal que $f = rg \circ \eta_{rB}$ é a co-unidade no objeto B , ou seja, $g = \varepsilon_B: \ell rB \rightarrow id_{\mathcal{B}} B$ (veja, por exemplo, [Barr & Wells 90, p.276]).

Proposição 6.2 O adjunto esquerdo de um functor, se existe, é único, a menos de isomorfismo.

Prova: Veja [Barr & Wells 90, p.282]. □

Adjunções são componíveis, como mostrado a seguir.

Proposição 6.3 Sejam $\langle \ell, r, \eta, \varepsilon \rangle: \mathcal{A} \rightarrow \mathcal{B}$ e $\langle \ell', r', \eta', \varepsilon' \rangle: \mathcal{B} \rightarrow \mathcal{C}$ duas adjunções. A composição das adjunções pode ser definida como $\langle \ell' \circ \ell, r' \circ r, \eta' \circ \ell \eta, \ell' \varepsilon' \circ \varepsilon \rangle: \mathcal{A} \rightarrow \mathcal{C}$.

Prova: Veja [Rydeheard & Burstall 88, p. 152]. □

Proposição 6.4 Considere a adjunção $\langle \ell, r \rangle$. Então, r preserva limites e ℓ preserva colimites.

Prova: Veja [Barr & Wells 90, p.283]. □

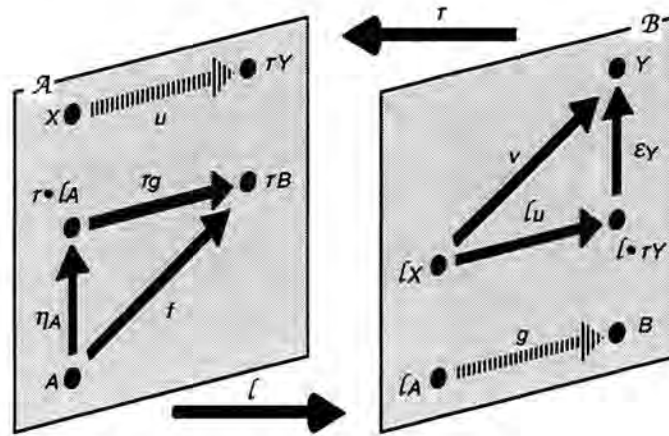


Figura 6.1 Adjunção

6.2 Reflexão e Co-Reflexão

Definição 6.5 Reflexão, Co-Reflexão. A adjunção $\langle \ell, r, \eta, \epsilon \rangle: \mathcal{A} \rightarrow \mathcal{B}$ é dita uma:

- a) reflexão, se ϵ (co-unidade) é um isomorfismo natural;
- b) co-reflexão, se η (unidade) é um isomorfismo natural. □

Intuitivamente, os conceitos de reflexão e co-reflexão significam que as composições $\ell \circ r$ e $r \circ \ell$ (no caso de reflexão e co-reflexão, respectivamente), quando aplicadas a um objeto, resultam nele próprio ou em um isomorfo. Ou seja, a "passagem de ida e volta" entre as categorias não modifica um objeto na sua essência. Em geral, o seguinte ocorre em relação aos funtores, o que induz a notação abaixo:

- no caso de reflexão, o adjunto direito é um funtor inclusão;
- no caso de co-reflexão, o adjunto esquerdo é um funtor inclusão.

Notação 6.6 Considere a adjunção $\langle \ell, r, \eta, \epsilon \rangle: \mathcal{A} \rightarrow \mathcal{B}$. É normal usar a seguinte notação para relacionar as categorias \mathcal{A} e \mathcal{B} :

- a) $\mathcal{A} \hookleftarrow \mathcal{B}$, se a adjunção é uma reflexão;
- b) $\mathcal{A} \hookrightarrow \mathcal{B}$, se a adjunção é uma co-reflexão. □

7 Construções sobre Funtores

7.1. Funtor Diagonal

Definição 7.1 Funtor Diagonal. Considere a categoria C . Seja C^2 a categoria onde os objetos e os morfismos são pares de C . O *funtor diagonal* $\Delta: C \rightarrow C^2$ associa a cada C -objeto A o C^2 -objeto $\langle A, A \rangle$ e a cada C -morfismo $f: A \rightarrow B$ o C^2 -morfismo $\langle f, f \rangle: \langle A, A \rangle \rightarrow \langle B, B \rangle$. □

