

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

**Relações Formais entre  
Gramáticas de Grafos e Redes de Petri**

por

MARCELO CUNHA SANTOS

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

Prof. Dr. Daltro José Nunes  
Orientador

Profa. Dra. Leila Ribeiro  
Co-orientadora

Porto Alegre, maio de 1999.

**CIP - Catalogação na Publicação**

Santos, Marcelo Cunha

Relações Formais entre Gramáticas de Grafos e Redes de Petri / por Marcelo Cunha Santos. – Porto Alegre: PPGC da UFRGS, 1999.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR - RS. Orientador: Nunes, Daltro José. Co-orientadora: Ribeiro, Leila.

1.Formalismos. 2.Gramáticas de Grafos. 3.Redes de Petri. 4.Semântica. 5.Teoria das Categorias. I.Nunes, Daltro José. II.Ribeiro, Leila. III.Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. Franz Rainer Semmelmann

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

Coordenadora do CPGCC: Profa. Carla Maria Dal Sasso Freitas

Bibliotecária-Chefe do Instituto de Informática: Beatriz Haro

## **Agradecimentos**

Uma finidade de pessoas que eu quero agradecer contribuíram, direta ou indiretamente, para a realização deste trabalho. Merecem destaque especial Sérgio M. Schneider, Rafael D. Lins, Daltro J. Nunes e Leila Ribeiro (a ordem é cronológica).

Agradecimentos também ao CPGCC/UFRGS pela disponibilidade de recursos humanos e materiais e ao apoio financeiro do CNPq.

## Sumário

<b>Lista de Símbolos</b> .....	<b>6</b>
<b>Lista de Figuras</b> .....	<b>7</b>
<b>Resumo</b> .....	<b>9</b>
<b>Abstract</b> .....	<b>10</b>
<b>1 Introdução</b> .....	<b>11</b>
1.1 Gramáticas de Grafos .....	11
1.2 Redes de Petri .....	11
1.3 Ferramenta Matemática e Semântica Utilizadas .....	12
1.4 Objetivos .....	12
1.5 Síntese dos Capítulos .....	13
<b>2 Redes de Petri</b> .....	<b>14</b>
2.1 Redes Lugar/Transição .....	15
2.2 Semântica Operacional de Redes L/T .....	20
2.3 Redes Seguras .....	21
2.4 Redes de Ocorrência .....	22
2.5 Unfolding de Redes Lugar/Transição .....	25
<b>3 Gramáticas de Grafos</b> .....	<b>29</b>
3.1 Grafos .....	30
3.2 Grafos Tipados .....	32
3.3 Regras .....	35
3.4 Gramáticas de Grafos .....	38
3.5 Semântica Operacional de Gramáticas de Grafos .....	41
3.6 Grafos Duplamente Tipados .....	42
3.7 Regras Duplamente Tipadas .....	46
3.8 Gramáticas de Grafos com Tipo Duplo .....	47
3.9 Relações numa Gramática de Grafos com Tipo Duplo .....	49
3.10 Gramáticas de Grafos de Ocorrência .....	54
3.11 Unfolding de Gramáticas de Grafos .....	58
<b>4 Transformando Gramáticas de Grafos em Redes L/T</b> .....	<b>64</b>
4.1 Transformação dos Objetos de GG .....	64
4.2 Transformação dos Morfismos de GGD .....	68
4.3 Transformação dos Objetos de $\overline{\text{OccGGD}}$ .....	70
4.4 Transformação dos Morfismos de $\overline{\text{OccGGD}}$ .....	74
<b>5 Transformando Redes L/T em Gramáticas de Grafos</b> .....	<b>76</b>
5.1 Transformação dos Objetos de PTNets .....	76
5.2 Transformação dos Morfismos de PTNets .....	79
5.3 Transformação dos Objetos de OccNets .....	85

5.4	Transformação dos Morfismos de OccNets .....	87
<b>6</b>	<b>Relações entre Gramáticas de Grafos e Redes L/T .....</b>	<b>89</b>
6.1	Relação Sintática .....	89
6.2	Relação Semântica .....	92
<b>7</b>	<b>Trabalhos Relacionados .....</b>	<b>99</b>
<b>8</b>	<b>Conclusão .....</b>	<b>101</b>
<b>A</b>	<b>Transformação de Gramáticas de Grafos em Gramáticas de Grafos Discretas .....</b>	<b>103</b>
A.1	Transformação Direta .....	103
A.2	Transformação Estruturada .....	103
A.3	Adjunção entre GG e GGD .....	104
A.4	Implicação Semântica .....	105
	<b>Bibliografia .....</b>	<b>106</b>

## Lista de Símbolos

$Cd(A)$ : cardinalidade do conjunto $A$ .	
$a \mapsto b$ : par ordenado $(a, b)$ .	
$\bigcup\{A_1, \dots, A_n\}$ : união dos conjuntos $A_1, \dots, A_n$ .	
$\uplus$ : união disjuntiva.	
$\dashrightarrow$ : morfismo total.	
$\hookrightarrow$ : morfismo total e injetor.	
$\Rightarrow$ : implicação lógica.	
$\Leftrightarrow$ : equivalência lógica.	
$(T, O_T)$ : conjunto apontado	16
$\bigoplus_{s \in \llbracket \mu \rrbracket} \mu(s)s$ : multiconjunto	16
$S^\oplus$ : conjunto de multiconjuntos	16
$u[\alpha \rangle v$ : passo	20
$\mathfrak{P}(a)$ : pré-transição	22
$\mathfrak{D}(a)$ : pós-transição	22
$\preceq$ : relação de dependência (redes L/T)	22
$\#$ : relação de conflito (redes L/T)	23
$Co(X)$ : elementos concorrentes	25
$\mathcal{U}_R^k(R)$ : <i>unfolding</i> finito de redes L/T	25
$\mathcal{U}_G^k(GG)$ : <i>unfolding</i> finito de gramáticas de grafos	59
$Unf^R(R)$ : <i>unfolding</i> (semântica) de redes L/T	26
$Unf^G(GG)$ : <i>unfolding</i> (semântica) de gramáticas de grafos	60
$\geq$ : comutatividade fraca	31
$\mathcal{T}_f$ : funtor de retipagem	34
$\subseteq^r$ : relação de subregra	37
$\xrightarrow{nr:m}$ : derivação direta	41
$\mathcal{V}_T$ : funtor de esquecimento de tipo	46
$Pre^R(a)$ : conjunto de pré-condições	49
$Min^R$ : conjunto de elementos mínimos	49
$R _S$ : restrição de $R$ ao conjunto $S$	49
$\leq$ : relação de dependência	50
$\overset{\#}{\dashrightarrow}$ : relação de conflito fraco	52
$\overset{\#}{\leftrightarrow}$ : relação de conflito (gramáticas de grafos)	52
$\leq^\#$ : relação de ocorrência (gramáticas de grafos)	53
$\leq_A^\#, \leq_a^\#$ relação de ocorrência local	53
$pre_a$ : pré-condição da regra $a$	46
$post_a$ : pós-condição da regra $a$	46

## Lista de Figuras

FIGURA 2.1 - Exemplo de redes Lugar/Transição .....	15
FIGURA 2.2 - Representação algébrica das redes da figura 2.1.....	18
FIGURA 2.3 - Exemplo de uma Rede L/T.....	20
FIGURA 2.4 - Disparo de uma transição numa rede L/T.....	21
FIGURA 2.5 - Exemplos de uma (a) rede segura e (b) rede de ocorrência. 22	
FIGURA 2.6 - Exemplos de uma (a) rede com ciclo e (b) rede com autoconflito.....	23
FIGURA 2.7 - Exemplo de uma rede de ocorrência decorada.....	24
FIGURA 3.1 - Exemplos de representação algébrica de grafos.....	31
FIGURA 3.2 - Exemplo de um morfismo entre os grafos $G$ e $H$ .....	32
FIGURA 3.3 - Grafo tipado $(G, t^G, T)$ .....	33
FIGURA 3.4 - Exemplo de um morfismo entre os grafos tipados $G^{T_0}$ e $H^{T_1}$ .....	34
FIGURA 3.5 - Retipagem do morfismo $g_1$ .....	36
FIGURA 3.6 - Exemplo de uma regra com tipo $T$ .....	37
FIGURA 3.7 - Exemplo de subregra.....	38
FIGURA 3.8 - Exemplo de uma gramática de grafos $GG_0$ .....	39
FIGURA 3.9 - Gramática de grafos $GG_1$ .....	40
FIGURA 3.10- Derivação direta do grafo $I_0$ usando a regra $r_2$ .....	42
FIGURA 3.11- Grafo duplamente tipado $G^{TG \nearrow T}$ .....	43
FIGURA 3.12- Morfismo entre grafos duplamente tipados.....	44
FIGURA 3.13- Retipagem do grafo duplamente tipado $G_1^{TG_1 \nearrow T_1}$ .....	45
FIGURA 3.14- Morfismo resultante da aplicação de $\mathcal{V}_T$ ao morfismo da figura 3.12.....	46
FIGURA 3.15- Regra duplamente tipada.....	47
FIGURA 3.16- Gramática de grafos com tipo duplo.....	48
FIGURA 3.17- Gramática duplamente tipada $GG_1$ .....	50
FIGURA 3.18- Gramática de grafos de ocorrência $GG_2$ .....	54
FIGURA 3.19- Gramática de Grafos $GG$ .....	62
FIGURA 4.1 - Exemplo de gramáticas de grafos discretas.....	67
FIGURA 4.2 - Redes L/T resultantes da aplicação do functor $\mathbb{R}$ às gramáticas da figura 4.1.....	67
FIGURA 5.1 - Transformação de uma rede L/T em uma gramática de grafos.....	78
FIGURA 5.2 - Duas redes L/T e suas representações algébricas.....	80
FIGURA 5.3 - Gramáticas resultantes da aplicação do functor $\mathbb{G}$ às redes da figura 5.2.....	81

FIGURA 6.1 - Exemplos de Redes de Ocorrência.....	92
FIGURA 6.2 - Gramáticas de Grafos de Ocorrência Discretas.....	93
FIGURA A.1 - Gramática de grafos discreta ( $GG_2$ ) e não-discreta ( $GG_1$ ).	103

## Resumo

Este trabalho vem a dar mais uma contribuição para o já consagrado uso de teoria das categorias para descrever e estabelecer relações entre formalismos diferentes. O uso de formalismos tem se tornado cada vez mais freqüente na Ciência da Computação, onde pesquisas têm sido realizadas direcionadas por domínios de aplicação, muitas vezes se chegando a modelos matemáticos que, em primeira instância, são totalmente distintos, não tendo, aparentemente, nada em comum entre si. Porém, analisados detalhadamente, tais formalismos podem revelar representarem intrinsecamente as mesmas idéias.

Esta dissertação tem como objetivo principal estabelecer uma relação (por meio de funtores) entre os formalismos de redes de Petri e gramáticas de grafos, a partir de suas já difundidas representações categóricas, utilizando, para isto, a linguagem da teoria das categorias. Com este intuito, é estabelecida uma relação entre as categorias sintáticas, onde é identificada uma subcategoria da categoria das gramáticas de grafos que é equivalente à categoria das redes de Petri. Para a semântica, será usado o modelo *unfolding semantics*, onde é estabelecida uma relação entre as categorias semânticas.

Palavras-chave: Formalismos, Gramáticas de Grafos, Redes de Petri, Semântica, Teoria das Categorias.

**TITLE:** “FORMAL RELATIONSHIPS BETWEEN GRAPH GRAMMAS AND PETRI NETS”

## Abstract

This work comes to be one more contribution to the well diffused use of category theory to describe and establish relationships between different formalisms. Recently, the use of formalisms has spread out to several areas of Computer Science, where research are directed by application domains, sometimes elaborating mathematical models that, at first sight, are totally different. Nevertheless, under detailed analysis, such formalisms can be seen as representing the same ideas.

The main goal of this dissertation is to establish a relationship (through functors) between the Petri nets and graph grammar formalisms, using category theory and their categorical representations found in the literature. With this in mind, it will be established a relationship between the syntactic categories, where a subcategory of the category of graph grammars will be identified and proved equivalent to the category of Petri nets. For the semantics, the model *unfolding semantics* will be used and a relationship between the semantic categories will also be established.

Keywords: Formalisms, Graph Grammars, Petri Nets, Semantics, Category Theory.

## 1 Introdução

### 1.1 Gramáticas de Grafos

As gramáticas de Chomsky, tais como as usadas em linguagens de programação, são definidas por um símbolo inicial e um conjunto de regras, onde o símbolo inicial e os lados direito e esquerdo das produções são *strings*. As **gramáticas de grafos** surgiram a partir destas gramáticas, utilizando grafos no lugar de *strings*: um grafo (grafo-estado) é usado para representar o símbolo inicial (estado inicial da gramática) e os lados esquerdo e direito das produções (regras da gramática) também são grafos. Deste modo, uma gramática de grafos consiste de um conjunto de produções e um grafo inicial. Analogamente à aplicação de regras nas gramáticas de Chomsky, uma regra em gramáticas de grafos pode ser aplicada num grafo-estado se houver uma ocorrência do lado esquerdo da regra no grafo-estado. Assim, quando um sistema é descrito utilizando gramáticas de grafos, seu estado (inicial) é representado por um grafo e uma mudança de estado é representada por relações entre grafos, descritas por meio das regras.

A semântica operacional descrevendo o comportamento de um sistema descrito por uma gramática de grafos é dada pela aplicação das regras da gramática a grafos-estado (iniciando-se no estado inicial). A aplicação de uma regra a um grafo-estado é chamada **derivação direta**.

Diferentes modelos matemáticos para representar grafos e gramáticas de grafos têm sido propostos na literatura [RIB 96b, TAE 96, KOR 93]. Neste trabalho, serão usados grafos dirigidos, representados algebricamente por meio de funções e conjuntos. Para as gramáticas de grafos, será seguida a abordagem algébrica SPO (*Single PushOut*). Nesta abordagem, as regras são representadas por um morfismo parcial entre grafos e as derivações diretas são definidas por meio de um *pushout*, em contraste com DPO (*Double PushOut*), onde uma derivação direta é definida por meio de dois *pushouts* [LÖW 93, EHR 96, COR 96]. Esta abordagem é chamada *algébrica* porque os grafos são considerados uma álgebra e uma derivação (aplicação de regras) é definida pela construção algébrica chamada *pushout* na categoria de grafos e morfismos de grafos. Com isto, pode-se aplicar resultados gerais da álgebra e teoria das categorias em gramáticas de grafos. Esta abordagem tem sido bastante pesquisada e provê resultados para a análise de paralelismo [RIB 96b, TAE 96], avaliação eficiente de expressões funcionais [AND 96] e mecanismos de sincronização, sistemas distribuídos e implementação de tipos abstratos de dados [COR 96].

### 1.2 Redes de Petri

As **Redes de Petri** foram apresentadas por C.A. Petri em sua tese de doutorado [PET 62] e estão sendo largamente usadas como modelo (formalismo) para concorrência. Têm sido também usadas para modelar sistemas paralelos, concorrentes,

assíncronos e não-determinísticos por possuírem uma representação gráfica que permite a visualização dos processos e a comunicação entre eles. Tem surgido grande interesse neste modelo sob o ponto de vista teórico por causa de sua simplicidade e natureza concorrente.

Vários tipos de redes de Petri com diferentes representações matemáticas têm surgido na literatura [LIN 96, REI 82]. Neste trabalho, serão usadas as redes **Lugar/Transição**, que são formadas por dois tipos de componentes: um ativo, denominado **transição**, e um passivo, denominado **lugar**. A ativação de uma transição é determinada pela presença de *tokens* em certos lugares da rede. Estas redes podem ser usadas para modelar um sistema, onde os recursos são representados pelos lugares, as possíveis ações pelas transições e os recursos disponíveis num dado momento pelos *tokens*. Deste modo, a semântica operacional (comportamento) de um tal sistema é dada pelo disparo de transições, descrevendo uma movimentação de *tokens* na rede. Assim como em gramáticas de grafos, vários modelos semânticos (estruturas matemáticas) têm sido propostos para representar esta semântica. Neste trabalho, será usada também a abordagem algébrica para representar as redes L/T [MES 94a, MES 94b].

### 1.3 Ferramenta Matemática e Semântica Utilizadas

Uma semântica é definida de modo a refletir o comportamento de uma rede L/T (resp. gramática de grafos) através do disparo de transições (resp. aplicação de regras), iniciando-se na marcação inicial (resp. grafo inicial). Com esta finalidade, várias abordagens para a representação e semântica de redes de Petri e gramáticas de grafos têm sido propostas na literatura, que diferem basicamente na quantidade de informações que são representadas na semântica e no modo de se representar tais informações. Para um estudo comparativo de diversos modelos semânticos, vantagens e desvantagens, veja [RIB 96b, MES 94b, COR 95]. Neste trabalho, será usada a abordagem algébrica, utilizando **teoria das categorias**, e o modelo semântico a ser utilizado para ambas redes L/T e gramáticas de grafos neste trabalho será *unfolding semantics* [MES 94a, RIB 96b]. Suas principais características/objetivos são:

- semântica concorrente;
- não-determinismo, concorrência e exclusão mútua representados explicitamente;
- não apenas ações, mas também dados são representados no nível semântico;
- relações importantes que descrevem um sistema (tais como relação de dependência, de conflito e de ocorrência) possuem representação;
- representa todas as derivações (seqüenciais e concorrentes) de uma gramática de grafos/rede L/T.

### 1.4 Objetivos

O objetivo deste trabalho é estabelecer relações sintáticas e semânticas entre os formalismos de gramáticas de grafos e redes L/T, utilizando o modelo semântico (*unfolding semantics*) definido em [MES 94b] (redes L/T) e [RIB 96b] (gramáticas de grafos). Estas relações serão estabelecidas por meio dos funtores  $\mathbb{R}$ ,  $\mathbb{G}$ ,  $\mathbb{UR}$  e  $\mathbb{UG}$ .

O funtor  $\mathbb{G}$  descreverá como transformar redes L/T, no nível sintático, em gramática de grafos, e vice-versa, através do funtor  $\mathbb{R}$ . Isto será feito pela *escolha* de uma subcategoria (GGD) de gramática de grafos onde os grafos obedecem a um certo

*padrão*, identificando a classe das gramáticas de grafos que podem ser transformadas em redes L/T. Já o funtor  $\mathbb{UG}$  relaciona a categoria semântica de redes L/T com a categoria semântica de gramática de grafos, e vice-versa, através do funtor  $\mathbb{UR}$ . O diagrama abaixo mostra as relações destes funtores com as respectivas categorias.

$$\begin{array}{ccc} \text{PTNets} & \xrightarrow{Unf^R} & \text{OccNets} \\ \mathbb{G} \left( \begin{array}{c} \downarrow \\ \uparrow \end{array} \right) \mathbb{R} & & \mathbb{UG} \left( \begin{array}{c} \downarrow \\ \uparrow \end{array} \right) \mathbb{UR} \\ \text{GGD} & \xrightarrow{Unf^G} & \text{OccGGD} \end{array}$$

Será analisado no texto a comutatividade deste diagrama. Para isto, será visto o seguinte:

- as categorias GGD e PTNets são equivalentes, de modo que podem ser consideradas “essencialmente a mesma” e desfrutam, assim, das mesmas propriedades;
- ocorre “perda de informação” com a aplicação do funtor  $\mathbb{UR}$ : a semântica de uma gramática de grafos é mais informativa do que a semântica de uma rede L/T (para uma gramática de grafos de ocorrência discreta  $Occ$ , há um morfismo  $h : Occ \rightarrow \mathbb{UG}(\mathbb{UR}(Occ))$ , que, via de regra, não é um isomorfismo);
- os funtores  $\mathbb{UG}$  e  $\mathbb{UR}$  preservam a semântica (preservam os colimites especiais usados na definição da semântica).

A existência destes funtores significa que qualquer rede de Petri pode ser modelada por uma gramática de grafos preservando sua semântica operacional. Isto quer dizer que as redes L/T podem ser consideradas um caso especial de gramática de grafos. Será mostrado também que, para uma gramática de grafos (discreta) modelada por meio de uma rede L/T, a semântica operacional é “preservada” através de um morfismo que possui algumas propriedades especiais, garantindo uma equivalência em termos de comportamento.

## 1.5 Síntese dos Capítulos

- capítulo 2: definição da categoria de redes L/T  $\text{PTNets}$ , juntamente com a semântica de *unfoldings*, que também é uma rede L/T, porém, com características especiais: são redes seguras livres de autoconflitos e de ciclos;
- capítulo 3: apresentação dos conceitos e definições de grafos, regras, grafos tipados e gramáticas de grafos (categoria sintática  $\text{GG}$ ) que são usados neste trabalho. Definição da categoria  $\text{OccGG}$  das gramáticas de ocorrência, que servirão como um domínio semântico para as gramáticas de  $\text{GG}$ . Estas gramáticas também possuem a propriedade de serem livres de ciclos e autoconflitos;
- capítulo 4: definição da categoria  $\text{GGD}$  das gramáticas de grafos discretas, que são as gramáticas de  $\text{GG}$  que serão transformadas em redes L/T de modo a preservar uma correspondência sintática e semântica. Os funtores  $\mathbb{R} : \text{GGD} \rightarrow \text{PTNets}$  e  $\mathbb{UR} : \text{OccGGD} \rightarrow \text{OccNets}$  são também definidos;
- capítulo 5: transformação das redes L/T em gramáticas de grafos: definição dos funtores  $\mathbb{G} : \text{PTNets} \rightarrow \text{GGD}$  e  $\mathbb{UR} : \text{OccNets} \rightarrow \text{OccGGD}$ ;
- capítulo 6: interrelacionamento entre as categorias sintáticas e semânticas por meio de funtores: prova de que as categorias  $\text{GGD}$  e  $\text{PTNets}$  são equivalentes. Exemplos de que os funtores  $\mathbb{UR}$  e  $\mathbb{UG}$  preservam colimites e não formam adjunção (pelo fato de a semântica de uma rede ser mais abstrata que a semântica de uma gramática de grafos).

## 2 Redes de Petri

As **Redes de Petri** foram apresentadas por C.A. Petri em [PET 62] e desde então vários tipos com diferentes abordagens têm surgido na literatura [LIN 96]. Neste trabalho, serão usadas as redes **Lugar/Transição** seguindo a abordagem algébrica. No uso destas redes para modelar um sistema, o estado (recursos) do sistema é representado por lugares, a disponibilidade de recursos (estado atual) por *tokens* (marcas) nos lugares e uma mudança de estado por (disparo de) transições (i.e., a ocorrência de eventos é equivalente ao disparo de transições). Ao ser disparada, uma transição remove *tokens* (consome recursos) de alguns lugares (pré-condição da transição) e cria novos *tokens* (disponibiliza recursos) em outros lugares (pós-condição da transição). O comportamento de tal rede é então dado pelo fluxo de *tokens* através dos disparo de transições. O disparo de uma transição não consome tempo e, dependendo da disponibilidade de *tokens*, várias transições podem ser disparadas “ao mesmo tempo” (paralelismo). Uma outra situação ocorre quando existem duas transições “aptas” a serem disparadas mas apenas uma delas pode ser disparada: a escolha de qual delas disparar é não-determinística (esta característica favorece o uso das redes L/T para a modelagem de sistemas não-determinísticos).

Vários modelos semânticos (estruturas matemáticas) têm sido propostos para representar estas idéias. Neste trabalho, será usada a semântica de *unfolding*. O *unfolding* finito de uma rede  $R$  é construído indutivamente, de modo que o *unfolding* zero representa a marcação inicial de  $R$  e os outros *unfoldings* são construídos a partir do disparo de transições.

As definições e conceitos deste capítulo foram retiradas de [LIN 96, MES 94b] e [MES 94a], onde também podem ser encontradas as provas para as proposições e teoremas apresentados.

### Estrutura do capítulo:

- seção 1: definição da categoria PTNets de redes L/T: os conceitos de redes L/T são apresentados juntamente com sua representação gráfica usual e algébrica, em termos de multiconjuntos;
- seção 2: conceituação da semântica operacional das redes L/T, em termos do disparo de transições a partir da marcação inicial;
- seção 3: definição das redes seguras, que são redes L/T onde os lugares têm multiplicidade unitária e são todos alcançáveis;
- seção 4: definição das redes de ocorrência, que são redes seguras que desfrutam de certas propriedades: não possuem ciclos nem autoconflitos. Estas redes serão usadas como domínio semântico para as redes L/T;
- seção 5: apresentação do *unfolding* e definição da semântica de uma rede L/T em termos de *unfoldings* finitos.

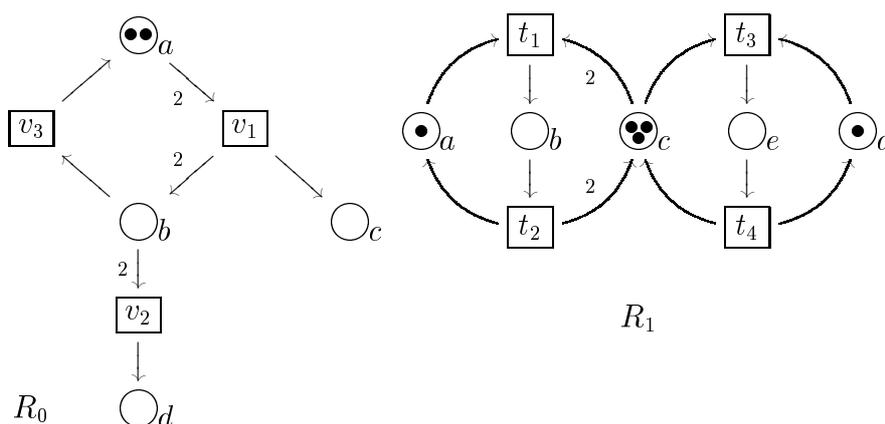


FIGURA 2.1- Exemplo de redes Lugar/Transição

## 2.1 Redes Lugar/Transição

As redes **Lugar/Transição** (ou, simplesmente, redes L/T) têm sido largamente usadas para modelar/especificar sistemas concorrentes. Estas redes são formadas por dois tipos de componentes: um ativo (**transições**) e um passivo (**lugares**). Na modelagem de um sistema, os lugares representam as variáveis de estado e as transições representam as ações/eventos realizados pelo sistema. A realização de ações está condicionada a um conjunto de pré-condições que a habilitam, resultando um conjunto de pós-condições, que por sua vez habilitam outras ações (vale ressaltar que nas redes clássicas não existe a noção de **tempo** para a realização de uma ação). A figura 2.1 mostra duas redes L/T na sua representação gráfica usual. Os círculos representam os lugares e os retângulos representam as transições. As marcas dentro dos círculos representam *tokens* (ou recursos) na rede e é chamada a **marcação** da rede. Arcos valorados que chegam nas transições representam os recursos **necessários** (pré-condição) para a ativação/habilitação da respectiva transição e os que saem delas representam os recursos gerados (valores unitários estão omitidos na figura). Num dado momento, as marcações de uma rede L/T representam as condições em que se encontra o sistema modelado e a presença de uma marca (*token*) num determinado lugar pode ser interpretada como a presença de um recurso de um determinado tipo. Os *tokens* disponíveis inicialmente numa rede são chamados **marcação inicial**. Uma vez ativada, uma transição pode consumir recursos (pré-condição) e gerar recursos (pós-condição) na rede.

**Exemplo 2.1:** Na rede  $R_0$  da figura 2.1, a transição  $v_1$  necessita de dois *tokens* do tipo  $a$  para ser ativada e disponibiliza na rede 2 *tokens* do tipo  $b$  e 1 do tipo  $c$ . Como existem 2 *tokens* no lugar  $a$ , a transição  $v_1$  está habilitada e pode ser disparada. ☺

Várias ferramentas matemáticas têm sido usadas para representar as redes L/T, como teoria dos conjuntos, teoria matricial ([LIN 96]) e teoria das categorias, entre outras. O uso de categorias possui a vantagem de se poder definir combinadores de redes (e outros conceitos) usando construções categóricas. Neste trabalho, serão usados multiconjuntos (*bags*) e funções (parciais) para representar as redes L/T. As funções parciais serão representadas por funções totais entre conjuntos apontados, definidos a seguir.

**DEFINIÇÃO 2.1. (Conjunto Apontado)** É um par  $(S, O_S)$  onde  $S$  é um conjunto e  $O_S \in S$  é o elemento apontado. Morfismos entre conjuntos apontados são funções que preservam os elementos apontados. ☺

Considerando o elemento apontado como um valor indefinido, a categoria de conjuntos e funções parciais é isomórfica à categoria de conjuntos apontados e seus morfismos.

**Exemplo 2.2:** A função parcial  $f$  definida abaixo pode ser representada pela função total  $g$  usando conjuntos apontados: os elementos com valor indefinido são mapeados para o elemento apontado.

$$f : A \rightarrow B$$

$x$	$f(x)$
$a$	1
$b$	undef
$c$	undef

$$A = \{a, b, c\}$$

$$B = \{1, 2, 3\}$$

$$g : A_1 \rightarrow B_1$$

$x$	$g(x)$
$O_A$	$O_B$
$a$	1
$b$	$O_B$
$c$	$O_B$

$$A_1 = \{O_A, a, b, c\}$$

$$B_1 = \{O_B, 1, 2, 3\}$$

☺

Os multiconjuntos (conjuntos com elementos repetidos, ou *bags*) serão usados para representar as multiplicidades dos *tokens* dos lugares, pré- e pós-condições de uma rede L/T. Serão definidos como uma função de um conjunto (de lugares) para o conjunto dos números naturais.

**DEFINIÇÃO 2.2. (Multiconjunto)** Dado um conjunto  $S$ ,  $S^\oplus$  denota o conjunto dos multiconjuntos de  $S$ , i.e., o conjunto de todas as funções de  $S$  para o conjunto  $\mathbb{N}$  dos números naturais. A função que mapeia todos os elementos de  $S$  para 0 será denotada por  $O_S$  e é definida como o elemento apontado de  $S^\oplus$ . Para  $\mu \in S^\oplus$ ,  $[\mu]$  denota o subconjunto de  $S$  constituído dos elementos  $s \in S$  tais que  $\mu(s) > 0$ . ☺

Notação: O multiconjunto  $\mu$  será representado pela soma  $\bigoplus_{s \in [\mu]} \mu(s)s$  (com a omissão das parcelas nulas).

**Exemplo 2.3:** Por exemplo, para o multiconjunto (função)  $\mu$  definido abaixo,  $\mu = b \oplus 2c \oplus 4d$  e  $[\mu] = \{b, c, d\}$ . O elemento apontado de  $S^\oplus$ ,  $O_S$ , que é também uma função, mapeia todos os elementos de  $S$  para zero.

$$S = \{a, b, c, d\}$$

$$\mu : S \rightarrow \mathbb{N}$$

$$O_S : S \rightarrow \mathbb{N}$$

$$O_S, \mu \in S^\oplus$$

$x$	$\mu(x)$
$a$	0
$b$	1
$c$	2
$d$	4

$x$	$O_S(x)$
$a$	0
$b$	0
$c$	0
$d$	0

☺

Como um conjunto  $S$  também pode ser um conjunto da forma  $A^\oplus$  (conjunto de multiconjuntos), será definida uma fórmula para transformar um elemento de  $S^\oplus = (A^\oplus)^\oplus$  num elemento de  $A^\oplus$ . Este conceito será usado na definição de morfismos entre redes L/T.

**DEFINIÇÃO 2.3. (Combinação Linear de Multiconjuntos)** Seja  $S = A^\oplus$ . Para  $n_\mu \in \mathbb{N}$ , a seguinte fórmula transforma um elemento de  $(A^\oplus)^\oplus$  num elemento de  $A^\oplus$ :

$$\bigoplus_{\mu \in S} n_\mu \mu = \bigoplus_{\mu} n_\mu \left( \bigoplus_{a \in A} \mu(a)a \right) = \bigoplus_{a \in A} \left( \sum_{\mu \in S} n_\mu \mu(a) \right) a$$

☺

**Exemplo 2.4:** Sejam  $A = \{a, b, c\}$  e  $S = A^\oplus$ . Sejam os multiconjuntos  $\mu_1 = a \oplus 2b$ ,  $\mu_2 = c \oplus 2b$  e  $\mu_3 = a \oplus c$ . O multiconjunto  $\mu = 2\mu_1 \oplus \mu_2 \oplus 3\mu_3$  pertence a  $S^\oplus$ , uma vez que  $\mu_1, \mu_2$  e  $\mu_3 \in S$ , e é calculado da seguinte maneira:

$$\begin{aligned}\mu &= 2(a \oplus 2b) \oplus (c \oplus 2b) \oplus 3(a \oplus c) \\ \mu &= (2 + 0 + 3)a \oplus (4 + 2 + 0)b \oplus (0 + 1 + 3)c \\ \mu &= 5a \oplus 6b \oplus 4c\end{aligned}$$

$x$	$\mu_1(x)$	$\mu_2(x)$	$\mu_3(x)$	$\mu(x)$
$a$	1	0	1	5
$b$	2	2	0	6
$c$	0	1	1	4

☺

Como os multiconjuntos serão usados na definição das redes L/T, é necessário definir morfismos entre conjuntos do tipo  $S^\oplus$ . Estes morfismos serão definidos em termos de combinação linear de multiconjuntos.

**DEFINIÇÃO 2.4. (Morfismo entre Conjuntos de Multiconjuntos)** Sejam  $A$  e  $B$  dois conjuntos e  $h : A \rightarrow B^\oplus$  uma função total. Um morfismo de  $A^\oplus$  para  $B^\oplus$  é uma função  $g : A^\oplus \rightarrow B^\oplus$  definida para todo  $\mu \in A^\oplus$  por

$$g(\mu) = \bigoplus_{s \in A} \mu(s)h(s)$$

Para  $h_1 : A \rightarrow B^\oplus$  e  $h_2 : B \rightarrow C^\oplus$ , a composição de  $h_1$  e  $h_2$  é dada por:  $\forall a \in A, h_2 \circ h_1(a) = \bigoplus_{b \in B} h_1(a)(b)h_2(b)$ . ☺

Isto significa que a função  $g$  é induzida pela função  $h$  e seu valor para cada elemento é calculado usando combinação linear de multiconjuntos. A composição destes morfismos representa “uma multiplicação de multiplicidades”: se  $h_1(a) = 2b \oplus 3c$ ,  $h_2(c) = 4d \oplus 2e$  e  $h_2(b) = 3k \oplus 2d$ , então  $h_2 \circ h_1(a)(d) = 3 \times 4 + 2 \times 2 = 16$ .

**Exemplo 2.5:** Para o morfismo  $g : A^\oplus \rightarrow B^\oplus$  induzido pela função  $h$  e o multiconjunto  $\mu = 2a \oplus b$  dados a seguir, tem-se que  $g(\mu) = \mu(a)h(a) \oplus \mu(b)h(b) \oplus \mu(c)h(c) = 2(u \oplus v) \oplus 1(2z \oplus u) \oplus 0(O_B) = (2+1+0)u \oplus (2+0+0)v \oplus (0+2+0)z \oplus (0+0+0)y = 3u \oplus 2v \oplus 2z$ .

$A = \{a, b, c\}$
$B = \{u, v, y, z\}$
$\mu = A \rightarrow \mathbb{N}$
$h : A \rightarrow B^\oplus$

$x$	$h(x)$
$a$	$u \oplus v$
$b$	$2z \oplus u$
$c$	$O_B$

$x$	$\mu(x)$
$a$	2
$b$	1
$c$	0

$x$	$g(\mu)(x)$
$u$	3
$v$	2
$z$	2
$y$	0

☺

Numa rede L/T, a marcação inicial e as pré- e pós-condições das transições serão representadas por multiconjuntos. Uma função de **origem** ( $p$ ) fornece a pré-condição de uma transição e uma de **destino** ( $d$ ) fornece a pós-condição. Será considerada aqui a restrição usual de que todas as transições devem possuir pré-condição diferente de vazio. Esta restrição é necessária para viabilizar a definição dos *unfoldings*. Exceção a esta regra é a **transição nula**, que possui pré- e pós-condições vazias.

**DEFINIÇÃO 2.5. (Rede Lugar/Transição)** Uma rede lugar/transição (rede L/T) é uma estrutura  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ , onde  $S_R$  é um conjunto cujos elementos são chamados **lugares**;  $(T_R, O_T)$  é um conjunto apontado cujos elementos

Rede  $R_0$ :

$$\begin{aligned}
R_0 &= (p_{R_0}, d_{R_0} : (T_{R_0}, O_{T_0}) \rightarrow S_{R_0}^\oplus, u_{R_0}) \\
S_{R_0} &= \{a, b, c, d\} \\
(T_{R_0}, O_{T_0}) &= \{O_{T_0}, v_1, v_2, v_3\} \\
\mu_{R_0} &= 2a
\end{aligned}$$

$x$	$p_{R_0}(x)$	$d_{R_0}(x)$
$O_{T_0}$	$O_{S_{R_0}}$	$O_{S_{R_0}}$
$v_1$	$2a$	$2b \oplus c$
$v_2$	$2b$	$d$
$v_3$	$b$	$a$

Rede  $R_1$ :

$$\begin{aligned}
R_1 &= (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1}) \\
S_{R_1} &= \{a, b, c, d, e\} \\
(T_{R_1}, O_{T_1}) &= \{O_{T_1}, t_1, t_2, t_3, t_4\} \\
\mu_{R_1} &= a \oplus 3c \oplus d
\end{aligned}$$

$x$	$p_{R_1}(x)$	$d_{R_1}(x)$
$O_{T_1}$	$O_{S_{R_1}}$	$O_{S_{R_1}}$
$t_1$	$2c \oplus a$	$b$
$t_2$	$b$	$a \oplus 2c$
$t_3$	$c \oplus d$	$e$
$t_4$	$e$	$c \oplus d$

FIGURA 2.2- Representação algébrica das redes da figura 2.1.

são chamados **transições**;  $p_R$  e  $d_R$  são morfismos entre conjuntos apontados que fornecem, respectivamente, **origem** e **destino** das transições;  $u_R \in S_N^\oplus$  é a **marcação inicial**. Além disso, há a seguinte restrição: para todo  $t \in (T_R, O_T)$ , se  $p_R(t) = O_S$  então  $t = O_T$ .  $\odot$

Pelo fato de  $p_R$  e  $d_R$  serem morfismos entre conjuntos apontados,  $p_R(O_T) = O_S$  e  $d_R(O_T) = O_S$ . Ou seja, a transição nula é a única que pode ter origem e destino nulos.

Exemplo 2.6: A figura 2.2 mostra a representação das redes  $R_0$  e  $R_1$  da figura 2.1 usando esta definição.  $\odot$

DEFINIÇÃO 2.6. (**Morfismo entre Redes L/T**) Um morfismo entre duas redes  $L/T$   $R_0 = (p_{R_0}, d_{R_0} : (T_{R_0}, O_{T_0}) \rightarrow S_{R_0}^\oplus, u_{R_0})$  e  $R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1})$  é um par de funções  $(f, h)$  que satisfaz as seguintes condições:

- 1:  $f : (T_{R_0}, O_{T_0}) \rightarrow (T_{R_1}, O_{T_1})$  é um morfismo entre conjuntos apontados;
- 2:  $g : S_{R_0}^\oplus \rightarrow S_{R_1}^\oplus$  é o morfismo entre multiconjuntos induzido pela função  $h : S_{R_0} \rightarrow S_{R_1}$ ;
- 3:  $g(u_{R_0}) = u_{R_1}$  ( $g$  preserva a marcação inicial);
- 4:  $\forall a \in S_{R_0}, \forall b \in S_{R_1}, g(a)(b) \leq 1$
- 5:  $\forall a \in S_{R_1}$  e  $\forall t \in T_{R_0}$ :
  - $(p_{R_1} \circ f(t))(a) = (g \circ p_{R_0}(t))(a)$
  - $(d_{R_1} \circ f(t))(a) = (g \circ d_{R_0}(t))(a)$

$$\begin{array}{ccccc}
S_{R_0}^\oplus & \xleftarrow{p_{R_0}} & T_{R_0} & \xrightarrow{d_{R_0}} & S_{R_0}^\oplus \\
\downarrow g & & \downarrow f & & \downarrow g \\
S_{R_1}^\oplus & \xleftarrow{p_{R_1}} & T_{R_1} & \xrightarrow{d_{R_1}} & S_{R_1}^\oplus
\end{array}$$

- 6:  $\forall a, b \in S_{R_0}$  se  $\llbracket g(a) \rrbracket \cap \llbracket g(b) \rrbracket \neq \emptyset$  então  $a = b$ .

A categoria de redes L/T é denotada por PTNets.  $\odot$

O item 4 da definição acima restringe a ação de  $g$ : a função  $h : S_{R_0} \rightarrow S_{R_1}$ , que induz o morfismo  $g$  é restrita a conjuntos, i.e.,  $\forall a \in S_{R_0}, h(a)$  é um conjunto.

Isto implica que o contradomínio de  $g$  restrito a  $S_{R_0}$  é restrito aos multiconjuntos  $\mu : S \rightarrow \{0, 1\} \subset \mathbb{N}$ . Esta restrição não é imposta aos morfismos definidos em [MES 94a] e é incluída aqui para permitir a existência de um funtor de PTNets para GG, a categoria de gramáticas de grafos. Isto significa que a quantidade de *tokens* da marcação inicial ou de pré- ou pós-condições das transições não pode ser “aumentada” por um morfismo, i.e., para  $d \in \llbracket g(b) \rrbracket$ , se  $b$  fizer parte da marcação inicial,  $d$  terá a mesma quantidade de *tokens* que  $b$  e se  $b$  pertencer à pós- ou pré-condição de alguma transição  $t$ , então a multiplicidade de  $d$  na pós- ou pré-condição da transição  $f(t)$  é a mesma que a de  $b$ .

O item 5 garante que os morfismos respeitarão a origem e o destino das transições. Esta é uma característica presente também nos morfismos entre grafos (definição 3.3).

O item 6 é incluído para permitir a existência do produto e coproduto na categoria. Esta condição impede que lugares pertencentes à marcação inicial ou à pós- ou pré-condição de alguma transição sejam mapeados para multiconjuntos tendo algum lugar em comum, limitando a ação dos morfismos. Em [MES 94a], esta restrição é imposta apenas à marcação inicial e às pós-condições das transições. Novamente, esta restrição é estendida neste trabalho para permitir a existência do funtor supracitado.

A categoria PTNets é bastante geral: os objetos são redes L/T nas quais as marcações podem ser infinitas e as transições podem ter pré- e pós-condições infinitas (mas com multiplicidades finitas para os lugares, i.e., o conjunto de lugares  $S_R$  pode ser infinito). A única restrição é que uma transição não pode ter pré-condição vazia. Os morfismos são bastantes gerais para permitirem construções categóricas: esta categoria possui objeto final e o produto e coproduto podem ser interpretados como as operações de composição paralela e não-determinística, respectivamente. O objeto terminal é uma rede com conjunto de lugares vazio, tendo como única transição a transição nula.

Exemplo 2.7: Para a rede  $R_2$  da figura 2.3, o conjunto de transições é  $(T_{R_2}, O_{T_2}) = \{O_{T_2}, t_1, t_2, t_3, t_4\}$ , o conjunto de lugares é  $S_{R_2} = \{k, l, m, n\}$  e a marcação inicial é  $u_{R_2} = 2k \oplus 2l$ . Um morfismo  $(f, h) : R_0 \rightarrow R_2$  entre as redes  $R_0$  da figura 2.1 e  $R_2$ , é dado pelas funções  $f$  e  $h$  dadas a seguir ( $g$  é induzida por  $h$ ). Se  $c$  fosse, por exemplo, mapeado para  $l$ , chegar-se-ia à contradição  $\llbracket g(a) \rrbracket \cap \llbracket g(c) \rrbracket = \{l\}$  (contraria o item 6 da definição). Como se pode notar, todos os elementos de  $\llbracket g(x) \rrbracket$  para  $x$  em  $S_{R_0}$  possuem multiplicidade menor ou igual a 1, i.e.,  $h(x)$  é um conjunto (item 4 da definição). Veja ainda que  $g(2a) = 2k \oplus 2l$ ,  $g(2b) = 2m$ ,  $g(b) = m$ ,  $g(2b \oplus c) = 2m$ ,  $g(d) = O_{S_{R_2}}$  e  $g(a) = k \oplus l$ , respeitando origem e destino das transições mapeadas.