Proposição 7.2 Considere a categoria C e o funtor diagonal $\Delta: C \rightarrow C^2$. Então:

- a) suponha C com produtos binários. Então, o funtor $\Pi: C^2 \rightarrow C$ é adjunto direito de Δ , onde Π associa a cada C^2 -objeto $\langle A, B \rangle$ o C -objeto resultante do produto $A \times B$ e a cada C^2 -morfismo $\langle f, g \rangle$ o C -morfismo $f \times g$ univocamente induzido pelo produto em C ;
- a) suponha C com coprodutos binários. Então, o funtor $\amalg: C^2 \rightarrow C$ é adjunto esquerdo de Δ , onde \amalg associa a cada C^2 -objeto $\langle A, B \rangle$ o C -objeto resultante do coproduto $A + B$ e a cada C^2 -morfismo $\langle f, g \rangle$ o C -morfismo $f + g$ univocamente induzido pelo coproduto em C .

Prova: Veja [Mac Lane 71, p. 85]. □

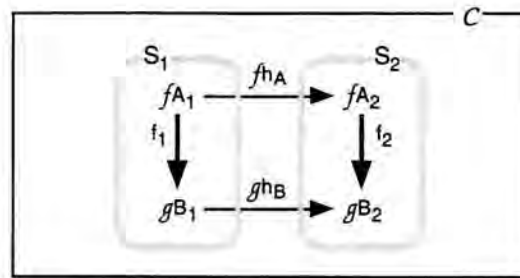


Figura 7.1 Diagrama comutativo para as categorias das setas

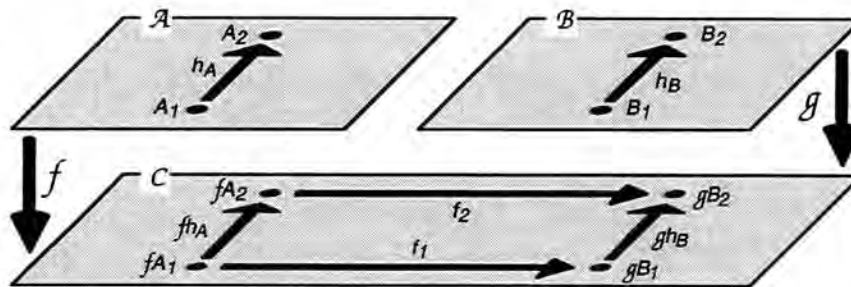


Figura 7.2 Categoria das setas

7.2 Categoria das Setas

Definição 7.3 Categoria das Setas. Considere os funtores $f: \mathcal{A} \rightarrow C$ e $g: \mathcal{B} \rightarrow C$. A categoria das setas $f \downarrow g$ é tal que:

- os objetos são triplas $S = \langle A, f, B \rangle$ onde A é um \mathcal{A} -objeto, B é um \mathcal{B} -objeto e $f: fA \rightarrow gB$ é um C -morfismo;
- um morfismo entre dois objetos $S_1 = \langle A_1, f_1, B_1 \rangle$ e $S_2 = \langle A_2, f_2, B_2 \rangle$ é um par $h = \langle h_A: A_1 \rightarrow A_2, h_B: B_1 \rightarrow B_2 \rangle$ onde h_A e h_B são morfismos em \mathcal{A} e \mathcal{B} , respectivamente, e são tais que o diagrama ilustrado na Figura 7.1 comuta;
- o morfismo identidade de um objeto $S = \langle A, f, B \rangle$ é $1_S = \langle 1_A: A \rightarrow A, 1_B: B \rightarrow B \rangle$;
- a composição de dois morfismos $f = \langle f_A, f_B \rangle: S_1 \rightarrow S_2$ e $g = \langle g_A, g_B \rangle: S_2 \rightarrow S_3$ é $g \circ f = \langle g_A \circ f_A, g_B \circ f_B \rangle: S_1 \rightarrow S_3$. □

A Figura 7.2 ilustra a construção de objetos e de morfismos em uma categoria das setas.

Proposição 7.4 Considere os funtores $f: \mathcal{A} \rightarrow C$, $g: \mathcal{B} \rightarrow C$ e a categoria das setas $f \downarrow g$.

- se \mathcal{A} , \mathcal{B} são completas e g preserva limites, então $f \downarrow g$ é completa;
- se \mathcal{A} , \mathcal{B} são cocompletas e f preserva colimites, então $f \downarrow g$ é cocompleta.

Prova: Veja [Casley 91, p. 46]. □

7.3 Funtor Produto Fibrado

Definição 7.5 Funtor Produto Fibrado. Considere uma categoria C com todos os produto fibrados. Sejam $f: A \rightarrow B$ um C -morfismo e \mathcal{A} e \mathcal{B} subcategorias de C induzidas por A e B as quais possuem um único objeto (A e B , respectivamente) e um único morfismo (o identidade). Sejam $c \downarrow a$ e $c \downarrow b$ categorias das setas onde $a: \mathcal{A} \rightarrow C$ e $b: \mathcal{B} \rightarrow C$ são funtores inclusão e $c: C \rightarrow C$ é o funtor identidade. O funtor produto fibrado induzido pelo morfismo $f: A \rightarrow B$ é $f^*: c \downarrow b \rightarrow c \downarrow a$ é tal que, para todos os objetos $g: D \rightarrow B$, $g': D' \rightarrow B$ em $c \downarrow b$ e para todo o morfismo $k: D \rightarrow D'$ em $c \downarrow b$, f^* é descrito pelo diagrama ilustrado na Figura 7.3, onde f^*k é univocamente induzido pelos produtos fibrados. □

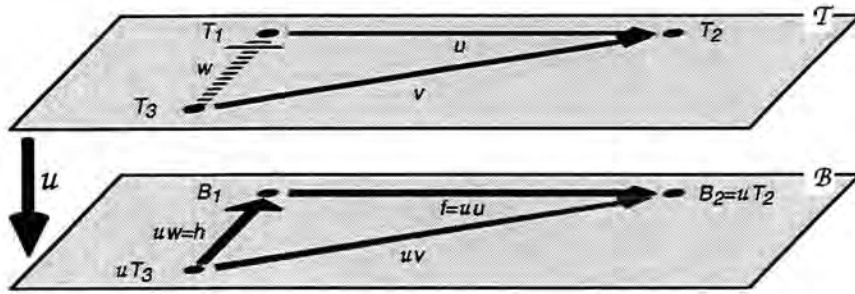


Figura 8.1 Morfismo cartesiano

Em uma fibração (cofibrção) $u: \mathcal{T} \rightarrow \mathcal{B}$, é usual afirmar que \mathcal{T} é a categoria total, \mathcal{B} é a categoria base e \mathcal{T} é fibrado (cofibrado) sobre \mathcal{B} . Um funtor que é simultaneamente uma fibração e uma cofibrção é denominado bifbração.

A escolha de uma determinada elevação cartesiana (cocartesiana) induz uma função denominada fenda (cofenda).

Definição 8.3 Fenda, Cofenda. Considere a fibração (cofibrção) $u: \mathcal{T} \rightarrow \mathcal{B}$.

- a) Uma fenda da fibração u é uma função que associa a cada \mathcal{B} -morfismo $f: B_1 \rightarrow B_2$ e a cada \mathcal{T} -objeto T_2 tal que $uT_2 = B_2$ um \mathcal{T} -morfismo $u: T_1 \rightarrow T_2$ cartesiano em relação ao f e T_2 .
- b) Uma cofenda da cofibrção u é uma função que associa a cada \mathcal{B} -morfismo $f: B_1 \rightarrow B_2$ e a cada \mathcal{T} -objeto T_1 tal que $uT_1 = B_1$ um \mathcal{T} -morfismo $u: T_1 \rightarrow T_2$ cocartesiano em relação ao f e T_2 . \square

Proposição 8.4 Considere uma fenda da fibração $u: \mathcal{T} \rightarrow \mathcal{B}$ e o \mathcal{B} -morfismo $f: B_1 \rightarrow B_2$. Então $f^*: u^{-1}B_2 \rightarrow u^{-1}B_1$ definido como abaixo e ilustrado na Figura 8.2 onde $u^{-1}B_1$ e $u^{-1}B_2$ são fibras determinadas por B_1 e B_2 , respectivamente, é um funtor induzido pela fenda da fibração u e pelo morfismo f :

- para cada $u^{-1}B_2$ -objeto T_1 , seja $f^*T_1 = D_{T_1}$ onde $f_{T_1}: D_{T_1} \rightarrow T_1$ é o morfismo cartesiano determinado pela fenda para f e T_1 ;
- para cada $u^{-1}B_2$ -morfismo $u: T_1 \rightarrow T_2$, f^*u é o único $u^{-1}B_1$ -morfismo tal que o diagrama ilustrado na Figura 8.3 comuta. \square