$x$	$h(x)$
$a$	$k \oplus l$
$b$	$m$
$c$	$O_{S_{R_2}}$
$d$	$O_{S_{R_2}}$

$x$	$f(x)$
$O_{T_0}$	$O_{T_2}$
$v_1$	$t_1$
$v_2$	$t_2$
$v_3$	$t_3$

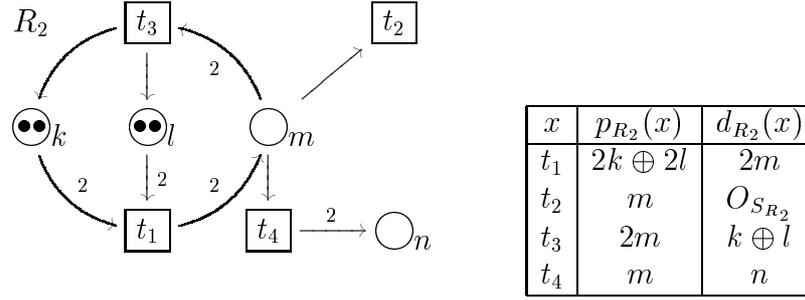


FIGURA 2.3- Exemplo de uma Rede L/T.

## 2.2 Semântica Operacional de Redes L/T

As transições formam a unidade básica de computação numa rede L/T. Uma transição  $t$  com origem  $p(t) = \mu_1$  e destino  $d(t) = \mu_2$  realiza uma computação consumindo os *tokens* em  $\mu_1$  e produzindo os *tokens* em  $\mu_2$ , i.e, a semântica operacional é dada pela movimentação dos *tokens* na rede. Para consumir *tokens*, uma transição precisa estar habilitada.

**DEFINIÇÃO 2.7. (Transição Habilitada)** Seja  $R$  uma rede L/T e  $u_R$  sua marcação. Uma transição  $t \in (T_R, O_T)$  está **habilitada** se, e somente se,  $\forall a \in S_R, p_R(t)(a) \leq u_R(a)$ . Com o **disparo** de  $t$  obtém-se uma nova marcação  $u_1 \in S_R^\oplus$  para  $R$  dada por:  $\forall a \in S_R, u_1(a) = u_R(a) - p_R(t)(a) + d_R(t)(a)$ . ☹

Uma transição habilitada é **disparada** para **consumir** os *tokens* de que necessita e **gerar** os *tokens* que disponibiliza. Num dado momento, mais de uma transição pode estar habilitada e qualquer uma pode ser (ou não) disparada, dando uma característica não-determinística à semântica operacional.

**Exemplo 2.8:** Na rede  $R_0$  da figura 2.4, a transição  $v_1$  pode ser disparada para produzir uma nova marcação  $u_1 = 2b \oplus c$  (mostrada na rede  $R'_0$ ):

$$\begin{aligned} u_1(a) &= 2 - 2 + 0 = 0 \\ u_1(b) &= 0 - 0 + 2 = 2 \\ u_1(c) &= 0 - 0 + 1 = 1 \\ u_1(d) &= 0 - 0 + 0 = 0 \end{aligned}$$

☹

Na rede  $R_1$  da figura 2.1, as transições  $t_1$  e  $t_3$  estão ambas habilitadas e, neste caso, podem ser disparadas ao mesmo tempo (i.e., em paralelo). Esta idéia será representada através de **passo**, que é formado por um número finito de transições compostas em paralelo. Um passo  $\alpha$  será representado por um multiconjunto de transições e tem origem e destino. A origem é a união das pré-condições das transições que formam o passo e o destino é a união das pós-condições.

**DEFINIÇÃO 2.8. (Passo)** Um passo  $\alpha$  é um multiconjunto de transições. Um passo com origem  $u$  e destino  $v$  é representado por  $u[\alpha \rangle v$ . O **conjunto de passos** de uma rede  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ ,  $\mathcal{P}(R)$ , é gerado pelas seguintes regras:

$$\frac{p_R(t) = u \text{ e } d_R(t) = v, t \in T_R \text{ e } w \text{ em } S^\oplus}{(u \oplus w)[t \rangle (v \oplus w) \text{ em } \mathcal{P}(R)} \qquad \frac{u[\alpha \rangle v \text{ e } u'[\beta \rangle v' \text{ em } \mathcal{P}(R)}{(u \oplus u')[\alpha \oplus \beta \rangle (v \oplus v') \text{ em } \mathcal{P}(R)}$$

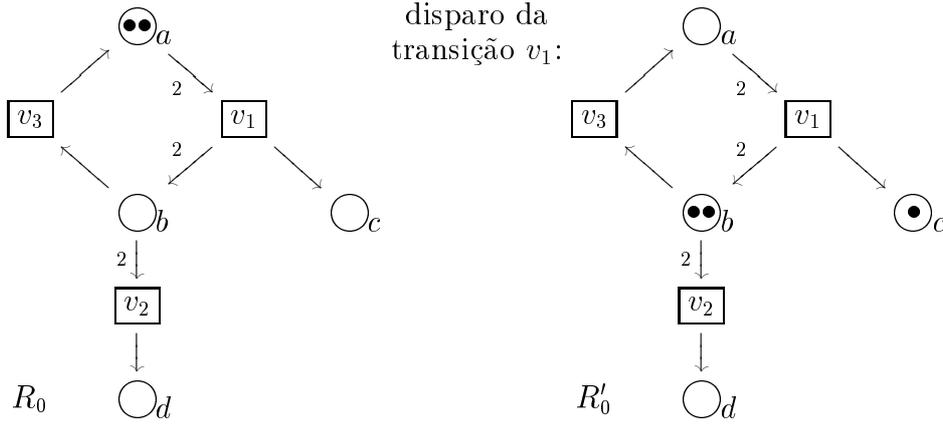


FIGURA 2.4- Disparo de uma transição numa rede L/T.

O conjunto  $\mathcal{PP}(R)$  de **seqüências de passos** é dado pela regra

$$\frac{u_R[\alpha_0 \rangle v_0, \dots, u_n[\alpha_n \rangle v_n \text{ em } \mathcal{P}(R) \text{ e } u_i = v_{i-1}, i = 1, \dots, n}{u_R[\alpha_0 \rangle [\alpha_1 \rangle \dots [\alpha_n \rangle v_n \text{ em } \mathcal{PP}(R)}$$

O conjunto de **marcações alcançáveis** de uma rede  $R$ ,  $\mathcal{A}(R)$ , é o conjunto das marcações que são destino de alguma seqüência de passos:

$$\mathcal{A}(R) = \{v | \exists (u_R[\alpha_0 \rangle [\alpha_1 \rangle \dots [\alpha_n \rangle v) \in \mathcal{PP}(R)\}$$

⊗

Portanto, uma seqüência de passos representa as transições que podem ser disparadas, iniciando-se com as transições habilitadas pela marcação inicial e um lugar alcançável é um lugar que pode ser atingido através dos disparo de transições, iniciando-se também na marcação inicial.

### 2.3 Redes Seguras

Um tipo especial de redes surge quando a marcação inicial e as pré- e pós-condições das transições de uma rede L/T forem conjuntos no lugar de multiconjuntos e todos os lugares forem acessíveis a partir da marcação inicial: as redes **seguras** que, como consequência, permitem apenas 1 *token* por vez em cada lugar. Estas redes serão usadas na definição de redes de ocorrência.

**DEFINIÇÃO 2.9. (Redes Seguras)** Uma rede  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  é **segura** se, e somente se as seguintes condições forem satisfeitas:

1.  $\forall t \in T_R, p_R(t)$  e  $d_R(t)$  são conjuntos;
2.  $\forall v \in \mathcal{A}(R), \forall a \in S_R, v(a) \leq 1$ , i.e.,  $v$  é um conjunto;
3.  $\forall a \in S_R, \exists v \in \mathcal{A}(R) | a \in \llbracket v \rrbracket$ , i.e., qualquer lugar ou pertence à marcação inicial ou é destino de alguma transição.

A categoria **Safe** é definida como a subcategoria completa de PTNets que tem as redes seguras como objetos.

⊗

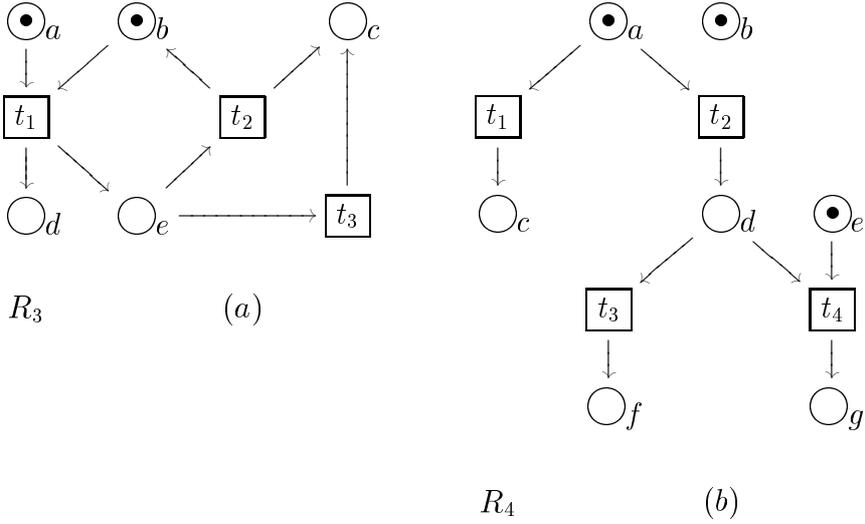


FIGURA 2.5- Exemplos de uma (a) rede segura e (b) rede de ocorrência.

Exemplo 2.9: A figura 2.5(a) mostra um exemplo de uma rede segura. Como exemplo de passos para esta rede, tem-se  $(a \oplus b \oplus c[t_1] \rangle d \oplus e \oplus c)$  e  $(a \oplus b[O_{T_3}] \rangle a \oplus b)$  (note que a origem e o destino do passo formado apenas pela transição nula,  $O_{T_3}$ , são sempre iguais, uma vez que esta transição possui origem e destino vazios). São seqüências de passos  $(a \oplus b[O_{T_3}] \rangle [t_1] \rangle d \oplus e)$  e  $(a \oplus b[t_1] \rangle [t_2] \rangle b \oplus c \oplus d)$ , por exemplo. O conjunto de marcações alcançáveis para esta rede é  $\mathcal{A}(R_3) = \{a \oplus b, d \oplus e, b \oplus c \oplus d, c \oplus d\}$ . ☺

## 2.4 Redes de Ocorrência

Redes de Ocorrência são redes seguras acíclicas sem conflitos (a serem definidos nesta seção) onde nenhum lugar da marcação inicial é destino de qualquer transição. O *unfolding* de uma rede L/T  $R$  possui estas características e, conseqüentemente, se  $R$  for uma rede de ocorrência, seu *unfolding* será isomórfico a  $R$ .

Estas redes irão constituir o domínio semântico para as redes L/T: suas transições representam possíveis disparos de transições da respectiva rede L/T e os lugares, lugares que são acessíveis a partir da marcação inicial.

De modo semelhante à pré- e pós-condição de uma transição, serão definidas as pré-transições de um lugar  $b$ , que são as transições que têm  $b$  como destino e as pós-transições, que são as transições que têm  $b$  como procedência.

DEFINIÇÃO 2.10. (**Pré-, Pós-Transição**) Sejam  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  uma rede L/T e  $a \in S_R$ . O conjunto das **pré-transições** do lugar  $a$  é definido como  $\mathfrak{P}(a) = \{t \in T_R | a \in \llbracket d_R(t) \rrbracket\}$ , e o conjunto das **pós-transições** do lugar  $a$  é definido como  $\mathfrak{D}(a) = \{t \in T_R | a \in \llbracket p_R(t) \rrbracket\}$ . ☺

As relações de dependência e de conflito, definidas a seguir, serão usadas para caracterizar uma rede de ocorrência.

DEFINIÇÃO 2.11. (**Relação de Dependência**) Sejam  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  uma rede L/T,  $\prec_1$  a relação definida por  $\prec_1 = \{(a, t) | a \in S_R, t \in T_R, t \in \mathfrak{D}(a)\} \cup \{(t, a) | a \in S_R, t \in T_R, t \in \mathfrak{P}(a)\}$  e  $\prec$  o fecho transitivo de  $\prec_1$ . A relação de dependência de  $R$ , denotada por  $\preceq$ , é definida como o fecho reflexivo de  $\prec$ . Se  $\exists a \in T_R \cup S_R | a \prec a$ , a rede  $R$  possui ciclos. ☺

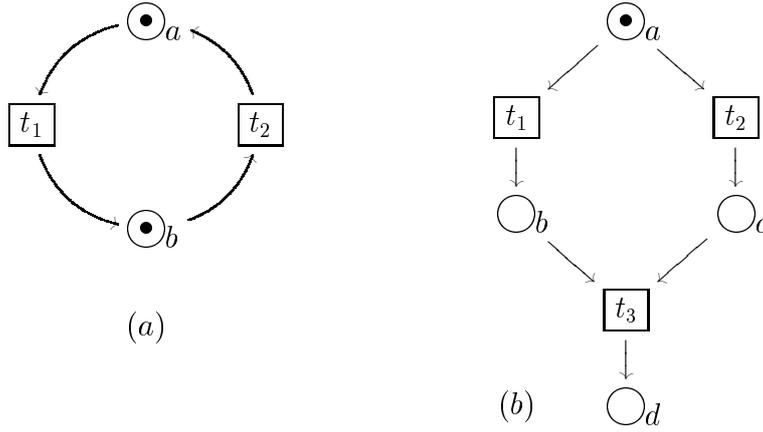


FIGURA 2.6- Exemplos de uma (a) rede com ciclo e (b) rede com autoconflito.

**Exemplo 2.10:** Para a rede da figura 2.6(a), a relação  $\prec_1$  é dada por  $\prec_1 = \{a \mapsto t_1, t_1 \mapsto a, b \mapsto t_2, t_2 \mapsto a\}$ . Seu fecho transitivo é  $\{a \mapsto b, t_1 \mapsto t_2, b \mapsto a, t_2 \mapsto t_1, a \mapsto a, b \mapsto b, t_1 \mapsto t_1, t_2 \mapsto t_2\}$ . Portanto,  $\prec$  não é irreflexiva, e a rede possui um ciclo. ☹

Chega-se a uma situação de **conflito** numa rede quando há possibilidade de indeterminismo, que é caracterizado quando um lugar é **origem** de mais de uma transição.

**DEFINIÇÃO 2.12. (Relação de Conflito)** Sejam  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  uma rede  $L/T$ ,  $\preceq$  o fecho reflexivo da relação  $\prec$  e  $\simeq$  a relação definida por  $\forall t_1, t_2 \in T_R, t_1 \simeq t_2 \Leftrightarrow \llbracket p_R(t_1) \rrbracket \cap \llbracket p_R(t_2) \rrbracket \neq \emptyset$  e  $t_1 \neq t_2$ . A relação de conflito no conjunto  $S_R \cup (T_R, O_T)$ , denotada por  $\#$ , é definida por

$$\forall x, y \in S_R \cup (T_R, O_T), x \# y \Leftrightarrow \exists t_1, t_2 \in T_R \text{ tais que } t_1 \simeq t_2 \text{ e } t_1 \preceq x \text{ e } t_2 \preceq y$$

Se a relação  $\#$  não for irreflexiva,  $R$  possui **autoconflitos**. ☹

**Exemplo 2.11:** Para a rede da figura 2.6(b), tem-se que  $\prec_1 = \{a \mapsto t_1, a \mapsto t_2, t_1 \mapsto b, b \mapsto t_3, t_2 \mapsto c\}$ . Seu fecho transitivo é  $\{a \mapsto b, a \mapsto c, t_1 \mapsto t_3, t_2 \mapsto t_3, b \mapsto d, c \mapsto d, a \mapsto b, a \mapsto t_3, a \mapsto d, a \mapsto c, t_1 \mapsto d, t_2 \mapsto d\}$ . Como  $t_1, t_2 \in T_R, t_1 \neq t_2$  e  $\llbracket p_R(t_1) \rrbracket \cap \llbracket p_R(t_2) \rrbracket = \{a\}$ , tem-se que  $t_1 \simeq t_2$ . Portanto,  $\# = \{t_1 \mapsto t_2, t_1 \mapsto c, t_1 \mapsto d, t_2 \mapsto b, t_2 \mapsto t_3, t_2 \mapsto d, b \mapsto c, b \mapsto t_3, b \mapsto c, c \mapsto t_3, c \mapsto d, t_3 \mapsto t_3, t_3 \mapsto d, d \mapsto d\}$  e os pares simétricos, uma vez que  $\#$  é simétrica. Neste caso, a relação  $\#$  não é irreflexiva, uma vez que  $d \# d$  e  $t_3 \# t_3$  e, portanto, a rede possui autoconflito. ☹

**DEFINIÇÃO 2.13. (Redes de Ocorrência)** Uma rede de ocorrência é uma rede segura  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  tal que:

1.  $a \in \llbracket u_R \rrbracket \Leftrightarrow \mathfrak{P}(a) = \emptyset$ ;
2.  $\forall a \in S_R, Cd(\mathfrak{P}(a)) \leq 1$ ;
3. a relação  $\prec$  é irreflexiva e,  $\forall t \in T_R$ , o conjunto  $\{t' \in T_R \mid t' \prec t\}$  é finito para todo  $t \in T_R$ .
4. a relação de conflito  $\#$  no conjunto  $T_R \cup S_R$  é irreflexiva.

A categoria **OccNets** é definida como a subcategoria completa de **Safe** que tem como objetos as redes de ocorrência. ☹

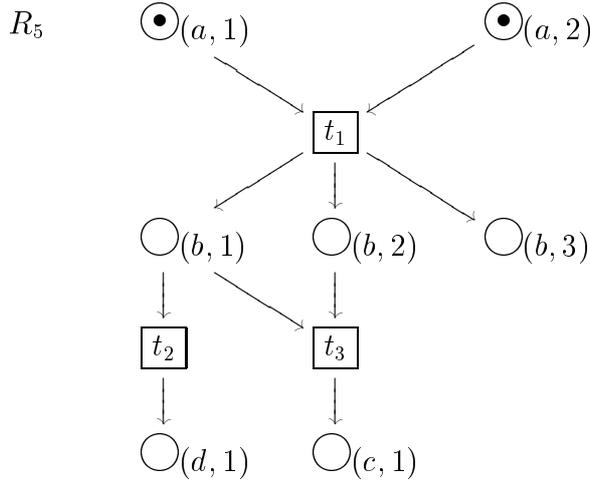


FIGURA 2.7- Exemplo de uma rede de ocorrência decorada.

O item 1 da definição proíbe que lugares pertencentes à marcação inicial sejam destino de transições. O item 2 proíbe que um lugar seja destino de mais de uma transição. O item 3 proíbe ciclos e causas infinitas e o item 4 proíbe autoconflitos.

Exemplo 2.12: A rede da figura 2.5(b) é um exemplo de uma rede de ocorrência: os lugares da marcação inicial não pertencem ao destino de qualquer transição (condição (1) da definição) e as relações  $\prec$  e  $\#$  são irreflexivas (a rede não possui ciclos nem autoconflitos). ☺

A seguir, serão definidas as redes de ocorrência decoradas, que serão usadas para representar os *unfoldings* de redes L/T, levando em consideração as multiplicidades dos lugares. São redes de ocorrência nas quais o conjunto de lugares pertencentes à pós-condição de uma mesma transição é particionado em famílias, que serão usadas para representar multiplicidade dos *tokens*.

DEFINIÇÃO 2.14. (**Rede de Ocorrência Decorada**) *Uma rede de ocorrência decorada é uma rede de ocorrência  $R$  que satisfaz as seguintes condições:*

1.  $S_R$  é da forma  $\bigcup_{a \in A_R} \{a\} \times \{1, \dots, n_a\}$  para algum conjunto  $A_R$ ,  $n_a \in \mathbb{N}$  e o conjunto  $\{a\} \times \{1, \dots, n_a\}$  é chamado de **família** de  $a$  e ;
2.  $\forall a \in A_R, \forall x, y \in \{a\} \times \{1, \dots, n_a\}, \mathfrak{P}(x) = \mathfrak{P}(y)$ .

A categoria  $\text{DecOcc}$  é definida como a subcategoria completa de  $\text{OccNets}$  que tem como objetos as redes de ocorrência decoradas. ☺

Exemplo 2.13: A rede  $R_5$  da figura 2.7 é uma rede de ocorrência decorada, onde  $A_R = \{a, b, c, d\}$ . Para todo  $x \in \{(a, 1), (a, 2)\}$  (família de  $a$ ),  $\mathfrak{P}(x) = \emptyset$ , uma vez que fazem parte da marcação inicial ( $R_5$  é uma rede de ocorrência) e, para todo  $x \in \{(b, 1), (b, 2), (b, 3)\}$ ,  $\mathfrak{P}(x) = \{t_1\}$ , i.e., os elementos de uma mesma família possuem as mesmas pré-transições. ☺

O funtor  $\mathbb{F}$ , definido a seguir, relaciona redes de ocorrência com as redes L/T, “juntando” os elementos de uma mesma família. Isto permitirá relacionar a categoria semântica com as redes L/T (lembre que  $\text{DecOcc}$  é uma subcategoria de  $\text{OccNets}$ ).

DEFINIÇÃO 2.15. (**Functor  $\mathbb{F}$** ) *Sejam  $R$  uma rede de ocorrência decorada, onde  $S_R = \bigcup_{a \in A_R} \{a\} \times \{1, \dots, n_a\}$  e  $h : S_R^\oplus \rightarrow A^\oplus$  um morfismo tal que  $h(a, j) = a$ . O funtor  $\mathbb{F} : \text{DecOcc} \rightarrow \text{OccNets}$  é definido por:*

**Objetos:**  $\mathbb{F}(R)$  é a rede  $(h \circ p, h \circ d : (T_R, O_T) \rightarrow A^\oplus, h(u_R))$ ;

**Morfismos:** dado um morfismo  $(f, g) : R \rightarrow R_1$ , o morfismo  $\mathbb{F}(f, g) : \mathbb{F}(R) \rightarrow \mathbb{F}(R_1)$  é definido por  $(f, h \circ g \circ \rho)$ , onde  $\rho : A^\oplus \rightarrow S_R^\oplus$  é o morfismo tal que  $\rho(a) = (a, 1)$ .

⊖

PROPOSIÇÃO 1. O funtor  $\mathbb{F} : \text{DecOcc} \rightarrow \text{PTNets}$  está bem definido.

## 2.5 Unfolding de Redes Lugar/Transição

Os *unfoldings* fornecem uma semântica concorrente para as redes L/T. Esta semântica é caracterizada pela escolha de explicar o comportamento da rede em termos das noções de *evento* (disparo de transições), *causa* (uma transição habilita outras transições) e *conflito* (exclusão mútua). Esta escolha tem a vantagem de fornecer uma explicação totalmente causal das computações numa rede, i.e., o disparo de uma transição causa o disparo de uma outra.

O *unfolding* de uma rede L/T  $R$  é definido como o colimite dos *unfoldings* finitos de  $R$  na categoria DecOcc. O *unfolding* finito é definido por regras de inferência e é construído indutivamente: o primeiro *unfolding* (*unfolding* de profundidade zero) representa a marcação inicial de  $R$  expandida em famílias; o  $k$ -ésimo *unfolding* (*unfolding* de profundidade  $k$ ) é obtido disparando-se todas as possíveis transições de  $R$  no  $(k-1)$ -ésimo *unfolding*. Isto implica que uma rede com ciclos pode ter *unfolding* infinito, dependendo da marcação inicial, possuindo várias instâncias do disparo de uma mesma transição. Os pares conflitantes também são representados no *unfolding*, i.e., se duas transições estão habilitadas, as duas são representadas, de modo que cada transição no *unfolding* corresponde a um possível disparo na rede  $R$ .

As multiplicidades dos *tokens* são representadas por famílias, de maneira que cada lugar do *unfolding* corresponde a um *token* de um lugar alcançável da rede  $R$ .

A seguir, serão definidos **elementos concorrentes**, que identificam quais lugares se pode usar como pré-condição para as transições dos *unfoldings* finitos. São constituídos de lugares alcançáveis que não estão em conflito e não são interdependentes.

DEFINIÇÃO 2.16. (**Elementos Concorrentes**) Dada uma rede de ocorrência  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ , são definidos:

- para  $x, y \in T_R \cup S_R$ ,  $x$  co  $y$  se  $(x \prec y$  ou  $y \prec x$  ou  $x \# y)$  é falso;
- para  $X \subseteq T_R \cup S_R$ ,  $\text{Co}(X)$  se

$\forall x, y \in X$ ,  $x$  co  $y$  e o conjunto  $\{t \in T_R \mid \exists z \in X, t \preceq z\}$  é finito.

⊖

DEFINIÇÃO 2.17. (**Unfolding Finito de Redes L/T**) Seja uma rede L/T  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ . O *unfolding* (finito) de profundidade  $k$  da rede  $R$ ,  $\mathcal{U}_R^k(R) = (p_k, d_k : (T_k, O_T) \rightarrow S_k^\oplus, u_k)$ , é definido para todo  $k \in \mathbb{N}$ , como:

$k = 0$  :

- $S_0 = \bigcup \{ \{(\emptyset, b)\} \times \{1, \dots, n\} \mid u_R(b) = n \}$ ;
- $T_0 = \{O_T\}$
- $u_0 = \bigoplus S_0$ .
- $p_0(O_T) = d_0(O_T) = O_{S_0}$ .

$k > 0$  ( $J$  é um conjunto de índices):

1. 
$$\frac{B = \{((x_j, b_j), i_j) \mid j \in J\} \subseteq S_{k-1}, Co(B), t \in T_R, \bigoplus_{j \in J} b_j = p_R(t)}{(B, t) \in T_k \text{ e } p_k(B, t) = \bigoplus B}$$
2. 
$$\frac{t_0 = (B, t) \in T_k, d_R(t) = \bigoplus_{j \in J} n_j b_j}{\{(\{t_0\}, b_j)\} \times \{1, \dots, n_j\} \subseteq S_k, \forall j \in J, \text{ e } d_k(t_0) = \bigoplus_{j, i \leq n_j} (\{t_0\}, b_j), i)}$$
3.  $u_k = \bigoplus S_0 = u_0.$

Para cada  $k \in \mathbb{N}$ , há uma inclusão  $in_k : \mathcal{U}_R^k(R) \rightarrow \mathcal{U}_R^{k+1}(R)$ . ☺

O objetivo da premissa da regra de inferência 1 é gerar todos os possíveis subconjuntos do conjunto de lugares e, através do predicado  $Co$ , eliminar aqueles que levam a conflito ou que já foram “usados” num passo anterior. Se este conjunto puder ser usado para disparar alguma transição de  $R$ , então uma transição é gerada na rede de ocorrência. A regra de inferência 2 gera as pós-condições para as transições geradas.

O  $k$ -ésimo *unfolding* de uma rede  $R$  reflete o comportamento de  $R$  até seqüências de passos de tamanho máximo  $k$  (i.e., corresponde ao disparo de no máximo  $k$  transições). A finalidade da definição é gerar todas as possíveis instâncias de lugares alcançáveis e todos os possíveis disparos de transições de  $R$ , “decorando-os” com sua história. As famílias de lugares em  $\mathcal{U}_R^k(R)$  representam lugares com a mesma história e são representados por pares  $(A, b)$  onde  $b \in S_R$  e  $A$  é um conjunto contendo a história de  $b$ . Daí, os lugares serão da forma  $((A, b), i)$ , onde  $i \in \mathbb{N}$  (para representar a multiplicidade de  $b$ ). De modo análogo, as transições serão pares  $(B, t)$ , onde  $t \in T_R$  e  $B$  representa a história de  $t$ .

A rede  $\mathcal{U}_R^0(R)$  é obtida formando famílias a partir da marcação inicial de  $R$ .  $\mathcal{U}_R^{k+1}(R)$  é obtida, indutivamente, gerando uma nova transição para cada subconjunto possível de lugares de  $\mathcal{U}_R^k(R)$  cujo multiconjunto dos correspondentes lugares em  $R$  constituem a origem de alguma transição  $t$  de  $R$ . As famílias formadas a partir do destino de  $t$  são também incluídas em  $\mathcal{U}_R^{k+1}(R)$ .

**PROPOSIÇÃO 2.** *Dada uma rede  $L/T R$ ,  $\mathcal{U}_R^k(R)$  é uma rede de ocorrência decorada (objeto de DecOcc).*

**DEFINIÇÃO 2.18. (Semântica de Redes L/T)** *O unfolding  $Unf^R(R)$  de uma rede  $L/T R$  é definido como sendo o objeto colimite do diagrama  $D : \mathbb{N} \rightarrow \text{DecOcc}$  tal que  $D(n) = \mathcal{U}_R^n(R)$  e  $D(n \rightarrow n+1) = in_n$ . A **semântica** de uma rede  $L/T R$  é o seu unfolding  $Unf^R(R)$ .* ☺

**PROPOSIÇÃO 3.** *O funtor  $Unf^R : \text{PTNets} \rightarrow \text{DecOcc}$  está bem definido, com o mapeamento dos morfismos induzido pela inclusão  $in_n$ .*

Pela definição, redes acíclicas finitas terão *unfoldings* finitos, i.e.,  $Unf^R(R)$  é uma rede finita se  $R$  for acíclica e finita.

**Exemplo 2.14:** O *unfolding*  $U^3(R_0)$  da rede  $R_0$  da figura 2.1, é obtido da seguinte maneira:

$\mathcal{U}_R^0(R_0)$ : esta rede é formada apenas por dois lugares, com um *token* em cada lugar, representando a expansão da marcação inicial em famílias:  $a_1 = ((\emptyset, a), 1)$  e  $a_2 = ((\emptyset, a), 2)$  (o conjunto vazio indica que o respectivo lugar não possui história).

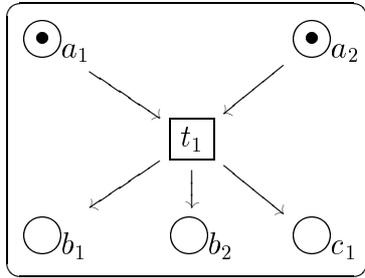
- $S_0 = \{a_1, a_2\}$
- $(T_0, O_T) = \{O_T\}$
- $u_0 = a_1 \oplus a_2$

$$\mathcal{U}_R^0(R_0) : \boxed{\begin{array}{cc} \odot_{a_1} & \odot_{a_2} \end{array}}$$

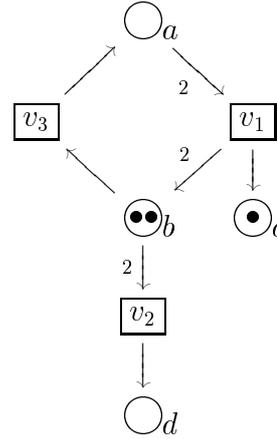
$\mathcal{U}_R^1(R_0)$ : O único subconjunto de  $S_0$  tais que seus elementos podem formar pré-condições de transições de  $R$  é  $B_1 = \{a_1, a_2\}$ , pois  $2a$  é a origem da transição  $v_1$ .  $Co(B_1)$  é verdadeiro, uma vez que ainda não há transições na rede  $\mathcal{U}_R^0(R_0)$ . Portanto,  $t_1 = (B_1, v_1) \in T_1$  e  $p_1(t_1) = a_1 \oplus a_2$ . Como  $d_R(v_1) = 2b \oplus c$ , tem-se que  $A_1 = \{b_1, b_2, c_1\} \subseteq S_1$  e  $d_1(t_1) = b_1 \oplus b_2 \oplus c_1$ , onde  $b_1 = ((t_1, b), 1)$ ,  $b_2 = ((t_1, b), 2)$  e  $c_1 = ((t_1, c), 1)$ . A história destes lugares é  $(B, t)$ , que significa que estes lugares foram gerados através dos disparos da transição  $t$ , com a pré-condição (e sua história) representada em  $B$ . Esta rede representa o disparo da transição  $v_1$  da rede  $R_0$ .

- $S_1 = \{a_1, a_2, b_1, b_2, c_1\}$
- $(T_1, O_T) = \{O_T, t_1\}$
- $u_1 = a_1 \oplus a_2$

$\mathcal{U}_R^1(R_0)$  :



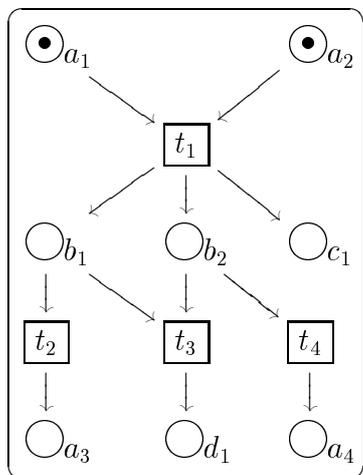
disparo da transição  $v_1$ :



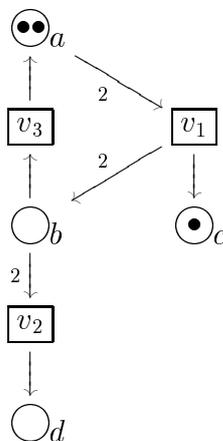
$\mathcal{U}_R^2(R_0)$ : Subconjuntos possíveis de  $S_1$  que formam elementos concorrentes são  $B_2 = \{b_1\}$  e  $B_3 = \{b_2\}$  que podem ser usados para o disparo da transição  $v_3$ . Portanto,  $t_1 = (B_2, v_3)$  e  $t_2 = (B_3, v_3) \in T_2$ . Como  $d_R(v_3) = a$ , tem-se os conjuntos  $A_2 = \{a_3\} \subseteq S_2$ ,  $A_3 = \{a_4\} \subseteq S_2$ ,  $d_2(t_1) = a_2$  e  $d_2(t_2) = a_4$ , onde  $a_3 = ((\{t_1\}, a), 1)$  e  $a_4 = ((\{t_2\}, a), 1)$ . Outro subconjunto possível de  $S_1$  que possui elementos concorrentes é  $B_4 = \{b_1, b_2\}$ , que pode ser usado para o disparo da transição  $v_2$ :  $t_3 = (B_4, v_2) \in T_2$  e, como  $d_R(v_2) = d$ , tem-se que  $A_4 = \{d_1\} \subseteq S_2$  e  $d_2(t_3) = d_1$ , onde  $d_1 = ((\{t_3\}, d), t)$ . Esta situação representa o disparo da transição  $v_2$  e aos dois disparos da transição  $v_3$  na rede  $R_0$ . Note-se que estas duas transições ( $v_2$  e  $v_3$ ) estão habilitadas e não podem ser disparadas ao mesmo tempo em  $R_0$  e ambas são representadas no *unfolding* (representação explícita do não-determinismo).

- $S_2 = \{a_1, a_2, b_1, b_2, c_1, a_3, a_4, d_1\}$
- $(T_2, O_T) = \{O_T, t_1, t_2, t_3, t_4\}$
- $u_2 = a_1 \oplus a_2$

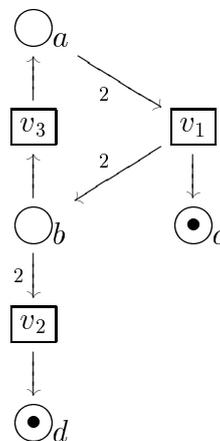
$\mathcal{U}_R^2(R_0)$



2 disparos da transição  $v_3$ :



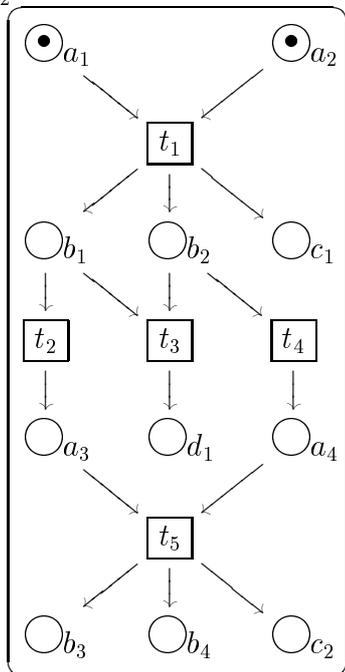
disparo da transição  $v_2$ :



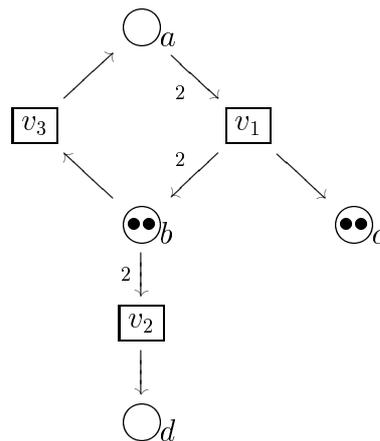
$\mathcal{U}_R^3(R_0)$ : O único subconjunto possível de  $S_2$  que possui elementos concorrentes é  $B_5 = \{a_3, a_4\}$  pois  $p_R(v_1) = 2a$ . Neste ponto, surge uma situação semelhante a  $\mathcal{U}_R^1(R_0)$ , ou seja, volta-se a disparar  $v_1$  (ciclo): o *unfolding* se desdobra indefinidamente, “repetindo-se” a rede  $\mathcal{U}_R^2(R_0)$ .

- $S_3 = \{a_1, a_2, b_1, b_2, c_1, a_3, a_4, d_1, b_3, b_4, c_2\}$
- $(T_3, O_T) = \{O_T, t_1, t_2, t_3, t_4, t_5\}$
- $u_3 = a_1 \oplus a_2$

$\mathcal{U}_R^3(R_0)$  :



disparo da transição  $v_1$ :



### 3 Gramáticas de Grafos

As gramáticas de grafos surgiram a partir das gramáticas de Chomsky pelo uso de grafos no lugar de *strings*, i.e., um grafo representa o símbolo inicial (símbolo inicial da gramática) e os lados esquerdo e direito das produções (regras da gramática) também são grafos. Analogamente à aplicação de regras nas gramáticas de Chomsky, uma regra em gramáticas de grafos pode ser aplicada num grafo-estado se houver uma ocorrência (definida em termos de subgrafo) do lado esquerdo da regra no grafo-estado.

A semântica operacional (comportamento) de um sistema descrito por uma gramática de grafos é dada pela aplicação das regras da gramática a grafos-estado (inciando-se no estado inicial). A aplicação de uma regra a um grafo-estado, chamada derivação direta, só é possível se houver uma ocorrência do lado esquerdo da regra no grafo-estado. Esta ocorrência será definida como um subgrafo do grafo-estado isomórfico ao lado esquerdo da regra.

Neste capítulo, será definida uma semântica para gramáticas de grafos que descreve todos os possíveis estados e mudanças de estado e que também é uma gramática de grafos. Tais gramáticas são as gramáticas de grafos de ocorrência, que são gramáticas que possuem dois grafos como tipo. Um destes grafos é chamado **grafo-núcleo** e será interpretado como possíveis grafos-estado (e as regras como possíveis derivações) de uma gramática de grafos da categoria  $GG$  (i.e., nestas gramáticas de ocorrência são representados os possíveis grafos-estado e as derivações necessárias para se chegar a tais grafos-estado). Para isto, serão definidas as gramáticas de grafos duplamente tipadas, que são gramáticas que possuem dois tipos e, a partir daí, serão definidas as gramáticas de grafos de ocorrência como sendo gramáticas de grafos duplamente tipadas que possuem algumas propriedades especiais (são acíclicas e livres de conflitos).

Uma gramática de grafos de ocorrência será então usada para representar a semântica de uma gramática de grafos tipada, i.e., todas as computações (derivações) possíveis serão representadas. Isto será feito através do *unfolding*, que é definido como o colimite do diagrama formado pelos *unfoldings* finitos de uma gramática de grafos  $GG$ . O *unfolding* finito é construído indutivamente a partir do grafo inicial de  $GG$ : o *unfolding* zero é uma gramática de grafos de ocorrência onde o grafo-núcleo é o grafo inicial de  $GG$  e o  $k$ -ésimo *unfolding* é construído aplicando-se todas as regras de  $GG$  que podem ser aplicadas ao grafo-núcleo do  $(k - 1)$ -ésimo *unfolding*. Desta maneira, o *unfolding* representará todas as possíveis derivações de  $GG$ .

Todas as definições e conceitos deste capítulo foram retirados de [EHR 96], [COR 96] e [RIB 96b], onde podem ser encontradas as provas para as proposições e teoremas apresentados.

### Estrutura do capítulo:

- seção 1: é apresentada a definição dos grafos que serão usados neste trabalho: são grafos dirigidos onde todos os vértices possuem um destino e uma origem. A seguir, é definida a categoria **GraphP**, dos grafos e morfismos parciais entre grafos. Os morfismos são definidos de modo a preservarem uma compatibilidade estrutural entre os grafos;
- seção 2: são definidos os grafos tipados, que são grafos onde seu tipo é também um grafo. Em seguida, é definido o funtor de retipagem, que pode ser visto como uma maneira de transformar um grafo com um certo tipo num outro grafo com outro tipo;
- seção 3: é apresentado o conceito das regras que serão usadas nas gramáticas de grafos a serem definidas: um morfismo parcial entre grafos tipados. Em seguida, o funtor de retipagem é usado para transformar uma regra com um certo tipo numa outra regra com outro tipo;
- seção 4: a categoria de gramáticas de grafos **GG** é definida: os objetos são gramáticas de grafos que possuem um grafo  $T$  como tipo, um grafo inicial tipado com tipo  $T$  e as regras também com tipo  $T$ . Os morfismos são definidos em função do funtor de retipagem, de modo a preservar uma compatibilidade estrutural e semântica;
- seção 5: a semântica operacional das gramáticas de grafos é apresentada, sendo definida em termos da aplicação de regras a grafos-estado, iniciando-se no grafo inicial.
- seções 6, 7 e 8: são definidos os mesmos conceitos das seções 1, 2 e 3, porém para o caso em que o grafo-tipo é um grafo tipado;
- seção 9: é definido o conceito de grafo-núcleo e são definidas as relações de ocorrência, conflito, dependência e conflito fraco para uma gramática de grafos com tipo duplo que possui grafo-núcleo;
- seção 10: as gramáticas de grafos de ocorrência são definidas usando as relações da seção 9. Estas gramáticas são livres de ciclos e autoconflitos;
- seção 11: a semântica de uma gramática de grafos é definida por meio de *unfoldings*. As gramáticas de grafos abstratas são definidas de modo que a semântica constitua um funtor.

### 3.1 Grafos

Além da representação gráfica de grafos, pode-se representá-los algebricamente por meio de conjuntos e funções: um grafo  $G$  é formado por um conjunto  $V$  de vértices e um conjunto  $A$  de arcos conectando os vértices. As conexões são representadas por duas funções totais  $p, d : A \rightarrow V$  que atribuem a cada arco uma **origem** e um **destino**, respectivamente.

**DEFINIÇÃO 3.1. (Grafo)** Um grafo  $G = (V_G, A_G, p_G, d_G)$  consiste de um conjunto de vértices  $V_G$ , um conjunto de arcos  $A_G$ , e duas funções totais  $p_G, d_G : A_G \rightarrow V_G$ , chamadas **origem** e **destino** dos arcos, respectivamente. Um **item**  $x \in G$  denota um elemento de  $V_G \cup A_G$ . Um grafo é finito se  $V_G$  e  $A_G$  forem ambos finitos.  $\odot$

**Exemplo 3.1:** A figura 3.1 mostra dois grafos  $G$  e  $H$  e suas representações gráficas e algébricas. Neste trabalho, quase sempre haverá mais de um grafo numa mesma figura. Por isso, serão sempre delimitados por um  $\square$ .  $\odot$

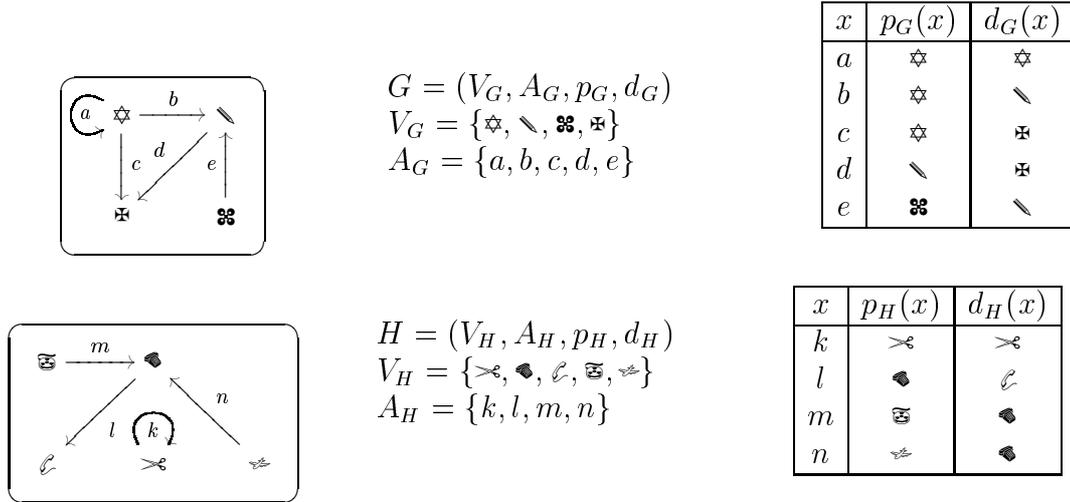


FIGURA 3.1- Exemplos de representação algébrica de grafos.