Proposição 8.5 Considere uma cofenda da cofibrção $u: \mathcal{T} \rightarrow \mathcal{B}$ e o \mathcal{B} -morfismo $f: B_1 \rightarrow B_2$. Então $f^+: u^{-1}B_1 \rightarrow u^{-1}B_2$ definido como abaixo onde $u^{-1}B_1$ e $u^{-1}B_2$ são fibras determinadas por B_1 e B_2 , respectivamente, é um funtor induzido pela cofenda da cofibrção u e pelo morfismo f :

- para cada $u^{-1}B_1$ -objeto T_1 , seja $f^+T_1 = C_{T_1}$ onde $f_{T_1}: T_1 \rightarrow C_{T_1}$ é o morfismo cocartesiano determinado pela cofenda para f e T_1 ;
- para cada $u^{-1}B_1$ -morfismo $u: T_1 \rightarrow T_2$, f^+u é o único morfismo w dado na definição de morfismo cocartesiano tal que o diagrama ilustrado na Figura 8.4 comuta. \square

Prova: Veja [Barr & Wells 90, p. 256]. \square

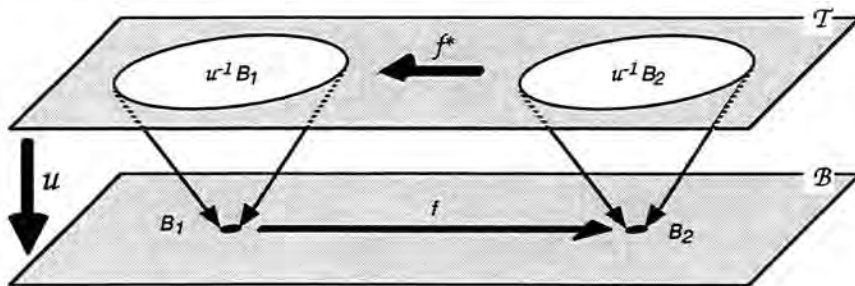


Figura 8.2 Funtor induzido por uma fibração

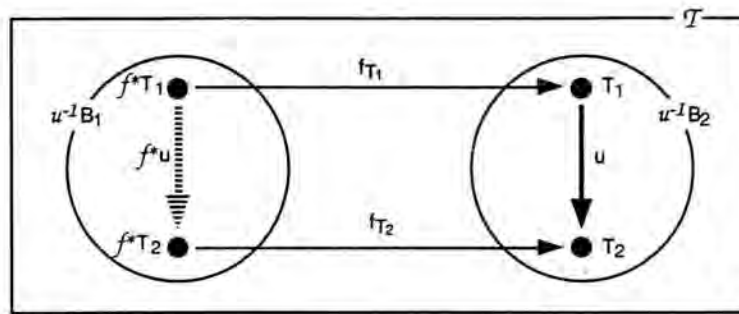


Figura 8.3 Diagrama comutativo para funtor induzido por uma fibração

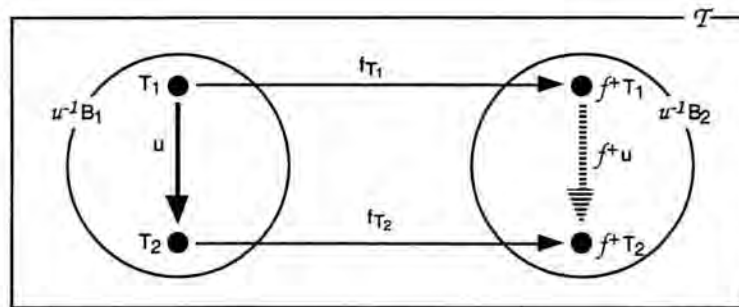


Figura 8.4 Diagrama comutativo para funtor induzido por uma cofibração

9 Categorias Concretas

Considere uma categoria abstrata cujos objetos estruturados e os correspondentes morfismos são definidos sobre os objetos e morfismo de uma categoria base. Se a categoria resultante for tratada puramente como uma categoria abstrata, algumas informações sobre os objetos e os morfismos da categoria base podem ser perdidas. Adicionalmente, muitas das propriedades da categoria abstrata são, de fato, propriedades da categoria base. A generalização desta idéia é conhecida como categoria concreta sobre uma categoria base, considerando a existência de um funtor fidedigno (em geral, um funtor esquecimento) da categoria abstrata para a base. Desta forma, estruturas complexas podem ser decompostas em estruturas mais simples, em diferentes níveis de abstração, de forma composicional. O texto que segue é baseado em [Adámek *et al* 90]. Por simplicidade, para uma coleção de índices I , é omitido que $i \in I$.

9.1 Categoria Concreta

Definição 9.1 Categoria Concreta. Considere as categorias C e B . Seja $u: C \rightarrow B$ um funtor fidedigno. Então $\langle C, u \rangle$ é uma categoria concreta sobre a categoria base B . \square

Considerando que o funtor é fidedigno, é usual usar a mesma identificação para os morfismo de C e os correspondentes morfismos em B .

9.2 Fonte e Sumidouro

Definição 9.2 Fonte, Sumidouro. Considere a categoria C .

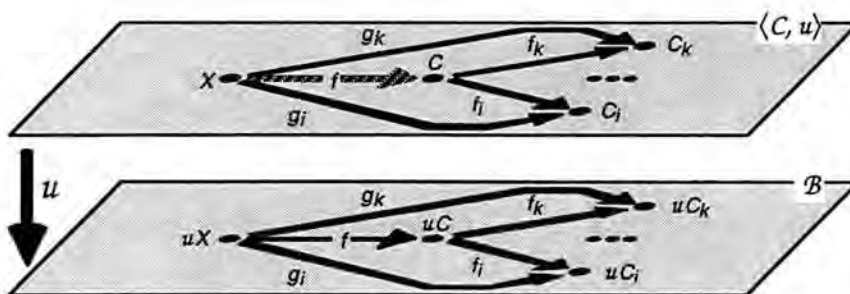


Figura 9.1 Fonte inicial

- a) Seja $\langle C, \{f_i: C \rightarrow C_i\} \rangle$ tal que C é um C -objeto e $\{f_i\}$ é uma coleção indexada de C -morfismos. Então, $\langle C, \{f_i\} \rangle$ é uma fonte de C onde C e $\{C_i\}$ são denominados domínio e codomínio da fonte, respectivamente.
- b) Seja $\langle \{f_i: C_i \rightarrow C\}, C \rangle$ tal que C é um C -objeto e $\{f_i: C_i \rightarrow C\}$ é uma coleção indexada de C -morfismos. Então, $\langle \{f_i\}, C \rangle$ é um sumidouro de C onde $\{C_i\}$ e C são denominados domínio e codomínio da fonte, respectivamente. □

Definição 9.3 Fonte Inicial, Sumidouro Final. Considere a categoria concreta $\langle C, u: C \rightarrow B \rangle$.

- a) Uma C -fonte $\langle C, \{f_i: C \rightarrow C_i\} \rangle$ é dita inicial se, para cada C -fonte $\langle X, \{g_i: X \rightarrow C_i\} \rangle$ e para cada B -morfismo $f: uX \rightarrow uC$ tal que, para todo i em I , $g_i = f_i \circ f: uX \rightarrow uC_i$ em B , existe o C -morfismo $f: X \rightarrow C$ tal que, para todo i em I , $g_i = f_i \circ f: X \rightarrow C_i$ em C , como ilustrado na Figura 9.1.
- b) Um C -sumidouro $\langle \{f_i: C_i \rightarrow C\}, C \rangle$ é dito final se, para cada C -sumidouro $\langle \{g_i: C_i \rightarrow X\}, X \rangle$ e para cada B -morfismo $f: uC \rightarrow uX$ tal que, para todo i em I , $g_i = f \circ f_i: uC_i \rightarrow uX$ em B , existe o C -morfismo $f: C \rightarrow X$ tal que, para todo i em I , $g_i = f \circ f_i: C_i \rightarrow X$ em C . □

Observação 9.4 Fonte Inicial e Sumidouro Final \times Morfismos Cartesianos e Cocartesianos. As definições fonte inicial e sumidouro final são similares às definições de morfismos cartesiano e cocartesiano, respectivamente, excetuando-se que a categoria considerada é concreta e as definições são em relação à fonte e sumidouro (e não morfismos). □

9.3 Limite e Colimite Concretos

No texto que segue, o limite (colimite) de um diagrama D é denotado simplesmente por \lim_D (colim_D), representando o objeto \lim_D (colim_D) juntamente com a correspondente coleção de morfismos.