Como as gramáticas de grafos serão construídas a partir de grafos, os morfismos acabarão por se resumir em morfismos entre grafos. Um morfismo  $f : G \rightarrow H$  entre os grafos  $G$  e  $H$  expressa uma compatibilidade estrutural entre os grafos e pode ser parcial ou total. Estes morfismos terão dois componentes: uma função  $f_V$  que mapeia os vértices de  $G$  para os vértices de  $H$  e uma função  $f_A$  que mapeia os arcos de  $G$  para os arcos de  $H$ . A compatibilidade estrutural se resume na restrição de que todo arco que é mapeado por  $f_A$  precisa ser compatível com o mapeamento de sua origem e seu destino por  $f_V$ , i.e., se  $p_G(a) = \succ$  e  $d_G(a) = \ell$ , então  $p_H(f_A(a)) = f_V(\succ)$  e  $d_H(f_A(a)) = f_V(\ell)$ . O diagrama abaixo resume esta compatibilidade.

$$\begin{array}{ccccc}
 \succ & \xleftarrow{p_G} & a & \xrightarrow{d_G} & \ell \\
 f_V \downarrow & & \downarrow f_A & & \downarrow f_V \\
 f_V(\succ) & \xleftarrow{p_H} & f_A(a) & \xrightarrow{d_H} & f_V(\ell)
 \end{array}$$

Como o morfismo  $f$  pode ser parcial, a compatibilidade é exigida apenas nos elementos mapeados pelos morfismos  $f_V$  e  $f_A$ . A **comutatividade fraca**, definida a seguir, será usada para representar esta compatibilidade e parcialidade dos morfismos.

**DEFINIÇÃO 3.2. (Comutatividade Fraca)** *Sejam  $f : A \rightarrow B$  uma função parcial com domínio  $dom(f)$ ,  $f'$  uma função parcial e  $a$  e  $b$  funções totais. Para o diagrama abaixo, será usada a notação  $f' \circ a \geq b \circ f$ . Este diagrama **comuta fracamente** se, e somente se,  $f' \circ a \circ f^* = b \circ f!$ . ( $f^* : dom(f) \rightarrow A$  é a inclusão do domínio de  $f$  e  $f! : dom(f) \rightarrow B$  é a restrição ao domínio de  $f$ .)*

$$\begin{array}{ccc}
 \begin{array}{ccc}
 A & \xleftarrow{f^*} & dom(f) & \xrightarrow{f!} & B \\
 a \downarrow & & & & \downarrow b \\
 A' & \xrightarrow{f'} & & & B'
 \end{array} & = & \begin{array}{ccc}
 A & \xrightarrow{f} & B \\
 a \downarrow & \geq & \downarrow b \\
 A' & \xrightarrow{f'} & B'
 \end{array}
 \end{array}
 \iff$$

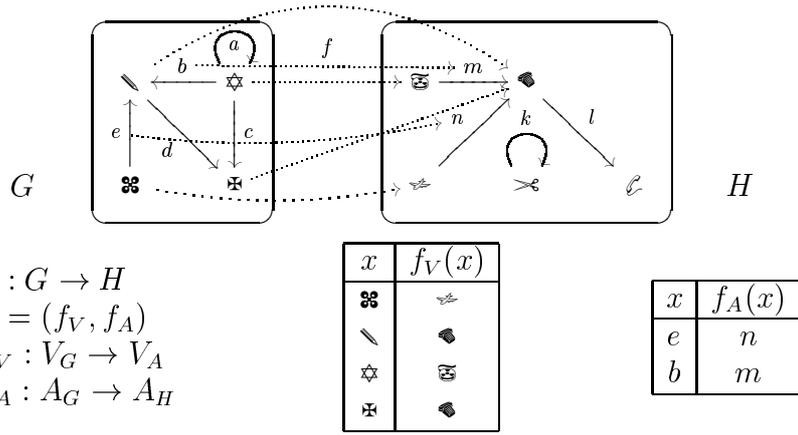


FIGURA 3.2- Exemplo de um morfismo entre os grafos  $G$  e  $H$ .

DEFINIÇÃO 3.3. (**Morfismo entre Grafos**) Um morfismo (parcial)  $g : G \rightarrow H$  de um grafo  $G$  para um grafo  $H$  é um par  $g = (g_V, g_A)$  consistindo de duas funções parciais  $g_V : V_G \rightarrow V_H$  e  $g_A : A_G \rightarrow A_H$  tais que os diagramas abaixo comutem fracamente:

$$\begin{array}{ccc}
 A_G & \xrightarrow{g_A} & A_H \\
 p^G \downarrow & \geq & \downarrow p^H \\
 V_G & \xrightarrow{g_V} & V_H
 \end{array}
 \qquad
 \begin{array}{ccc}
 A_G & \xrightarrow{g_A} & A_H \\
 d^G \downarrow & \geq & \downarrow d^H \\
 V_G & \xrightarrow{g_V} & V_H
 \end{array}$$

Um morfismo  $g$  é chamado total, injetor, sobrejetor, de inclusão ou vazio se ambos os componentes forem funções totais, injetoras, sobrejetoras, de inclusão ou vazias, respectivamente. Um grafo  $G$  é um subgrafo de  $H$  ( $G \subseteq H$ ) se houver uma inclusão  $g : G \rightarrow H$ .

A categoria de grafos e morfismos parciais de grafos é denotada por GraphP. A subcategoria de GraphP consistindo de todos os grafos e morfismos totais de grafos é denotada por Graph. ☺

Exemplo 3.2: Na figura 3.2, há um morfismo  $f$  entre os grafos  $G$  e  $H$ . Como se pode ver, o morfismo é parcial e estabelece uma compatibilidade estrutural entre os grafos. As funções parciais  $f_V$  e  $f_A$  satisfazem a comutatividade fraca, i.e,  $dom(f_A) = \{e, b\}$  e  $f_A \circ p_H \geq p_G \circ f_V$ :

- $p_H(f_A(e)) = p_H(n) = \text{⬢} = f_V(\text{☼}) = f_V(p_G(e))$
- $d_H(f_A(e)) = d_H(n) = \text{⬤} = f_V(\text{↘}) = f_V(d_G(e))$
- $p_H(f_A(b)) = p_H(m) = \text{⊞} = f_V(\text{⊛}) = f_V(p_G(b))$
- $d_H(f_A(b)) = d_H(m) = \text{⬤} = f_V(\text{↘}) = f_V(d_G(b))$

☺

### 3.2 Grafos Tipados

Um grafo tipado é um grafo que possui um outro grafo como tipo, i.e., cada elemento (arco ou vértice) possui um tipo. Neste trabalho, serão usadas gramáticas de grafos nas quais o grafo-estado é um grafo tipado e as regras são descritas por morfismos entre grafos tipados onde o tipo, chamado grafo-tipo, também é um grafo.

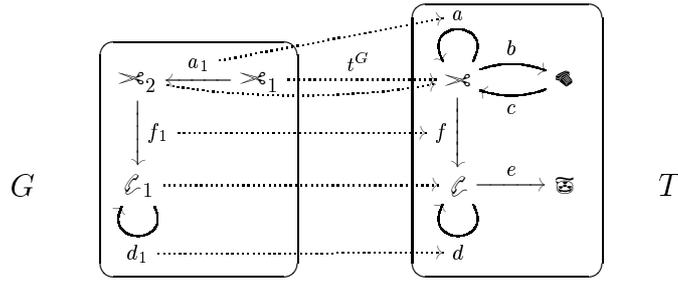


FIGURA 3.3- Grafo tipado  $(G, t^G, T)$ .

O uso de grafos tipados tem a vantagem de que algumas inconsistências nos estados de um sistema são eliminadas pela **compatibilidade de tipos**. Por exemplo, se não houver um arco entre dois vértices  $\star$  e  $\star$  no grafo-tipo, não poderá haver na gramática de grafos arcos conectando vértices que têm  $\star$  e  $\star$  como tipos.

**DEFINIÇÃO 3.4. (Grafo Tipado)** Um grafo tipado  $G^T$  é uma tripla  $G^T = (G, t^G, T)$  onde  $G$  e  $T$  são grafos e  $t^G : G \rightarrow T$  é um morfismo total entre grafos.  $T$  é chamado **grafo-tipo** de  $G^T$ . ☺

**Exemplo 3.3:** Na figura 3.3, há o grafo tipado  $G^T = (G, t^G, T)$ , onde  $T$  é o grafo-tipo e  $t^G$  é um morfismo de tipo (morfismo total entre os grafos  $G$  e  $T$ ). Como na maioria das vezes serão dados grafos tipados neste trabalho, para facilitar a representação do morfismo de tipo, será adotada a convenção de que os itens do grafo-tipo não terão índices, os itens dos grafos terão índices e seu tipo será o item sem índices. Isto elimina a necessidade de representação gráfica (setas pontilhadas) do morfismo de tipo. ☺

**DEFINIÇÃO 3.5. (Morfismo entre Grafos Tipados)** Um morfismo entre grafos tipados  $g^t : G^{T_1} \rightarrow H^{T_2}$  é um par de morfismos  $g^t = (g, t)$  com  $g : G \rightarrow H$  e  $t : T_1 \rightarrow T_2$  tal que o diagrama abaixo comute fracamente em GraphP, i.e.,  $t \circ t^G \circ g^* = t^H \circ g!$ .

$$\begin{array}{ccc} G & \xrightarrow{g} & H \\ t^G \downarrow & \geq & \downarrow t^H \\ T_1 & \xrightarrow{t} & T_2 \end{array}$$

Um **automorfismo** entre grafos tipados é um isomorfismo  $f^t : G^T \rightarrow G^T$  que seja diferente da identidade em  $G^T$ .

A categoria dos grafos tipados e dos morfismos definidos acima é denotada por TGraphP. Fixando um grafo-tipo  $T$ , a subcategoria TGraphP( $T$ ) de TGraphP tem como objetos todos os grafos tipados com tipo  $T$  e como morfismos todos os morfismos de TGraphP nos quais o componente de tipo é a função identidade. A restrição das categorias acima a morfismos totais são denotadas por TGraph e TGraph( $T$ ), respectivamente. ☺

**Exemplo 3.4:** O morfismo  $g^t = (g, t)$  entre os grafos  $G^{T_0} = (G, t_0, T_0)$  e  $H^{T_1} = (H, t_1, T_1)$  da figura 3.4 é um morfismo entre grafos tipados. Note-se que os morfismos  $g$  e  $t$  obedecem à compatibilidade estrutural. ☺

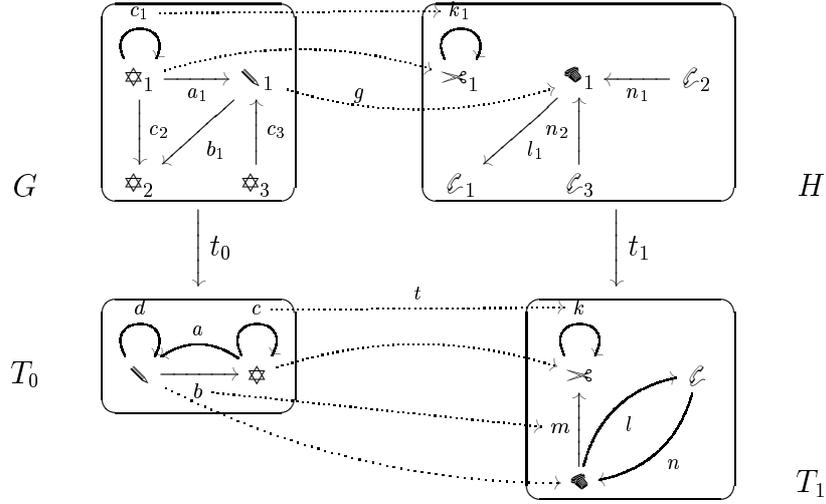


FIGURA 3.4- Exemplo de um morfismo entre os grafos tipados  $G^{T_0}$  e  $H^{T_1}$ .

Na definição de morfismos entre gramáticas de grafos, serão relacionadas gramáticas de grafos com tipos diferentes. Para esta finalidade, será definido como transformar um morfismo entre grafos tipados num outro morfismo entre grafos tipados onde os grafos-tipo sejam diferentes. A transformação de um grafo tipado  $G_0^{T_0}$  em  $G_1^{T_1}$  será definida em termos de um morfismo  $f : T_1 \rightarrow T_0$  entre os grafos-tipo  $T_1$  e  $T_0$ . A transformação será feita de tal modo que  $G_1$  seja (isomórfico a) o maior subgrafo de  $G_0$  cujos elementos possuem como tipo elementos da imagem de  $f$ .

A definição de retipagem é feita em termos da construção categórica *pullback* na categoria  $\text{GraphP}$ . Esta propriedade assegura a preservação de derivações (uma derivação é a aplicação de uma regra a um grafo-estado) através dos morfismos entre gramáticas de grafos (compatibilidade semântica). Como os *pullbacks* são únicos a menos de isomorfismo, é preciso um resultado concreto para definir uma construção determinística. A definição a seguir formaliza esta idéia.

**DEFINIÇÃO 3.6. (Escolha de Pullback)** Uma escolha de *pullback* numa categoria  $\text{Cat}$  que tenha *pullbacks* é um *pullback* fixo  $\overline{PB}(G_3 \xrightarrow{g} G_1 \xleftarrow{f} G_2) = (G_3 \xleftarrow{f'} G_4 \xrightarrow{g'} G_2)$  para cada diagrama  $(G_3 \xrightarrow{g} G_1 \xleftarrow{f} G_2)$  em  $\text{Cat}$ .  $\odot$

**DEFINIÇÃO 3.7. (Functor de Retipagem)** Seja  $f : T_2 \rightarrow T_1$  um morfismo em  $\text{GraphP}$ . Então há um functor  $\mathcal{T}_f : \text{TGraphP}(T_1) \rightarrow \text{TGraphP}(T_2)$ , chamado **functor de retipagem**, induzido por  $f$ .

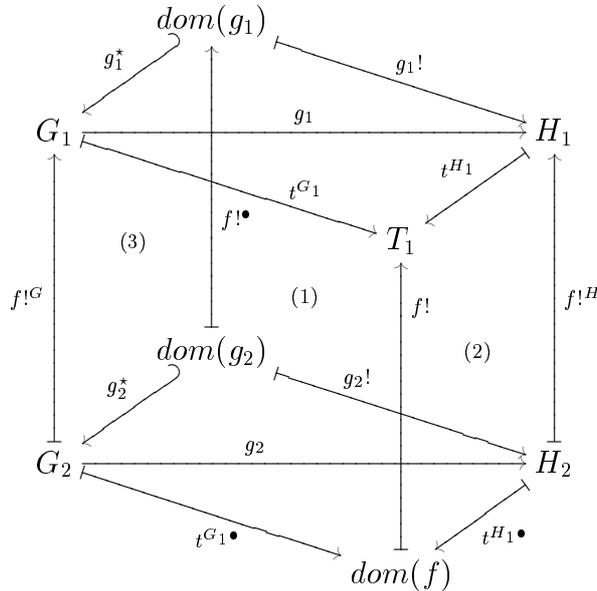
$$\begin{array}{ccc} T_1 & & \text{TGraphP}(T_1) \\ f \uparrow & & \downarrow \mathcal{T}_f \\ T_2 & & \text{TGraphP}(T_2) \end{array}$$

$\mathcal{T}_f$  é definido para cada objeto  $(G_1, t^{G_1}, T_1)$  e morfismo  $g^{T_1} : G_1 \rightarrow H_1$  em  $\text{TGraphP}(T_1)$  como:

- **Objetos:**  $\mathcal{T}_f(G_1, t^{G_1}, T_1) = (G_2, f^* \circ t^{G_1}, T_2)$ , onde (1) abaixo é uma escolha de pullback em Graph, i.e.,  $(G_1 \xleftarrow{f!^G} G_2 \xrightarrow{t^{G_1}} \text{dom}(f)) = \overline{PB}(G_1 \xrightarrow{t^{G_1}} T_1 \xleftarrow{f!} \text{dom}(f))$ .

$$\begin{array}{ccc} G_1 & \xleftarrow{t^{G_1}} & T_1 \\ f!^G \uparrow & (1) & \uparrow f! \\ G_2 & \xleftarrow{t^{G_1}} & \text{dom}(f) \xrightarrow{f^*} T_2 \end{array}$$

- **Morfismos:**  $\mathcal{T}_f(g_1^{T_1}) = g_2^{T_2}$ , onde  $g_1 : G_1^{T_1} \rightarrow H_1^{T_1}$  e  $g_2 = g_2! \circ g_2^{*-1}$  é assim definido: seja (3) o pullback em GraphP de  $g_1^* : \text{dom}(g_1) \rightarrow G_1$  e  $f!^G : G_2 \rightarrow G_1$  onde o morfismo  $g_2^* : \text{dom}(g_2) \rightarrow G_2$  é uma inclusão. O morfismo  $g_2! : \text{dom}(g_2) \rightarrow H_2$  é o morfismo universal induzido pelo pullback (2).



⊙

Exemplo 3.5: A figura 3.5 mostra a aplicação do functor de retipagem ao morfismo  $g_1 : G_1^{T_1} \rightarrow H_1^{T_1}$ ,  $\mathcal{T}_f(g_1) = g_2 : G_2 \rightarrow H_2$ . Note que os grafos  $G_2$  e  $H_2$  são obtidos através da aplicação do functor  $\mathcal{T}_f$  aos grafos  $G_1$  e  $H_1$ , respectivamente. Na figura está representado também o morfismo  $f!^G$ , obtido em  $\mathcal{T}_f(G_1)$ . ⊙

### 3.3 Regras

Como foi dito na introdução, uma regra em gramáticas de grafos é um morfismo entre grafos (no nosso caso, grafos tipados) e são compostas por um lado direito e um lado esquerdo. No uso de regras para descrever uma mudança de estado, o lado esquerdo representa o que **deve** estar presente no (grafo-)estado para que a regra seja aplicada e o morfismo representa as **mudanças** que ocorrem no estado com a aplicação da regra. A relação (morfismo) entre os lados direito e esquerdo expressa as operações envolvidas numa mudança de estado, que são **preservação**, **eliminação** e **adição** de itens do grafo-estado.

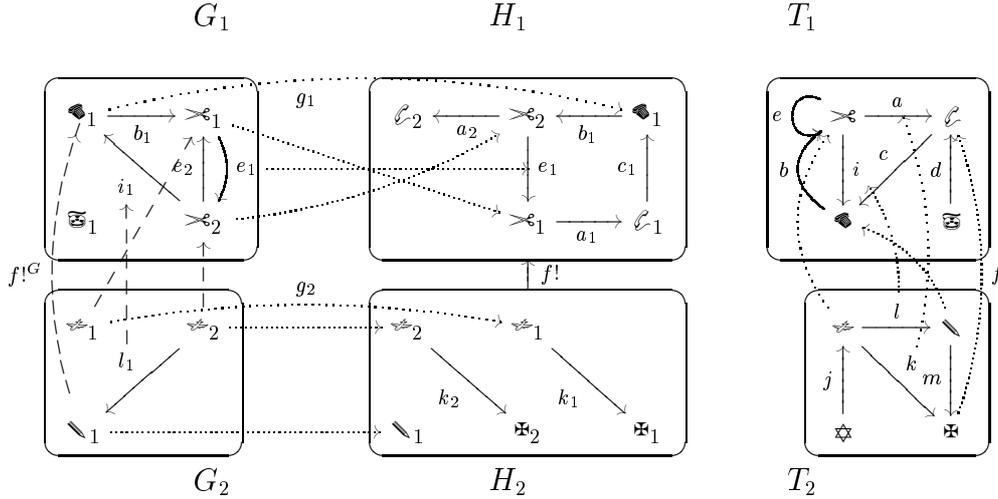


FIGURA 3.5- Retipagem do morfismo  $g_1$ .

Seja  $r : L \rightarrow R$  uma regra. O significado das operações envolvidas numa mudança de estado em termos do morfismo são:

- **preservação:** todo item que é mapeado por  $r$  é preservado pela regra;
- **eliminação:** todo item que está em  $L$  e não é mapeado por  $r$  é eliminado pela regra;
- **adição:** todo item em  $R$  que não pertencem à imagem de  $r$  é criado/adicionado pela regra.

Para permitir a definição dos *unfoldings*, duas restrições serão impostas às regras: a primeira é que as regras não podem identificar itens (i.e., o morfismo precisa ser injetor) e a segunda é que as regras devem eliminar algum item (i.e., o morfismo precisa ser parcial).

**DEFINIÇÃO 3.8. (Regra)** *Seja  $T$  um grafo. Uma regra em relação a  $T$  é um morfismo  $r^T : L^T \rightarrow R^T$  em  $\text{TGraphP}(T)$  tal que:*

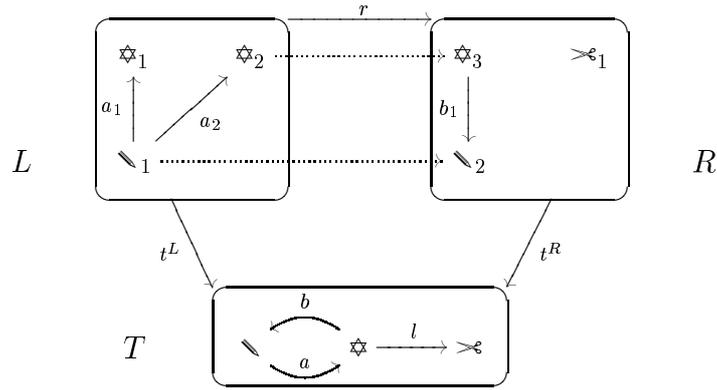
1.  $r^T$  é um morfismo injetor;
2.  $r^T$  não é total;

O conjunto de todas a regras em relação a um grafo  $T$  é denotado por  $\text{Rules}(T)$ . Se substituirmos o segundo item por (2'.  $r$  não é um isomorfismo), será obtida uma **regra geral**. O conjunto de todas as regras gerais com um grafo-tipo  $T$  fixo é denotado por  $\text{GRules}(T)$ . ☺

**Exemplo 3.6:** A regra  $r : L^T \rightarrow R^T = (f, id_T)$ , onde  $f$  é o morfismo entre os grafos  $L$  e  $R$  dado por  $f : L \rightarrow R = (\{\searrow_1 \mapsto \searrow_2, \star_2 \mapsto \star_3\}, \emptyset)$  da figura 3.6 preserva os itens  $\searrow_1$  e  $\star_2$ , elimina os itens  $a_1, \star_1$  e  $a_2$  e cria os itens  $\succ_1$  e  $b_1$ . ☺

No relacionamento entre gramáticas de grafos, a relação entre as regras será definida em termos de **subregras**. Se um regra  $r_1$  é subregra de uma regra  $r_2$ , então elas criam os mesmos itens. Além disso, deve haver uma inclusão de  $r_1$  em  $r_2$  que satisfaça a condição de segurança, definida a seguir.

**DEFINIÇÃO 3.9. (Condição de Segurança)** *Sejam  $L_1^T = (L_1, t^{L_1}, T)$  e  $L_2^T = (L_2, t^{L_2}, T)$  dois grafos tipados com grafo-tipo  $T$ . Um morfismo total e injetor  $i_L : L_1 \rightarrow L_2$  satisfaz a condição de segurança se, e somente se, para todo elemento  $e \in L_2$ , se  $t^{L_2}(e) \in \text{img}(t^{L_1})$  então  $\exists p \in L_1$  tal que  $i_L(p) = e$ .* ☺

FIGURA 3.6- Exemplo de uma regra com tipo  $T$ .

A condição de segurança assegura que  $L_2$  não contenha elementos que tenham o mesmo tipo que um elemento da imagem de  $i_L$  e que não esteja na imagem de  $i_L$ .

**DEFINIÇÃO 3.10. (Subregra)** Sejam  $r_1 : L_1^T \rightarrow R_1^T$  e  $r_2 : L_2^T \rightarrow R_2^T$  duas regras com grafo-tipo  $T$ . Então  $r_1$  será uma **subregra** de  $r_2$ , denotado por  $r_1 \subseteq^r r_2$ , se, e somente se, houver morfismos totais e injetores  $i_L : L_1 \rightarrow L_2$  e  $i_R : R_1 \rightarrow R_2$  tais que o diagrama abaixo seja um *pushout* em  $\mathbb{T}\text{GraphP}(T)$  e que satisfaçam a condição de segurança.

$$\begin{array}{ccc} L_1 & \xrightarrow{r_1} & R_1 \\ i_L \downarrow & PO & \downarrow i_R \\ L_2 & \xrightarrow{r_2} & R_2 \end{array}$$

Duas regras  $r_1$  e  $r_2$  são **isomórficas**, denotado por  $r_1 \cong r_2$ , se  $r_1 \subseteq^r r_2$  e  $i_L$  e  $i_R$  forem isomorfismos. A classe das regras isomórficas a uma regra  $r$  é denotada por  $[r]$ . Será denotado por  $[r_1] \subseteq^R [r_2]$  a extensão da relação de subregra a classes de regras isomórficas. Os conjuntos de classes de regras isomórficas com tipo  $T$  são denotados por  $IRules(T)$  e, para as regras gerais,  $IGRules(T)$ . ☺

**Exemplo 3.7:** Na figura 3.7, a regra  $r_1 : L_1^T \rightarrow R_1^T$  é subregra da regra  $r_2 : L_2^T \rightarrow R_2^T$ . Observe que o diagrama comuta e satisfaz a condição de segurança. Se houvesse em  $L_2$  mais algum item do tipo  $\sphericalangle$  (ou  $\clubsuit$  ou  $a$ ), deveria haver mais um item deste tipo em  $L_1$  para ser mapeado por  $i_L$  e satisfazer, assim, a condição de segurança. ☺

A seguir, será definido um functor de regras, que será usado da definição de morfismos entre gramáticas de grafos. Este functor associa a cada grafo  $T$  o conjunto de regras com grafo-tipo  $T$  e a cada morfismo entre grafos-tipo  $T_1$  e  $T_2$  a correspondente tradução das regras através do functor de retipagem  $\mathcal{T}$ .

**DEFINIÇÃO 3.11. (Functor de Regras)**  $IGRules$  define um functor  $\mathcal{R} : \text{GraphP}^{OP} \rightarrow \text{SetP}$ , definido para todos os objetos  $T_1, T_2$  e morfismo  $f^{OP} : T_1 \rightarrow T_2$  em  $\text{GraphP}^{OP}$  da seguinte maneira:

- **Objetos:**  $\mathcal{R}(T_1) = IGRules(T_1)$
- **Morfismos:**  $\mathcal{R}(f) = \mathcal{R}_f : IGRules(T_1) \rightarrow IGRules(T_2)$  é definido para todo  $[r] \in IGRules(T_1)$  como

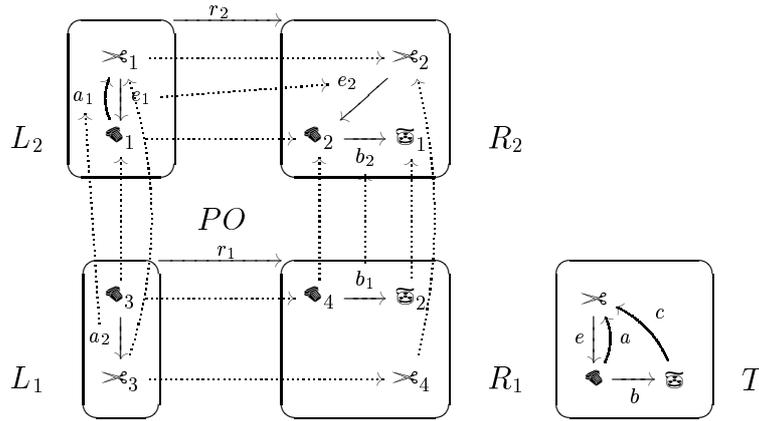
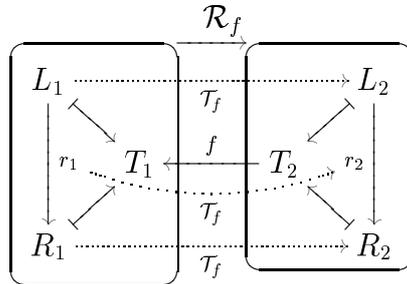


FIGURA 3.7- Exemplo de subregra.

$$\mathcal{R}_f([r]) = \begin{cases} [\mathcal{T}_f(r)], & \text{se } \mathcal{T}_f(r) \text{ não for um isomorfismo,} \\ \text{undef,} & \text{caso contrário} \end{cases}$$



Através do functor  $\mathcal{R}$ , uma regra  $r_1$  com grafo-tipo  $T_1$  pode ser “traduzida” para uma regra  $r_2$  com grafo-tipo  $T_2$  usando para isto um morfismo  $f : T_2 \rightarrow T_1$ .  $\mathcal{R}_f$  relaciona a regra  $r_1 : L_1^{T_1} \rightarrow R_1^{T_1}$  com a regra  $r_2 : L_2^{T_2} \rightarrow R_2^{T_2}$ , onde  $r_2 = \mathcal{T}_f(r_1)$ ,  $L_2^{T_2} = \mathcal{T}_f(L_1^{T_1})$  e  $R_2^{T_2} = \mathcal{T}_f(R_1^{T_1})$ .



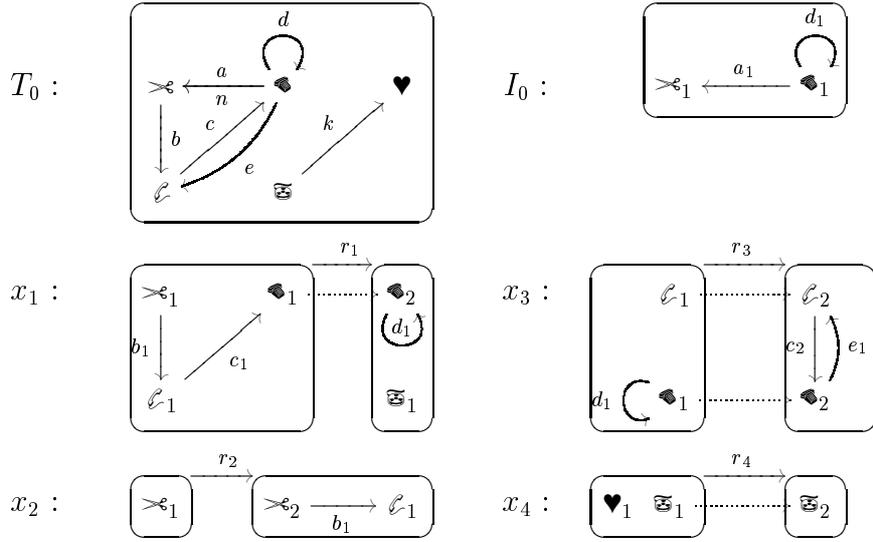
### 3.4 Gramáticas de Grafos

Neste trabalho, serão usadas gramáticas de grafos onde os grafos são grafos tipados. Uma gramática de grafos consiste basicamente de um grafo inicial e um conjunto de regras. Sua definição consiste de quatro itens:

1. um grafo-tipo que especifica o tipo de todos os grafos envolvidos na gramática;
2. um grafo inicial, que representa o estado inicial do sistema;
3. um conjunto de nomes de regras, para identificar as regras da gramática;
4. uma função que associa a cada nome de regra uma regra.

DEFINIÇÃO 3.12. **(Gramática de Grafos)** Uma gramática de grafos é uma tupla  $GG = (T, I, N, n)$  onde:

- $T$  é um grafo (o tipo da gramática);
- $I$  é um grafo tipado em  $\text{TGraphP}(T)$  (o grafo inicial da gramática),
- $N$  é um conjunto de nomes de regras;
- $n : N \rightarrow \text{Rules}(T)$  é uma função total (a função de nomeação, atribuindo a cada nome de regra uma regra em relação ao tipo  $T$ ).

FIGURA 3.8- Exemplo de uma gramática de grafos  $GG_0$ .

Será denotado por  $\bar{n}$  a extensão de  $n$  a classes de regras isomórficas, i.e.,  $\bar{n}(x) = [n(x)]$  para cada  $x \in N$ . ☺

Exemplo 3.8: Seja a gramática de grafos da figura 3.8. O conjunto de nomes de regras é  $N_0 = \{x_1, x_2, x_3, x_4\}$  e  $T_0$  é o grafo-tipo (o tipo da gramática). A regras são:

- $r_1 = (f_1, id_T)$ ,  $f_1 = (\{\clubsuit_1 \mapsto \clubsuit_2\}, \emptyset)$ : esta regra preserva um item com tipo  $\clubsuit$ , elimina itens com os tipos  $b$ ,  $c$ ,  $\succ$  e  $\ell$  e cria itens com tipo  $d$  e  $\boxtimes$ ;
- $r_2 = (f_2, id_T)$ ,  $f_2 = (\emptyset, \emptyset)$ : esta regra não preserva nenhum item, elimina um item do tipo  $\succ$  e cria itens com tipo  $\succ$ ,  $b$  e  $\ell$ ;
- $r_3 = (f_3, id_T)$ ,  $f_3 = (\{\ell_1 \mapsto \ell_2, \clubsuit_1 \mapsto \clubsuit_2\}, \emptyset)$ : esta regra preserva itens com tipo  $\ell$  e  $\clubsuit$ , elimina um item com tipo  $d$  e cria itens com tipo  $c$  e  $e$ ;
- $r_4 = (f_4, id_T)$ ,  $f_4 = (\{\boxtimes_1 \mapsto \boxtimes_2\}, \emptyset)$ : esta regra preserva um item com tipo  $\boxtimes$  e elimina um item com tipo  $\heartsuit$ .

☺

Os morfismos entre gramáticas de grafos são definidos de modo a expressar uma compatibilidade estrutural entre as gramáticas e a implicar uma correspondente relação semântica: se houver um morfismo entre as gramáticas  $GG_0$  e  $GG_1$ , então há uma correspondente relação entre as semânticas de  $GG_0$  e  $GG_1$ , induzida pelo morfismo.

Um morfismo  $f : GG_1 \rightarrow GG_2$  entre as gramáticas  $GG_1$  e  $GG_2$  é composto de dois componentes: um morfismo  $f_T : T_2 \rightarrow T_1$  entre os grafos-tipo  $T_1$  e  $T_2$  e uma função parcial  $f_N : N_1 \rightarrow N_2$  entre os conjuntos dos nomes das regras.

**DEFINIÇÃO 3.13. (Morfismo entre Gramáticas de Grafos)** *Sejam as gramáticas  $GG_1 = (T_1, I_1^{T_1}, N_1, n_1)$  e  $GG_2 = (T_2, I_2^{T_2}, N_2, n_2)$ . Um morfismo entre gramáticas de grafos  $f : GG_1 \rightarrow GG_2$  é um par  $f = (f_T^{OP}, f_N)$  onde  $f_T^{OP} : T_1 \rightarrow T_2$  é um morfismo em  $\text{GraphP}^{OP}$  e  $f_N : N_1 \rightarrow N_2$  é uma função parcial tais que as seguintes condições sejam satisfeitas:*

1. **Subcomutatividade:**  $\bar{n}_2 \circ f_N \subseteq^R \mathcal{R}_{f_T} \circ \bar{n}_1$ , onde  $\bar{n}_1$  e  $\bar{n}_2$  são as extensões de  $n_1$  e  $n_2$  a classes de regras isomórficas e  $\subseteq^R$  é a relação de subgrafia.

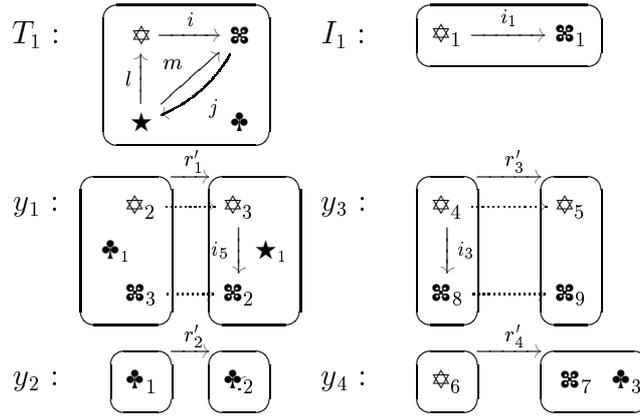


FIGURA 3.9- Gramática de grafos  $GG_1$ .

$$\begin{array}{ccc}
 N_1 & \xrightarrow{\bar{n}_1} & \mathcal{R}(T_1) \\
 f_N \downarrow & \subseteq^R & \downarrow \mathcal{R}_{f_T} \\
 N_2 & \xrightarrow{\bar{n}_2} & \mathcal{R}(T_2)
 \end{array}$$

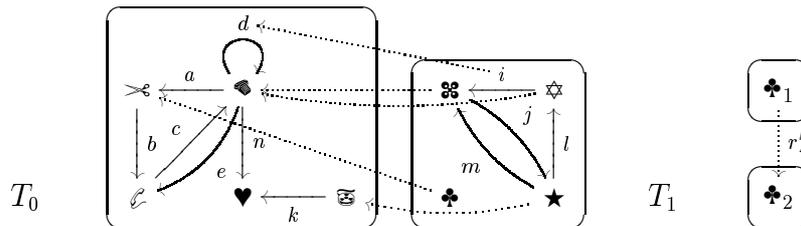
2.  $I_2^{T_2} \cong \mathcal{T}_{f_T}(I_1^{T_1})$



Observações:

- Sempre que uma regra  $x$  for mapeada por  $f_N$ , a regra associada a  $f_N(x)$  precisa ser uma subregra de  $\mathcal{T}_f(n(x))$ , a tradução de  $n(x)$  através do morfismo  $f_T$ . Com isto, um morfismo em gramáticas de grafos pode mapear uma regra  $r$  para uma subregra de  $\mathcal{T}_f(r)$
- A condição de subcomutatividade garante que se uma regra  $r$  não for mapeada, então  $\mathcal{T}_f(r)$  é um isomorfismo. Esta condição é necessária para garantir uma compatibilidade semântica (preservação de derivações).
- O grafo inicial  $I_2$  precisa ser isomórfico à tradução  $\mathcal{T}_f(I_1)$  do grafo inicial  $I_1$ .

Exemplo 3.9: Um morfismo entre as gramáticas de grafos  $GG_0$  da figura 3.8 e  $GG_1$  da figura 3.9 é dado por:  $f_T : T_1 \rightarrow T_0 = (\{i \mapsto d, \diamond \mapsto \clubsuit, \spadesuit \mapsto \heartsuit, \star \mapsto \heartsuit, \clubsuit \mapsto \heartsuit\}, \emptyset)$  e  $f_N = \{x_1 \mapsto y_1, x_2 \mapsto y_2, x_3 \mapsto y_3, x_4 \mapsto \text{undef}\}$ . A tradução da regra  $r_4$  resulta na regra  $r'_4$  abaixo, que é um isomorfismo e, portanto, não é mapeada.



### 3.5 Semântica Operacional de Gramáticas de Grafos

A idéia básica de gramáticas de grafos quando usadas para especificar sistemas é representar o estado inicial do sistema por um grafo tipado e as possíveis mudanças de estado por regras. O comportamento do sistema é então dado pela aplicação das regras a grafos representando os estados do sistema (grafos-estado). A aplicação de uma regra a um grafo-estado só é possível se houver um ocorrência para esta regra, i.e., se houver um subgrafo do grafo-estado que corresponda ao lado esquerdo da regra.

O comportamento de uma gramática de grafos é dado pela aplicação de uma regra a um grafo-estado. A aplicação de uma regra, chamada **derivação direta**, só é possível se houver um ocorrência para a regra, i.e., um subgrafo do grafo-estado que seja isomórfico ao lado esquerdo da regra. Esta relação entre o subgrafo e o lado esquerdo da regra não precisa ser um isomorfismo. Porém, neste trabalho, esta exigência (isomorfismo) é imposta para permitir o relacionamento com as redes L/T. Ademais, esta restrição tem sentido do ponto de vista prático: por exemplo, se uma regra elimina dois itens de um tipo  $\clubsuit$ , é de se esperar que devam existir no grafo-estado dois itens deste mesmo tipo para que a regra possa ser aplicada.

A semântica operacional de uma gramática de grafos é dada em termos de derivações diretas, que são aplicações das regras a algum grafo-estado. A aplicação de uma regra  $r : L \rightarrow R$  numa ocorrência  $m : L \rightarrow IN$ , onde  $IN$  é um grafo-estado, consiste em retirar do grafo  $IN$  tudo que é removido por  $r$  e acrescentar tudo que é criado pela regra. Como algumas situações de conflito podem ocorrer neste processo ([EHR 96, COR 96]), será feito primeiro a adição de itens e depois a eliminação:

1. acrescentar a  $IN$  todos os itens que são adicionados para regra  $r$ , i.e., os elementos que estão no lado direito  $R$  e que não pertencem à imagem de  $r$ ;
2. remover do grafo-estado resultante todos os itens que são removidos por  $r$ , i.e., os elementos que não pertencem ao domínio de  $r$ . Também são removidos os vértices que porventura ficarem sem origem ou sem destino.

**DEFINIÇÃO 3.14. (Ocorrência)** Dada uma regra  $r : L^T \rightarrow R^T$  com grafo-tipo  $T$ , uma ocorrência  $m : L^T \rightarrow IN^T$  de  $r$  num grafo  $IN^T$  é um morfismo total e injetor em  $\text{TGraphP}(T)$ .  $\odot$

A derivação direta é definida a seguir por um *pushout* na categoria  $\text{TGraphP}(T)$ , daí o nome SPO (*Single PushOut*).

**DEFINIÇÃO 3.15. (Derivação Direta)** Uma derivação direta  $s$  de um grafo  $IN^T$  usando a regra  $r$  de nome  $nr$  na ocorrência  $m$  é uma tupla  $s = (nr, S)$ , onde  $S$  é um *pushout* de  $m$  e  $r$  em  $\text{TGraphP}(T)$ .

$$\begin{array}{ccc} L & \xrightarrow{r} & R \\ m \downarrow & S & \downarrow m^\bullet \\ IN & \xrightarrow{r^\bullet} & OUT \end{array}$$

Uma derivação direta é denotada por  $IN \xrightarrow{nr:m} OUT$ .  $IN$ ,  $OUT$ ,  $r^\bullet$  e  $m^\bullet$  são chamados **grafo de entrada**, **grafo de saída**, **co-regra** e **co-match**, respectivamente.

$\odot$

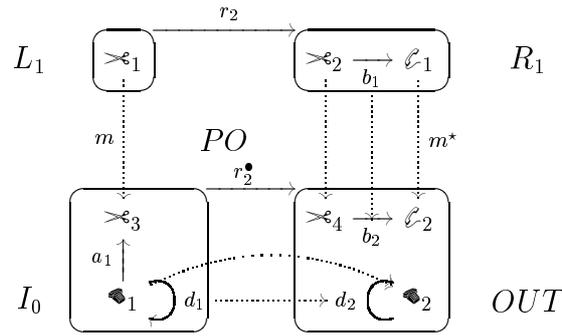


FIGURA 3.10- Derivação direta do grafo  $I_0$  usando a regra  $r_2$ .

Exemplo 3.10: A figura 3.10 mostra uma derivação direta usando a regra  $r_2$  e usando como grafo de entrada o grafo inicial  $I_0$ , ambos da gramática da figura 3.8. O grafo de saída  $OUT$  pode ser usado como grafo de entrada para uma outra derivação usando a regra  $r_3$ , formando uma seqüência de derivações (derivação seqüencial). ☺

Uma derivação direta descreve uma unidade de computação (mudança de estado) numa gramática de grafos. Estas derivações podem ser combinadas seqüencialmente (seqüências de derivações) de tal maneira que o grafo de saída de uma derivação é usado como grafo de entrada para a derivação seguinte. Estas seqüências são chamadas **derivações seqüenciais** e formam a **semântica seqüencial** de uma gramática de grafos.

### 3.6 Grafos Duplamente Tipados

Um grafo tipado é um grafo  $G$  que possui dois grafos-tipo  $TG$  e  $T$ , onde  $T$  também é o grafo-tipo de  $TG$ .