Definição 9.5 Limite, Colimite Concretos. Considere a categoria concreta $\langle C, u \rangle$.

- a) Um limite de um diagrama em $\langle C, u \rangle$ é dito concreto se é preservado por u .
- a) Um colimite de um diagrama em $\langle C, u \rangle$ é dito concreto se é preservado por u . □

Proposição 9.6 Considere a categoria concreta $\langle C, u: C \rightarrow B \rangle$. Seja D um diagrama em C . Então:

- a) \lim_D é um limite concreto de D em C sse $u\lim_D$ é um limite de uD em B e \lim_D é uma fonte inicial em $\langle C, u \rangle$;
- b) colim_D é um colimite concreto de D em C sse $u\text{colim}_D$ é um colimite de uD em B e colim_D é um sumidouro final em C .

Prova: Veja [Adámek *et al* 90, p. 210]. □

Portanto, a construção de um limite (colimite) concreto pode ser determinada, como segue:

- determinação do limite (colimite) do diagrama correspondente na categoria base;
- elevação do limite (colimite) da categoria base como uma fonte inicial (um sumidouro final) na categoria concreta.

Definição 9.7 Elevação de Limites e Colimites. Considere o funtor $f: C \rightarrow B$.

- a) f eleva os limites (univocamente) se, para cada diagrama D em C e para cada limite $\lim f_D$ em B , existe um (único) limite \lim_D em C tal que $f \lim_D = \lim f_D$.
- b) f eleva os colimites (univocamente) se, para cada diagrama D em C e para cada colimite $\text{colim} f_D$ em B , existe um (único) colimite colim_D em C tal que $f \text{colim}_D = \text{colim} f_D$. \square

Proposição 9.8 Considere o funtor $f: C \rightarrow B$.

- a) Se f eleva limites e B é completa, então C é completa.
- b) Se f eleva colimites e B é cocompleta, então C é cocompleta.

Prova: Veja [Adámek *et al* 90, p. 210]. \square

Corolário 9.9 Considere a categoria concreta $\langle C, u: C \rightarrow B \rangle$.

- a) Se B é completa e se para cada diagrama D em C o limite de uD em B pode ser elevado como uma fonte inicial em C , então $\langle C, u \rangle$ é completa.
- b) Se B é cocompleta e se para cada diagrama D em C o colimite de uD em B pode ser elevado como um sumidouro final em C , então $\langle C, u \rangle$ é cocompleta. \square

Corolário 9.10 Considere a categoria concreta $\langle C, u: C \rightarrow B \rangle$.

- a) Se B é completa e, para toda a coleção de C -objetos $\{C_i\}$, para todo B -objeto B e para toda B -fonte $\langle B, \{f_i: B \rightarrow C_i\} \rangle$, se $\langle B, \{f_i\} \rangle$ pode ser elevada como uma fonte inicial em C , então $\langle C, u \rangle$ é completa.
- b) Se B é cocompleta e, para toda a coleção de C -objetos $\{C_i\}$, para todo B -objeto B e para todo B -sumidouro $\langle B, \{f_i: C_i \rightarrow B\} \rangle$, se $\langle \{f_i\}, B \rangle$ pode ser elevado como um sumidouro final em C , então $\langle C, u \rangle$ é cocompleta. \square

10 Mônadas e Categorias Kleisli

Um monóide $\langle A, \bullet, e \rangle$ com suporte A , operação binária e associativa \bullet e elemento neutro e (à esquerda e à direita em relação à operação \bullet) pode ser considerado como uma tripla $\langle A, \mu, \eta \rangle$ onde $\mu: A^2 \rightarrow A$ descreve a operação e $\eta: 1 \rightarrow A$, onde 1 é um conjunto de um só elemento, identifica a identidade em A . A associatividade e a unidade são determinadas pelos diagramas comutativo ilustrados na Figura 10.1. A generalização desta abordagem considerando um endofuntor como suporte e transformações naturais como μ e η , define uma mônada.