DEFINIÇÃO 3.16. (**Grafo Duplamente Tipado**) Seja  $T$  um grafo. Um grafo duplamente tipado  $G^{TG/\nearrow T}$  é uma tupla  $G^{TG/\nearrow T} = (G^T, t, TG^T)$  onde  $G^T$  e  $TG^T$  são grafos tipados com grafo-tipo  $T$  e  $t : G^T \rightarrow TG^T$  é um morfismo total entre grafos tipados em  $\text{TGraphP}(T)$ . O grafo tipado  $TG^T$  é denominado grafo-tipo duplo. Um item  $x \in G^{TG/\nearrow T}$  denota um elemento de  $V_G \cup A_G$ . ☺

Exemplo 3.11: No grafo duplamente tipado  $G^{TG/\nearrow T}$  da figura 3.11,  $t' = (t, id_T)$  é um morfismo total entre  $G^T$  e  $TG^T$ ,  $t_1$  é o morfismo de tipo de  $G^T$  e  $t_2$  é o morfismo de tipo de  $TG^T$ . Como se pode notar,  $t_1 = t_2 \circ t$ .

$$\begin{array}{ccc} G & \xrightarrow{t} & TG, \\ \downarrow t_1 & & \downarrow t_2 \\ T & \xrightarrow{id_T} & T \end{array}$$

Serão considerados três tipos de morfismos entre grafos duplamente tipados: o primeiro entre grafos de tipos diferentes; o segundo entre grafos com o mesmo grafo-tipo e o terceiro entre grafos com mesmo grafo-tipo duplo (estes dois últimos casos especiais do primeiro). Estes três tipos serão usados para definir morfismos entre gramáticas de grafos duplamente tipadas e morfismos entre *unfoldings*. ☺

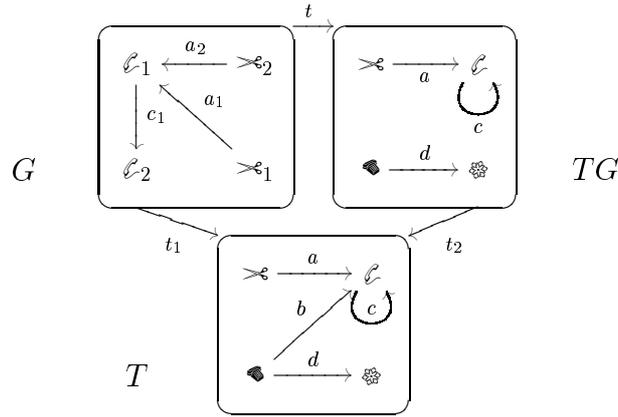


FIGURA 3.11- Grafo duplamente tipado  $G^{TG/\mathcal{T}}$ .

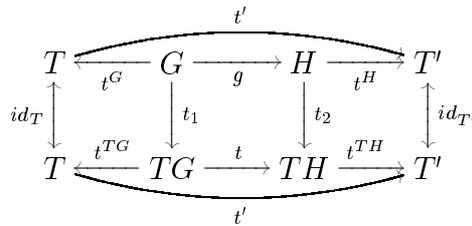
DEFINIÇÃO 3.17. (**Morfismo entre Grafos Duplamente Tipados**) Seja  $G^{TG/\mathcal{T}}$  e  $H^{TH/\mathcal{T}'}$  dois grafos duplamente tipados e  $g^t : G^T \rightarrow H^T$  e  $t^t : TG^T \rightarrow TH^T$  dois morfismos entre grafos tipados (morfismos em TGraphP).

$$\begin{array}{ccc}
 G^T \xrightarrow{g^t} H^T & G^T \xrightarrow{g^T} H^T & G^T \xrightarrow{g^T} H^T \\
 \downarrow t^{G^T} & \downarrow t^{G^T} & \downarrow t^{G^T} \\
 TG^T \xrightarrow{t^t} TH^T & TG^T \xrightarrow{t^T} TH^T & TG^T \xrightarrow{id_{TG^T}} TG^T
 \end{array}
 \quad (1) \quad (2) \quad (3)$$

1. o par  $(g^t, t^t)$  será um **morfismo entre grafos duplamente tipados**, denotado por  $g^{t/\mathcal{T}'} : G^{TG/\mathcal{T}} \rightarrow H^{TH/\mathcal{T}'}$ , se, e somente se, o diagrama (1) comutar fracamente;
2. se  $g^{t/\mathcal{T}'}$  for um morfismo e  $t^t = id_T$  (i.e.,  $T = T'$ ), será usada a notação  $g^{t/\mathcal{T}}$ . Neste caso, a comutatividade fraca se reduz à do diagrama (2);
3. se  $g^{t/\mathcal{T}}$  for um morfismo e  $t = id_{TG}$  (i.e.,  $T = T'$  e  $TG^T = TH^T$ ), será usada a notação  $g^{TG/\mathcal{T}}$ . Neste caso, a comutatividade fraca se reduz à do diagrama (3).

A categoria de grafos duplamente tipados é denotada por DTGraphP. ☺

Exemplo 3.12: A figura 3.12 apresenta um exemplo de um morfismo  $(g^t, t^t)$  entre os grafos duplamente tipados  $G^{TG/\mathcal{T}} = (G^T, t^{G^T}, TG^T)$  e  $H^{TH/\mathcal{T}'} = (H^T, t^{H^T}, TH^T)$  onde  $g^t = (g, t')$  e  $t^t = (t, t')$ .



Analogamente ao functor de retipagem, é definido o functor de retipagem dupla, definido em função de  $\mathcal{T}$ . ☺

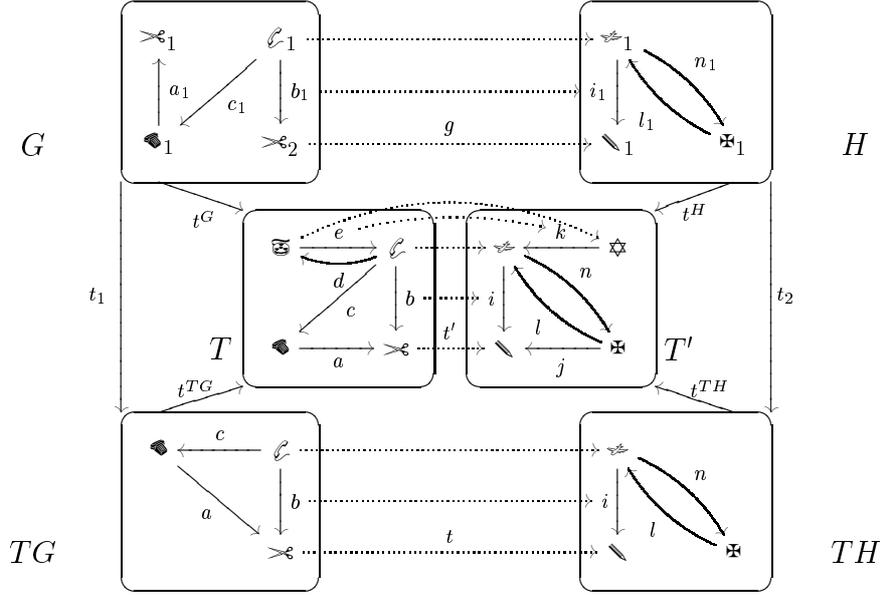


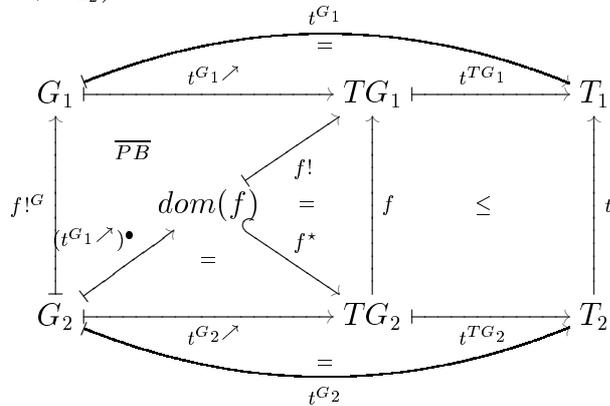
FIGURA 3.12- Morfismo entre grafos duplamente tipados.

DEFINIÇÃO 3.18. **(Funtor de Retipagem Dupla)** Seja  $f^t : TG_2^{T_2} \rightarrow TG_1^{T_1}$  um morfismo em  $TGraphP$ . Há um funtor  $\mathcal{DT}_{f^t} : DTGraphP(TG_1^{T_1}) \rightarrow DTGraphP(TG_2^{T_2})$  induzido por  $f^t$ .

$$\begin{array}{ccc}
 TG_1^{T_1} & & DTGraphP(TG_1^{T_1}) \\
 f^t \uparrow & & \downarrow \mathcal{DT}_{f^t} \\
 TG_2^{T_2} & & DTGraphP(TG_2^{T_2})
 \end{array}$$

$\mathcal{DT}_{f^t}$  é definido para todo objeto  $G_1^{TG_1 \nearrow T_1} = (G_1^{T_1}, t^{G_1^{T_1}}, TG_1^{T_1})$  com morfismo de tipo  $t^{G_1^{T_1}} = (t^{G_1 \nearrow}, id_{T_1})$  e para todo morfismo  $g^{TG_1 \nearrow T_1} = (g^{TG_1}, id_{TG_1^{T_1}}) : G_1^{TG_1 \nearrow T_1} \rightarrow H_1^{TG_1 \nearrow T_1}$  em  $DTGraphP(TG_1^{T_1})$  como:

- **Objetos:**  $\mathcal{DT}_{f^t}(G_1^{TG_1 \nearrow T_1}) = (G_2^{T_2}, t^{G_2^{T_2}}, TG_2^{T_2})$ , onde  $G_2^{TG_2} = \mathcal{T}_f(G_1^{TG_1})$  com morfismo de tipo  $t^{G_2 \nearrow} : G_2 \rightarrow TG_2$ ,  $t^{G_2} = t^{TG_2} \circ t^{G_2} : G_2 \rightarrow T_2$  e  $t^{G_2^{T_2}} = (t^{G_2 \nearrow}, id_{T_2})$ .



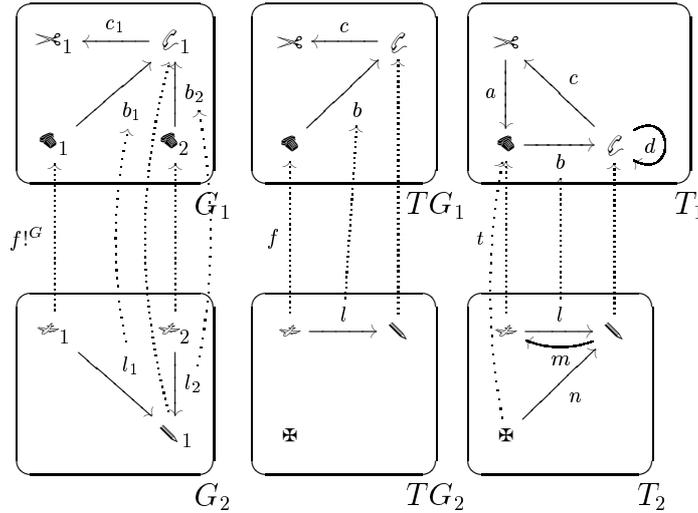
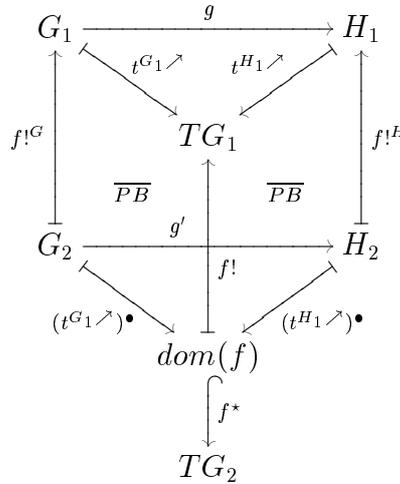


FIGURA 3.13- Retipagem do grafo duplamente tipado  $G_1^{TG_1 \nearrow T_1}$ .

- **Morfismos:**  $\mathcal{DT}_{f^t}(g^{TG_1 \nearrow T_1}) = g^{TG_2 \nearrow T_2} = (g^{TG_2}, id_{TG_2^{T_2}})$ , onde  $g^{TG_2} = \mathcal{T}_f(g^{TG_1})$ .



Exemplo 3.13: A figura 3.13 apresenta a retipagem do grafo duplamente tipado  $G_1^{TG_1 \nearrow T_1}$  através do morfismo  $f^t = (f, t) : TG_2^{T_2} \rightarrow TG_1^{T_1}$ . O resultado é o grafo duplamente tipado  $G_2^{TG_2 \nearrow T_2}$ , i.e.,  $\mathcal{DT}_{f^t}(G_1^{TG_1 \nearrow T_1}) = G_2^{TG_2 \nearrow T_2}$ . Note que  $G_2$  é subgrafo de  $G_1$ . ☺

A próxima definição estabelece uma relação entre grafos duplamente tipados e grafos tipados chamada **functor de esquecimento de tipo**. Este functor elimina um dos tipos do grafo duplamente tipado: pela definição,  $G^{TG \nearrow T}$  consiste de dois grafos tipados  $G^T$  e  $TG^T$  e um morfismo  $t^{GT} = (t, id_T) : G^T \rightarrow TG^T$  que os conecta. O functor elimina o componente  $TG^T$  e o morfismo  $t^{GT}$ , resultando o grafo tipado  $G^T$ . Este functor será usado na definição de gramáticas de grafos de ocorrência.

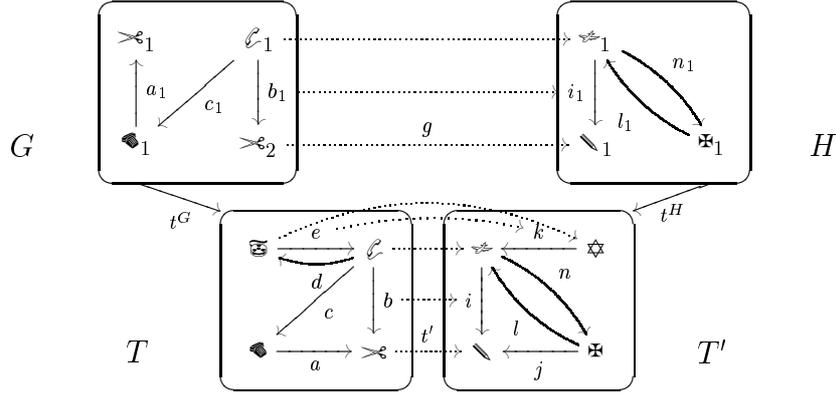
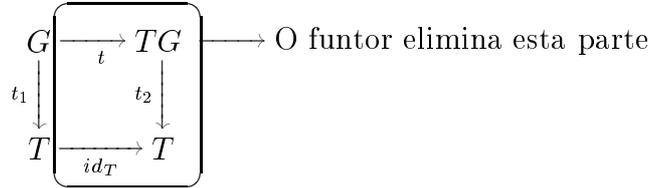


FIGURA 3.14- Morfismo resultante da aplicação de  $\mathcal{V}_T$  ao morfismo da figura 3.12.



DEFINIÇÃO 3.19. (**Functor de Esquecimento de Tipo**) Sejam  $\text{DTGraphP}$  e  $\text{TGraphP}$  as categorias de grafos duplamente tipados e grafos tipados, respectivamente. O **functor de esquecimento de tipo**  $\mathcal{V}_T : \text{DTGraphP} \rightarrow \text{TGraphP}$  é definido por:

- **Objetos:**  $\mathcal{V}_T(G^{TG/\wedge T'}) = G^{T'}$ , para todo objeto  $G^{TG/\wedge T'} = (G^{T'}, t^{G^{T'}}, TG^{T'})$  em  $\text{DTGraphP}$ .
- **Morfismos:**  $\mathcal{V}_T(f^{t/\wedge t'}) = f^{t'}$ , para todo morfismo  $f^{t/\wedge t'} = (f^{t'}, t')$  em  $\text{DTGraphP}$ .

☺

Exemplo 3.14: A figura 3.14 mostra o resultado da aplicação do functor  $\mathcal{V}_T$  ao morfismo apresentado na figura 3.12.

☺

### 3.7 Regras Duplamente Tipadas

As regras duplamente tipadas são definidas em termos de regras.

DEFINIÇÃO 3.20. (**Regra Duplamente Tipada**) Dado um grafo tipado  $TG^T$ , uma regra com tipo duplo  $TG^T$  é um morfismo  $a = r^{TG/\wedge T} = (r^T, id_{TG^T}) : L^{TG/\wedge T} \rightarrow R^{TG/\wedge T}$  em  $\text{DTGraphP}(TG^T)$  se, e somente se,  $r^T$  for uma regra em  $\text{TGraphP}$ .

Será denotado por  $\text{Rules}(TG^T)$  o conjunto das regras que têm o grafo  $TG^T$  como tipo duplo e  $\text{DRules}(TG^T)$  a correspondente extensão para classes de equivalência de regras (gerais).

Uma regra  $r_1^{TG/\wedge T}$  é uma subregra de uma regra  $r_2^{TG/\wedge T}$ , denotado por  $r_1^{TG/\wedge T} \subseteq^r r_2^{TG/\wedge T}$ , se, e somente se,  $r_1^T$  for uma subregra de  $r_2^T$ .

☺

Notação: Sejam  $t_2^L$  e  $t_2^R$  os morfismos de tipo em relação a  $TG$  de  $L$  e  $R$ , respectivamente. Então são definidos:

- $L_a = L^T$  e  $R_a = R^T$ , os lados esquerdo e direito da regra  $a = r^{TG/\wedge T}$
- $pre_a = t_2^L : L \rightarrow TG$ , a **pré-condição** da regra  $a = r^{TG/\wedge T}$
- $post_a = t_2^R : R \rightarrow TG$ , a **pós-condição** da regra  $a = r^{TG/\wedge T}$

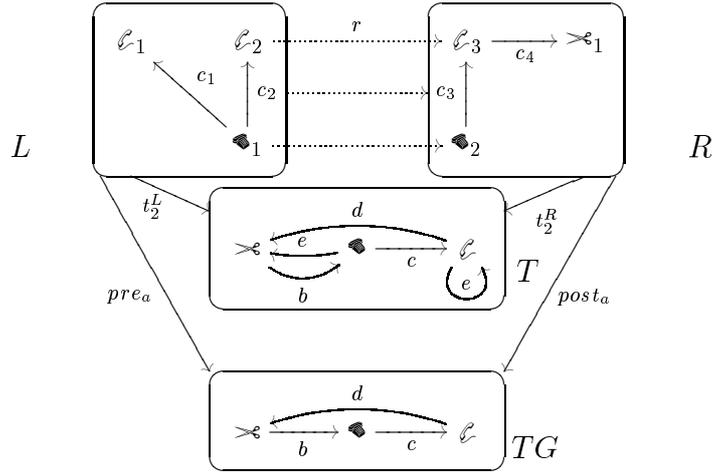


FIGURA 3.15- Regra duplamente tipada.

- $r_a = r^T$ , o **padrão** da regra  $a = r^{TG} \nearrow^T$

Exemplo 3.15: A figura 3.15 mostra a regra duplamente tipada  $a = r^{TG} \nearrow^T$ . Note que  $r^T$  é uma regra, pois o morfismo é parcial (os itens  $l_1$  e  $c_1$  não são mapeados, i.e., são eliminados pela regra). ☹

De modo análogo ao funtor de regras ( $\mathcal{T}$ ), é definido o funtor de regras com tipo duplo.

DEFINIÇÃO 3.21. (**Functor de Regras Duplamente Tipadas**)  $DRules$  define um funtor  $\mathcal{DR} : DTGraphP^{OP} \rightarrow SetP$ , definido para todo objeto  $TG^T$  e morfismo  $f^t : TG_2^{T_2} \rightarrow TG_1^{T_1}$  de  $DTGraphP$ :

- **Objetos:**  $\mathcal{DR}(TG^T) = DRules(TG^T)$
- **Morphisms:**  $\mathcal{DR}(f^t) = \mathcal{DR}_{f^t} : DRules(TG_1^{T_1}) \rightarrow DRules(TG_2^{T_2})$  é definido para todo  $[r] \in DRules(TG_1^{T_1})$  como

$$\mathcal{R}_{f^t}([r]) = \begin{cases} [\mathcal{DT}_{f^t}(r)], & \text{se } \mathcal{DT}_{f^t}(r) \text{ não for um isomorfismo,} \\ \text{undef,} & \text{caso contrário} \end{cases}$$

☹

### 3.8 Gramáticas de Grafos com Tipo Duplo

Uma gramática de grafos com tipo duplo é uma gramática de grafos onde o grafo-tipo é um grafo tipado, o grafo inicial é um grafo com tipo duplo e as regras são duplamente tipadas.

DEFINIÇÃO 3.22. (**Gramáticas com Tipo Duplo**) Uma gramática de grafos com tipo duplo é uma tupla  $GG = (TG^T, I^{TG} \nearrow^T, N, n)$  onde:

- $TG^T$  é um grafo tipado (tipo da gramática);
- $I^{TG} \nearrow^T$  é um grafo com tipo duplo em  $DTGraphP(TG^T)$  (o grafo inicial da gramática);
- $N$  é um conjunto de nomes de regras;
- $n : N \rightarrow Rules(TG^T)$  é uma função total (função de nomeação) que atribui a cada nome de regra uma regra com tipo duplo  $TG^T$ .

☹

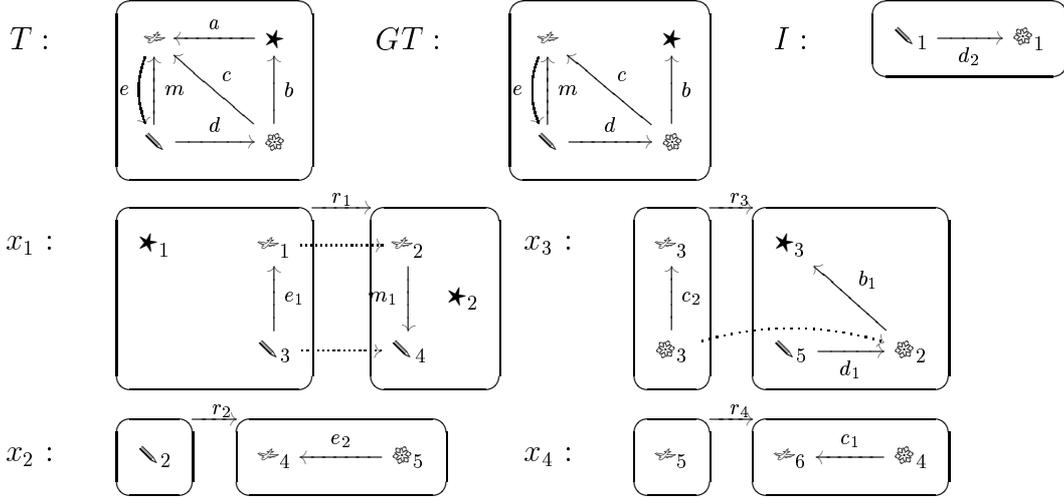


FIGURA 3.16- Gramática de grafos com tipo duplo.

Notação: Será denotado por  $in_{GG}$  o morfismo de tipo do grafo inicial em relação a  $TG$ , i.e.,  $in_{GG} = t' : I \rightarrow TG$ .

Exemplo 3.16: A figura 3.16 apresenta um exemplo de uma gramática de grafos com tipo duplo. ☹

DEFINIÇÃO 3.23. (**Morfismo entre Gramáticas com Tipo Duplo**) Sejam duas gramáticas  $GG_1 = (TG_1^{T_1}, I_1^{TG_1 \nearrow T_1}, N_1, n_1)$  e  $GG_2 = (TG_2^{T_2}, I_2^{TG_2 \nearrow T_2}, N_2, n_2)$ . Um **morfismo entre gramáticas com tipo duplo** é um par  $f = (f_T^{OP}, f_N)$  onde  $f_T^{OP} : TG_1^{T_1} \rightarrow TG_2^{T_2}$  é um morfismo em  $TGraphP^{OP}$  e  $f_N$  é uma função tal que as seguintes condições sejam verdadeiras:

1. O diagrama a seguir subcomuta, i.e., para todo  $x \in dom(f_N)$  e  $x \in dom(\mathcal{DR}_{f_T} \circ \bar{n}_1)$ , tem-se que  $\bar{n}_2 \circ f_N(x) \subseteq^R \mathcal{DR}_{f_T} \circ \bar{n}_1(x)$  ( $\bar{n}_1$  e  $\bar{n}_2$  são as extensões de  $n_1$  e  $n_2$  a classes de isomorfismos de regras)

$$\begin{array}{ccc} N_1 & \xrightarrow{\bar{n}_1} & \mathcal{DR}(TG_1^{T_1}) \\ f_N \downarrow & \subseteq^R & \downarrow \mathcal{DR}_{f_T} \\ N_2 & \xrightarrow{\bar{n}_2} & \mathcal{DR}(TG_2^{T_2}) \end{array}$$

2.  $I_2^{TG_2 \nearrow T_2} \cong \mathcal{DT}_{f_T}(I_1^{TG_1 \nearrow T_1})$ .

A categoria de gramáticas de grafos com tipo duplo é denominada DTGG. ☹

DEFINIÇÃO 3.24. (**Match**) Dada uma regra  $r : L \rightarrow R$  com tipo duplo  $TG^T$ , um **match**  $m : L \rightarrow IN$  de  $r$  num grafo duplamente tipado  $IN$  é um morfismo total e injetor em  $DTGraphP(TG^T)$ . ☹

DEFINIÇÃO 3.25. (**Derivação**) Um **derivação** de um grafo  $IN$  com a regra  $r$  de nome  $nr$  no ocorrência  $m$ , denotado por  $IN \xrightarrow{nr:m} OUT$ , é uma tupla  $s = (n_s, S)$ , onde  $S$  é um pushout  $IN \xrightarrow{r^\bullet} OUT \xleftarrow{m^\bullet} R$  de  $m$  e  $r$  em  $DTGraphP(TG^T)$ . Os componentes  $IN$ ,  $OUT$ ,  $r^\bullet$  e  $m^\bullet$  são chamados **grafo de entrada**, **grafo de saída**, **co-regra** e **co-match**, respectivamente.

$$\begin{array}{ccc}
L_s & \xrightarrow{r} & R \\
\downarrow m & PO & \downarrow m \bullet \\
IN & \xrightarrow{r \bullet} & OUT
\end{array}$$



### 3.9 Relações numa Gramática de Grafos com Tipo Duplo

Nesta seção serão definidas relações entre regras e entre itens do grafo-tipo duplo de uma gramática de grafos com tipo duplo. Estas relações serão usadas para definir uma subclasse de gramáticas de grafos com tipo duplo que podem ser consideradas como comportamento dos objetos de  $GG$ . A idéia básica é considerar o grafo  $TG$  do tipo duplo  $TG^{TT}$  como um grafo que descreve ocorrências de itens com tipo  $T$  (daí esta subclasse ser chamada de gramáticas de grafos de ocorrência). Estes itens representarão vértices ou arcos que podem ocorrer nos grafos-estado de uma gramática. Estas ocorrências serão representadas num grafo chamado **grafo-núcleo**. Todo item de um grafo-núcleo precisa ter uma única origem: ou estava presente no grafo inicial da gramática ou foi criado por alguma regra. No *unfolding* de uma gramática  $GG$ , o grafo-núcleo representará todos os grafos-estado que podem ser atingidos pela aplicação das regras de  $GG$  (iniciando-se no grafo inicial).

**DEFINIÇÃO 3.26. (Grafo-núcleo)** *Seja  $GG = (C^T, IC^{\nearrow T}, N, n)$  uma gramática de grafos com tipo duplo.  $C^T$  será um grafo-núcleo se, e somente se, satisfizer a seguinte condição:*

$$\forall x \in C^T: \exists! y \in (I^T \uplus (\biguplus_{a \in \text{img}(n)} R_a)) \text{ tal que } x = \begin{cases} \text{in}_{GG}(y), & \text{if } y \in I^T, \\ \text{post}_a(y), & \text{se } y \in R_a \text{ e } y \notin \text{dom}(r_a) \end{cases}$$

*Se  $C^T$  for um grafo-núcleo, cada elemento de  $\text{img}(n)$  é chamado **ação** de  $GG$ . Uma ação  $a$  **cria** um elemento  $b \in (V_C \cup A_C)$  se, e somente se,  $b \in \text{dom}(\text{post}_a)$  e  $b \notin \text{dom}(\text{post}_a \circ r_a)$ . Uma ação  $a$  **elimina**  $b$  se, e somente se  $b \in \text{img}(\text{pre}_a)$  e existe  $b_1 \in L_a$  tal que  $\text{pre}_a(b_1) = b$  e  $b_1 \notin \text{dom}(r_a)$ .* ☺

**Exemplo 3.17:** O grafo  $GT$  da gramática da figura 3.16 não é um grafo-núcleo pois, para o item  $\ast$ ,  $\exists y_1 = \ast_1 \in I$  e  $y_2 = \ast_5 \in R_2$  (o lado direito da regra  $r_2$ ) tal que  $\ast = \text{in}_{GG}(y_1) = \text{post}_{r_2}(y_2)$  e  $y_2 \notin \text{dom}(r_3)$  (i.e.,  $\ast$  tem mais de uma origem: já estava no grafo inicial e foi criado pela ação  $r_2$ ). Já o grafo  $GT$  da gramática  $GG_1$  da figura 3.17 é um grafo-núcleo: todos os itens do grafo  $TG$  ou estão presentes no grafo inicial ou são criados por alguma ação. ☺

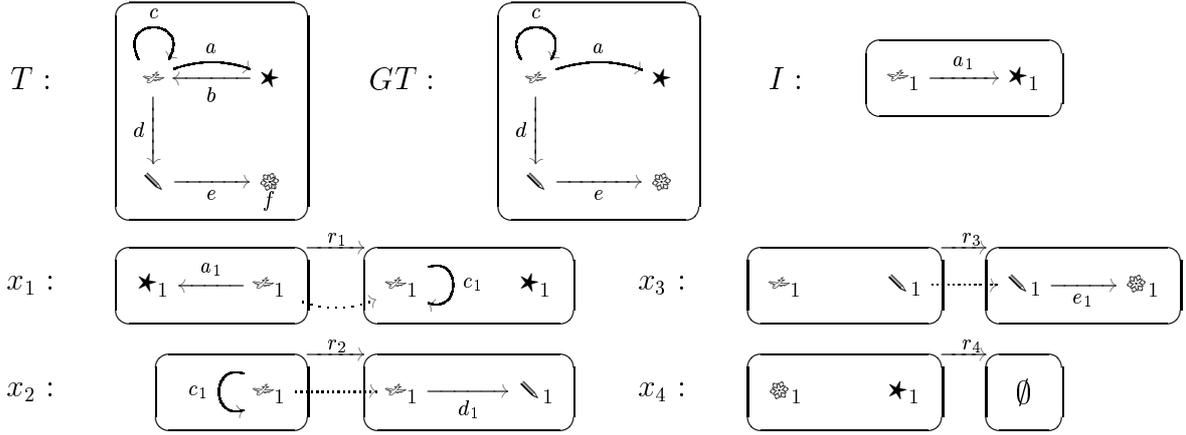
Notação:

- o conjunto de *pré-condições* de um elemento  $a \in A$  restrito a uma relação  $R \subseteq A \times A$ , denotado por  $\text{Pre}^R(a)$ , é definido por

$$\text{Pre}^R(a) = \{a' \mid a'Ra\}$$

- o conjunto de *elementos mínimos* de uma relação  $R$ , denotado por  $\text{Min}^R$ , é definido por

$$\text{Min}^R = \{x \mid \nexists y : yRx\}$$

FIGURA 3.17- Gramática duplamente tipada  $GG_1$ .

3. a *restrição* de uma relação  $R$  a um conjunto de elementos  $S \subseteq A$ , denotada por  $R|_S \subseteq S \times S$ , é definida por

$$R|_S = \{aRb \mid a, b \in S\}$$

Neste trabalho, serão definidas apenas relações onde o grafo-tipo duplo da gramática for um grafo-núcleo. Estas relações serão usadas na definição de gramáticas de grafos de ocorrência e na construção dos *unfoldings*.

**Relação de Dependência:** Uma ação  $a$  é **causa** de uma ação  $b$  se  $a$  **cria** algum item de que  $b$  necessita (para preservar ou eliminar). Isto implica que  $b$  pode ocorrer apenas após a ocorrência de  $a$  (ou seja,  $b$  depende de  $a$ ), i.e., esta relação impõe uma ordem de ocorrência das ações em derivações seqüenciais. A relação de dependência é também estendida a tipos: um item  $x$  do grafo-núcleo é causa de um item  $y$  se a eliminação de  $x$  causa a criação de  $y$  (i.e., existe uma ação que elimina  $x$  e cria  $y$ ). Isto significa que  $y$  só pode ocorrer após a ocorrência de  $x$  ( $y$  depende de  $x$ ).

A formalização desta idéia de dependência é baseada na aplicação do inverso de uma regra: para saber se uma ação  $a_2$  depende de uma ação  $a_1$ , remove-se do grafo-núcleo tudo que for adicionado por  $a_1$ ; se todos os itens de que  $a_2$  necessita ainda estiverem presentes no grafo-núcleo resultante,  $a_2$  não depende de  $a_1$ ; caso contrário,  $a_2$  depende de  $a_1$ . Portanto, se uma ação  $a$  precisa de itens que já estão no grafo inicial, ela não depende de nenhuma outra ação que crie tais itens.

**DEFINIÇÃO 3.27. (Relação de Dependência)** *Sejam  $GG = (C^T, I, N, n)$  uma gramática de grafos com tipo duplo e  $C^T$  um grafo-núcleo. Sejam  $n_1, n_2 \in N$  e  $n(n_1) = a_1 = r_1^{TG \nearrow T}$ ,  $n(n_2) = a_2 = r_2^{TG \nearrow T}$ ,  $a_1 \neq a_2$ . Sejam  $e_1, e_2 \in C^T$ ,  $e_1 \neq e_2$  ( $e_1, e_2 \in C^T$  significa que  $e_1, e_2 \in V_C \cup A_C$ ).*

1. a ação  $n_2$  é **diretamente dependente** de  $n_1$ , denotado por  $n_1 \triangleleft^N n_2$ , se, e somente se,  $(r_{a_1})^{-1} \circ pre_{a_2}$  não é total, onde (1) é um *pushout* em  $\text{TGraphP}(T)$ . Caso contrário,  $n_2$  **não é diretamente dependente** de  $n_1$ , denotado por  $n_1 \not\triangleleft^N n_2$ . Se  $n_1 \not\triangleleft^N n_2$  e  $n_2 \not\triangleleft^N n_1$ , diz-se que  $n_1$  e  $n_2$  são **diretamente independentes**.

$$\begin{array}{ccccc}
L_{a_1} & \xleftarrow{(r_{a_1})^{-1}} & R_{a_1} & & L_{a_2} & \xrightarrow{r_{a_2}^T} & R_{a_2} \\
\downarrow & & \swarrow \text{post}_{a_1} & & \nearrow \text{pre}_{a_2} & & \\
& & (1) & & & & \\
X_1 & \xleftarrow{(r_{a_1})^{-1} \bullet} & C^T & & & & 
\end{array}$$

2. a **relação de dependência entre ações** de uma gramática com tipo duplo,  $\leq^N \subseteq (N \times N)$ , é o fecho reflexivo e transitivo de  $\triangleleft^N$ ;
3. o elemento  $e_2$  é **diretamente dependente** de  $e_1$  denotado por  $e_1 \triangleleft^T e_2$ , se, e somente se, houver uma ação  $n_1 \in N$  que elimina  $e_1$  e cria  $e_2$ ;
4. a **relação de dependência entre tipos** de uma gramática de grafos com tipo duplo,  $\leq^T \subseteq (C \times C)$ , é o fecho transitivo e reflexivo de  $\triangleleft^T$ ;
5. a **relação de dependência** de uma gramática de grafos com tipo duplo,  $\leq \subseteq (N \times N) \cup (C \times C)$ , é definida por  $(\leq^N \cup \leq^T)$ .

☺

Exemplo 3.18: Para a gramática  $GG_1$  da figura 3.17, deriva-se a seguinte tabela relacionando itens e regras:

regra	cria	elimina	preserva
$x_1$	$c_1$	$a_1, \star_1$	$\varnothing_1$
$x_2$	$\backslash_1$	$c_1$	$\varnothing_1$
$x_3$	$e_1, \otimes_1$	$\varnothing_1$	$\backslash_1$
$x_4$	—	$\otimes_1, \star_1$	—

De acordo com a definição 3.27, a relação de dependência entre as ações é:  $x_1 \leq x_2, x_2 \leq x_3, x_3 \leq x_4$  e, por transitividade,  $x_1 \leq x_3, x_2 \leq x_4$ , e  $x_1 \leq x_4$  (e os pares reflexivos). Esta relação pode ser interpretada como: se houver uma derivação sequencial que usa as 4 ações, as derivações diretas podem ser feitas usando as ações  $x_1, x_2, x_3$  e  $x_4$ , nesta ordem (porém, isto depende da relação de conflito, que será vista adiante).

A relação de dependência entre os itens do grafo-núcleo é:  $\star_1 \leq c_1, a_1 \leq c_1, c_1 \leq \backslash_1, \varnothing_1 \leq e_1$  e  $\varnothing_1 \leq \otimes_1$ . E, por transitividade,  $a_1 \leq \backslash_1$  e  $\star_1 \leq \backslash_1$  (e os pares reflexivos).

☺

**Relação de Conflito Fraco:** Uma ação  $a$  está em **conflito fraco** com uma ação  $b$  se  $a$  **elimina** algum item de que  $b$  necessita (para preservar ou eliminar). Isto implica que  $b$  **não** pode ocorrer se  $a$  já houver ocorrido (ou seja,  $a$  impede a ocorrência de  $b$ ). No caso de  $b$  também estar em conflito fraco com  $a$ , então estão em conflito (situação de exclusão mútua). A possibilidade de ocorrência de apenas conflito fraco (i.e., sem exclusão mútua) vem do fato de que itens podem ser considerados como “apenas para leitura” e assim ações que usam estes itens podem ocorrer em paralelo no modelo SPO (i.e., é permitido que eliminação possa ocorrer em paralelo com preservação). Portanto então que esta relação não é simétrica: o fato de uma ação  $a$  estar em conflito fraco com  $b$  não implica que  $b$  esteja em conflito fraco com  $a$ . Obviamente, esta relação é simétrica se as regras não preservarem itens.

A relação de conflito fraco também é estendida aos itens do grafo-núcleo: um item  $a$  está em conflito com um item  $b$  se a **criação** de  $b$  **exclui** a criação de  $a$ , i.e., a regra que cria  $b$  elimina algum item que é necessário para a regra que cria  $a$ .

DEFINIÇÃO 3.28. (**Relação de Conflito Fraco**) Sejam  $GG = (C^T, I, N, n)$  uma gramática de grafos com tipo duplo e  $C^T$  um grafo-núcleo. Sejam  $n_1, n_2 \in N$  e  $n(n_1) = a_1 = r_1^{TG \nearrow T}$ ,  $n(n_2) = a_2 = r_2^{TG \nearrow T}$ ,  $a_1 \neq a_2$ . Sejam  $e_1, e_2 \in C$ ,  $e_1 \neq e_2$ .

1. a ação  $n_2$  está em **conflito fraco** com  $n_1$ , denotado por  $n_1 \xrightarrow{\#^N} n_2$ , se, e somente se,  $r_{a_2}^\bullet \circ pre_{a_1}$  não for total, onde (1) é um pushout em  $\mathbf{TGraphP}(T)$ . Caso contrário,  $n_2$  **não está em conflito fraco** com  $n_1$ , denotado por  $n_1 \dashrightarrow^N n_2$ .

$$\begin{array}{ccccc}
 R_{a_1} & \xleftarrow{r_{a_1}} & L_{a_1} & & L_{a_2} & \xrightarrow{r_{a_2}} & R_{a_2} \\
 & & \searrow^{pre_{a_1}} & & \nearrow^{pre_{a_2}} & & \downarrow^{pre_{a_2}^\bullet} \\
 & & & C^T & & & H_2 \\
 & & & \xrightarrow{r_{a_2}^\bullet} & & & 
 \end{array}
 \quad (1)$$

2. o elemento  $e_2$  está em **conflito fraco** com  $e_1$ , denotado por  $e_1 \xrightarrow{\#^T} e_2$ , se, e somente se, existirem ações  $n_1$  e  $n_2$  que criam  $e_1$  e  $e_2$ , respectivamente, e  $n_1 \xrightarrow{\#^N} n_2$ .
3. a **relação de conflito fraco** de uma gramática de grafos com tipo duplo,  $\xrightarrow{\#} \subseteq (N \times N) \cup (C \times C)$ , é definida por  $(\xrightarrow{\#^N} \cup \xrightarrow{\#^T})$ .

⊙

Exemplo 3.19: Considerando a gramática  $GG_1$  da figura 3.17, e a tabela do exemplo anterior, chega-se à seguinte relação de conflito fraco:

- entre as ações:  $x_1 \xrightarrow{\#} x_4$ ,  $x_3 \xrightarrow{\#} x_2$  e  $x_4 \xrightarrow{\#} x_1$ , i.e., se a ação  $x_1$  ocorrer, o item  $\star_1$  é eliminado e, portanto, a ação  $x_4$  não pode mais ocorrer (e vice-versa:  $x_1$  e  $x_4$  estão em conflito mútuo). Se a ação  $x_3$  ocorrer, o item  $\varphi_1$  é eliminado e a ação  $x_1$  não pode mais ocorrer, uma vez que ela preserva este mesmo item.
- entre os itens:  $e_1 \xrightarrow{\#} \setminus_1$  e  $\otimes_1 \xrightarrow{\#} \setminus_1$ , uma vez que  $x_3 \xrightarrow{\#} x_2$  e  $x_3$  cria os itens  $e_1$  e  $\otimes_1$  e  $x_2$  cria o item  $\setminus_1$ .

⊙

**Relação de Conflito:** Quando simétricas, situações de conflito fraco levam a situações de conflito, razão esta que será definida a relação de conflito em sua função: se duas ações estão em conflito entre si, uma exclui a ocorrência da outra (exclusão mútua). Os conflitos são herdados através da relação de dependência: se uma ação  $a$  está em conflito com uma ação  $b$ , então todas as ações que dependem de  $a$  também estarão em conflito com  $b$  (se  $b$  ocorrer, nenhuma delas pode mais ocorrer). Se uma ação  $a$  está em conflito com uma ação  $b$ , então **não** existe uma derivação seqüencial que inclua ambas as ações, uma vez que uma exclui a ocorrência da outra.

DEFINIÇÃO 3.29. (**Relação de Conflito**) Sejam  $GG = (C^T, IN^{C \nearrow T}, N, n)$  uma gramática de grafos com tipo duplo e  $C^T$  um grafo-núcleo. Sejam  $x_1, x_2, x_3 \in N \cup C$ .

1. a **relação de conflito fraco herdado** de  $GG$ , denotada por  $\xrightarrow{\#}$ , é definida por  $\xrightarrow{\#} \subseteq (N \times N) \cup (C \times C)$  tal que

$$x_1 \xrightarrow{\#} x_2 \text{ se, e somente se, } \exists x_3 | x_1 \xrightarrow{\#} x_3 \text{ e } x_3 \leq x_2$$

2. a **relação de conflito** de  $GG$ , denotada por  $\overset{\#}{\longleftrightarrow}$ , é definida por  $\overset{\#}{\longleftrightarrow} \subseteq (N \times N) \cup (C \times C)$  tal que

$$x_1 \overset{\#}{\longleftrightarrow} x_2 \text{ se, e somente se, } x_1 \overset{\#}{\implies} x_2 \text{ e } x_2 \overset{\#}{\implies} x_1$$

Se  $x_1 \overset{\#}{\longleftrightarrow} x_1$ ,  $GG$  possui **autoconflito**.

☺

Exemplo 3.20: Com as relações de conflito fraco e de dependência encontradas nos dois exemplos anteriores para a gramática da figura 3.17, pode-se encontrar a seguinte relação de conflito para  $GG_1$ :

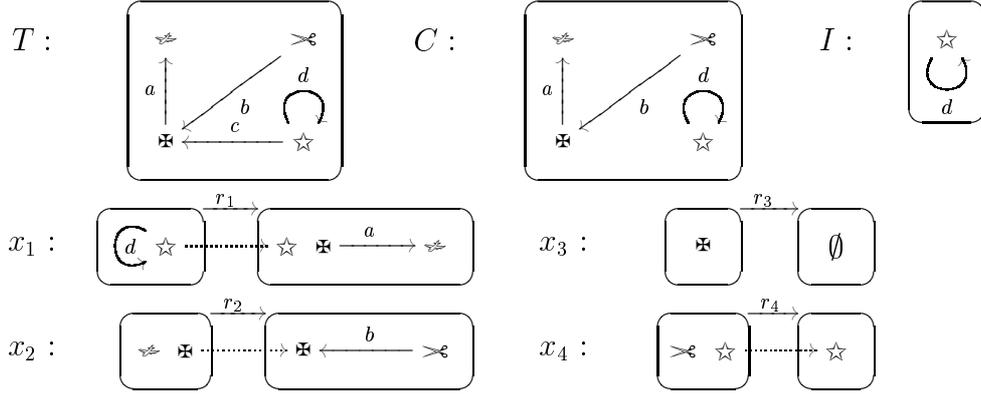
$\overset{\#}{\longrightarrow}$	$\leq$	$\overset{\#}{\implies}$	$\overset{\#}{\longleftrightarrow}$
$x_1 \overset{\#}{\longrightarrow} x_4$	$x_4 \leq x_4$	$x_1 \overset{\#}{\implies} x_4$	$x_1 \overset{\#}{\longleftrightarrow} x_4$
$x_4 \overset{\#}{\longrightarrow} x_1$	$x_1 \leq x_1$	$x_4 \overset{\#}{\implies} x_1$	$x_3 \overset{\#}{\longleftrightarrow} x_4$
	$x_1 \leq x_2$	$x_4 \overset{\#}{\implies} x_2$	$x_4 \overset{\#}{\longleftrightarrow} x_4$
	$x_1 \leq x_3$	$x_4 \overset{\#}{\implies} x_3$	
	$x_1 \leq x_4$	$x_4 \overset{\#}{\implies} x_4$	
$x_3 \overset{\#}{\longrightarrow} x_2$	$x_2 \leq x_2$	$x_3 \overset{\#}{\implies} x_2$	$x_3 \overset{\#}{\longleftrightarrow} x_3$
	$x_2 \leq x_3$	$x_3 \overset{\#}{\implies} x_3$	
	$x_2 \leq x_3$	$x_3 \overset{\#}{\implies} x_4$	
	$x_3 \leq x_3$		
	$x_3 \leq x_4$		

A relação de conflito para os itens do grafo-núcleo é  $e_1 \overset{\#}{\longleftrightarrow} \setminus_1$  e  $\otimes_1 \overset{\#}{\longleftrightarrow} \setminus_1$ , uma vez que os únicos itens em conflito fraco são  $e_1 \overset{\#}{\longrightarrow} \setminus_1$  e  $\otimes_1 \overset{\#}{\longrightarrow} \setminus_1$  e nenhum outro item depende de  $\setminus_1$ .

☺

**Relação de Ocorrência:** Observe que as relações  $\leq$  e  $\overset{\#}{\longrightarrow}$  impõem restrições na ordem na qual as ações podem ocorrer em seqüências de derivações. Por exemplo, se  $a \leq b$ , então  $a$  deve ocorrer antes de  $b$  em qualquer seqüência de derivações (porque  $b$  precisa de algum item que é criado por  $a$ ). O mesmo é válido para a relação de conflito fraco, pois, se  $a \overset{\#}{\longrightarrow} b$ ,  $b$  exclui a ocorrência de  $a$  e a ação  $b$  não pode ocorrer antes de  $a$  numa mesma seqüência de derivações. Portanto, numa gramática de grafos, é necessário considerar ambas estas relações para estabelecer alguma ordem de ocorrência entre as ações da gramática, i.e., a combinação destas duas relações nos fornece uma possível ordem de ocorrência das ações. Esta combinação é a **relação de ocorrência** e a idéia básica é que se um par  $(a, b)$  de ações pertence a esta relação, então  $a$  deve ocorrer antes de  $b$  em qualquer derivação seqüencial que incluir ambas as ações. Será usada para reconhecer possíveis seqüências de derivações num conjunto de ações.

DEFINIÇÃO 3.30. (**Relação de Ocorrência**) Sejam  $GG = (C^T, I, N, n)$  uma gramática de grafos com tipo duplo,  $C^T$  um grafo-núcleo e  $\leq$  e  $\overset{\#}{\longrightarrow}$  as relações de dependência e conflito fraco, respectivamente. A **relação de ocorrência** de  $GG$ ,

FIGURA 3.18- Gramática de grafos de ocorrência  $GG_2$ 

denotada por  $\leq^\#$ , é definida por  $\leq^\# \subseteq (N \times N) \cup (C \times C)$  tal que  $\leq^\#$  seja o fecho reflexivo e transitivo de  $(\leq \cup \xrightarrow{\#})$ .

Seja  $a \in N$ . A **relação de ocorrência local** restrita à ação  $a$ , denotada por  $\leq_a^\#$ , é o fecho reflexivo e transitivo de  $(\leq \cup \xrightarrow{\#})|_{Pre \leq(a)}$ . De modo análogo, para um conjunto de ações  $A \subseteq N$ ,  $\leq_A^\#$  é o fecho reflexivo e transitivo de  $(\leq \cup \xrightarrow{\#})|_P$ , onde  $P = \{a' | a \in A \text{ e } a' \leq a\}$ . ☺

Exemplo 3.21: Para a gramática  $GG_1$  da figura 3.17, a relação de ocorrência é  $N \times N$ , não havendo uma ordem total que seja compatível com tal relação. Portanto, não há uma derivação seqüencial que inclua todas as ações de  $GG_1$ . Para a gramática  $GG_2$  da figura 3.18, chega-se às seguintes relações:

regra	cria	elimina	preserva
$x_1$	$\boxplus, a, \boxtimes$	$d$	$k$
$x_2$	$b, \boxtimes$	$\boxtimes$	$\boxplus$
$x_3$	—	$\boxplus$	—
$x_4$	—	$\boxtimes$	$\star$

- relação de dependência entre as regras:  $x_1 \leq x_3, x_1 \leq x_2, x_2 \leq x_4, x_1 \leq x_4$  e entre os itens do grafo-núcleo:  $\boxplus \leq d, a \leq d, \boxtimes \leq d, b \leq \boxtimes, \boxtimes \leq \boxtimes$  e  $\text{sy}a \leq d$  (e os pares reflexivos);
- relação de conflito fraco:  $x_2 \xrightarrow{\#} x_3$ . Como  $x_3$  não cria nenhum item, a relação de conflito fraco entre os itens do grafo-núcleo é vazia;
- relação de conflito: como  $\xrightarrow{\#} = \{x_2 \rightarrow x_3\}$ ,  $\xleftarrow{\#} = \emptyset$ ;
- relação de ocorrência é fecho reflexivo e transitivo de  $\leq \cup \xrightarrow{\#}$ :  $\trianglelefteq = \{x_1 \rightarrow x_2, x_1 \rightarrow x_3, x_1 \rightarrow x_4, x_2 \rightarrow x_4, x_3 \rightarrow x_2\}$  e os pares reflexivos.

Portanto, uma possível ordem para a ocorrência das ações é  $x_1 < x_2 < x_3 < x_4$ . ☺

### 3.10 Gramáticas de Grafos de Ocorrência

A idéia para a semântica que será definir é que ela deve descrever todos os possíveis grafos-estado e todas as possíveis derivações e que seja também uma gramática de grafos.

A seguir, serão definidas as gramáticas de grafos de ocorrência, que são gramáticas de grafos duplamente tipadas com algumas propriedades especiais e que podem ser

interpretadas como comportamento (semântica) dos objetos de  $GG$ . Serão interpretadas como um domínio semântico que representa todo o comportamento das gramáticas de  $GG$  e serão definidas em termos das relações definidas na seção anterior.

Uma gramática de grafos de ocorrência que representa as computações de uma gramática de grafos  $GG = (T, I^T, N, n)$  possui dois grafos-tipo: o grafo-tipo  $T$  de  $GG$  e um grafo-tipo  $C$  onde são representadas as ocorrências dos itens de  $T$  que aparecem nas derivações de  $GG$ . A idéia é que o grafo-núcleo represente todos os grafos-estado envolvidos em derivações, i.e., seus itens representam os vértices ou arcos que podem ocorrer nos grafos-estado. Isto significa que os itens de  $C$  ou pertencem ao grafo inicial de  $GG$  ou foram criados em alguma derivação, uma vez que um mesmo item não pode ser criado duas vezes (i.e.,  $C$  é um grafo-núcleo). Além disso, cada ação de uma gramática de grafos de ocorrência representará uma derivação em alguma seqüência de derivações de  $GG$ .

**DEFINIÇÃO 3.31. (Gramáticas de Grafos de Ocorrência)** *Seja  $Occ = (C^T, I^{C^T}, N, n)$  uma gramática de grafos com tipo duplo. Então  $Occ$  será uma **gramática de grafos de ocorrência** se, e somente se, satisfizer as seguintes condições:*

1. *relações de ocorrência locais acíclicas:  $\forall a \in N : \leq_a^\#$  é anti-simétrica;*
2. *não possuir auto-conflitos:  $\left\langle \overset{\#}{\rightleftharpoons} \right\rangle$  é irreflexiva;*
3.  *$\forall a \in N : Pre^{\leq_a^\#}(a)$  é finito;*
4.  *$C^T$  é um grafo-núcleo;*
5.  *$\forall (nr, a), (nr, b) \in N : pre_a = pre_b$  e  $r_a = r_b \Rightarrow post_a = post_b$ ;*
6. *os elementos de  $N$  são do tipo  $(x, a)$ , onde  $a$  é um morfismo em  $DTGraphP$ ,  $n(x, a) = a$  e para todo  $(x, a_1), (x, a_2) \in N : \mathcal{V}_T(a_1) = \mathcal{V}_T(a_2)$ .*



O significado destes itens é:

1. este item assegura que a relação de ocorrência local associada a cada ação de  $Occ$  seja uma ordem parcial (i.e., não há ciclos, de modo que qualquer ação pode fazer parte de alguma seqüência de derivações) e, portanto, a relação de dependência de  $Occ$  (tanto para as ações quanto para os itens do grafo-núcleo) é também uma ordem parcial.
2. não há ações em autoconflito. Como a relação de conflito é herdada através da relação de dependência, sempre que duas ações  $a$  e  $b$  forem conseqüência de uma terceira, i.e.,  $c \leq b$  e  $c \leq a$ ,  $a$  e  $b$  **não** estarão em conflito.
3. cada ação tem um número finito de causas, i.e., precisa ocorrer um número finito de ações (derivações) antes que cada ação ocorra.
4. cada item de  $C^T$  tem uma única origem: é imagem ou de algum item do grafo inicial ou do lado direito de uma única ação de  $Occ$ . Isto implica que se um item do grafo inicial (ou que foi criado por uma alguma ação) aparecer no lado direito de alguma outra ação  $b$ , este item é preservado por  $b$ , i.e., todas as ações precisam usar itens ou do grafo inicial ou que foram criados por outras ações. Como todas as ações sempre eliminam algum item, todos os elementos mínimos de  $C^T$  em relação a  $\leq$  pertencem ao grafo inicial.

5. este axioma implica que uma regra é aplicada no máximo uma vez usando uma mesma ocorrência. Isto garante que as aplicações das regras serão determinísticas: a aplicação de uma regra a uma ocorrência resulta num único resultado, ao invés de infinitos resultados isomórficos.
6. este item justifica o nome “gramáticas de grafos de ocorrência”: cada item  $(x, a)$  de  $N$  é uma ocorrência da regra  $x$ . Os nomes das ocorrências são pares contendo o nome da regra que foi usada e a própria ocorrência, que é usada para diferenciar diferentes ocorrências de uma mesma regra. Ademais, todas as ocorrências que têm o mesmo nome de regra têm que usar a mesma regra.

**Exemplo 3.22:** A gramática da figura 3.18 é uma gramática de grafos de ocorrência. A relação de conflito para esta gramática é vazia, uma vez que as únicas ações em conflito fraco são  $x_2$  e  $x_3$  e nenhuma ação depende de  $x_3$ . Como  $\leq$  é anti-simétrica, as relações de ocorrência locais também são anti-simétricas. ☺

Relações entre gramáticas de grafos de ocorrência serão expressas por morfismos, que são morfismos entre gramáticas de grafos duplamente tipadas que preservam a estrutura da gramática de grafos de ocorrência. Como cada item do grafo-núcleo  $C^T$  representa uma ocorrência de algum item de  $T$ , o morfismo deve mapear estes itens de modo compatível e ainda deve respeitar os nomes das ações: se  $(n_a, a)$  for mapeado para  $(n_b, b)$ , todas as outras ações que usam o nome  $n_a$  precisam ser mapeadas para as ações que usam o nome  $n_b$ .

**DEFINIÇÃO 3.32. (Morfismos entre Gramáticas de Grafos de Ocorrência)** *Sejam  $Occ_1$  and  $Occ_2$  duas gramáticas de grafos de ocorrência e  $f = (f_T^{OP}, f_N) : Occ_1 \rightarrow Occ_2$  um morfismo em DTGG, onde  $f_T = c^t : C_1^{T_1} \rightarrow C_2^{T_2}$ . Então  $f$  será um morfismo entre gramáticas de grafos de ocorrência se, e somente se, as seguintes condições forem satisfeitas:*

1.  $\forall (rn, a_1), (rn, b_1) \in \text{dom}(f_N), f_N(rn, a_1) = (rn_1, a_2)$  e  $f_N(rn, b_1) = (rn_2, b_2) \Rightarrow rn_1 = rn_2$ .
2. no diagrama abaixo, (1) é um pullback em GraphP, onde  $t^{\text{dom}(c)} = (t^*)^{-1} \circ t^{C_2} \circ c^*$  (este morfismo precisa ser total pois  $c^t$  comuta fracamente).

$$\begin{array}{ccc}
 C_1 & \xrightarrow{t^{C_1}} & T_1 \\
 c^! \uparrow & (1) & \uparrow t^! \\
 \text{dom}(c) & \xrightarrow{t^{\text{dom}(c)}} & \text{dom}(t) \\
 c^* \downarrow & = & \downarrow t^* \\
 C_2 & \xrightarrow{t^{C_2}} & T_2
 \end{array}$$

A categoria de gramáticas de grafos de ocorrência é denotada por OccGG. ☺

Os elementos do grafo-núcleo de uma gramática de grafos de ocorrência representam itens do grafo-tipo que foram criados por alguma ação ou que já estavam presentes no grafo inicial. Como serão usadas gramáticas para representar computações de modo que o grafo-núcleo represente todos os (grafos) estados alcançados pelas computações (i.e., o grafo-núcleo representa todos os possíveis grafos-estado), surge a questão de quais subgrafos podem ser realmente encontrados em algum grafo-estado.

Por exemplo, se um item  $x$  foi criado como conseqüência da eliminação de um item  $y$ , estes dois elementos nunca poderão ocorrer num mesmo grafo-estado. A próxima definição estabelece as condições para que um subgrafo  $G$  de um grafo-núcleo  $C$  possa ser um subgrafo de alguma computação. Estes grafos são chamados **grafos concorrentes**. Em tal grafo, os itens são independentes entre si e não estão em conflito entre si.

**DEFINIÇÃO 3.33. (Grafo Concorrente)** *Sejam  $Occ = (C, I, N, n)$  um objeto de  $OccGG$  e  $G$  um subgrafo de  $C$ . Seja  $A = \{a \in N \mid a \leq b \text{ e } x \in \text{img}(\text{pre}_b)\}$  e  $b \in N$  e  $x \in G$ .  $G$  será um **grafo concorrente** se, e somente se, as seguintes condições forem verdadeiras para todo  $x, y \in G$ :*

1.  $x \not\leq y$
2.  $x \xrightarrow{\#} y$  é falso (i.e., os itens de  $G$  não estão em conflito)
3.  $\leq_A^\#$  é anti-simétrica.
4.  $\forall a \in \{b \in N \mid x \in \text{img}(\text{pre}_b)\}, \text{Pre}^{\leq_A^\#}(a)$  é finito.

☺

**Exemplo 3.23:** Para a gramática da figura 3.18,  $N$  é finito,  $\leq$  é anti-simétrica e a relação de conflito é vazia para os itens de  $C^T$ . Portanto, qualquer subgrafo de  $C$  que não possuem itens dependentes entre si é um grafo concorrente. ☺

Nem todo morfismo entre gramáticas de grafos de ocorrência preserva as relações de ocorrência e de conflito. Os morfismos prefixos, definidos a seguir, preservam tais relações e grafos concorrentes. Além disso, este morfismo, se existir, é único. A prova para estas propriedades dos morfismos prefixos pode ser encontrada em [RIB 96b].

**DEFINIÇÃO 3.34. (Morfismo Prefixo)** *Sejam  $Occ_1 = (C_1^{T_1}, I_1^{C_1 \nearrow T_1}, N_1, n_1)$  e  $Occ_2 = (C_2^{T_2}, I_2^{C_2 \nearrow T_2}, N_2, n_2)$  duas gramáticas de grafos de ocorrência e  $p = (p_T^{OP}, p_N) : Occ_1 \rightarrow Occ_2$  um morfismo em  $OccGG$ , onde  $p_T = c^t : C_2^{T_2} \rightarrow C_1^{T_1}$ . Então,  $p$  será um morfismo prefixo se, e somente se, as seguintes condições forem satisfeitas:*

1.  $I_2^{T_2} = I_1^{T_1}$ .
2.  $t = \text{id}_{T_1}$ .
3.  $p_N(rn, a_1) = (rn', a_2) \Rightarrow rn = rn' \text{ e } r_{a_1} = r_{a_2}$ .

☺

Como as gramáticas de grafos de ocorrência servirão como um domínio semântico para os objetos de  $GG$ , será definido um functor  $\mathcal{F} : OccGG \rightarrow GG$ . A idéia é que se uma gramática de grafos de ocorrência  $Occ$  se relaciona com uma gramática de grafos tipada  $G$ , então  $Occ$  descreve computações de  $G$ .

**DEFINIÇÃO 3.35. (Functor  $\mathcal{F}$ )** *Sejam  $G = (C^T, I, N, n)$  um objeto de  $GG$  e  $f = (f_T^{OP}, f_N) : Occ_1 \rightarrow Occ_2$  um morfismo em  $OccGG$ , onde  $f_T = c^t$ . Então o functor  $\mathcal{F} : OccGG \rightarrow GG$  é definido por:*

- **Objetos:**  $\mathcal{F}(G) = (T, \mathcal{V}_T(I), N', n')$ , onde  $N' = \{x \mid (x, a) \in N\}$ ,  $n'(x) = \mathcal{V}_T(n(x, a))$  e  $\mathcal{V}_T : DTGraphP \rightarrow TGraphP$  é o functor de esquecimento de tipo.
- **Morfismos:**  $\mathcal{F}(f) = (t^{OP}, f'_N)$ , onde  $f'_N(x) = x'$  if  $f_N(x, a) = (x', a')$ .

☺

### 3.11 Unfolding de Gramáticas de Grafos

Nesta seção, será construída uma gramática de grafos de ocorrência a partir de uma gramática de grafos  $GG$  e que representará todas as computações de  $GG$  e será chamada de *unfolding* da gramática  $GG$ .

A idéia do *unfolding* é construir um objeto que represente todos os possíveis grafos-estado e todas as possíveis derivações de uma gramática de grafos  $GG = (T, I^T, N, n)$ . Este objeto será uma gramática de grafos de ocorrência e é definido como o colimite do diagrama formado pelos *unfoldings* finitos de  $GG$ . O *unfolding* finito de  $GG$  é definido indutivamente através da aplicação de regras, de maneira que o *unfolding* zero é definido como uma gramática de grafos de ocorrência que possui mesmo grafo inicial que  $GG$  e como grafo-núcleo também o grafo inicial. O  $k$ -ésimo *unfolding* é obtido aplicando-se todas as regras possíveis (e em paralelo) ao grafo-núcleo do  $(k - 1)$ -ésimo *unfolding*, não removendo, porém, os elementos que porventura sejam eliminados por alguma regra (com isto o grafo núcleo vai representar todos os possíveis grafos-estado de  $GG$ ).

A semântica de um objeto  $GG$  de  $GG$  é definida como o *unfolding* de  $GG$ , que é construído indutivamente da seguinte forma:

**Base da Indução(Profundidade 0):** O *unfolding* de profundidade 0 é o *unfolding* vazio, que consiste do grafo inicial  $I^T$  de  $GG$  como o grafo-núcleo,  $I^T$  tipado com o morfismo identidade como grafo inicial, conjunto de nomes de regras vazio e função de nomeação vazia.

**Passo da Indução(Profundidade  $i + 1$ ):** O *unfolding* de profundidade  $i + 1$  é construído em 4 passos:

1. construir o conjunto  $ApplRules^{i+1}$  de regras que são aplicáveis a algum grafo de saída do *unfolding* do passo  $i$ . Este conjunto consiste de tuplas contendo um nome para a regra, uma regra e o correspondente ocorrência
2. aplicar todas as regras de  $ApplRules^{i+1}$  ao grafo-núcleo do passo  $i$  de tal maneira que nada é removido, i.e., apenas são inseridos os itens que são criados pelas regras. Isto significa aplicar uma regra paralela contendo todas as regras de  $ApplRules^{i+1}$  no correspondente *match* paralelo, porém, iniciando não do lado esquerdo, mas do domínio da regra paralela (deste modo, nenhum item é removido).
3. retipar os lados esquerdos das regras de  $ApplRules^{i+1}$  em relação ao novo grafo-núcleo.
4. retipar todas as ações que estavam presentes no *unfolding*  $i$  em relação ao novo grafo-núcleo.

DEFINIÇÃO 3.36. **(Conjunto de Regras Aplicáveis)** Sejam  $Occ$  uma gramática de grafos de ocorrência com grafo núcleo  $C^T$  e  $GG = (T, I, N, n)$  uma gramática de grafos. O conjunto de regras aplicáveis de  $GG$  em  $C^T$  é definido como:

$$ApplRules(Occ, GG) = \{(nr, r^T, m^T) | nr \in N, n(nr) = r^T : L^T \rightarrow R^T,$$

$m^T = m^{OUT} : L^T \rightarrow C^T, m^{OUT}$  é um ocorrência e  $dom(m^{OUT})$  é um grafo concorrente}

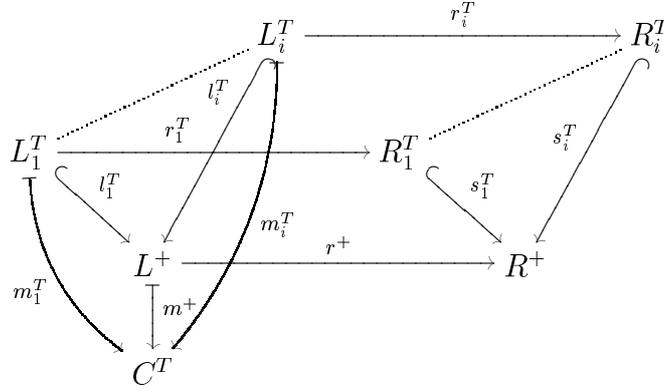


DEFINIÇÃO 3.37. **(Conjunto de Regras Aplicadas)** Seja  $Occ = (C^T, I, N, n)$  uma gramática de grafos de ocorrência. O conjunto de regras aplicadas de  $Occ$  é definido como

$$Applied(Occ) = \{(n_a, r_a, pre_a) | (n_a, a) \in N\}$$

⊖

DEFINIÇÃO 3.38. **(Regra Paralela Aplicável)** Seja  $A = ApplRule(Occ, GG)$  o conjunto de regras aplicáveis de  $GG$  no grafo núcleo  $C^T$  de  $Occ$ . Seja  $L^+$  o coproduto em  $TGraphP(T)$  de todos os lados esquerdos das regras em  $A$  e  $R^+$  o coproduto em  $TGraphP(T)$  de todos os lados direitos das regras em  $A$ . A regra paralela aplicável é um par  $(r^+, m^+)$  onde  $r^+ : L^+ \circ R^+$  e  $m^+ : L^+ \circ C$  são os morfismos universais induzidos pelo coproduto que gera o lado esquerdo  $L^+$ .



⊖

DEFINIÇÃO 3.39. **(Unfolding de Profundidade  $d$ )** O *unfolding* (finito)  $\mathcal{U}_G^d(GG)$  de profundidade  $d$  de uma gramática de grafos tipada  $GG = (T, I^T, N, n)$  é obtido indutivamente como se segue:

**Base da Indução:** *Unfolding* de profundidade 0:

$$\mathcal{U}_G^0(GG) = (I^T, I^T \nearrow^T, \emptyset, \emptyset)$$

**Hipótese da Indução:** Seja  $\mathcal{U}_G^i = (C_i^T, IN^{C_i \nearrow^T}, N_i, n^i)$  um *unfolding* de profundidade  $i$ .

**Passo da Indução:** *Unfolding* de profundidade  $i + 1$ :

O *unfolding* de profundidade  $i + 1$ ,  $\mathcal{U}_G^{i+1} = (C_{i+1}^T, IN^{C_{i+1} \nearrow^T}, N_{i+1}, n^{i+1})$  é construído em 4 passos:

1. construir o conjunto de regras aplicáveis  $ApplRules^{i+1}$ :

$$ApplRules^{i+1} = ApplRules(\mathcal{U}_G^i, GG) - Applied(\mathcal{U}_G^i)$$

2. construir o novo grafo núcleo  $C^{i+1}$  e a inclusão  $c_{i+1}^T$ :

Seja  $(r^+, m^+)$  a regra paralela aplicável de  $ApplRules^{i+1}$ . O grafo núcleo do passo  $i + 1$  é definido como o *pushout* do diagrama (1) abaixo (*pushout* em  $TGraphP(T)$ ), e a correspondente inclusão é dada pelo morfismo (definido pelo *pushout*)  $c_{i+1}^T : C_i^T \rightarrow C_{i+1}^T$ .

$$\begin{array}{ccc}
L^+ & \xleftarrow{r^{+\bullet}} \text{dom}(r^+) \xrightarrow{r^{+!}} & R^+ \\
& \searrow^{m^+} \quad \downarrow m' \quad (1) \quad \downarrow m'^\bullet & \\
& & C_i^T \xrightarrow{c_{i+1}^T} C_{i+1}^T
\end{array}$$

3. acrescentar as regras de  $\text{ApplRules}^{i+1}$  em  $\mathcal{U}_G^{i+1}(GG)$ :

Para todo  $(nr, r^T, m^T) \in \text{ApplRules}^{i+1}$ ,  $(nr, a) \in N_{i+1}$ , onde  $r_a = r^T$ ,  $pre_a = c_{i+1}^T \circ m^T$ ,  $post_a = m'^\bullet \circ s^T$ .

$$\begin{array}{ccc}
L^T & \xrightarrow{r^T} & R^T \\
\downarrow m^T & \searrow pre_a & \downarrow post_a \\
C_i^T & \xrightarrow{c_{i+1}^T} & C_{i+1}^T
\end{array}
\quad
\begin{array}{ccc}
R^T & \xrightarrow{s^T} & R^+ \\
\downarrow m'^\bullet & \swarrow & \\
C_{i+1}^T & & 
\end{array}$$

4. incluir as ações de  $N_i$  em  $\mathcal{U}_G^{i+1}(GG)$ :

Para todo  $(nr, b) \in N_i$ ,  $(nr, a) \in N_{i+1}$ , onde  $r_a = r_b$ ,  $pre_a = c_{i+1}^T \circ pre_b$ ,  $post_a = c_{i+1}^T \circ post_b$ .

$$\begin{array}{ccc}
L_a = L_b & \xrightarrow{r_a = r_b} & R_a = R_b \\
\downarrow pre_b & \swarrow post_b & \downarrow post_a \\
C_i^T & \xrightarrow{c_{i+1}^T} & C_{i+1}^T
\end{array}$$

Entre os *unfoldings*  $\mathcal{U}_G^k(GG)$  e  $\mathcal{U}_G^{k+1}(GG)$  há uma inclusão  $in_n$  induzida pelo morfismo de inclusão entre os grafos-núcleo  $C_k^T$  e  $C_{k+1}^T$   $\ominus$

DEFINIÇÃO 3.40. (**Semântica de Gramáticas de Grafos**) O *unfolding*  $Unf^G(GG)$  de uma gramática de grafos  $GG$  é definido como o colimite em  $\text{OccGG}$  do diagrama formado pelos *unfoldings* finitos de  $GG$  e as correspondentes inclusões.

A **semântica** de uma gramática de grafos  $GG$  é o seu *unfolding*  $Unf^G(GG)$ .  $\ominus$

Deste modo, o *unfolding* representa todas as computações determinísticas (derivações) e não determinísticas possíveis (através das ações). Sendo definido como colimite, é único a menos de isomorfismo.

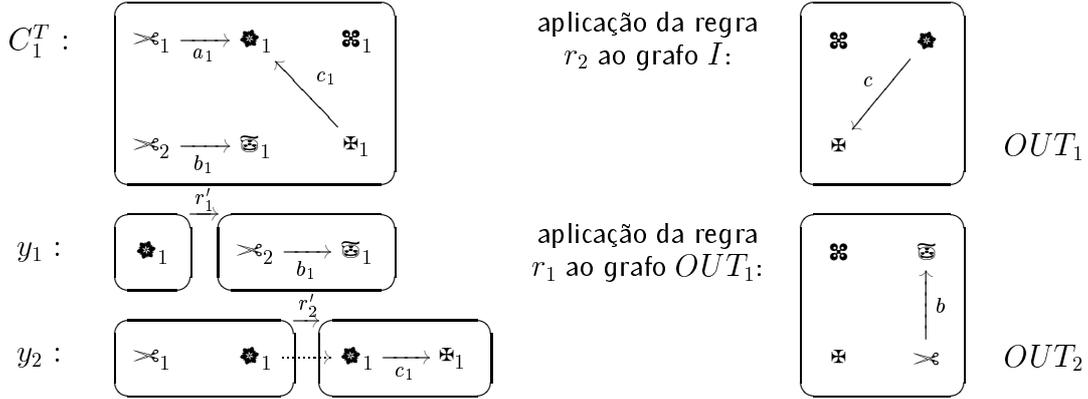
Exemplo 3.24: Como exemplo, será encontrado o *unfolding* de profundidade 3 da gramática  $GG$  da figura 3.19. Neste exemplo, os morfismos das regras/grafos inicial em relação ao grafo-núcleo serão representados usando itens com mesmos índices.

$\mathcal{U}_G^0(GG)$ : no *unfolding* de profundidade zero, o grafo-núcleo  $C_0^T$  é igual ao grafo inicial de  $GG$  e o grafo inicial  $I^1 \nearrow^T$  é o grafo duplamente tipado  $(I^T, id, I^T)$  e o conjunto de nomes de regras é vazio ( $N_0 = \emptyset$ ).

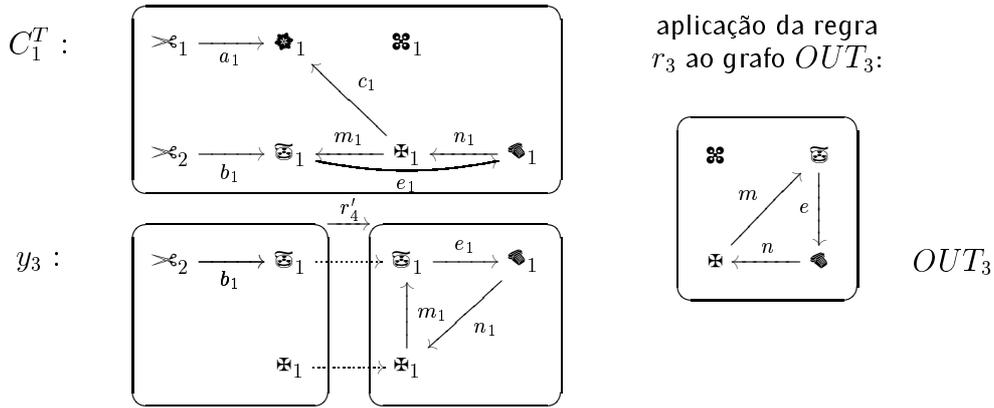
$$C_0^T : \quad \boxed{\begin{array}{ccc} \triangleright \varepsilon_1 & \xrightarrow{a_1} & \star_1 \quad \star\star_1 \end{array}}$$

$\mathcal{U}_G^1(GG)$ : o conjunto de regras aplicáveis em  $C_0^T$  é  $\text{ApplRules}^1 = \{(nr_1, r_1, m_1), (nr_2, r_2, m_2)\}$ , onde  $img(m_1) = \boxed{\star_1}$  e  $img(m_2) = \boxed{\triangleright \varepsilon_1 \quad \star_1}$ , que são grafos concorrentes,

uma vez que  $N_0 = \emptyset$ . Portanto,  $N_1 = \{y_1, y_2\}$ , onde  $y_1 = (nr_1, r'_1)$  e  $y_2 = (nr_2, r'_2)$ . Esta situação representa a aplicação da regra  $r_2$  ao grafo inicial de  $GG$ , seguida da aplicação de  $r_1$  (derivação seqüencial). Observe que se  $r_1$  for aplicada em primeiro lugar,  $r_2$  não pode mais ser aplicada ao grafo de saída resultante (apesar das regras poderem ser aplicadas em paralelo, só existe uma ordem na qual podem ser aplicadas seqüencialmente).



$U_G^2(GG)$ : o conjunto de regras aplicáveis em  $C_1^T$  é  $\{(nr_3, r_3, m_3)\}$ , onde a imagem do ocorrência  $m_3$  é o grafo-concorrente  $img(m_3) = \boxed{\begin{array}{c} >e_2 \xrightarrow{b_1} & \textcircled{1} & \textcircled{1} \\ & & \textcircled{1} \end{array}}$ . Observe que os itens de  $C_1^T$  que são eliminados por ações ( $*_4$  e  $>e_4$ ) não podem ser itens de grafo concorrente. Então  $N_2 = \{y_1, y_2, y_3\}$ , onde  $y_3 = (nr_3, r'_3)$ . A regra  $r_3$ , aplicada ao grafo  $OUT_2$ , resulta no grafo de saída  $OUT_3$  a seguir.



$U_G^3(GG)$ : a única regra que pode ser aplicada ao grafo  $OUT_3$  é  $r_4$ . Portanto, o conjunto de regras aplicáveis em  $C_2^T$  é  $\{(nr_4, r_4, m_4)\}$ , onde  $img(m_4) = \boxed{\begin{array}{c} & \textcircled{1} & \xrightarrow{e_1} & \textcircled{1} \\ & \textcircled{1} & & \end{array}}$ . Então  $N_3 = \{y_1, y_2, y_3, y_4\}$ , onde  $y_4 = (nr_4, r'_4)$ . A regra  $r_4$ , aplicada ao grafo  $OUT_3$ , resulta no grafo-estado  $OUT_4$ . Note que os grafos-estado  $OUT_1, OUT_2, OUT_3, OUT_4$  e  $I$  são subgrafos do grafo-núcleo  $C_3^T$ . A única regra que pode ser aplicada ao grafo-estado  $OUT_4$  é  $r_1$  e, em seqüência,  $r_3$ , que são as regras que serão usadas para construir  $U_G^4(GG)$  e  $U_G^5(GG)$ , respectivamente.

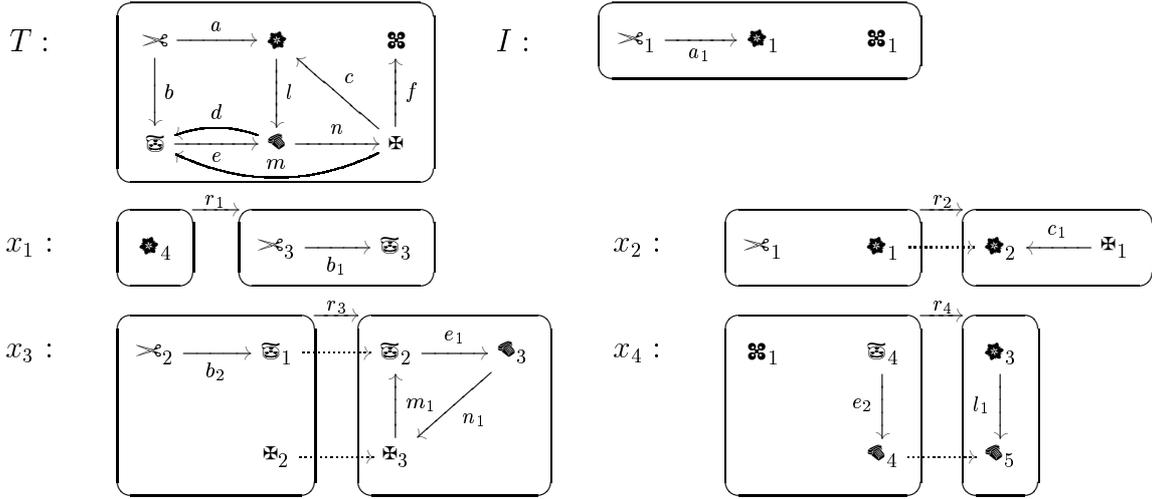
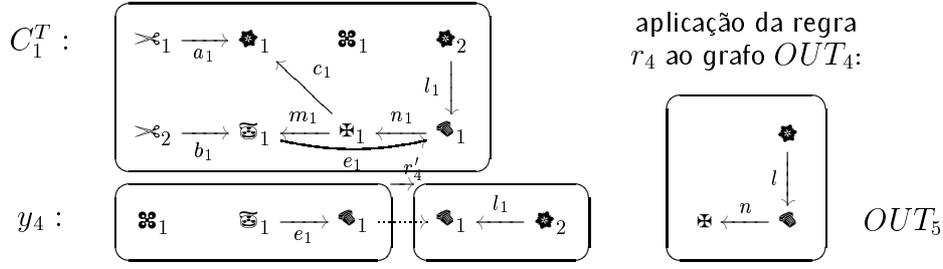


FIGURA 3.19- Gramática de Grafos  $GG$ .



☺

Pelo fato do *unfolding* de uma gramática de grafos ser único a menos de isomorfismo,  $GG$  terá várias semânticas isomórficas, fato este que não ocorre nas redes L/T, que possuem *unfoldings* determinísticos. Para definir a semântica como um funtor, serão definidas as gramáticas de grafos de ocorrência abstratas, que são definidas como uma classe de equivalência onde os elementos da classe estão relacionados por morfismos prefixos (definição 3.34) que são isomorfismos. Definidos desta maneira, as gramáticas de uma destas classes usarão as mesmas regras, possuirão o mesmo grafo inicial e terão grafos-núcleo isomórficos. Esta condição para os grafos-núcleo (serem isomórficos) reflete o fato das derivações serem únicas a menos de isomorfismo.

DEFINIÇÃO 3.41. **(Gramática de Grafos de Ocorrência Abstrata)** Uma gramática de grafos de ocorrência abstrata é uma classe de equivalência definida por:

$$[Occ] = \{Occ' | p : Occ' \rightarrow Occ \text{ é um isomorfismo e um morfismo prefixo}\}$$

Um morfismo  $[f] : [Occ_1] \rightarrow [Occ_2]$  é definido por

$$[f] = \{g | \mathcal{F}(g) = \mathcal{F}(f)\}$$

onde  $f, g : Occ_1 \rightarrow Occ_2$  são morfismos em  $OccGG$ .

A categoria das gramática de grafos de ocorrência abstratas é denotada por  $\overline{OccGG}$ . O funtor  $\overline{Unf}^G : GG \rightarrow \overline{OccGG}$  é definido por:  $\overline{Unf}^G(GG) = \overline{Occ} \Leftrightarrow Unf^G(GG) = Occ$

☺

Observação:  $\overline{Unf^G}(\overline{OccGG})$  serão usados como sinônimo de  $Unf^G(OccGG)$  quando puder ser inferido, do contexto, ao qual se refere.

## 4 Transformando Gramáticas de Grafos em Redes L/T

Neste capítulo, as gramáticas de grafos serão relacionadas com as redes L/T através de dois funtores  $\mathbb{R}$  e  $\mathbb{UR}$ :

$$\begin{array}{ccc}
 \text{PTNets} & \xrightarrow{\text{Unf}^R} & \text{OccNets} \\
 \uparrow \mathbb{R} & & \uparrow \mathbb{UR} \\
 \text{GGD} & \xrightarrow{\text{Unf}^G} & \overline{\text{OccGGD}}
 \end{array}$$

GGD é a categoria de gramáticas de grafos discretas, que são objetos de GG tais que os grafos não possuem arcos e as regras não preservam itens, que são as características (a serem discutidas na seção 1) das gramáticas de grafos que serão transformadas em redes L/T: os vértices dos grafos serão mapeados para os lugares e as regras, mapeadas para as transições (tanto para a categoria sintática GGD quanto para a categoria semântica  $\overline{\text{OccGGD}}$ ). No capítulo 6, será mostrado que este diagrama comuta a menos de isomorfismo.

### Estrutura do Capítulo:

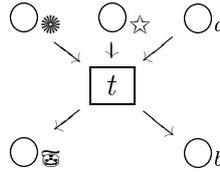
- seção 1: define como transformar uma gramática de grafos em uma rede L/T, analisando os casos;
- seção 2: mostra como transformar um morfismo entre gramáticas de grafos em morfismos entre redes L/T. Em seguida, é provado que  $\mathbb{R}$  é um funtor;
- seção 3: mostra como transformar uma gramática de grafos de ocorrência em uma rede de ocorrência. Demonstração de que o resultado da transformação é uma rede de ocorrência;
- seção 4: mostra como transformar morfismos entre gramáticas de grafos de ocorrência em morfismos entre redes de ocorrência.

### 4.1 Transformação dos Objetos de GG

Nas redes L/T, mudanças de estado são representadas por transições, que determinam a movimentação (ou melhor, eliminação/criação) de *tokens* na rede. Em gramáticas de grafos, este conceito é representado pelas regras, que podem eliminar, criar ou preservar itens. Então, uma regra será representada através de uma transição. Deste modo, preservação, eliminação e criação de itens terão que ser representados em termos de eliminação/criação de *tokens*. Surge então a questão de como representar a preservação de itens. Uma possibilidade seria “eliminação seguida de criação” (equivalente a “esquecer” o morfismo das regras). Mas isto implicaria em tornar equivalentes as regras  $x_1 : \langle \text{token} \rangle \xrightarrow{r_1} \langle \text{token} \rangle$  e  $x_2 : \langle \text{token} \rangle \xrightarrow{r_2} \langle \text{token} \rangle$ . Porém, estas regras possuem semânticas diferentes, pois, por exemplo, a regra  $x_3 : \langle \text{token} \rangle \xrightarrow{r_3} \langle \text{token} \rangle$  pode ser aplicada em paralelo com  $r_1$

(uma vez que eliminação de itens pode ocorrer em paralelo com criação), mas não o pode com  $r_2$ , pois eliminam o mesmo item  $\searrow$ , sendo mutuamente exclusivas. Disto, concluímos que não é conveniente simplesmente “esquecer” a preservação de itens, uma vez que se procura uma equivalência semântica entre redes L/T e gramáticas de grafos. Portanto, para uma gramática de grafos poder ser transformada numa rede L/T, nenhuma de suas regras pode preservar itens.

Para a representação de recursos em uma gramática de grafos, pode ser usado tanto vértices quanto arcos, mas nas redes L/T só é possível representá-los por meio dos lugares. Tem-se agora um outro problema: como representar os vértices e arcos em termos de lugares. Uma opção é simplesmente ignorar os arcos e transformar somente os vértices em lugares, o que leva a semânticas bem diferentes, uma vez que menos recursos precisam estar presentes para o disparo da transição equivalente. Outro problema que surge neste caso é uma regra deixar de ser regra: por exemplo, ignorando o arco da regra  $x_4 : \left( \text{C} \xrightarrow{r_4} \text{C} \right)$ , chega-se à “regra”  $x_5 : \left( \text{C} \xrightarrow{r_5} \text{C} \right)$ , que não é mais uma regra, uma vez que é um isomorfismo. Outra opção é transformar tanto arcos quanto vértices em lugares. Neste caso, a regra  $x_6 : \left( \text{C} \xrightarrow{a} \text{C} \right)$  seria transformada na transição  $t$ :



Uma possível marcação válida para uma rede com esta transição é  $\text{C} \oplus a$ , o que implicaria em um grafo-estado numa gramática equivalente a tal rede que apresente um arco sem destino, i.e., apresenta recursos representados por  $\left( \text{C} \xrightarrow{a} \right)$ , o que, pela definição de grafo (deste trabalho), não é possível, uma vez que os arcos precisam ter origem e destino. Portanto, neste caso as semânticas também serão diferentes. Como um dos objetivos deste trabalho é obter uma equivalência semântica entre gramáticas de grafos e redes L/T, os grafos usados na gramática a ser transformada devem possuir apenas vértices.

Os (tipos dos) possíveis itens que podem ocorrer nos grafos das regras ou no grafo inicial de uma gramática de grafos são exatamente aqueles do grafo-tipo. Nas redes L/T, esta idéia é representada pelo conjunto de lugares, i.e., só se pode ter *tokens* nos lugares que pertencem ao conjunto de lugares. Portanto, o grafo-tipo será transformado no conjunto de lugares.

Os morfismos de tipo tanto das regras quanto do grafo inicial serão usados para encontrar a multiplicidade dos *tokens* na rede L/T resultante da transformação: se um elemento (vértice, pois estão sendo considerados apenas os grafos que não têm arcos)  $x$  do grafo-tipo  $T$  de um grafo tipado  $(G, t, T)$  da gramática de grafos é imagem de  $n$  elementos do grafo  $G$ ,  $n$  será a multiplicidade do “lugar”  $x$  (i.e.,  $t(x_1) = \dots = t(x_n), x_i \in G, 1 \leq i \leq n$  e  $x \in T$ ). Uma vez que os lugares em uma rede L/T admitem multiplicidade finita,  $n$  deve ser finito.