Definição 10.1 Mônada. Uma mônada ou tripla sobre uma categoria C é uma tripla $T = \langle t, \mu, \eta \rangle$ onde $t: C \rightarrow C$ é um endofuntor, $\mu: t^2 \rightarrow t$ é uma transformação natural denominada multiplicação ou operação interna e $\eta: id_C \rightarrow t$ é uma transformação natural denominada unidade ou neutro tal que os diagramas ilustrados na Figura 10.2 comutam, onde

- t^2 e t^3 denotam $t \circ t$ e $t \circ t \circ t$, respectivamente. Considerando que a operação é associativa, $t \circ (t \circ t)$ e $(t \circ t) \circ t$ são identificados;
- $t: t \rightarrow t$ é a transformação natural identidade;
- $\mu t, t\mu, \eta t$, e $t\eta$ representam composições horizontais de transformações naturais. \square

Proposição 10.2 Seja $\langle \ell, r, \eta, \varepsilon \rangle: \mathcal{A} \rightarrow \mathcal{B}$ uma adjunção. Então $T = \langle t, \eta, \mu \rangle$ é uma mônada onde $t = r \circ \ell: \mathcal{A} \rightarrow \mathcal{A}$ e $\mu = r\varepsilon\ell: t^2 \rightarrow t$ tal que $r\varepsilon\ell$ é a composição horizontal de transformações naturais, como ilustrada na Figura 10.3.

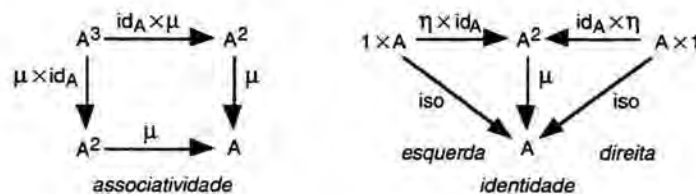


Figura 10.1 Diagramas comutativos para monóides

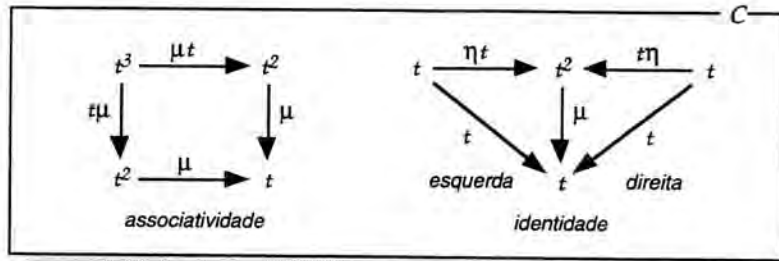


Figura 10.2 Diagramas comutativos para mônadas

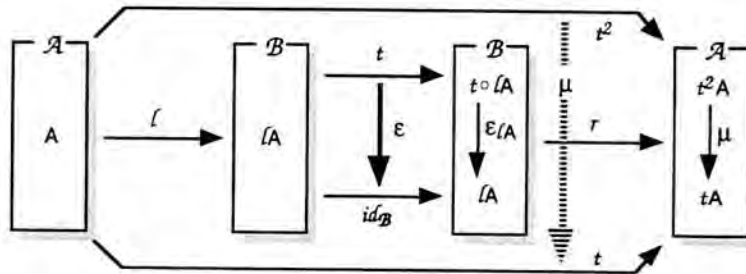


Figura 10.3 Construção da multiplicação de uma mônadas

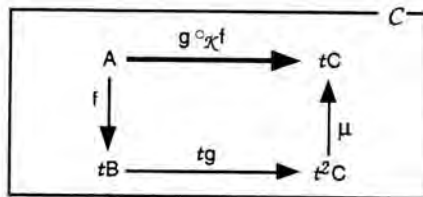


Figura 10.4 Composição de morfismos em uma categoria Kleisli

Prova: Veja [Asperti & Longo 91, p. 109]. □

Definição 10.3 Categoria Kleisli. Seja $T = \langle t, \eta, \mu \rangle$ uma mônada sobre a categoria C . A Categoria Kleisli $\mathcal{K} = \mathcal{K}(T)$ é tal que os objetos são os mesmos de C , para cada C -morfismo $f: A \rightarrow B$, $f: A \rightarrow B$ é um \mathcal{K} -morfismo, para cada C -objeto A , $\eta_A: A \rightarrow tA$ é o correspondente \mathcal{K} -morfismo identidade (determinado por $\eta_A: A \rightarrow tA$) e a para os \mathcal{K} -morfismos $f: A \rightarrow B$ e $g: B \rightarrow C$, a composição $g \circ_{\mathcal{K}} f: A \rightarrow C$ é determinada em C , como ilustrado na Figura 10.4. □