A seguir, será definida uma categoria GGD onde os objetos são os objetos de GG tais que os grafos não possuam arcos e as regras não preservem itens e será definido, então, um funtor  $\mathbb{R}$  desta categoria para a categoria PTNets.

DEFINIÇÃO 4.1. **(Gramáticas de Grafos Discretas)** Seja  $GG = (T, I^T, N, n)$  uma gramática de grafos com grafo-tipo  $T$ .  $GG$  será uma gramática de grafos discreta se, e somente se, satisfizer as seguintes condições:

- 1: Os conjuntos de arcos dos grafos são vazios (i.e., nenhum grafo de  $GG$  deve possuir arcos);
- 2:  $\forall x \in N, n(x) = (\emptyset, id_T)$ , i.e., os morfismos  $r = (g, t) : L^T \rightarrow R^T$  em  $TGraphP(T)$  onde  $g = \emptyset$  e  $t = id_T$ , com a restrição de que  $\forall x \in T$ , os conjuntos  $\{y \in L | t^L(y) = x\}$ ,  $\{y \in R | t^R(y) = x\}$  e  $\{y \in I | t^I(y) = x\}$  sejam finitos, i.e., existe um número finito de elementos que possuem tipo  $x$ .

A categoria  $GGD$  é definida como a subcategoria completa de  $GG$  que possui como objetos as gramáticas de grafos discretas.  $\odot$

DEFINIÇÃO 4.2. **(Transformação dos Objetos de  $GGD$ )** Sejam  $GG = (T, I^T, N, n)$  um objeto da categoria  $GGD$ ,  $I = (V_I, \emptyset, \emptyset, \emptyset)$  e  $t^I : I \rightarrow T$  o morfismo de tipo de  $I$ . A rede  $L/T \mathbb{R}(Occ) = R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  é definida por:

1.  $S_R = V_T$ , o conjunto de vértices do grafo-tipo  $T = (V_T, \emptyset, \emptyset, \emptyset)$ ;
2.  $(T_R, O_T) = N \cup \{O_T\}$ , onde  $O_T \notin N$  é o elemento apontado de  $T_R$ ;
3.  $\forall x \in N, n(x) = r^T : L^T \rightarrow R^T$ , sejam os morfismos de tipo  $t^L : L \rightarrow T$  e  $t^R : R \rightarrow T$ . Os morfismos  $p_R$  e  $d_R$  são definidos da seguinte maneira:  $\forall a \in V_T$  ( $Cd(A) =$  cardinalidade do conjunto  $A$ ),
  - $p_R(x)(a) = \begin{cases} Cd(\{y \in V_L | t^L(y) = a\}), & \text{se } a \in \text{img}(t^L) \\ 0, & \text{caso contrário} \end{cases}$
  - $d_R(x)(a) = \begin{cases} Cd(\{y \in V_R | t^R(y) = a\}), & \text{se } a \in \text{img}(t^R) \\ 0, & \text{caso contrário} \end{cases}$
4.  $p_R(O_T) = d_R(O_T) = O_{S_R}$ ;
5.  $\forall a \in V_T, u_R(a) = \begin{cases} Cd(\{y \in V_I | t^I(y) = a\}), & \text{se } a \in \text{img}(t^I) \\ 0, & \text{caso contrário} \end{cases}$

$\odot$

Exemplo 4.1: Para transformar a gramática  $GG_0$  da figura 4.1 numa rede  $L/T R_0$ , o seguinte passo pode ser seguido: como  $T_0 = (\{a, b, c, d, e\}, \emptyset, \emptyset, \emptyset)$ ,  $S_{R_0} = \{a, b, c, d, e\}$  e, como  $N = \{v_1, v_2, v_3, v_4\}$ ,  $T_{R_0} = \{O_{T_0}, v_1, v_2, v_3, v_4\}$ . Os elementos que pertencem a  $\text{img}(t^{L_1})$ , onde  $L_1$  é o lado direito da regra  $v_1$ , são  $c$  e  $e$ . Portanto, estes serão os únicos que terão cardinalidade diferente de 0 em  $p_{R_0}(v_1)$ , a origem da transição  $v_1$ :  $p_{R_0}(v_1)(e) = Cd(\{e_1\}) = 1$ , uma vez que  $t^{L_1}(e_1) = e$  e  $p_{R_0}(v_1)(c) = Cd(\{c_1, c_2\}) = 2$ , dado que  $t^{L_1}(c_1) = c = t^{L_1}(c_2)$ . Portanto,  $p_{R_0}(v_1) = e \oplus 2c$ . Analogamente, para todas as transições:

$x$	$p_{R_0}(x)$	$d_{R_0}(x)$
$v_1$	$e \oplus 2c$	$a \oplus 2b$
$v_2$	$\oplus b$	$d \oplus c$
$v_3$	$2d$	$O_{S_1}$
$v_4$	$b \oplus a$	$c \oplus e$

A rede  $L/T$  resultante desta transformação é a rede  $R_0$  da figura 4.2.  $\odot$

A prova de que, para uma gramática de grafos  $GG$  que seja objeto de  $GGD$ ,  $\mathbb{R}(GG)$  é uma rede  $L/T$ , é simples: é necessário provar que as condições da definição 2.5 são satisfeitas, i.e., que  $p_R$  e  $d_R$  são morfismos entre conjuntos apontados, que  $u_R$  é um elemento de  $S_R^\oplus$  e que as pré-condições das transições não são vazias.

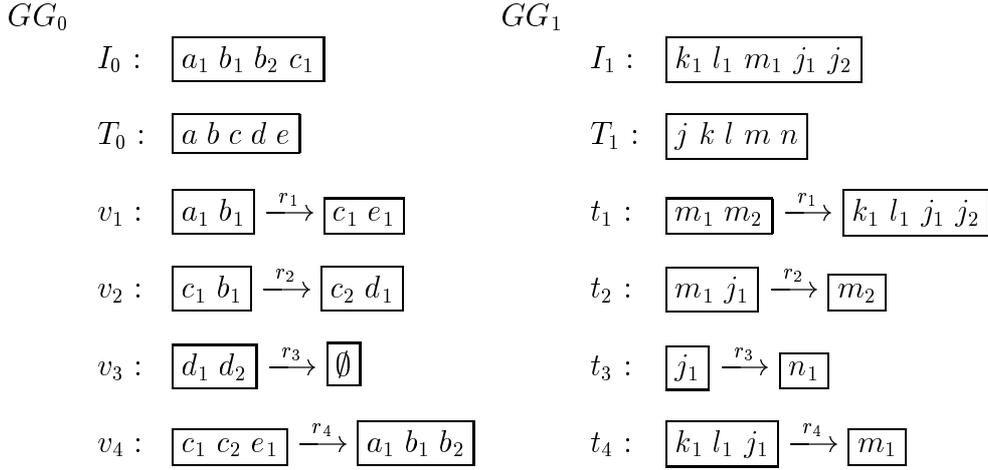
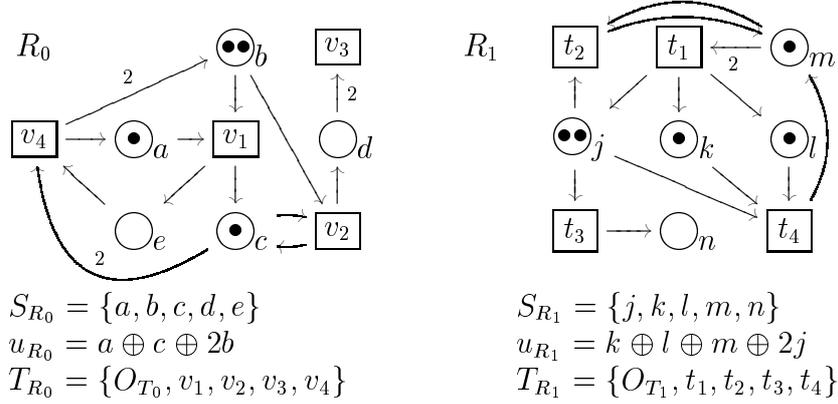


FIGURA 4.1- Exemplo de gramáticas de grafos discretas.

FIGURA 4.2- Redes L/T resultantes da aplicação do functor  $\mathbb{R}$  às gramáticas da figura 4.1.

PROPOSIÇÃO 4. Dada uma gramática de grafos discretas  $GG$ ,  $\mathbb{R}(GG)$  é uma rede L/T.

*Demonstração:*

- 1:  $T_R$  é um conjunto apontado, pela definição de  $\mathbb{R}$ .
- 2:  $p_R$  e  $d_R$  são morfismos entre conjuntos apontados, pois, pela definição,  $p_R$  e  $d_R$  são funções e  $p_R$  e  $d_R$  preservam os elementos apontados (i.e.,  $p_R(O_T) = d_R(O_T) = O_{S_R}$ ).
- 3:  $\forall t \in T, p_R(t), d_R(t)$  e  $u_R \in S_R^\oplus$ , i.e., são funções de  $S_R$  para  $\mathbb{N}$ : decorre da definição.
- 4:  $\forall t \in T, p_R(t) = O_{S_R} \Rightarrow t = O_T$ , pois como as regras da gramática são apenas consumidoras (i.e., não preservam itens), seus lados esquerdos são diferentes de  $\emptyset$ , i.e., possuem ao menos 1 vértice e, daí, as origens das transições são diferentes de  $O_{S_R}$ . Além disso, por definição,  $p_R(O_T) = O_{S_R}$ .

☺

## 4.2 Transformação dos Morfismos de GGD

Na transformação de uma gramática de grafos numa rede L/T, o grafo-tipo foi transformado no conjunto de lugares. Como um morfismo entre duas gramáticas é definido em função de um morfismo entre seus grafos-tipo, este morfismo será usado na transformação. A função que mapeia as transições será definida em termos da função entre as regras e o morfismo indutor será definido em termos do morfismo entre os grafos-tipo.

**DEFINIÇÃO 4.3. (Transformação dos Morfismos de GGD)** *Sejam  $GG_0 = (T_0, I_0, N_0, n_0)$  e  $GG_1 = (T_1, I_1, N_1, n_1)$  objetos de GGD e  $R_0 = (p_{R_0}, d_{R_0} : (T_{R_0}, O_{T_0}) \rightarrow S_{R_0}^\oplus, u_{R_0})$  e  $R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1})$  duas redes L/T tais que  $\mathbb{R}(GG_0) = R_0$  e  $\mathbb{R}(GG_1) = R_1$ . Seja  $(f_T, f_N)$  um morfismo entre  $GG_0$  e  $GG_1$ . O morfismo  $\mathbb{R}(f_T, f_N)$  entre  $R_0$  e  $R_1$  será o par  $(f : T_{R_0} \rightarrow T_{R_1}, h : S_{R_0} \rightarrow S_{R_1}^\oplus)$  definido por:*

1.  $\forall x \in T_{R_0}, f(x) = \begin{cases} f_N(x), & \text{se } x \in \text{dom}(f_N) \\ O_{T_1}, & \text{caso contrário} \end{cases}$
2.  $\forall a \in S_{R_0}, h(a) = \begin{cases} \bigoplus_{b \in A} b, & \text{onde } A = \{b \mid f_T(b) = a\} \text{ se } a \in \text{img}(f_T); \\ O_{S_1}, & \text{caso contrário.} \end{cases}$

Deste modo,  $h$  induz o morfismo entre multiconjuntos  $g : S_{R_0}^\oplus \rightarrow S_{R_1}^\oplus$ .

☺

**Exemplo 4.2:** Para exemplificar esta transformação, considere as gramáticas  $GG_0$  e  $GG_1$  da figura 4.1 e o morfismo  $(f_T, f_N) : GG_0 \rightarrow GG_1$  dado pelos morfismos parciais  $f_T : T_1 \rightarrow T_0$ ,  $f_T = \{j \mapsto b, k \mapsto a, l \mapsto a, m \mapsto c, n \mapsto \text{undef}\}$  e  $f_N : N_0 \rightarrow N_1$ ,  $f_N = \{v_1 \mapsto t_1, v_2 \mapsto t_2, v_3 \mapsto \text{undef}, v_4 \mapsto t_4\}$ . O morfismo  $\mathbb{R}(f_T, f_N)$  é dado pelo par  $(f, g)$ , onde  $f : T_{R_0} \rightarrow T_{R_1}$ ,  $f = \{v_1 \mapsto t_1, v_2 \mapsto t_2, v_3 \mapsto O_{T_1}, v_4 \mapsto t_4, O_{T_0} \mapsto O_{T_1}\}$  e  $h : S_{R_0} \rightarrow S_{R_1}^\oplus$ ,  $h = \{a \mapsto k \oplus l, b \mapsto j, c \mapsto m, d \mapsto O_{S_1}, e \mapsto O_{S_1}\}$ . ☺

Para provar que  $(f, g)$  é um morfismo em PTNets, é necessário provar que as condições da definição 2.6 são satisfeitas.

**PROPOSIÇÃO 5.** *Dadas duas gramáticas de grafos discretas  $GG_0$  e  $GG_1$  e  $(f_T, f_N) : GG_0 \rightarrow GG_1$  um morfismo entre  $GG_0$  e  $GG_1$ ,  $\mathbb{R}(f_T, f_N)$  é um morfismo entre as redes L/T  $\mathbb{G}(GG_0) = R_0 = (p_{R_0}, d_{R_0} : (T_{R_0}, O_{T_0}) \rightarrow S_{R_0}^\oplus, u_{R_0})$  e  $\mathbb{R}(GG_1) = R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1})$ .*

*Demonstração:*

- 1:  $f$  é um morfismo entre conjuntos apontados, pois segue definição de  $\mathbb{R}$  que  $f$  é uma função total e, como  $f(O_{T_0}) = O_{T_1}$  (uma vez que  $O_{T_0} \notin \text{dom}(f_N)$ ),  $f$  preserva os elementos apontados;
- 2:  $g(u_{R_0}) = u_{R_1}$ , i.e.,  $\forall a \in S_{R_1}, g(u_{R_0})(a) = u_{R_1}(a)$ . Como  $(f_T, f_N)$  é um morfismo em GG, o diagrama abaixo comuta,  $i$  é um isomorfismo e (1) é um pullback. Sejam  $a_1 \in S_{R_1}$  e  $m = u_{R_1}(a_1)$ . Há então dois casos: os casos de  $m$  ser igual ou maior que zero.

$$\begin{array}{ccccc}
 & & \text{---} f_T \text{---} & & \\
 & & \text{---} f_T! \text{---} & \text{---} f_T^* \text{---} & \\
 T_0 & \xleftarrow{f_T!} & \text{dom}(f_T) & \xrightarrow{f_T^*} & T_1 \\
 \uparrow t^0 & & \uparrow v & & \uparrow t^1 \\
 I_0 & \xleftarrow{u} & I' & \xleftarrow{i} & I_1
 \end{array}$$

**Caso 1:**  $m > 0$ . Pela construção de  $u_{R_1}, \exists b_1, \dots, b_m \in I_1 | t^{I_1}(b_j) = a_1, 1 \leq j \leq m$ . Como  $i$  é um isomorfismo,  $\exists X_1, \dots, X_m \in I' | i(b_j) = X_j$ . Dado que  $v$  é total, (2) comuta e  $f_T^*$  é uma inclusão, tem-se que  $v(X_j) = a_1$ , i.e.,  $a_1 \in \text{dom}(f_T)$ . Seja  $a_0 \in T_0 | f_T(a_1) = a_0$ . Pelo fato de que (1) é pullback,  $\exists c_1, \dots, c_m \in I_0 | u(X_j) = c_j$  e  $t^{I_0}(c_j) = a_0$ . Pela construção de  $u_{R_0}, u_{R_0}(a_0) = m$ . Como  $g(a_0)(a_1) = 1$  (por definição e do fato de que  $f_T(a_1) = a_0$ ) e pela definição 2.3 (multiconjuntos), tem-se que  $g(u_{R_0})(a_1) = m$ .

**Caso 2:**  $m = 0$ . Neste caso,  $a_1 \notin \text{img}(t^{I_1})$ . Suponha que  $g(u_{R_0})(a_1) > 0$ . Pela definição de  $g, \exists a_0 \in \llbracket u_{R_1} \rrbracket | f_T(a_1) = a_0$ , ie  $a_0 \in \text{img}(f_T!)$  e  $a_0 \in \text{img}(t^{I_0})$ . Como (1) é pullback,  $\exists X \in I' | f_T! \circ v(X) = t^{I_0} \circ u(X) = a_0$  e  $v(X) = a_1$ . Como  $i$  é um isomorfismo e (2) comuta,  $\exists b \in I_1 | t^{I_1}(b) = a_1$ , contrariando a hipótese de que  $a_1 \notin \text{img}(t^{I_1})$ . Portanto,  $g(u_{R_0})(a_1) = 0$ .

3:  $\forall a \in S_{R_0}, \forall b \in S_{R_1}, g(a)(b) \leq 1$ : segue pela construção de  $g$  e do fato de  $f_T$  ser uma função (parcial).

4:  $\forall a \in S_{R_1}$  e  $\forall r_0 \in T_{R_0}$ :

4.1:  $(p_{R_1} \circ f(r_0))(a) = (g \circ p_{R_0}(r_0))(a)$ . Seja  $r_0 \in N_0$ . Para provar esta igualdade, serão considerados os casos de  $r_0$  pertencer ou não ao domínio de  $f_N$ . Considere o diagrama a seguir, que comuta pelo fato de  $(f_T, f_N)$  ser um morfismo na categoria GG, onde os diagramas (1) e (3) são pullbacks e (2) é um pushout.

$$\begin{array}{ccccccc}
 T_1 & \xleftarrow{t^{L_1}} & L_1 & \xrightarrow{r_1} & R_1 & \xrightarrow{t^{R_1}} & T_1 \\
 \downarrow f_T^* & (4) & \downarrow i_L & (2) & \downarrow i_R & (5) & \downarrow f_T^* \\
 \text{dom}(f_T) & \xleftarrow{u_1} & L' & \xrightarrow{r'} & R' & \xrightarrow{u_2} & \text{dom}(f_T) \\
 \downarrow f_T! & (1) & \downarrow v_1 & v_2 & \downarrow & (3) & \downarrow f_T! \\
 T_0 & \xleftarrow{t^{L_0}} & L_0 & \xrightarrow{r_0} & R_0 & \xrightarrow{t^{R_0}} & T_0
 \end{array}$$

**Caso 1:**  $r_0 \notin \text{dom}(f_N)$ : neste caso,  $r'$  é um isomorfismo e, daí,  $L' = R' = \emptyset$  (pois  $r_0$  não preserva itens). Como  $i_L$  e  $i_R$  também são isomorfismos,  $L_1 = R_1 = \emptyset$ . Portanto,  $r_1 = O_{T_1}$ , i.e.,  $p_{R_1}(f(r_0))(a) = p_{R_1}(O_{T_1})(a) = 0$ . Como (1) é pullback e  $L' = \emptyset$ ,  $\text{img}(t^{L_0}) \cap \text{img}(f_T!) = \emptyset$  (i.e., nenhum elemento da imagem de  $f_T$  pertence à imagem de  $t^{L_0}$ ). Daí, pela definição de  $h, h \circ p_{R_0}(r_0) = O_{S_1}$ . Portanto,  $g(p_{R_0}(r_0))(a) = 0$ .

**Caso 2:**  $r_0 \in \text{dom}(f_N)$ : neste caso,  $r' = \emptyset$  não é um isomorfismo e a prova fica análoga ao item (3) acima.

4.2:  $(d_{R_1} \circ f(t))(a) = (g \circ d_{R_0}(t))(a)$ : prova de modo análogo ao item anterior.

5:  $\forall a, b \in S_{R_0}$  se  $\llbracket g(a) \rrbracket \cap \llbracket g(b) \rrbracket \neq \emptyset$  então  $a = b$ . Suponha que  $\llbracket g(a) \rrbracket \cap \llbracket g(b) \rrbracket \neq \emptyset$ . Seja  $c \in \llbracket g(a) \rrbracket$ , i.e.,  $f_T(c) = a$ . Como  $c \in \llbracket g(b) \rrbracket$ ,  $f_T(c) = b$ . Pelo fato de  $f_T$  ser uma função,  $a = b$ .

☺

O próximo passo é provar que  $\mathbb{R}$  é um funtor.

PROPOSIÇÃO 6. O funtor  $\mathbb{R} : \text{GGD} \rightarrow \text{PTNets}$  é bem definido.

*Demonstração:*

Sejam  $GG, GG_0, GG_1$  e  $GG_2$  gramáticas de grafos discretas e  $h_0 = (f_{T_0}, f_{N_0}) : GG_0 \rightarrow GG_1$  e  $h_1 = (f_{T_1}, f_{N_1}) : GG_1 \rightarrow GG_2$  morfismos na categoria GGD. Sejam

$\mathbb{R}(h_0) = (f_0, g_0)$ ,  $\mathbb{R}(h_1) = (f_1, g_1)$  e  $id_{GG} = (id_T, id_N)$  o morfismo identidade de  $GG$ . (Nesta prova, será usada a notação  $afb$  para indicar que  $f(a) = b$ .)

- 1:  $\mathbb{R}(h_0) : \mathbb{R}(GG_0) \rightarrow \mathbb{R}(GG_1)$ : Pela proposição 4,  $\mathbb{R}(GG_0)$  e  $\mathbb{R}(GG_1)$  são redes  $L/T$  e Pela proposição 11,  $\mathbb{R}(h_0)$  é um morfismo em  $PTNets$ . Segue diretamente da definição de  $\mathbb{R}$  que  $\mathbb{R}(h_0)$  é um morfismo entre  $\mathbb{R}(GG_0)$  e  $\mathbb{R}(GG_1)$ .
- 2:  $\mathbb{R}(h_1 \circ h_0) = \mathbb{R}(h_1) \circ \mathbb{R}(h_0)$ : como  $h_1 \circ h_0 : GG_0 \rightarrow GG_2$ , pela definição de  $\mathbb{R}$ ,  $\mathbb{R}(h_1 \circ h_0) : \mathbb{R}(GG_0) \rightarrow \mathbb{R}(GG_2)$ .
  - (transições): sejam  $t_0$  uma transição de  $\mathbb{R}(GG_0)$  e  $t_2$  uma transição de  $\mathbb{R}(GG_2)$ .
 
$$t_0 \mathbb{R}(h_1 \circ h_0) t_2 \Leftrightarrow t_0 (f_{N_1} \circ f_{N_0}) t_2 \text{ (pela definição de } \mathbb{R})$$

$$\Leftrightarrow \exists t_1 \in N_1 | (t_0 f_{N_0} t_1) \text{ e } (t_1 f_{N_1} t_2) \text{ (pela definição de morfismos em GGD)}$$

$$\Leftrightarrow (t_0 f_0 t_1) \text{ e } (t_1 f_1 t_2) \text{ (pela definição de } \mathbb{R})$$

$$\Leftrightarrow t_0 (f_1 \circ f_0) t_2 \text{ (pela composição de morfismos em PTNets)}$$

$$\Leftrightarrow t_0 (\mathbb{R}(h_1) \circ \mathbb{R}(h_0)) t_2$$
  - (lugares): sejam  $a_0$  um lugar de  $\mathbb{R}(GG_0)$  e  $a_2$  um lugar de  $\mathbb{R}(GG_2)$ . Pela definição de  $\mathbb{R}$ ,  $\mathbb{R}(h_1 \circ h_0)(a_0)$  é um conjunto, i.e.,
 
$$\mathbb{R}(h_1 \circ h_0)(a_0)(a_2) = 1 \Leftrightarrow (f_{T_1} \circ f_{T_0})(a_2) = a_0 \text{ (pela definição de } \mathbb{R})$$

$$\Leftrightarrow \exists a_1 \in T_1 \text{ (o grafo-tipo de } GG_1) | (a_1 f_{T_0} a_0) \text{ e } (a_2 f_{T_1} a_1) \text{ (pela definição de morfismos em GGD)}$$

$$\Leftrightarrow g_0(a_0)(a_1) = 1 \text{ e } g_1(a_1)(a_2) = 1 \text{ (pela definição de } \mathbb{R})$$

$$\Leftrightarrow g_0 \circ g_1(a_0)(a_2) = 1 \text{ (pela composição de morfismos entre multiconjuntos)}$$

$$\Leftrightarrow \mathbb{R}(h_1) \circ \mathbb{R}(h_0)(a_0)(a_2) = 1$$
- 3:  $\mathbb{R}(id_{GG}) = id_{\mathbb{R}(GG)}$ 
  - (transições): sejam  $t_1$  e  $t_2$  transições de  $\mathbb{R}(GG)$ .
 
$$t_1 \mathbb{R}(id_{GG}) t_2 \Leftrightarrow t_1 id_N t_2 \text{ (pela definição de } \mathbb{R})$$

$$\Leftrightarrow t_1 = t_2 \text{ (uma vez que } id_N \text{ é identidade)}$$

$$\Leftrightarrow t_1 id_{\mathbb{R}(GG)} t_2, \text{ uma vez que } t_1 \text{ e } t_2 \text{ transições de } \mathbb{R}(Occ).$$
  - (lugares): sejam  $a$  e  $b$  lugares de  $\mathbb{R}(GG)$ . Pela definição de  $\mathbb{R}$ ,  $\mathbb{R}(id_{GG})(b)$  é um conjunto, i.e.,
 
$$\mathbb{R}(id_{GG})(b)(a) = 1 \Leftrightarrow a id_T b$$

$$\Leftrightarrow a = b$$

$$\Leftrightarrow id_{\mathbb{R}(GG)}(b)(a) = 1, \text{ uma vez que } a \text{ e } b \text{ lugares de } \mathbb{R}(GG).$$

☺

### 4.3 Transformação dos Objetos de $\overline{OccGGD}$

Nesta seção, será visto como transformar uma gramática de grafos de ocorrência numa rede de ocorrência. Aqui valem as mesmas discussões já feitas a respeito de transformar arcos e preservação de itens. Será então definida uma categoria  $\overline{OccGGD}$  com as mesmas características de  $GGD$ : os grafos não possuem arcos e as regras não preservam itens.

**DEFINIÇÃO 4.4. (Gramáticas de Grafos de Ocorrência Discretas)** Seja  $Occ = (C^T, IN^{C^T}, N, n)$  um objeto de  $OccGG$ .  $Occ$  será uma gramática de grafos de ocorrência discreta se, e somente se, satisfizer as seguintes condições:

- 1: Os conjuntos de arcos dos grafos são vazios (i.e., nenhum grafo de  $Occ$  deve possuir arcos);

**2:**  $\forall x \in N, n(x) = a$  não preserva itens e  $\forall x \in C^T$ , os conjuntos  $\{y \in L_a | pre_a(y) = x\}$ ,  $\{y \in R_a | post_a(y) = x\}$  e  $\{y \in I | in_{GG}(y) = x\}$  sejam finitos.

A categoria  $\text{OccGGD}$  é definida como a subcategoria completa de  $\text{OccGG}$  que possui como objetos as gramáticas de grafos de ocorrência discretas. De modo análogo a gramáticas de ocorrência abstratas (definição 3.41), é definida a categoria  $\overline{\text{OccGGD}}$ .  $\odot$

A transformação de um objeto de  $\overline{\text{OccGGD}}$  numa rede de ocorrência é análoga à transformação de gramáticas de grafos discretas em redes L/T; porém, o grafo núcleo é transformado no conjunto de lugares e o grafo-tipo é “esquecido”.

**DEFINIÇÃO 4.5. (Transformação dos Objetos de  $\overline{\text{OccGGD}}$ )** Seja  $[Occ = (C^T, IN^{C \nearrow T}, N, n)]$  um objeto de  $\overline{\text{OccGGD}}$ . A rede de ocorrência  $\text{UR}([Occ]) = R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  é definida por:

1.  $T_R = N \cup \{O_T\}$ , onde  $O_T \notin N$ .
2.  $S_R = V_C$ , o conjunto de vértices do grafo-núcleo  $C^T$ .
3.  $\forall x \in C^T$ :

$$u_R(x) = \begin{cases} 1, & \text{se } x \in \text{img}(IN^{C \nearrow T}) \\ 0, & \text{caso contrário} \end{cases}$$

4.  $\forall t \in N$  e  $n(t) = a$ ,  $\forall x \in C^T$ :

$$p_R(t)(x) = \begin{cases} 1, & \text{se } x \in \text{img}(pre_a) \\ 0, & \text{caso contrário} \end{cases}$$

$$d_R(t)(x) = \begin{cases} 1, & \text{se } x \in \text{img}(post_a) \\ 0, & \text{caso contrário} \end{cases}$$

$\odot$

Deste modo, os objetos de  $\overline{\text{OccGGD}}$  que se diferenciam apenas pelo grafo-tipo são mapeados para a mesma rede de ocorrência. Qualquer gramática de grafos  $Occ'$  que pertença à classe de equivalência  $[Occ]$ , a igualdade  $\text{UR}([Occ']) = \text{UR}([Occ])$  se verifica, uma vez que as gramáticas da classe  $[Occ]$  estão relacionadas por isomorfismos que são morfismos prefixos (veja a definição 3.34).

A proposição a seguir estabelece uma equivalência entre as relações de conflito e de ocorrência em gramáticas de grafos de ocorrência e as relações de conflito e ocorrência em redes de ocorrência.

**PROPOSIÇÃO 7.** *Sejam  $Occ = (C^T, IN^{C \nearrow T}, N, n)$  um objeto de  $\overline{\text{OccGGD}}$  e uma rede L/T  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  tal que  $\text{UR}([Occ]) = R$ . Então, as seguintes implicações são verdadeiras para todo  $x_1, x_2 \in C$  ou  $x_1, x_2 \in N$ .*

1.  $x_1 \leq x_2 \Rightarrow x_1 \preceq x_2$
2.  $x_1 \preceq x_2 \Rightarrow x_1 \leq x_2$
3.  $x_1 \xrightarrow{\#} x_2 \Rightarrow x_1 \# x_2$
4.  $x_1 \xRightarrow{\#} x_2 \Rightarrow x_1 \# x_2$
5.  $x_1 \# x_2 \Rightarrow \exists x, x' \in C | x \xrightarrow{\#} x' \text{ e } x \leq x_1 \text{ e } x' \leq x_2$

*Demonstração:*

Serão provadas as asserções para os itens do grafo-núcleo. A prova para as ações é semelhante. Sejam  $x_1, x_2 \in C$ :

- 1: Suponha que  $x_1 \leq^T x_2$ . Como  $\leq^T$  é o fecho reflexivo e transitivo de  $\triangleleft^T$ , tem-se que  $\exists y_1, \dots, y_n \in C^T, n \geq 0 | x_1 \triangleleft^T y_1 \triangleleft^T y_2 \cdots y_n \triangleleft^T x_2$ . Pela definição de

- $\triangleleft^T$ ,  $\exists a_0, \dots, a_n \in N$  tais que  $a_0$  elimina  $x_1$  e cria  $y_1$ ,  $a_1$  elimina  $y_1$  e cria  $y_2$ ,  $\dots$ ,  $a_n$  elimina  $y_n$  e cria  $x_2$ , i.e.,  $x_1 \in \text{img}(\text{pre}_{a_0})$ ,  $y_1 \in \text{img}(\text{pre}_{a_1})$ ,  $\dots$ ,  $y_n \in \text{img}(\text{pre}_{a_n})$  e  $y_1 \in \text{img}(\text{post}_{a_0})$ ,  $\dots$ ,  $y_n \in \text{img}(\text{post}_{a_{n-1}})$ ,  $x_2 \in \text{img}(\text{post}_{a_n})$ . Pela definição de  $\mathbb{UR}$ ,  $x_1 \in \llbracket p_R(a_0) \rrbracket$ ,  $y_1 \in \llbracket p_R(a_1) \rrbracket$ ,  $\dots$ ,  $y_n \in \llbracket p_R(a_n) \rrbracket$  e  $y_1 \in \llbracket d_R(a_0) \rrbracket$ ,  $\dots$ ,  $x_n \in \llbracket d_R(a_{n-1}) \rrbracket$ ,  $x_2 \in \llbracket d_R(a_n) \rrbracket$ . Pela definição de  $\prec_1$ , tem-se que  $x_1 \prec_1 a_0$ ,  $y_1 \prec_1 a_1$ ,  $\dots$ ,  $y_n \prec_1 a_n$  e  $a_0 \prec_1 y_1$ ,  $\dots$ ,  $a_{n-1} \prec_1 y_n$ ,  $a_n \prec_1 x_2$ . Como  $\preceq$  é o fecho reflexivo e transitivo de  $\prec_1$ , tem-se que  $x_1 \preceq a_0 \preceq y_1 \preceq a_1 \cdots a_{n-1} \preceq y_n \preceq a_n \preceq x_2$ . Portanto  $x_1 \preceq x_2$ .
- 2: Suponha que  $x_1 \preceq x_2$ . Como  $\preceq$  é o fecho reflexivo e transitivo de  $\prec_1$ , tem-se que  $\exists t_0, \dots, t_n \in T_R$  e  $\exists b_1, \dots, b_n, n \geq 0 \mid x_1 \prec_1 t_0 \prec_1 b_1 \cdots b_n \prec_1 t_n \prec_1 x_2$ , i.e.,  $x_1 \in \llbracket p_R(t_0) \rrbracket$ ,  $b_1 \in \llbracket p_R(t_1) \rrbracket$ ,  $\dots$ ,  $b_n \in \llbracket p_R(t_n) \rrbracket$  e  $b_1 \in \llbracket d_R(t_0) \rrbracket$ ,  $\dots$ ,  $b_n \in \llbracket d_R(t_{n-1}) \rrbracket$ ,  $x_2 \in \llbracket d_R(t_n) \rrbracket$ . Pela definição de  $\mathbb{UR}$ , tem-se que  $x_1 \in \text{img}(\text{pre}_{t_0})$ ,  $b_1 \in \text{img}(\text{pre}_{t_1})$ ,  $\dots$ ,  $b_n \in \text{img}(\text{pre}_{t_n})$  e que  $b_1 \in \text{img}(\text{post}_{t_0})$ ,  $\dots$ ,  $b_n \in \text{img}(\text{post}_{t_{n-1}})$ ,  $x_2 \in \text{img}(\text{post}_{t_n})$ . Como as regras não preservam itens, elas eliminam os itens da imagem de  $\text{pre}$  e criam os itens da imagem de  $\text{post}$ . Pela definição de  $\triangleleft^T$ , tem-se que  $x_1 \triangleleft^T b_1 \triangleleft^T b_2 \cdots b_{n-1} \triangleleft^T b_n \triangleleft^T x_2$ . Como  $\leq^T$  é o fecho reflexivo e transitivo de  $\triangleleft^T$ , tem-se que  $x_1 \leq^T x_2$ .
- 3: Suponha que  $x_1 \xrightarrow{\#} x_2$ . Pela definição de  $\xrightarrow{\#}$ , existem ações  $a_1, a_2 \in N$ ,  $a_1 \neq a_2$  tais que  $a_1$  cria o item  $x_1$  (i.e.,  $x_1 \in \text{img}(\text{post}_{a_1})$ ),  $a_2$  cria o item  $x_2$  (i.e.,  $x_2 \in \text{img}(\text{post}_{a_2})$ ) e  $a_1 \xrightarrow{\#} a_2$  (i.e.,  $\text{img}(\text{pre}_{a_1}) \cap \text{img}(\text{pre}_{a_2}) \neq \emptyset$ ). Seja  $x_3 \in \text{img}(\text{pre}_{a_1}) \cap \text{img}(\text{pre}_{a_2})$ . Pela definição de  $\mathbb{UR}$ ,  $x_3 \in \llbracket p_R(a_1) \rrbracket$ ,  $x_3 \in \llbracket p_R(a_2) \rrbracket$ ,  $x_1 \in \llbracket d_R(a_1) \rrbracket$  e  $x_2 \in \llbracket d_R(a_2) \rrbracket$  (lembre-se que as regras não preservam itens). Portanto,  $a_1 \prec_1 x_1$  e  $a_2 \prec_1 x_2$ . Como  $\preceq$  é o fecho reflexivo e transitivo de  $\prec_1$ ,  $a_1 \preceq x_1$  e  $a_2 \preceq x_2$ . Como  $a_1 \neq a_2$  e  $\{x_3\} \subseteq \llbracket p_R(a_1) \rrbracket \cap \llbracket p_R(a_2) \rrbracket$ , tem-se que  $a_1 \not\approx a_2$  e, portanto,  $x_1 \# x_2$ .
- 4: Suponha que  $x_1 \xrightarrow{\#} x_2$ . Pela definição de  $\xrightarrow{\#}$ ,  $\exists x_3 \in C^T \mid x_1 \xrightarrow{\#} x_3$  e  $x_3 \leq x_2$ . Portanto,  $\exists a_1, a_2 \in N \mid a_1$  cria  $x_1$ ,  $a_2$  cria  $x_2$  e  $a_1 \xrightarrow{\#} a_2$ . Concluimos então que  $x_1 \# x_3$ ,  $x_3 \preceq x_2$  e  $x_1 \preceq x_1$ . Pela definição de  $\#$ ,  $\exists t_1, t_2 \in T_R \mid t_1 \not\approx t_2$  e  $t_1 \preceq x_1$  e  $t_2 \preceq x_3$ . Pela transitividade de  $\preceq$ ,  $t_2 \preceq x_2$  e, portanto,  $x_1 \# x_2$ .
- 5: Suponha que  $x_1 \# x_2$ . Pela definição de  $\#$  e  $\not\approx$ , tem-se que  $\exists a_1, a_2 \in T_R \mid \llbracket p_R(a_1) \rrbracket \cap \llbracket p_R(a_2) \rrbracket \neq \emptyset$ ,  $a_1 \neq a_2$ ,  $a_1 \preceq x_1$  e  $a_2 \preceq x_2$ . Seja  $x_3 \in \llbracket p_R(a_1) \rrbracket \cap \llbracket p_R(a_2) \rrbracket$ . Tem-se então que  $x_3 \prec_1 a_1$  e  $x_3 \prec_1 a_2$ . Como  $\preceq$  é o fecho reflexivo e transitivo de  $\prec_1$ , tem-se que  $x_3 \preceq x_1$  e  $x_3 \preceq x_2$ . Tem-se então que  $\exists t_1, \dots, t_n \in T_R$  e  $\exists b_1, \dots, b_n, n \geq 0 \mid a_1 \prec_1 b_1 \prec_1 t_1 \cdots b_n \prec_1 t_n \prec_1 x_1$  e  $\exists t'_1, \dots, t'_n \in T_R$  e  $\exists b'_1, \dots, b'_n, n \geq 0 \mid a_1 \prec_1 b'_1 \prec_1 t'_1 \cdots b'_n \prec_1 t'_n \prec_1 x_2$ . Pela transitividade de  $\preceq$ , tem-se que  $x_3 \preceq b_1 \cdots b_n \preceq x_1$  e  $x_3 \preceq b'_1 \cdots b'_n \preceq x_2$ . De  $x_3 \prec_1 a_1 \prec_1 b_1$  e  $x_3 \prec_1 a_2 \prec_1 b'_1$ , tem-se que  $x_3 \in \llbracket p_R(a_1) \rrbracket \cap \llbracket p_R(a_2) \rrbracket$ ,  $b_1 \in \llbracket d_R(a_1) \rrbracket$  e  $b'_1 \in \llbracket d_R(a_2) \rrbracket$ . Portanto, tem-se que  $x_3 \in \text{img}(\text{pre}_{a_1}) \cap \text{img}(\text{pre}_{a_2})$  (i.e.,  $a_1$  e  $a_2$  eliminam o mesmo elemento  $x_3$ ),  $b_1 \in \text{img}(\text{post}_{a_1})$  (i.e.,  $a_1$  cria  $b_1$ ) e  $b'_1 \in \text{img}(\text{post}_{a_2})$  (i.e.,  $a_2$  cria  $b'_1$ ). Portanto,  $b_1 \xrightarrow{\#} b'_1$ ,  $b'_1 \leq^T x_2$  e  $b_1 \leq^T x_1$ .

◻

A seguir, será provado que  $\mathbb{UR}([Occ])$  é uma rede de ocorrência, i.e., que satisfaz as condições da definição 2.13: é uma rede segura, acíclica e sem autoconflitos.

**PROPOSIÇÃO 8.** *Sejam  $[Occ = (C^T, IN^{C/\wedge^T}, N, n)]$  uma gramática de grafos em  $\overline{OccGGD}$ . A rede  $L/TR = \mathbb{UR}([Occ])$  é uma rede de ocorrência.*

*Demonstração:*

1:  $R$  é uma rede segura, i.e.,  $\forall t \in T_R, p_R(t)$  e  $d_R(T)$  são conjuntos e  $\forall v \in \mathcal{A}(R), v$  é um conjunto: pela definição de  $\mathbb{R}$ ,  $p_R$  e  $d_R$  são conjuntos. Seja  $v \in \mathcal{A}(R)$ . Então, existe uma seqüência de passos  $\alpha_0, \alpha_1, \dots, \alpha_n$  tais que  $u_R[\alpha_0] \succ v_0, \dots, u_n[\alpha_n] \succ v$ . Isto significa que  $v$  é destino de uma ou mais transições. Suponha que  $v$  não seja um conjunto, i.e.,  $\exists a \in \llbracket v \rrbracket | v(a) \geq 2$ . Seja  $n = v(a)$ . Existem dois casos: se  $a$  faz parte do destino de uma única transição ou mais de uma:

**Caso 1:**  $\exists t \in T_R | d_R(t)(a) = n$ : neste caso,  $d_R(t)$  não seria um conjunto, mas o é pela definição de  $\mathbb{UR}$ .

**Caso 2:**  $\exists t_1, \dots, t_n \in T_R | d_R(t_i)(a) = 1, 1 \leq i \leq n$  e  $n \geq 2$ : como  $U(Occ)$  é uma gramática de grafos de ocorrência,  $C_k^T$  é um grafo-núcleo. Pela definição de  $\mathbb{UR}$ ,  $a$  é um vértice do grafo-núcleo e é imagem do vértice  $a_i$  do lado direito da regra  $t_i$ . Pela definição de grafo-núcleo, só existe uma tal regra e, portanto,  $i = 1$ ; uma contradição, já que  $n \geq 2$ . Portanto,  $v$  é um conjunto.

2:  $a \in \llbracket u_R \rrbracket$  se, e somente se,  $\mathfrak{P}(a) = \emptyset$ , i.e., lugares do grafo inicial não são destino de nenhuma transição: seja  $a \in \llbracket u_R \rrbracket$ . Isto significa que  $a$  é um vértice de  $IN^{C_k} \nearrow^T$ . Como  $C_k^T$  é um grafo-núcleo, não existe regra  $r$  em  $Occ$  tal que  $a$  seja imagem de algum vértice do lado direito de  $r$ . Portanto, pela definição de  $\mathbb{UR}$ ,  $\nexists t \in T_R | a \in \llbracket d_R(t) \rrbracket$ , i.e.,  $\mathfrak{P}(a) = \emptyset$ .

3:  $\forall a \in S_R, Cd(\mathfrak{P}(a)) \leq 1$ , i.e., todo lugar da rede pertence ao destino de no máximo uma transição: suponha que  $\exists a \in S_R | Cd(\mathfrak{P}(a)) > 1$ . Sejam  $t_1, t_2 \in T_R$  tais que  $a \in \llbracket d_R(t_1) \rrbracket \cap \llbracket d_R(t_2) \rrbracket$ . Isto significa que existe um vértice  $a_1$  no lado direito da regra  $t_1$  e existe  $a_2$  no lado direito da regra  $t_2$  que possuem a mesma imagem  $a$  em  $C_k^T$ . Como  $C_k^T$  é um grafo-núcleo, isto não é possível. Portanto,  $Cd(\mathfrak{P}(a)) \leq 1$ .

4:  $\prec$  é irreflexiva: segue da proposição 7 e do fato de que  $\leq^\#$  é irreflexiva em uma gramática de grafos de ocorrência.

5:  $\forall t \in T_R$ , o conjunto  $\{t' \in T_R | t' \prec t\}$  é finito: segue do fato de que  $\forall a \in N$ , o conjunto  $Pre^{\leq^N}(a)$  é finito.

6: a relação  $\#$  (conflito binário) no conjunto  $T_R \cup S_R$  é irreflexiva: segue da proposição 7 e do fato de que  $\longleftrightarrow^\#$  é irreflexiva em uma gramática de grafos de ocorrência.

☺

Na proposição a seguir, será estabelecida a relação entre grafo concorrente e elementos concorrentes. Este resultado será usado para estabelecer um relacionamento entre as semânticas de gramáticas de grafos e redes L/T.

**PROPOSIÇÃO 9.** *Seja  $[Occ = (C^T, IN^{C \nearrow^T}, N, n)]$  um objeto de  $\overline{OccGGD}$  e uma rede L/T  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  tal que  $\mathbb{UR}([Occ]) = R$ . Seja  $G = (V_G, \emptyset, \emptyset, \emptyset)$  um subgrafo do grafo-núcleo de  $[Occ]$ . Então,  $G$  é um grafo concorrente se, e somente se,  $Co(V_G)$ .*

*Demonstração:*

1:  $(\Rightarrow)$  Suponha que  $G = (V_G, \emptyset, \emptyset, \emptyset)$  seja um subgrafo concorrente de  $C^T$ . Pela definição de  $Co$ , é necessário que provar que  $\forall x, y \in V_G, x$  co  $y$  e que o conjunto  $\{t' \in T_R | t' \preceq z\}$  é finito para todo  $z \in V_G$ . Pela definição de  $co$ ,  $x$  co  $y$  implica que  $(x \prec y$  ou  $y \prec x$  ou  $x \# y)$  é falso. Suponha que  $x$  co  $y$ . É

necessário então provar que  $x \prec y$  e  $y \prec x$  e  $x\#y$  são os três falsos para  $x, y \in V_G$ :

- 1.1: Como  $x \not\leq y$ , i.e.,  $x \leq y$  é falso, tem-se, pela proposição 7, que  $x \preceq y$  é falso. Como  $\preceq$  é o fecho reflexivo de  $\prec$ ,  $x \prec y$  é falso. Do mesmo modo,  $y \prec x$  é falso.
- 1.2: Para os itens de um grafo concorrente,  $x \not\longleftrightarrow y$  é falso. Suponha que  $x\#y$  seja verdadeiro. Pela proposição 7, tem-se que  $\exists b, b' \in T_R | b \xrightarrow{\#} b', b \xrightarrow{\#} x$  e  $b' \xrightarrow{\#} y$ . Portanto, existem ações  $a_1, a_2 \in N | a_1$  cria  $b$ ,  $a_2$  cria  $b'$ , e  $a_1 \xrightarrow{\#} a_2$ . Como as regras não preservam itens,  $a_2 \xrightarrow{\#} a_1$ , o que é uma contradição uma vez que  $\leq_A^\#$  é anti-simétrica. Portanto,  $x\#y$  é falso.
- 1.3: Como  $G$  é um grafo concorrente,  $\forall a \in \{b \in N | x \in \text{img}(\text{pre}_b)\}$ , o conjunto  $\text{Pre}^{\leq_A^\#}(a)$  é finito. Portanto, pela definição de  $\text{Pre}^{\leq_A^\#}$ , o conjunto  $A = \{a \in N | a \leq b \text{ e } x \in \text{img}(\text{pre}_b) \text{ e } b \in N \text{ e } x \in G\}$  é finito. Então, pela definição de  $\mathbb{U}R$  e pela proposição 7, o conjunto  $B = \{a \in N | a \preceq b \text{ e } x \in \llbracket d_R(b) \rrbracket \text{ e } b \in N \text{ e } x \in V_G\}$  também finito. Pela definição de  $\prec_1$ ,  $b \prec_1 x$  e, portanto,  $a \preceq x$ , i.e.,  $B = \{a \in N | a \preceq b \text{ e } b \prec_1 x \text{ e } b \in N \text{ e } x \in V_G\} = \{a \in N | a \preceq x \text{ e } x \in V_G\}$ .
- 2: ( $\Leftarrow$ ) Suponha que  $\text{Co}(V_G)$  é verdadeiro. Então o conjunto  $\{t' \in T_R | t' \prec t\}$  é finito e  $x \prec y$  e  $y \prec x$  e  $x\#y$  são os três falsos para  $x, y \in V_G$ . Sejam  $x, y \in V_G$ :
  - 2.1: como  $x \prec y$  é falso,  $x \preceq y$  também é falso. Pela proposição 7,  $x \leq y$  é falso.
  - 2.2: Suponha que  $x \longleftrightarrow y$  seja verdadeiro. Pela definição de  $\longleftrightarrow$ ,  $x \xrightarrow{\#} y$  e  $x \xrightarrow{\#} y$ . Pela proposição 7, tem-se que  $x\#y$ , uma contradição, uma vez que  $x\#y$  é falso. Portanto,  $x \not\longleftrightarrow y$  é falso, i.e., os itens de  $G$  não estão em conflito.
  - 2.3: Sejam  $a_1, a_2 \in N | a_1 \leq_A^\# a_2, a_1 \neq a_2$ . Suponha que  $a_2 \leq_A^\# a_1$ . Como a relação  $\xrightarrow{\#}$  é vazia para os elementos de  $A$ , tem-se que  $a_1 \leq a_2$  e  $a_2 \leq a_1$ . Pela proposição 7,  $a_1 \prec a_2$  e  $a_2 \prec a_1$ , uma contradição, uma vez que  $\prec$  é irreflexiva numa rede de ocorrência. Portanto,  $\leq_A^\#$  é anti-simétrica.
  - 2.4:  $\forall a \in \{b \in N | x \in \text{img}(\text{pre}_b)\}$ , o conjunto  $\text{Pre}^{\leq_A^\#(a)}$  é finito: pela definição 3.30,  $\text{Pre}^{\leq_A^\#(a)} = \{b \in N | b \leq_A^\# a\}$ . Como  $\leq_A^\#$  é o fecho reflexivo e transitivo de  $(\leq \cup \xrightarrow{\#})|_P$ , onde  $P = \{a' \in N | a' \leq a'' \text{ e } a'' \in A\}$ . Como  $\{t' \in T_R | t' \prec x\}$  é finito, tem-se, de modo análogo ao item 1.4 acima, que  $P$  também é finito. Portanto,  $P = \{a' \in N | a' \leq a'' \text{ e } a'' \in A\}$  e  $\text{Pre}^{\leq_A^\#}(a)$  são ambos finitos.

□

#### 4.4 Transformação dos Morfismos de $\overline{\text{OccGGD}}$

A transformação dos morfismos de  $\overline{\text{OccGGD}}$  é feita de modo análogo à transformação dos morfismos de  $\text{GGD}$ , porém o grafo-núcleo será usado no lugar do grafo-tipo, uma vez que o grafo-núcleo é usado para encontrar o conjunto de lugares.

**DEFINIÇÃO 4.6. (Transformação dos Morfismos de  $\overline{\text{OccGGD}}$ )** Sejam  $[Occ_0 = (C_0^{T_0}, IN^{C_0 \nearrow T_0}, N_0, n_0)]$  e  $[Occ_1 = (C_1^{T_1}, IN^{C_1 \nearrow T_1}, N_1, n_1)]$  objetos de  $\overline{\text{OccGGD}}$  e  $R_0 = (p_{R_0}, d_{R_0} : (T_{R_0}, O_{T_0}) \rightarrow S_{R_0}^\oplus, u_{R_0})$  e  $R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1})$  duas redes L/T tais que  $\mathbb{UR}([Occ_0]) = R_0$  e  $\mathbb{UR}([Occ_1]) = R_1$ . Seja  $[(f_T, f_N)]$  um morfismo entre  $[Occ_0]$  e  $[Occ_1]$ ,  $f_T = c^t : C_0^{T_0} \rightarrow C_1^{T_1}$ . O morfismo  $\mathbb{UR}([(f_T, f_N)])$  entre  $R_0$  e  $R_1$  será o par  $(f : T_{R_0} \rightarrow T_{R_1}, h : S_{R_0} \rightarrow S_{R_1}^\oplus)$  definido por:

1.  $\forall x \in T_{R_0}, f(x) = \begin{cases} f_N(x), & \text{se } x \in \text{dom}(f_N) \\ O_{T_1}, & \text{caso contrário} \end{cases}$
2.  $\forall a \in S_{R_0}, h(a) = \begin{cases} \bigoplus_{b \in A} b, & \text{onde } A = \{b | c(b) = a\} \text{ se } a \in \text{img}(c); \\ O_{S_1}, & \text{caso contrário.} \end{cases}$

Deste modo,  $h$  induz o morfismo entre multiconjuntos  $g : S_{R_0}^\oplus \rightarrow S_{R_1}^\oplus$ .

☺

Note que, para qualquer  $(f'_T, f'_N)$  pertencente à classe de equivalência  $[(f_T, f_N)]$ , o morfismo  $\mathbb{UR}([(f'_T, f'_N)])$  é único (veja a definição 3.41). Como  $\text{OccNets}$  é uma subcategoria de  $\text{PTNets}$ , a prova de que  $\mathbb{UR}$  é um funtor análoga à prova de que  $\mathbb{R}$  é um funtor (proposição 6). Note-se que  $\mathbb{UR}$  é como um funtor de esquecimento, uma vez que “esquece” um dos tipos das gramáticas e, conseqüentemente, um dos morfismos (de tipo).

## 5 Transformando Redes L/T em Gramáticas de Grafos

Neste capítulo, serão definidos os funtores  $\mathbb{G} : \text{PTNets} \rightarrow \text{GGD}$  e  $\mathbb{UG} : \text{OccNets} \rightarrow \overline{\text{OccGGD}}$ , estabelecendo um relacionamento entre redes L/T e gramáticas de grafos.

$$\begin{array}{ccc}
 \text{PTNets} & \xrightarrow{U_n f^R} & \text{OccNets} \\
 \mathbb{G} \downarrow & & \downarrow \mathbb{UG} \\
 \text{GGD} & \xrightarrow{U_n f^G} & \overline{\text{OccGGD}}
 \end{array}$$

Os lugares serão transformados em vértices e as transições em regras, de modo que a gramática de grafos resultante é tal que nenhum de seus grafos possui arcos (grafos discretos) e nenhuma regra preserva itens. Esta mesma representação é usada em [KOR 96], que relaciona redes de Petri com gramáticas de grafos, usando árvores como domínio semântico (relaciona apenas os objetos). No capítulo 6, será mostrado que  $\mathbb{UG}$  preserva a semântica. Será visto também que este diagrama não comuta pelo fato de as redes de ocorrência serem mais abstratas que as gramáticas de grafos de ocorrência.

### Estrutura do capítulo:

- seção 1: definição do objeto  $GG = \mathbb{G}(R)$  a partir de uma rede L/T  $R$  e prova de que  $GG$  é uma gramática de grafos;
- seção 2: transformação dos morfismos de  $\text{PTNets}$ , i.e., dado um morfismo entre as redes  $R_0$  e  $R_1$   $g : R_0 \rightarrow R_1$ , o morfismo  $\mathbb{G}(g) = f : \mathbb{G}(R_0) \rightarrow \mathbb{G}(R_1)$  é definido. Em seguida, a prova de que  $\mathbb{G}$  é um funtor;
- seção 3: transformação das redes de ocorrência em gramáticas de grafos de ocorrência e prova de que, para uma rede de ocorrência  $R$ ,  $\mathbb{UG}(R)$  é uma gramáticas de grafos de ocorrência.
- seção 4: definição de como transformar morfismos entre redes de ocorrência em morfismos entre gramáticas de grafos de ocorrência.

### 5.1 Transformação dos Objetos de PTNets

Como foi visto anteriormente, a marcação inicial numa rede L/T pode ser usada para representar o estado inicial de um sistema e as transições podem ser usadas para representar possíveis mudanças de estado do sistema. Seus equivalentes em gramáticas de grafos são o grafo inicial e as regras, respectivamente. Portanto, a marcação inicial será transformada no grafo inicial e as transições juntamente com origem e destino serão transformadas em regras.

Os únicos lugares que podem fazer parte de uma rede estão no conjunto de lugares da rede; por esta razão será transformado no tipo da gramática. A seguir estão as regras para a transformação. Note que nenhum grafo da gramática resultante possui arcos. Para a representação dos vértices dos grafos tipados, serão usados pares  $(a, n)$

onde  $a$  é um lugar e  $n$  um número entre 1 e a multiplicidade de  $a$  na rede  $L/T$ , de modo que o tipo de  $(a, n)$  seja  $a$  (i.e., a multiplicidade do lugar  $a$  é representada pelo número de itens que têm tipo  $a$ ).

**DEFINIÇÃO 5.1. (Transformação dos Objetos de PTNets)** Dada uma rede  $L/T$   $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ , o objeto  $\mathbb{G}(R) = (T, I^T, N, n)$  é definido da seguinte maneira:

1.  $T = (S_R, \emptyset, d_T, p_T)$ , onde  $p_T : \emptyset \rightarrow S_R$  e  $d_T : \emptyset \rightarrow S_R$ ;
2. Para  $\mu \in S_R^\oplus$ ,  $\mathcal{TR}(\mu)$  transforma um elemento  $\mu$  de  $S_R^\oplus$  no grafo tipado  $G^T = (G, t, T)$ , com morfismo de tipo  $t = (t_V, t_A) : G \rightarrow T$ :
  - $G = (V, \emptyset, \emptyset, \emptyset)$ , onde  $V = \bigcup \{ \{a\} \times \{1, \dots, \mu(a)\} \mid a \in S_R \}$ ;
  - $t_A : \emptyset \rightarrow \emptyset$ ;
  - $t_V : V \rightarrow S_R$  é a função dada por:  $\forall (a, n) \in V, t_V(a, n) = a$ ;
3.  $I^T = \mathcal{TR}(u_R)$ ;
4.  $N = T_R - \{O_T\}$ ;
5. a função  $n : N \rightarrow \text{Rules}(T)$  é definida por  $n(x) = r : L^T \rightarrow R^T = (g, t)$ , onde:
  - $t = id_T : T \rightarrow T$ ;
  - $g = (g_V : V_L \rightarrow V_R, g_A : A_L \rightarrow A_R) : L \rightarrow R = (\emptyset, \emptyset)$
  - $L^T = \mathcal{TR}(p_R(x))$ ;
  - $R^T = \mathcal{TR}(d_R(x))$ .

☺

**Exemplo 5.1:** A figura 5.1 mostra a transformação da rede  $R$  na gramática de grafos  $GG$  (note que os grafos contêm apenas vértices). A forma algébrica da rede  $R$  é dada por:

$R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$	$x$	$p_R(x)$	$p_R(x)$
$S_R = \{a, b, c, d\}$	$O_{T_0}$	$O_{S_0}$	$O_{S_0}$
$T_R = \{O_{T_0}, t_1, t_2, t_3\}$	$t_1$	$2a$	$2b \oplus c$
$\mu_R = 2a$	$t_2$	$2b$	$d$
	$t_3$	$b$	$a$

A gramática  $\mathbb{G}(R) = GG = (T, I^T, N, n)$  resultante é dada por:

1. grafo-tipo:  $T = (\{a, b, c, d\}, \emptyset, \emptyset, \emptyset)$ ;
2. grafo inicial:  $I^T = (G, t, T)$  onde  $G = (\{(a, 1), (a, 2)\}, \emptyset, \emptyset, \emptyset)$  e  $t(a, 1) = a = t(a, 2)$ ;
3. conjunto de nomes de regras:  $N = \{t_1, t_2, t_3\}$ ;
4. regras:
  - $n(t_1) = (\emptyset, id_T) = r_1 : (L_1, t^{L_1}, T) \rightarrow (R_1, t^{R_1}, T)$ , onde  
 $L_1 = (\{(a, 1), (a, 2)\}, \emptyset, \emptyset, \emptyset)$ ,  $t^{L_1}(a, 1) = a = t^{L_1}(a, 2)$ ,  
 $R_1 = (\{(b, 1), (b, 2), (c, 1)\}, \emptyset, \emptyset, \emptyset)$ ,  $t^{R_1}(b, 1) = b = t^{R_1}(b, 2)$  e  $t^{R_1}(c, 1) = c$ ;
  - $n(t_2) = (\emptyset, id_T) = r_2 : (L_2, t^{L_2}, T) \rightarrow (R_2, t^{R_2}, T)$ , onde  
 $L_2 = (\{(b, 1), (b, 2)\}, \emptyset, \emptyset, \emptyset)$ ,  $t^{L_2}(b, 1) = b = t^{L_2}(b, 2)$ ,  
 $R_2 = (\{(d, 1)\}, \emptyset, \emptyset, \emptyset)$  e  $t^{R_2}(d, 1) = d$ ;
  - $n(t_3) = (\emptyset, id_T) = r_3 : (L_3, t^{L_3}, T) \rightarrow (R_3, t^{R_3}, T)$ , onde  
 $L_3 = (\{(b, 1)\}, \emptyset, \emptyset, \emptyset)$ ,  $t^{L_3}(b, 1) = b$ ,  $R_3 = (\{(a, 1)\}, \emptyset, \emptyset, \emptyset)$  e  $t^{R_3}(a, 1) = a$ .

☺

Observação: A transição nula ( $O_T$ ) não é traduzida, pois isto resultaria numa regra onde o lado direito seria um grafo vazio, o que implicaria que a respectiva

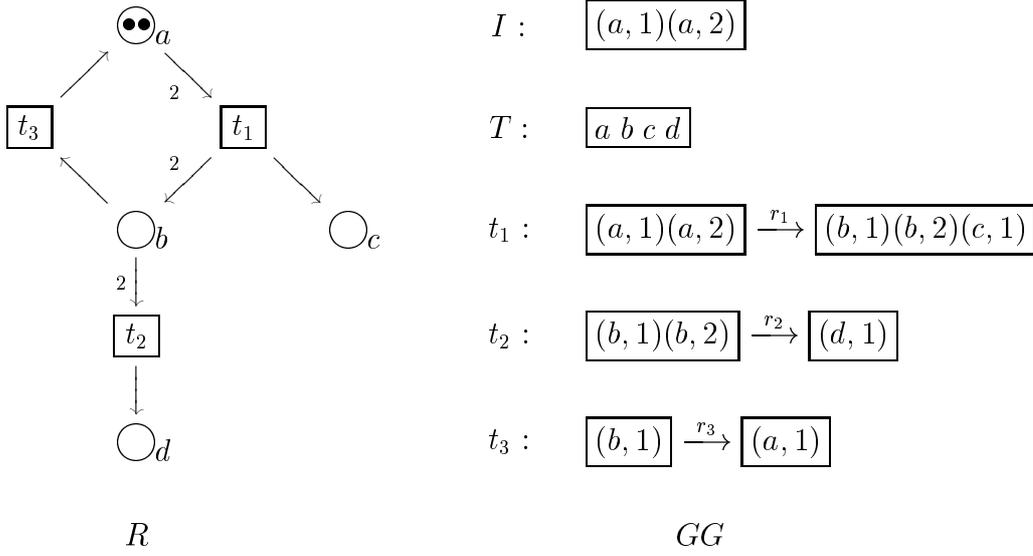


FIGURA 5.1- Transformação de uma rede L/T em uma gramática de grafos.

regra não eliminaria nenhum item, desrespeitando o fato de que uma regra **deve** eliminar algum item (ver definição 3.8).

O próximo passo é provar que, dada uma rede L/T  $R$  qualquer,  $\mathbb{G}(R)$  é uma gramática de grafos, i.e., que satisfaz as condições da definição 3.12. De acordo com esta definição, é necessário provar que  $T$  é um grafo e que  $I$  é um grafo tipado. Para as regras, é necessário provar que elas pertencem a  $Rules(T)$  (o conjunto das regras com tipo  $T$ ), i.e., que são morfismos parciais e injetores na categoria  $\mathbb{T}GraphP(T)$ . Como as regras não preservam quaisquer itens, para provar a parcialidade, será provado que o lado direito não é um grafo vazio (i.e., será provado que o conjunto de vértices dos lados esquerdos das regras não é vazio).

**PROPOSIÇÃO 10.** *Para qualquer rede L/T  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ ,  $\mathbb{G}(R) = GG = (T, I^T, N, n)$  é uma gramática de grafos.*

*Demonstração:*

- 1:  $T$  é um grafo:  $S_R$  e  $\emptyset$  são conjuntos e  $d_R$  e  $p_R$  são funções.
- 2:  $I^T = (I, t, T)$  é um grafo tipado em  $\mathbb{T}GraphP(T)$ , pois  $I$  e  $T$  são grafos e  $t = (t_V, t_A) : I \rightarrow T$  é um morfismo total entre grafos, pois  $t_A : \emptyset \rightarrow \emptyset$  e  $t_V : V_I \rightarrow V_T$  são totais (pela definição de  $\mathcal{TR}$ ). Além disso, os diagramas abaixo comutam fracamente (pois os conjuntos de arcos dos grafos  $I$  e  $T$  são vazios):

$$\begin{array}{ccc}
 \emptyset & \xrightarrow{t_A = \emptyset} & \emptyset \\
 p_I \downarrow & \geq & \downarrow p_T \\
 V_I & \xrightarrow{t_V} & V_T
 \end{array}
 \qquad
 \begin{array}{ccc}
 \emptyset & \xrightarrow{t_A = \emptyset} & \emptyset \\
 d_I \downarrow & \geq & \downarrow d_T \\
 V_I & \xrightarrow{t_V} & V_T
 \end{array}$$

- 3:  $n : N \rightarrow Rules(T)$  é uma função total pois, como  $p_R$  e  $d_R$  são funções totais  $L^T$  e  $R^T$  estão definidos para todo  $x \in N$ . Portanto,  $n$  é um morfismo total.
- 4:  $\forall x \in N, n(x) = r = (g, t) \in Rules(T)$ , i.e.,  $r$  é um morfismo parcial e injetor na categoria  $\mathbb{T}GraphP(T)$ :

- 4.1:  $L^T$  e  $R^T$  são grafos tipados: prova análoga ao item (2) acima.  
 4.2:  $r$  é um morfismo em  $\text{TGraphP}(T)$ , pois o diagrama (1) abaixo comuta fracamente em  $\text{GraphP}$  (vem do fato de que os conjuntos de vértices dos grafos  $L$ ,  $R$  e  $T$  são vazios, o diagrama (2) comuta fracamente e, como  $\text{dom}(g_V) = \emptyset$ , o diagrama (3) também comuta fracamente).

$$\begin{array}{ccc}
 L \xrightarrow{g} R & & \emptyset \xrightarrow{g_A} \emptyset \\
 \downarrow t^L & (1) & \downarrow t_A^L \\
 T \xrightarrow{id_T} T & & \emptyset \xrightarrow{id} \emptyset
 \end{array}
 \qquad
 \begin{array}{ccc}
 V_L \xrightarrow{g_V = \emptyset} V_R & & \\
 \downarrow t_V^L & (3) & \downarrow t_V^R \\
 V_T \xrightarrow{id} V_T & & 
 \end{array}$$

- 4.3:  $r = (g_V, g_A)$  é um morfismo injetor, pois  $g_V : V_L \rightarrow V_R$  e  $g_A : A_L \rightarrow A_R$  são, por construção funções vazias ( $\emptyset$ ).  
 4.4:  $r$  não é um morfismo total, pois como  $g_V : V_L \rightarrow V_R \equiv \emptyset$  (função vazia), o domínio de  $g_V$  também é vazio ( $\text{dom}(g_V) = \emptyset$ ). Suponha que  $V_L = \emptyset$ : Seja  $p_R(x) = \mu$  o elemento de  $S_R^\oplus$  que foi transformado em  $L^T$ . Como  $V_L = \bigcup \{\{a\} \times \{1, \dots, \mu(a)\} \mid a \in S_R\}$ , isto só é possível se  $S_R = \emptyset$  (neste caso,  $T_R = \{O_T\}$  e, daí,  $N = \emptyset$ ) ou se  $\forall a \in S_R, \mu(a) = 0$  (neste caso,  $\mu = O_S$ , ou seja,  $p_R(x) = O_S$ , o que implica que  $x = O_T$ ; mas  $O_T \notin N$ ): portanto,  $V_L \neq \emptyset$ , i.e.,  $g_V$  não é total.

□

## 5.2 Transformação dos Morfismos de PTNets

Quando uma rede L/T foi transformada numa gramática de grafos, o conjunto de lugares foi usado para formar o grafo-tipo. Como um morfismo entre gramáticas de grafos é definido em função de um morfismo entre os grafos-tipo das gramáticas, o morfismo  $g$  entre os lugares da rede será usado para definir o morfismo  $f_T$  entre os grafos-tipo das gramáticas, porém em “sentido contrário”, i.e., será usada a relação inversa de  $g$ , pois num morfismo  $(f_T, f_N) : GG_0 \rightarrow GG_1$ , o morfismo  $f_T$  **parte** do grafo-tipo de  $GG_1$  **para** o grafo-tipo de  $GG_0$ . Para a definição do morfismo  $f_N$ , será usado o morfismo  $f$  entre os conjuntos de transições das redes.

**DEFINIÇÃO 5.2. (Transformação dos Morfismos de PTNets)** *Sejam duas redes  $L/T R_0 = (p_{R_0}, d_{R_0} : (T_{R_0}, O_{T_0}) \rightarrow S_{R_0}^\oplus, u_{R_0})$  e  $R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1})$  e  $GG_0 = (T_0, I_0, N_0, n_0)$  e  $GG_1 = (T_1, I_1, N_1, n_1)$  gramáticas de grafos discretas tais que  $\mathbb{G}(R_0) = GG_0$  e  $\mathbb{G}(R_1) = GG_1$ . Seja  $(f, g) : R_0 \rightarrow R_1$  um morfismo entre as redes  $R_0$  e  $R_1$ . Da transformação dos objetos, tem-se que:  $T_0 = (S_{R_0}, \emptyset, \emptyset, \emptyset)$ ,  $T_1 = (S_{R_1}, \emptyset, \emptyset, \emptyset)$ ,  $N_0 = T_{R_0} - \{O_{T_0}\}$  e  $N_1 = T_{R_1} - \{O_{T_1}\}$ . O morfismo  $\mathbb{G}(f, g) = (f_T, f_N)$  entre as gramáticas de grafos  $GG_0$  e  $GG_1$  será o par  $(f_T, f_N) : GG_0 \rightarrow GG_1$ ,  $f_T = (g_V, g_A) : T_1 \rightarrow T_0$  e  $f_N : N_0 \rightarrow N_1$ , definido por:*

1.  $g_A : A_{T_1} \rightarrow A_{T_0}$  e  $g_A = \emptyset$
2.  $g_V : V_{T_1} \rightarrow V_{T_0}$  e  $\forall b \in S_{R_1}, g_V(b) = \begin{cases} a, & \text{se } b \in \llbracket g(a) \rrbracket; \\ \text{undef}, & \text{caso contrário.} \end{cases}$
3.  $\forall x \in N_0, f_N(x) = \begin{cases} f(x), & \text{se } f(x) \neq O_{T_1}; \\ \text{undef}, & \text{caso contrário.} \end{cases}$

□

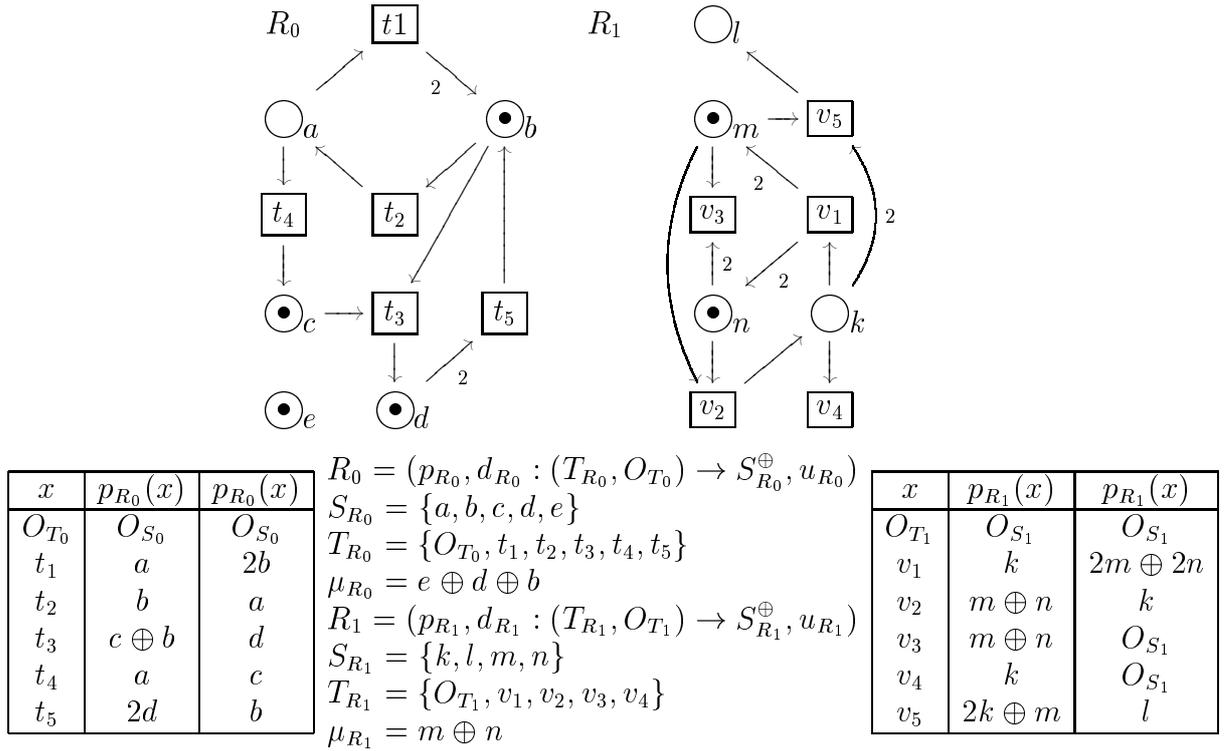


FIGURA 5.2- Duas redes L/T e suas representações algébricas.

Esta opção para a transformação dos morfismos justifica algumas das restrições impostas aos morfismos em PTNets (veja a definição 2.6). Por exemplo, suponha que  $g(a) = 2b \oplus c$ . É necessário representar  $g$  em termos de um morfismo entre grafos. Como pela transformação o grafo-tipo contém apenas os elementos  $b$  e  $c$ , não há como representar esta idéia (de que  $g(a)$  é um multi-conjunto, ao invés de um conjunto) por um morfismo.

Exemplo 5.2: A figura 5.3 mostra as gramáticas de grafos resultantes da aplicação do functor  $\mathbb{G}$  às redes  $R_0$  e  $R_1$  da figura 5.2. Um morfismo entre as redes  $R_0$  e  $R_1$  é o par  $(f, g) : R_0 \rightarrow R_1$  dado por:

$x$	$g(x)$
$a$	$k$
$b$	$m \oplus n$
$c$	$O_{S_1}$
$d$	$O_{S_1}$
$e$	$O_{S_1}$

$x$	$f(x)$
$t_1$	$v_1$
$t_2$	$v_2$
$t_3$	$v_3$
$t_4$	$v_4$
$t_5$	$O_{T_1}$

O morfismo  $\mathbb{G}(f, g) = (f_T, f_N) : GG_0 \rightarrow GG_1$  é dado por:

morfismo de tipos	morfismo de regras
$f_T(k) = a$ , pois $k \in \llbracket g(a) \rrbracket = \{k\}$	$f_N(t_1) = v_1$ , pois $f(t_1) = v_1$
$f_T(l) = \text{undef}$ , pois $\nexists x \in S_{R_0}   l \in \llbracket g(x) \rrbracket$	$f_N(t_2) = v_2$ , pois $f(t_2) = v_2$
$f_T(m) = b$ , pois $m \in \llbracket g(b) \rrbracket = \{m, n\}$	$f_N(t_3) = v_3$ , pois $f(t_3) = v_3$
$f_T(n) = b$ , pois $n \in \llbracket g(b) \rrbracket = \{n, m\}$	$f_N(t_4) = v_4$ , pois $f(t_4) = v_4$
	$f_N(t_5) = \text{undef}$ , pois $f(t_5) = O_{T_1}$

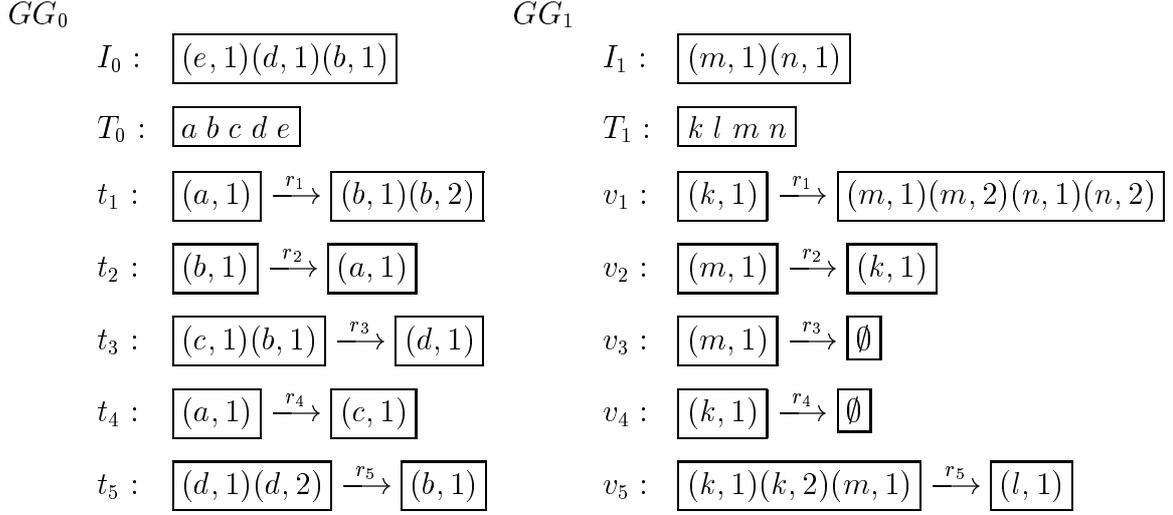


FIGURA 5.3- Gramáticas resultantes da aplicação do functor  $\mathbb{G}$  às redes da figura 5.2.

Para provar que  $\mathbb{G}(f, g) = (f_T, f_N)$  é um morfismo em  $\mathbb{G}\mathbb{G}$  é necessário provar que  $f_T$  é um morfismo entre grafos tipados e que as regras mapeadas por  $f_N$  são subregras (no nosso caso as funções de inclusão serão isomorfismos) das regras traduzidas pelo functor  $\mathcal{T}_{f_T}$  induzido por  $f_T$  (veja a definição 3.13).

PROPOSIÇÃO 11. *Dado um morfismo  $(f, g) : R_0 \rightarrow R_1$  na categoria PTNets,  $\mathbb{G}(f, g)$  é um morfismo na categoria GGD.*

*Demonstração:*

1:  $f_T$  é um morfismo em GraphP:

1.1:  $g_A$  e  $g_V$  são funções (parciais), pois como  $g_A = \emptyset$ ,  $g_A$  é trivialmente um morfismo. A funcionalidade de  $g_V$  é implicada pela definição dos morfismos entre redes  $L/T$ : sejam  $a \in V_{T_1}$ ,  $a_0, a_1 \in V_{T_0}$ ,  $g_V(a) = a_0$  e  $g_V(a) = a_1$ ; isto implica que  $a \in \llbracket g(a_0) \rrbracket$  e  $a \in \llbracket g(a_1) \rrbracket$ . Como  $\llbracket g(a_0) \rrbracket \cap \llbracket g(a_1) \rrbracket \neq \emptyset$  tem-se que  $a_0 = a_1$  (pela definição 2.6). Portanto,  $g_V$  é um morfismo.

1.2: os diagramas abaixo comutam fracamente pois  $g_V \circ p_{T_0} \circ g_A^* = p_{T_1} \circ g_A! = g_V \circ d_{T_0} \circ g_A^* = d_{T_1} \circ g_A! = \emptyset$ .

$$\begin{array}{ccc}
 \emptyset & \xrightarrow{g_A = \emptyset} & \emptyset \\
 p_{T_0} \downarrow & \geq & \downarrow p_{T_1} \\
 S_{R_0} & \xrightarrow{g_V} & S_{R_1}
 \end{array}
 \qquad
 \begin{array}{ccc}
 \emptyset & \xrightarrow{g_A = \emptyset} & \emptyset \\
 d_{T_0} \downarrow & \geq & \downarrow d_{T_1} \\
 S_{R_0} & \xrightarrow{g_V} & S_{R_1}
 \end{array}$$

2:  $f_N$  é uma função parcial: segue do fato de que o primeiro componente de um morfismo entre redes  $L/T$  ser uma função: sejam  $t' \in \text{dom}(f_N)$ ,  $t_0, t_1 \in T_0$ .  $f_N(t') = t_0$  e  $f_N(t') = t_1 \Leftrightarrow f(t') = t_0$  e  $f(t') = t_1$ . Como  $f$  é uma função,  $t_0 = t_1$ .

3: Subcomutatividade: é necessário provar que o diagrama abaixo possui a propriedade da subcomutatividade, i.e., que  $\overline{n_1} \circ f_N \subseteq^R \mathcal{R}_{f_T} \circ \overline{n_0}$ :

$$\begin{array}{ccc}
N_0 & \xrightarrow{\overline{n_0}} & \mathcal{R}(T_0) \\
f_N \downarrow & \subseteq^R & \downarrow \mathcal{R}_{f_T} \\
N_1 & \xrightarrow{\overline{n_1}} & \mathcal{R}(T_1)
\end{array}$$

Para isto, sejam:

- $t_0 \in N_0$
- $n_0(t_0) = r_0 : L_0^{T_0} \rightarrow R_0^{T_0}$ ;
- $\mathcal{T}_{f_T}(r_0) = r' : L' \rightarrow R'$ ,  $\mathcal{T}_{f_T}(L_0) = L'$  e  $\mathcal{T}_{f_T}(R_0) = R'$ , onde os diagramas (1) e (2) abaixo são pullbacks em Graph. Pela transformação de  $t_0$ ,  $r_0 = ((\emptyset : V_L \rightarrow V_R, \emptyset), id)$ . Portanto,  $r' = ((\emptyset : V_{L'} \rightarrow V_{R'}, \emptyset), id)$ .

$$\begin{array}{ccc}
L_0 & \xrightarrow{t^{L_0}} & T_0 \\
\uparrow u_1 & (1) & \uparrow f_N! \\
L' & \xrightarrow[v_1]{} & \text{dom}(f_N) \xrightarrow[f_N^*]{} T_1
\end{array}
\qquad
\begin{array}{ccc}
R_0 & \xrightarrow{t^{R_0}} & T_0 \\
\uparrow u_2 & (2) & \uparrow f_N! \\
R' & \xrightarrow[v_2]{} & \text{dom}(f_N) \xrightarrow[f_N^*]{} T_1
\end{array}$$

Há dois casos, caso  $t_0$  seja ou não um elemento do domínio de  $f_N$ :

**Caso 1:**  $t_0 \in \text{dom}(f_N)$ . Sejam:

- $t_1 = f_N(t_0)$ ;
- $n_1(t_1) = r_1 : L_1^{T_1} \rightarrow R_1^{T_1}$ ;

Portanto:

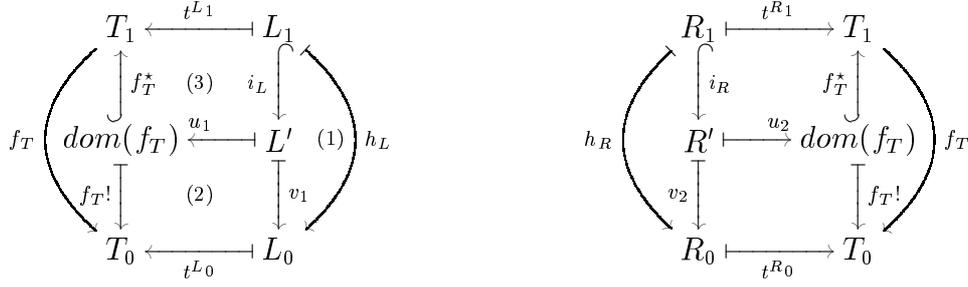
$$\overline{n_1} \circ f_N(t_0) = \overline{n_1}(f_N(t_0)) = \overline{n_1}(t_1) = [r_1]$$

$$\mathcal{R}_{f_T} \circ \overline{n_0}(t_0) = \mathcal{R}_{f_T}(\overline{n_0}(t_0)) = \mathcal{R}_{f_T}([r_0]) = [\mathcal{T}_{f_T}(r_0)] = [r']$$

É necessário provar que  $r'$  não é um isomorfismo e que  $[r_1] \subseteq^R [r']$ , ou seja, que  $r_1 \subseteq^r r'$ . Para isto, é preciso encontrar, para o diagrama abaixo, morfismos  $i_L$  e  $i_R$  que sejam totais e injetores tal que (1) seja pushout. Como  $i_R$  é definido de maneira análoga, será definido apenas  $i_L$ .

$$\begin{array}{ccc}
L_1 & \xrightarrow{r_1} & R_1 \\
i_L \downarrow & (1) & \downarrow i_R \\
L' & \xrightarrow{r'} & R'
\end{array}$$

3.1: definição de  $i_L$ : seja  $h_L : L_1 \rightarrow L_0$  o morfismo definido por  $h_L(a, n) = (c, n) \Leftrightarrow f_T(a) = c$ . Será provado que  $h_L$  é uma função total e sobrejetora, que  $(c, n) \in L_0$  e que o diagrama externo abaixo comuta. Isto implica na existência do morfismo universal  $i_L : L_1 \rightarrow L'$  tal que (1) e (3) comutem. Será então provado que  $i_L$  é um isomorfismo, i.e., é total, injetor e sobrejetor.



3.1.1:  $h_L$  é uma função:  $f(a, n) = (c_0, n)$  e  $f(a, n) = (c_1, n) \Leftrightarrow f_T(a) = c_0$  e  $f_T(a) = c_1 \Leftrightarrow c_0 = c_1$  (pois  $f_T$  é uma função parcial).

3.1.2:  $h_L$  é um morfismo total:  $f(a, n) = \text{undef} \Leftrightarrow f_T(a) = \text{undef} \Leftrightarrow a \notin \text{dom}(f_T)$ . Como  $(a, n) \in L_1, a \in \llbracket p_{R_1}(t_1) \rrbracket$ , i.e.,  $a \in \llbracket p_{R_1}(f(t_0)) \rrbracket$ . Isto implica que  $p_{R_1}(t_1)(a) > 0$ . Pela condição 5 da definição 2.6,  $(p_{R_1} \circ f(t))(a) = (g \circ p_{R_0}(t))(a)$ , i.e.,  $g(p_{R_0}(t_0))(a) > 0$ . Portanto,  $a \in \llbracket g(p_{R_0}(t_0)) \rrbracket$ , i.e.,  $\exists b \in S_{R_0} | a \in \llbracket g(b) \rrbracket$  (pela definição 2.4). Isto implica que  $g_V(a) = b$ , ou seja,  $f_T(a) = b$ , contrariando a hipótese de que  $a \notin \text{dom}(f_T)$ . Portanto,  $f$  é total.

3.1.3:  $h_L$  é um morfismo sobrejetor: seja  $(a_0, n_0) \in L_0$ . É necessário provar que  $\exists (a_1, n_0) \in L_1 | h_L(a_1, n_0) = (a_0, n_0)$ . Pela definição de  $g$ ,  $a_0 \in \text{img}(f_T)$ , i.e.,  $a_0 \in \llbracket u_0 \rrbracket$  e  $f_T(a_1) = a_0$ . Como  $(p_{R_1} \circ f(t_0))(a) = (g \circ p_{R_0}(t_0))(a)$ ,  $a_1 \in \llbracket u_1 \rrbracket$  e  $u_0(a_0) = u_1(a_1)$ , i.e.,  $(a_1, a_0) \in L_1$ . Pela definição de  $h_L$ ,  $h_L(a_1, n_0) = (a_0, n_0)$ .

3.1.4:  $(c, n) \in L_0$ : pela construção de  $L_0$ , é necessário provar que  $p_{R_0}(t_0)(c) \geq n$ .  $f_T(a) = c \Leftrightarrow g_V(a) = c \Leftrightarrow a \in \llbracket g(c) \rrbracket \Rightarrow g(c)(a) = 1$  (pela condição 4, definição 2.6).

$(a, n) \in L_1 \Leftrightarrow a \in \llbracket p_{R_1}(t_1) \rrbracket \Leftrightarrow a \in \llbracket p_{R_1}(f(t_0)) \rrbracket \Leftrightarrow p_{R_1}(f(t_0))(a) > 0$

Pela construção de  $L_1$ ,  $n \leq p_{R_1}(t_1)(a)$ . Pela condição 5 da definição 2.6,  $p_{R_1}(f(t_0))(a) = g(p_{R_0}(t_0))(a)$ , i.e.,  $n \leq g(p_{R_0}(t_0))(a)$ . Pela definição 2.4,  $\exists d \in \llbracket p_{R_0}(t_0) \rrbracket | a \in \llbracket g(d) \rrbracket$ . Como  $d$  é único (pela condição 6 da definição 2.6),  $d = c$  e  $g(p_{R_0}(t_0))(g(c)) = p_{R_0}(t_0)(c)$ . Como  $a \in \llbracket g(c) \rrbracket$ ,  $g(p_{R_0}(t_0))(a) = p_{R_0}(t_0)(c)$ . Portanto,  $g(c)(a) = p_{R_0}(t_0)(c)$ . Como  $p_{R_1}(t_1) = g(p_{R_0}(t_0))$ , tem-se que  $p_{R_0}(t_0)(c) \geq n$ .

3.1.5: o diagrama externo comuta: claramente,  $f_T \circ t^{L_1} = t^{L_0} \circ h_L$ .

3.2:  $i_L$  e  $i_R$  são morfismos totais: como  $h_L$  é total e (2) comuta,  $i_L$  é total.

3.3:  $i_L$  é um morfismo injetor: suponha que  $i_L(a_0, n_0) = i_L(a_1, n_1) = X$ . Seja  $v_1(X) = (c, n)$ . Como (1) comuta,  $h_L(a_0, n_0) = (c, n)$  e  $h_L(a_1, n_1) = (c, n)$ , i.e.,  $n_0 = n = n_1$  e  $f_T(a_0) = c$  e  $f_T(a_1) = c$ . Como (3) comuta,  $f_T^* \circ u_1 \circ i_L = t^{L_1} \Leftrightarrow t^{L_1}(a_0, n) = a_0 = f_T^* \circ u_1(X) = t^{L_1}(a_1, n) = a_1$ .

3.4:  $i_L$  é um morfismo sobrejetor: seja  $X \in I'$ . Como  $u$  é total e  $h_L$  é sobrejetor,  $u(x) \in \text{img}(h)$ . Como (1) comuta,  $\exists a_1 \in L_1 | i_L(a_1) = X$ .

3.5: o diagrama (4) abaixo é pushout em  $\mathbb{T}\text{GraphP}(T)$ : para isto é preciso provar que (4) comuta e que, dado que (1) é pushout, provar que existe um isomorfismo  $h : R \rightarrow R'$ , i.e., mostrar que o morfismo  $h$  existe, que é total, injetor e sobrejetor. Como  $r_1 = (\emptyset : V_{L_1} \rightarrow V_{R_1}, \emptyset : \emptyset \rightarrow \emptyset)$  e  $r' = (\emptyset : V_{L'} \rightarrow V_{R'}, \emptyset : \emptyset \rightarrow \emptyset)$ , (4) comuta. Como (4) comuta e (1)

é *pushout*,  $h$  será o morfismo universal onde (2) e (3) comutam. Será provado que  $i$  é um isomorfismo. Como  $i_R$  também é um isomorfismo e  $i_R = i \circ h$ ,  $h$  também é um isomorfismo.

$$\begin{array}{ccc}
 L_1 & \xrightarrow{r_1} & R_1 \\
 \downarrow i_L & (4) & \downarrow i_R \\
 L' & \xrightarrow{r'} & R'
 \end{array}
 \qquad
 \begin{array}{ccc}
 L_1 & \xrightarrow{r_1} & R_1 \\
 \downarrow i_L & (1) & \downarrow i \\
 L' & \xrightarrow{r} & R \\
 & (2) & \searrow h \\
 & & R' \\
 & & \nearrow i_R \\
 & & R
 \end{array}$$

Como  $i_L$  é um isomorfismo, e  $r_1 = \emptyset$ ,  $r = \emptyset$ . Como (1) é *pushout*,  $i$  é sobrejetor. Como (3) comuta e  $i_R$  é um isomorfismo,  $i$  é total. Suponha que  $i$  não seja injetor, i.e.,  $a \neq b$  e  $i(a) = i(b)$ . Como (3) comuta,  $h(i(a)) = i_R(a)$  e  $h(i(b)) = i_R(b)$ . Como  $h$  é um morfismo,  $h(i(a)) = h(i(b))$ , i.e.,  $i_R(a) = i_R(b)$ , uma contradição, pois  $i_R$  é um isomorfismo. Portanto,  $i$  é um morfismo injetor, i.e., é um isomorfismo.

3.6:  $i_L$  e  $i_R$  são seguras: é necessário provar que  $\forall X \in L', t_{L'}(X) \in \text{img}(t^{L_1}) \Rightarrow \exists p \in L_1 | i_L(p) = X$ . Seja  $t^{L'} = f_T^* \circ u_1$ .

$t^{L'}(X) = a \in \text{img}(t^{L_1}) \Rightarrow \exists n | (a, n) \in L_1$ .  $t^{L_1}(a, n) = a \Leftrightarrow f_T^* \circ u_1 \circ i_L(a, n) = a \Leftrightarrow t^{L'} \circ i_L(a, n) = a \Leftrightarrow t^{L'}(i_L(a, n)) = a$ . Como (3) comuta,  $i_L(a, n) = X$ .

3.7:  $r'$  não é um isomorfismo: como  $r' = \emptyset$ , é necessário mostrar que  $L' \neq \emptyset$ , i.e.,  $V_{L'} \neq \emptyset$ . Claramente isto se verifica, dado que (3) comuta e  $L_1 \neq \emptyset$ .

**Caso 2:**  $t_0 \notin \text{dom}(f_N)$ .

3.8:  $r'$  é um isomorfismo: como  $r' = (\emptyset, id)$ , é necessário mostrar que  $L' = R' = \emptyset$ . Como, neste caso  $f_N(t_0) = \text{undef}$ ,  $f(t_0) = O_{T_1}$ . Por definição,  $p_{R_1}(O_{T_1}) = O_{S_1} = d_{R_1}(O_{T_1})$ . Como  $\llbracket p_{R_1} \circ f(t_0) \rrbracket = \llbracket g \circ p_{R_0}(t_0) \rrbracket$  e  $\llbracket p_{R_1}(O_{T_1}) \rrbracket = \llbracket p_{R_1}(f(t_0)) \rrbracket = \emptyset$ , tem-se que  $\llbracket g \circ p_{R_0}(t_0) \rrbracket = \emptyset$  e, portanto,  $\llbracket p_{R_1} \circ f(t_0) \rrbracket = \emptyset$ . No caso de  $p_{R_0} = \emptyset$ ,  $L_0 = \emptyset$ . Daí,  $L' = \emptyset$  dado que é um objeto *pullback*. Suponha que  $p_{R_0}(t_0) \neq \emptyset$ : seja  $a_0 \in \llbracket p_{R_0}(t_0) \rrbracket$ . Como  $\llbracket g \circ p_{R_0}(t_0) \rrbracket = \emptyset$ ,  $\llbracket g(a_0) \rrbracket = \emptyset$ , i.e.,  $a_0 \notin \text{img}(f_T)$ . Portanto,  $\text{img}(t^{L_0}) \cap \text{img}(f_N!) = \emptyset$  e, daí,  $L' = \emptyset$ , dado que é um objeto *pullback*.

4:  $I_1^{t_1} \cong \mathcal{T}_{f_T}(I_0^{T_0})$ , i.e., é preciso encontrar um isomorfismo  $i : I_1 \rightarrow I'$  no diagrama abaixo, onde (2) é *pullback* em *Graph*. Para isto, é necessário definir o morfismo  $i$  e provar que é total, injetor e sobrejetor.  $i$  e  $h$  são definidos de modo análogo a  $i_L$ , i.e.,  $i$  é total, injetor e sobrejetor.

$$\begin{array}{ccccc}
 & & \xrightarrow{f_T} & & \\
 T_0 & \xleftarrow{f_T!} & \text{dom}(f_T) & \xrightarrow{f_T^*} & T_1 \\
 \uparrow t^{L_0} & & \uparrow v & & \uparrow t^{I_1} \\
 I_0 & \xleftarrow{u} & I' & \xleftarrow{i} & I_1 \\
 & & \downarrow h & & \\
 & & (1) & & 
 \end{array}$$



O próximo passo é provar que  $\mathbb{G}$  é um funtor, o que pode ser feito de modo análogo à prova da proposição 6.

PROPOSIÇÃO 12. *O funtor  $\mathbb{G} : \text{PTNets} \rightarrow \text{GGD}$  é bem definido.*

### 5.3 Transformação dos Objetos de OccNets

Será usada uma idéia semelhante à transformação de redes L/T em gramáticas de grafos para transformar uma rede de ocorrência numa gramática de grafos de ocorrência: o conjunto de lugares é transformado no grafo-núcleo e também no grafo-tipo. Portanto, o grafo-tipo será o mesmo que o grafo-núcleo e o morfismo de tipo será a função identidade.

DEFINIÇÃO 5.3. (**Transformação dos objetos de OccNets**) *Dada uma rede de ocorrência decorada  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ , seja  $GG = \mathbb{G}(R) = (T_1, I^{T_1}, N_1, n_1)$ . O objeto  $\mathbb{U}\mathbb{G}(R) = [(C^T, IN^{C^T}, N, n)]$  é definido por:*

1.  $T = T_1$
2.  $C^T = (T, id_T, T)$
3.  $IN^{C^T} = (I^T, t', C^T)$ ,  $(t' = t^I, id_T)$ .
4. regras:  $N = \{(x, r) | x \in N_1 \text{ e } r = \beta(x)\}$ , onde  $\beta(x)$  é dado por:

$$\beta(x) = r^{t \nearrow t'} = (n_1(x), id_T) : L^{C^T} \rightarrow R^{C^T}$$

onde, para  $n_1(x) = r^T : L^T \rightarrow R^T$ :

$$\begin{cases} L^{C^T} = (L^T, t_1, C^T) \\ L^T = (L, t^L, T) \\ t_1 = (t^L, id_T) \end{cases} \quad e \quad \begin{cases} R^{C^T} = (R^T, t_2, C^T) \\ R^T = (R, t^R, T) \\ t_2 = (t^R, id_T) \end{cases}$$

5.  $n(x, a) = a$



A proposição a seguir estabelece uma equivalência entre as relações de conflito e de ocorrência em gramáticas de grafos de ocorrência e as relações de conflito e ocorrência em redes de ocorrência, tal como no capítulo anterior (proposição 7). A prova também é análoga e será omitida.

PROPOSIÇÃO 13. *Seja  $[Occ = (C^T, IN^{C^T}, N, n)]$  e  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  uma rede L/T tal que  $\mathbb{U}\mathbb{R}([Occ]) = R$ . Então, as seguintes implicações são verdadeiras para todo  $x_1, x_2 \in C$  ou  $x_1, x_2 \in N$ .*

1.  $x_1 \leq x_2 \Rightarrow x_1 \preceq x_2$
2.  $x_1 \preceq x_2 \Rightarrow x_1 \leq x_2$
3.  $x_1 \xrightarrow{\#} x_2 \Rightarrow x_1 \# x_2$
4.  $x_1 \xrightarrow{\#} x_2 \Rightarrow x_1 \# x_2$
5.  $x_1 \# x_2 \Rightarrow \exists x, x' \in C | x \xrightarrow{\#} x' \text{ e } x \leq x_1 \text{ e } x' \leq x_2$

O próximo objetivo é provar que, para uma rede de ocorrência  $R$  qualquer, o objeto  $[Occ] = \mathbb{U}\mathbb{G}(R)$  é uma gramática de grafos de ocorrência, satisfazendo as condições da definição 3.31.

PROPOSIÇÃO 14. *Dada uma rede de ocorrência  $R$ , o objeto  $[Occ = \mathbb{U}\mathbb{G}(R) = dtgg]$  é uma gramática de grafos de ocorrência.*

*Demonstração:*

1: *Occ é uma gramática de grafos duplamente tipada:*

- 1.1:  $C^T$  é um grafo tipado, pois  $T = C$  e o morfismo de tipo é a identidade.  
 1.2:  $IN^{C/\wedge T}$  é um objeto de  $DTGraphP(C^T)$ , i.e., é um grafo duplamente tipado, uma vez que  $t^I$ , o morfismo de tipo do grafo inicial, é total e o diagrama abaixo comuta.

$$\begin{array}{ccc} I & \xrightarrow{t^I} & C = T \\ t^I \downarrow & & \downarrow id_T \\ T & \xrightarrow{id_T} & T \end{array}$$

- 1.3:  $n$  é uma função total: segue da definição.  
 2:  $\forall a \in N, \leq_a^\#$  é anti-simétrica: seja  $a \in N$ . Suponha que  $\exists a_1, a_2 \in N | a_1 \neq a_2, a_1 \leq_a^\# a_2$  e  $a_2 \leq_a^\# a_1$ . Como  $\leq_a^\#$  é o fecho reflexivo e transitivo de  $(\leq \cup \xrightarrow{\#})|_{Pre \leq(a)}$ , tem-se que  $a_1 \leq a_2$  e  $a_2 \leq a_1$ , uma vez que os elementos de  $Pre \leq(a)$  não estão em conflito (se  $\exists x, y \in Pre \leq(a) | x \xrightarrow{\#} y$ , pela proposição 13,  $x \# y$  e como  $x \leq a$  e  $y \leq a$ , tem-se que  $a \# a$ , o que é uma contradição uma vez que uma rede de ocorrência não possui autoconflitos). Portanto,  $a_1 \preceq a_2$  e  $a_2 \preceq a_1$ . Como  $a_1 \neq a_2$ , tem-se que  $a_1 \prec a_2$  e  $a_2 \prec a_1$ . Pela transitividade de  $\prec$ , tem-se que  $a_1 \prec a_1$ , uma contradição, uma vez que  $\prec$  é irreflexiva numa rede de ocorrência. Portanto,  $\forall a \in N, \leq_a^\#$  é anti-simétrica.  
 3:  $\xleftrightarrow{\#}$  é irreflexiva: Suponha que, do contrário,  $\exists x \in N \cup C | x \xleftrightarrow{\#} x$ . Pela definição de conflito,  $x \xrightarrow{\#} x$ , i.e.,  $\exists y | x \xrightarrow{\#} y$  e  $y \leq x$ . Pela proposição 13,  $x \# x$ , uma contradição, uma vez que  $R$  é uma rede de ocorrência e, por conseguinte, livre de autoconflitos.  
 4:  $\forall a \in N, Pre \leq^N(a)$  é finito: segue do fato de que o conjunto  $\{t \in T_R | t \prec a\}$  é finito.  
 5:  $C^T$  é um grafo-núcleo: i.e.,  $\forall x \in C^T, \exists! y \in (I^T \uplus (\bigoplus_{a \in \text{img}(n)} R_a))$  tal que

$$x = \begin{cases} in_{GG}(y), & \text{if } y \in I^T, \\ post_a(y), & \text{se } y \in R_a \text{ e } y \notin \text{dom}(r_a) \end{cases}$$

Como uma rede de ocorrência é uma rede segura, todos os lugares fazem parte ou da marcação inicial ou da pós-condição de alguma transição. Sejam  $y_1$  e  $y_2$  que são mapeados para  $x$ . Existem três casos, dependendo de  $y_1$  ou  $y_2$  pertencerm ao grafo inicial ou a lado direito de regras.

**Caso 1:**  $y_1$  e  $y_2$  pertencem ao grafo inicial. Pela definição de  $\mathbb{UG}$  isto não é possível, uma vez que a marcação inicial de  $R$  é um conjunto.

**Caso 2:**  $y_1$  pertence ao grafo inicial e  $y_2$  pertence ao lado direito de alguma ação  $a$ . Isto também não é possível, pois, pela definição de rede de ocorrência, a marcação inicial não faz parte do destino de transições.

**Caso 3:**  $y_1$  e  $y_2$  pertencem ambos ao lado direito de alguma ação  $a$  ou a ações diferentes  $b$  e  $c$ . Como numa rede de ocorrência os lugares são destino de no máximo uma transição e, além disso, a multiplicidade dos lugares é 1, tem-se que  $y_1$  e  $y_2 \in S_k$ , o que implica que  $y_1$  e  $y_2 \in C^T$  e, portanto, não possuem a mesma imagem.

- 6:  $\forall (n_r, a), (n_r, b) \in N, pre_a = pre_b$  e  $r_a = r_b \Rightarrow post_a = post_b$ : segue diretamente da definição de  $\mathbb{UG}$ .

7:  $\forall(x, a) \in N$ :

7.1:  $a$  é um morfismo entre grafos duplamente tipados, pois  $a = (\emptyset, id_T)$ .

7.2:  $\forall(x, a_1), (x, a_2) \in N, \mathcal{V}_T(a_1) = \mathcal{V}_T(a_2)$ : segue diretamente da definição de  $\mathbb{UG}$ .

☺

Na proposição a seguir, será estabelecida uma relação entre grafo concorrente e elementos concorrentes, tal como no capítulo anterior, e a prova pode ser feita de maneira análoga.

PROPOSIÇÃO 15. *Sejam  $R$  uma rede  $L/T$  e  $Occ$  um objeto de  $\overline{\text{OccGGD}}$  tais que  $\mathbb{UG}(R) = Occ$ . Seja  $G = (V_G, \emptyset, \emptyset, \emptyset)$  um subgrafo do grafo-núcleo de  $Occ$ . Então,  $G$  é um grafo concorrente se, e somente se,  $Co(V_G)$ .*

#### 5.4 Transformação dos Morfismos de OccNets

A transformação dos morfismos é feita de modo análogo à definição 5.2: o morfismo entre os conjuntos de lugares será usado para encontrar os morfismo entre os grafos-núcleo, uma vez que é o conjunto de lugares que é transformado no grafo-núcleo. Este mesmo morfismo será usado para encontrar o morfismo entre os grafos-tipo.

DEFINIÇÃO 5.4. (**Transformação dos Morfismos de OccNets**) *Sejam as redes de ocorrência  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  e  $R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^\oplus, u_{R_1})$  e as gramáticas de grafos de ocorrência  $[Occ = (C^T, IN^{C \nearrow T}, N, n)]$  e  $[Occ_1 = (C_1^{T_1}, IN^{C_1 \nearrow T_1}, N_1, n_1)]$  tais que  $\mathbb{UG}(R) = [Occ]$  e  $\mathbb{UG}(R_1) = [Occ_1]$ . Das definições de  $\mathbb{UR}$  e  $\mathbb{UG}$ , tem-se que  $T = C = (S_R, \emptyset, \emptyset, \emptyset)$ ,  $T_1 = C_1 = (S_{R_1}, \emptyset, \emptyset, \emptyset)$ ,  $T_R = N \cap \{O_T\}$  e  $T_{R_1} = N_1 \cap \{O_{T_1}\}$  (veja a definição 5.3).  $[(f_T, f_N)]$  é definido como o morfismo dado por  $f_N : N \rightarrow N_1$  e  $f_T = (c, t) : C_1^{T_1} \rightarrow C^T, c = t = (c', \emptyset) : C_1 \rightarrow C$ , onde:*

- $\forall x \in T_R - \{O_T\}, f(x, r) = \begin{cases} f(x), & \text{se } f(x) \neq O_{T_1} \\ \text{undef}, & \text{caso contrário} \end{cases}$
- $\forall a \in S_{R_1}, c'(a) = \begin{cases} b, & \text{se } a \in \llbracket g(b) \rrbracket; \\ \text{undef}, & \text{caso contrário.} \end{cases}$

☺

É importante observar que, definido deste modo,  $\mathbb{UG}$  não é sobrejetor em relação aos morfismos: pode haver mais morfismos entre  $\mathbb{UG}(R_1)$  e  $\mathbb{UG}(R_2)$  do que entre  $R_1$  e  $R_2$ .

PROPOSIÇÃO 16. *Sejam  $R_1$  e  $R_2$  duas redes de ocorrência e  $g : R_1 \rightarrow R_2$  um morfismo entre  $R_1$  e  $R_2$ . Então  $[(f_T, f_N)] = \mathbb{UG}(g) : \mathbb{UG}(R_1) \rightarrow \mathbb{UG}(R_2)$  é um morfismo em  $\text{OccGGD}$ .*

*Demonstração:*

1: Pela definição 5.3,  $\forall(nr_1, a), (nr_2, b) \in N, (nr_1, a) \neq (nr_2, b) \Rightarrow nr_1 \neq nr_2$  e, portanto, a condição 1 da definição 3.32 é satisfeita. Para a condição 2 da mesma definição, tem-se o seguinte diagrama:

$$\begin{array}{ccccc}
& & c & & \\
& \swarrow & & \searrow & \\
C & \xrightarrow{c!} & \text{dom}(c) & \xrightarrow{c^*} & C_1 \\
\downarrow \text{id}_C & & \downarrow f & & \downarrow \text{id}_{C_1} \\
(1) & & (2) & & \\
T = C & \xrightarrow{t!} & \text{dom}(t) & \xrightarrow{t^*} & T_1 = C_1 \\
& \swarrow & t=c & \searrow & 
\end{array}$$

Como  $f$  também é a função identidade, (1) é pullback e (2) comuta.

Resta mostrar que  $(f_T, f_N)$  é um morfismo em DTGG, i.e., que  $I_1 \cong \mathcal{DT}_{f_T}(I)$  e que o diagrama a seguir subcomuta. O restante desta prova é análoga à prova da proposição 11.

$$\begin{array}{ccc}
N & \xrightarrow{\bar{n}} & \mathcal{DR}(C^T) \\
f_N \downarrow & \subseteq^R & \downarrow \mathcal{DR}_{f_T} \\
N_1 & \xrightarrow{\bar{n}_1} & \mathcal{DR}(C_1^{T_1})
\end{array}$$



## 6 Relações entre Gramáticas de Grafos e Redes L/T

Neste capítulo, serão estabelecidas relações entre as categorias PTNets, GGD, OccNets e OccGGD:

$$\begin{array}{ccc}
 \text{PTNets} & \xrightarrow{U_{nf}^R} & \text{OccNets} \\
 \mathbb{G} \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \mathbb{R} & & \mathbb{UG} \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \mathbb{UR} \\
 \text{GGD} & \xrightarrow{U_{nf}^G} & \text{OccGGD}
 \end{array}$$

Este diagrama não comuta. Porém, as categorias PTNets e GGD são equivalentes. Será visto que os funtores UR e UG preservam o colimite dos diagramas usados da definição da semântica e também contra exemplos para uma possível adjunção: a semântica de uma rede L/T é mais abstrata que a semântica de uma gramática de grafos.

### Estrutura do Capítulo:

- seção 1: prova de que as categorias GGD e PTNets são equivalentes;
- seção 2: prova de que os funtores UR e UG preservam colimites e contra-exemplos para uma possível adjunção. Estabelecimento de uma relação entre as semânticas de redes L/T e de gramáticas de grafos.

### 6.1 Relação Sintática

Nesta seção, será mostrado que as categorias GGD e PTNets são essencialmente a mesma [ASP 91], i.e., são equivalentes. Para isto, será mostrado que, para uma rede L/T  $R$ ,  $\mathbb{R} \circ \mathbb{G}(R) \cong R$ , usando as definições de  $\mathbb{R}$  e  $\mathbb{G}$ : será aplicada a definição de  $\mathbb{R}$  no resultado de  $\mathbb{G}(R)$  e será obtida uma rede isomórfica a  $R$  (neste caso específico, será obtida a igualdade). De modo análogo, será obtido um isomorfismo para as gramáticas de grafos discretas.

PROPOSIÇÃO 17. *As categorias PTNets e GGD são equivalentes.*

*Demonstração:*

Sejam  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$  um objeto de PTNets e  $GG = (T, I^T, N, n)$  um objeto de GGD. É necessário provar que  $\mathbb{G} \circ \mathbb{R}(GG) \cong GG$  e que  $\mathbb{R} \circ \mathbb{G}(R) \cong R$ .

1: (Redes L/T) Seja  $GG = (T, I^T, N, n)$  um objeto de GGD tal que  $\mathbb{G}(R) = GG$ .

Pela definição 5.1, a gramática  $GG$  é dada por:

1.  $T = (S_R, \emptyset, d_T, p_T)$ , onde  $p_T : \emptyset \rightarrow S_R$  e  $d_T : \emptyset \rightarrow S_R$ ;
2. Para  $\mu \in S_R^\oplus$ ,  $\mathcal{TR}(\mu)$  transforma um elemento de  $S_R^\oplus$  no grafo tipado  $G^T = (G, t, T)$ , com morfismo de tipo  $t = (t_V, t_A) : G \rightarrow T$ :
  - $G = (V, \emptyset, \emptyset, \emptyset)$ , onde  $V = \bigcup \{ \{a\} \times \{1, \dots, \mu(a)\} \mid a \in S_R \}$ ;
  - $t_A : \emptyset \rightarrow \emptyset$ ;

- $t_V : V \rightarrow S_R$  é a função dada por:  $\forall (a, n) \in V, t_V(a, n) = a$ ;
- 3.  $I^T = \mathcal{TR}(u_R)$ ;
- 4.  $N = T_R - \{O_T\}$ ;
- 5. a função  $n : N \rightarrow \text{Rules}(T)$  é definida por  $n(x) = r : L^T \rightarrow R^T = (g, t)$ , onde:
  - $t = id_T : T \rightarrow T$ ;
  - $g = (g_V : V_L \rightarrow V_R, g_A : A_L \rightarrow A_R) : L \rightarrow R = (\emptyset, \emptyset)$
  - $L^T = \mathcal{TR}(p_R(x))$ ;
  - $R^T = \mathcal{TR}(d_R(x))$ .

Será mostrado que as redes  $R$  e  $\mathbb{R}(GG) = R_1 = (p_{R_1}, d_{R_1} : (T_{R_1}, O_{T_1}) \rightarrow S_{R_1}^{\oplus}, u_{R_1})$  são iguais, i.e., possuem o mesmo conjunto de lugares, o mesmo conjunto (apontado) de transições, que  $p_R = p_{R_1}$ ,  $d_R = d_{R_1}$  e que possuem a mesma marcação inicial. Pela definição 4.2, a rede  $R_1$  é dada por:

1.  $S_{R_1} = S_R$ , o conjunto de vértices do grafo-tipo  $T = (S_R, \emptyset, \emptyset, \emptyset)$ ;
2.  $(T_{R_1}, O_{T_1}) = N \cup \{O_{T_1}\} = (T_R, O_T) - \{O_T\} \cup \{O_{T_1}\}$ , onde  $O_T \notin N$  é o elemento apontado de  $T_R$ . Tomando  $O_{T_1} = O_T$ , uma vez que são valores quaisquer, tem-se que  $(T_{R_1}, O_{T_1}) = (T_R, O_T)$ .
3.  $\forall x \in N, n(x) = r^T : L^T \rightarrow R^T$ , sejam os morfismos de tipo  $t^L : L \rightarrow T$  e  $t^R : R \rightarrow T$ .  $\forall a \in V_T$  ( $Cd(A)$  = cardinalidade do conjunto  $A$ ),

$$\bullet p_{R_1}(x)(a) = \begin{cases} Cd(\{y \in V_{LT} | t^L(y) = a\}) = \\ = Cd(\{(a, 1), \dots, (a, p_R(x)(a))\}) = \\ = p_R(x)(a), \text{ se } a \in \text{img}(t^L) \\ 0 = p_R(x)(a), \text{ caso contrário} \end{cases}$$

$$\bullet d_{R_1}(x)(a) = \begin{cases} Cd(\{y \in V_{RT} | t^R(y) = a\}) = \\ = Cd(\{(a, 1), \dots, (a, d_R(x)(a))\}) = \\ = d_R(x)(a), \text{ se } a \in \text{img}(t^R) \\ 0 = d_R(x)(a), \text{ caso contrário} \end{cases}$$

$$4. p_{R_1}(O_T) = d_{R_1}(O_T) = O_{S_R};$$

$$5. \forall a \in V_T = S_R, u_{R_1}(a) = \begin{cases} Cd(\{y \in V_I | t^I(y) = a\}) = \\ = Cd(\{(a, 1), \dots, (a, u_R(a))\}) = \\ = u_R(a), \text{ se } a \in \text{img}(t^I) \\ 0 = u_R(a), \text{ caso contrário} \end{cases}$$

2: (Gramáticas de Grafos) Seja  $R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^{\oplus}, u_R)$  um objeto de PTNets tal que  $\mathbb{R}(GG) = R$ . De acordo com a definição 4.2, a rede  $R$  é dada por:

1.  $S_R = V_T$ , o conjunto de vértices do grafo-tipo  $T = (V_T, \emptyset, \emptyset, \emptyset)$ ;
2.  $(T_R, O_T) = N \cup \{O_T\}$ , onde  $O_T \notin N$  é o elemento apontado de  $T_R$ ;
3.  $\forall x \in N, n(x) = r^T : L^T \rightarrow R^T$ , sejam os morfismos de tipo  $t^L : L \rightarrow T$  e  $t^R : R \rightarrow T$ . Os morfismos  $p_R$  e  $d_R$  são definidos da seguinte maneira:  $\forall a \in V_T$  ( $Cd(A)$  = cardinalidade do conjunto  $A$ ),

$$\bullet p_R(x)(a) = \begin{cases} Cd(\{y \in V_{LT} | t^L(y) = a\}), \text{ se } a \in \text{img}(t^L) \\ 0, \text{ caso contrário} \end{cases}$$

$$\bullet d_R(x)(a) = \begin{cases} Cd(\{y \in V_{RT} | t^R(y) = a\}), \text{ se } a \in \text{img}(t^R) \\ 0, \text{ caso contrário} \end{cases}$$

$$4. p_R(O_T) = d_R(O_T) = O_{S_R};$$

$$5. \forall a \in V_T, u_R(a) = \begin{cases} Cd(\{y \in V_I | t^I(y) = a\}), \text{ se } a \in \text{img}(t^I) \\ 0, \text{ caso contrário} \end{cases}$$

Será definido um isomorfismo  $i : \mathbb{G}(R) \rightarrow GG$ . De acordo com a definição 5.1, a gramática  $\mathbb{G}(R) = GG_1 = (T_1, I_1^{T_1}, N_1, n_1)$  é dada por:

1.  $T_1 = (S_R, \emptyset, d_T, p_T)$ , onde  $p_T : \emptyset \rightarrow S_R$  e  $d_T : \emptyset \rightarrow S_R$ . Como  $S_R = V_T$ , tem-se que  $T_1 = T$ , i.e.,  $GG$  e  $GG_1$  possuem o mesmo grafo-tipo;
2. Para  $\mu \in S_R^\oplus$ ,  $\mathcal{TR}(\mu)$  transforma um elemento de  $S_R^\oplus$  no grafo tipado  $G^T = (G, t, T)$ , com morfismo de tipo  $t = (t_V, t_A) : G \rightarrow T$ :
  - $G = (V, \emptyset, \emptyset, \emptyset)$ , onde  $V = \bigcup \{\{a\} \times \{1, \dots, \mu(a)\} \mid a \in S_R\}$ ;
  - $t_A : \emptyset \rightarrow \emptyset$ ;
  - $t_V : V \rightarrow S_R$  é a função dada por:  $\forall (a, n) \in V, t_V(a, n) = a$ ;
3.  $I_1^{T_1} = \mathcal{TR}(u_R)$ ;
4.  $N_1 = (T_R, O_T) - \{O_T\} = N \cup \{O_T\} = N$ . Portanto,  $GG$  e  $GG_1$  possuem o mesmo conjunto de nomes de regras;
5. a função  $n_1 : N_1 \rightarrow \text{Rules}(T)$  é definida por  $n_1(x) = r : L^T \rightarrow R^T = (g, t)$ , onde:
  - $t = id_T : T \rightarrow T$ ;
  - $g = (g_V : V_L \rightarrow V_R, g_A : A_L \rightarrow A_R) : L \rightarrow R = (\emptyset, \emptyset)$ ;
  - $L^T = \mathcal{TR}(p_R(x))$ ;
  - $R^T = \mathcal{TR}(d_R(x))$ .

Note que  $\forall a \in S_R, u_R(a) = Cd(\{y \in V_I \mid t^I(y) = a\})$ , para  $a \in \text{img}(t^I)$ . Seja  $m_a$  um isomorfismo entre os conjuntos  $A = \{y \in V_I \mid t^I(y) = a\}$  e  $B = \{(a, 1), \dots, (a, u_R(a))\}$  (note que  $A$  e  $B$  possuem a mesma cardinalidade). Os elementos de  $A$  são vértices do grafo  $I$  e os elementos de  $B$  são vértices do grafo  $I_1$  e todos possuem tipo  $a$ . O isomorfismo  $m : I \rightarrow I_1$  é definido por  $m = \bigcup_{a \in S_R} m_a$ . Portanto, o diagrama abaixo comuta.

$$\begin{array}{ccc} I & \xrightarrow{m} & I_1 \\ \downarrow t^I & & \downarrow t^{I_1} \\ T & \xrightarrow{id_T} & T \end{array}$$

Os isomorfismos entre as regras são definidos de maneira análoga, o que induz um isomorfismo  $i_N : N \rightarrow N_1$ . O isomorfismo  $i$  é então dado por  $i = (id_T, i_N)$  e satisfaz as condições da definição 3.13:

1. **Subcomutatividade:**  $\overline{n_2} \circ f_N \subseteq^R \mathcal{R}_{f_T} \circ \overline{n_1}$ . De fato, as regras são isomórficas:  $n_2 \circ f_N \cong \mathcal{R}_{f_T} \circ n_1$ .

$$\begin{array}{ccc} N & \xrightarrow{\overline{n}} & \mathcal{R}(T) \\ id_N \downarrow & \subseteq^R & \downarrow \mathcal{R}_{id_T} \\ N_1 & \xrightarrow{\overline{n_1}} & \mathcal{R}(T_1) \end{array}$$

2.  $I_1^{T_1} \cong \mathcal{T}_{id_T}(I^T)$

3: de modo análogo, usando as definições de  $\mathbb{R}$  e  $\mathbb{G}$ , pode-se mostrar que para todo morfismo  $f$  de PTNets ( $g$  de GGD),  $\mathbb{R} \circ \mathbb{G}(f) \cong f(\mathbb{G} \circ \mathbb{R}(g) \cong g)$ . ☺

## 6.2 Relação Semântica

Os funtores  $\mathbb{UR}$  e  $\mathbb{UG}$  não constituem adjunção, uma vez que os conjuntos  $A = \overline{\text{OccGGD}}[\mathbb{UG}(R), GG]$  (para uma categoria qualquer  $C$ ,  $C[a, b]$  denota a classe dos

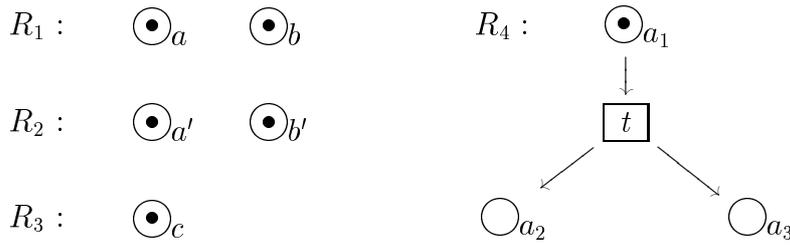
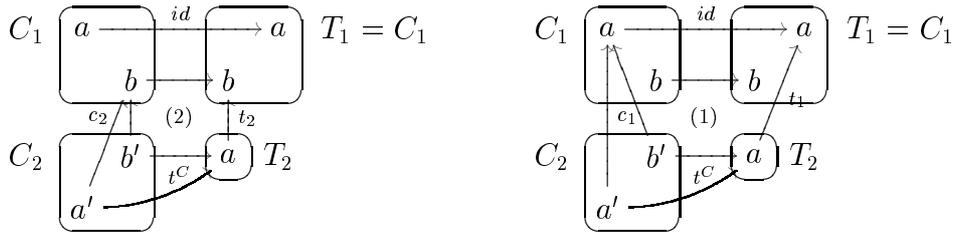


FIGURA 6.1- Exemplos de Redes de Ocorrência.

morfismos  $f : a \rightarrow b$ ) e  $B = \text{OccNets}[R, \mathbb{UR}(GG)]$  não são isomórficos. No exemplo a seguir, são dadas uma rede  $R$  e uma gramática  $GG$  de modo que os conjuntos  $A$  e  $B$  tenham cardinalidades diferentes.

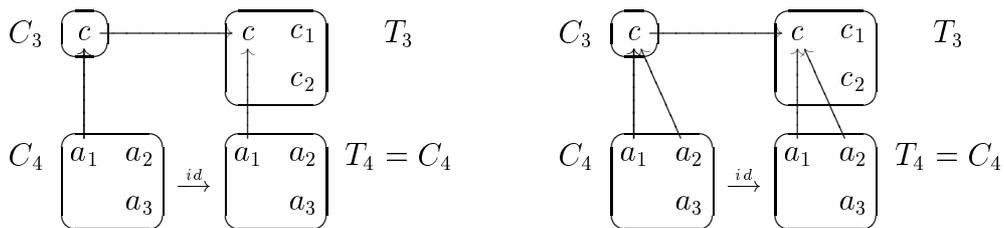
Exemplo 6.1: Nas figuras 6.1 e 6.2, as gramáticas  $GG_1, GG_2$  e  $GG_3$  possuem conjunto de regras vazio,  $\mathbb{UR}([GG_2]) = R_2$  e  $\mathbb{UG}(R_1) = [GG_1]$ . É fácil observar que existem dois isomorfismos  $f_1, f_2 : R_1 \rightarrow R_2$ . Porém, não existe nenhum morfismo  $f' : [GG_1] \rightarrow [GG_2]$  entre as gramáticas  $[GG_1] = \mathbb{UG}(R_1)$  e  $[GG_2]$ , uma vez que os únicos morfismos não vazios entre  $C_2^{T_2}$  e  $C_1^{T_1}$  são:



Neste caso,  $C_2 = \text{dom}(c_1) = \text{dom}(c_2)$  e os diagramas (1) e (2) não são *pullbacks*. Portanto, estes morfismos não induzem nenhum morfismo entre as gramáticas de grafos de ocorrência  $GG_1$  e  $GG_2$  (veja o item (2) da definição 3.32).

De modo análogo, os conjuntos de morfismos  $A = \overline{\text{OccGGD}}[GG, \mathbb{UG}(R)]$  e  $B = \text{OccNets}[\mathbb{UR}(GG), R]$  não são isomórficos. Como contra-exemplo, também são dadas uma rede  $R$  e uma gramática  $GG$  de modo que os conjuntos  $A$  e  $B$  tenham cardinalidades diferentes.

Exemplo 6.2: Observe também nas figuras 6.1 e 6.2 que  $\mathbb{UR}([GG_3]) = R_3$  e que  $\mathbb{UG}(R_4) = [GG_4]$ . Veja que existe apenas um morfismo (não vazio)  $f : R_3 \rightarrow R_4$  entre as redes  $R_3$  e  $R_4$  e que existe mais de um morfismo  $f' : [GG_3] \rightarrow [GG_4]$ . Por exemplo, os morfismos  $g_1, g_2 : [GG_3] \rightarrow [GG_4]$  induzidos pelos dois morfismos abaixo (note que a condição (2) da definição 3.32 é satisfeita).



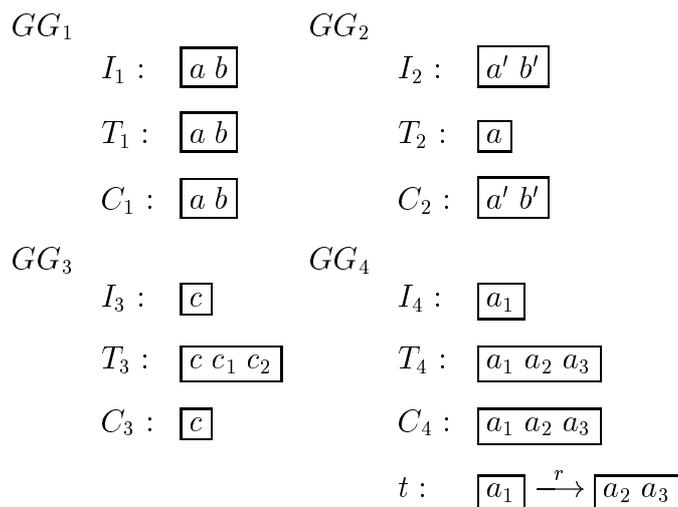


FIGURA 6.2- Gramáticas de Grafos de Ocorrência Discretas.

Note que, se estes funtores constituíssem uma adjunção onde a unidade (ou co-unidade) fosse um isomorfismo, o diagrama abaixo comutaria a menos de isomorfismo.

$$\begin{array}{ccc}
 \text{PTNets} & \xrightarrow{\text{Unf}^R} & \text{OccNets} \\
 \text{G} \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \mathbb{R} & & \text{UG} \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \mathbb{UR} \\
 \text{GGD} & \xrightarrow{\text{Unf}^G} & \overline{\text{OccGGD}}
 \end{array}$$

Estes funtores não formam adjunção porque, devido ao fato de as gramáticas de grafos de ocorrência possuírem dois (grafos) tipos, elas são menos abstratas do que as redes de ocorrência e daí não é sempre possível obter um isomorfismo na relação semântica. Esta situação pode ser contornada de duas maneiras:

1. definir uma nova categoria semântica UNets (para as redes L/T) onde os objetos sejam redes de ocorrência que sejam *unfoldings* (finitos e infinitos) das redes L/T. Como pode ser notado na definição dos *unfoldings*, os lugares da rede L/T original são “guardados” no conjunto de lugares dos *unfoldings* (veja a definição 2.17). Daí é possível retirar destes objetos-*unfoldings* informações para definir o grafo-tipo, obtendo assim comutatividade para o diagrama abaixo.

$$\begin{array}{ccccc}
 \text{PTNets} & \longrightarrow & \text{OccNets} & \longrightarrow & \text{UNets} \\
 \updownarrow & & & & \updownarrow \\
 \text{GGD} & \longrightarrow & & \longrightarrow & \overline{\text{OccGGD}}
 \end{array}$$

2. definir uma nova categoria OccGGA para as gramáticas de grafos de ocorrência, onde seus objetos possuem como tipo apenas um grafo (sendo, portanto, objetos de GG). Esta categoria se relaciona com OccGG através de um funtor que “esquece” um dos tipos, deixando o grafo-núcleo. Assim, é obtida comutatividade para o diagrama abaixo.



Como  $\mathbb{U}G$  é um funtor,  $(\mathbb{U}G(Unf_R(R)), \{\mathbb{U}G(f_0), \mathbb{U}G(f_1), \dots, \mathbb{U}G(f_k), \dots\})$  é um co-cone deste mesmo diagrama. Seja  $(Occ, \{g_0, g_1, \dots, g_k, \dots\})$  o colimite de  $D$ . Então, o morfismo  $f : Occ \rightarrow \mathbb{U}G(Unf_R(R))$  existe e é o único que faz o diagrama abaixo comutar:

$$\begin{array}{ccccccc}
 \mathbb{U}G(\mathcal{U}_R^0(R)) & \xrightarrow{\mathbb{U}G(in_0)} & \mathbb{U}G(\mathcal{U}_R^1(R)) & \xrightarrow{\mathbb{U}G(in_1)} & \dots & \longrightarrow & \mathbb{U}G(\mathcal{U}_R^k(R)) & \longrightarrow & \dots \\
 & \searrow & \downarrow & \searrow & & & \downarrow & & \\
 & & \mathbb{U}G(Unf_R(R)) & \xleftarrow{f} & & & Occ & & 
 \end{array}$$

Como os morfismos  $in_k$  são inclusões, pela definição de  $\mathbb{U}G$ ,  $\mathbb{U}G(in_k)$  também são inclusões. Portanto, os morfismos  $g_k$  do colimite de  $D$  também são inclusões (veja [RIB 96b]). Como cada morfismo  $\mathbb{U}G(f_k)$  é uma inclusão, uma vez que, pela definição de  $Occ$  em [RIB 96b], cada elemento de  $Occ$  pertence ao contradomínio de algum morfismo  $g_k$ : isto implica que  $f$  é injetor. O morfismo  $f$  é também sobrejetor, pois suponha que  $\exists X \in \mathbb{U}G(Unf_R(R)) | X \notin \text{img}(f)$ . Isto implica que  $X$  não pertence à imagem de nenhum morfismo  $\mathbb{U}G(f_k)$ . Pela definição de  $\mathbb{U}G$ ,  $\exists X' \in Unf_R(R) | X' \notin \text{img}(f_k)$ . Mas isto é uma contradição, uma vez que, pela definição de colimite em [MES 94b], todo elemento (transição ou lugar) de  $Unf_R(R)$  é imagem de algum  $f_k$ . Portanto,  $f$  é um isomorfismo.

□

Com um raciocínio análogo, pode-se provar que

**PROPOSIÇÃO 19.** *O funtor  $\mathbb{U}R$  preserva o colimite dos diagramas usados na definição da semântica de gramáticas de grafos discretas.*

### Relacionamento entre *unfoldings* Finitos:

Para a proposição a seguir, será usada indução para mostrar que a rede  $\mathbb{U}R(\mathcal{U}_G^k(GG))$  é isomórfica à rede  $\mathcal{U}_R^k(\mathbb{R}(GG))$ .

**PROPOSIÇÃO 20.** *Sejam  $GG$  uma gramática de grafos em GGD,  $GG^k = \mathcal{U}_G^k(GG)$  o seu  $k$ -ésimo *unfolding* e  $R$  e  $R^k$  duas redes  $L/T$  tais que  $R = \mathbb{R}(GG)$  e  $R^k = \mathbb{U}R(GG^k)$ . Há um isomorfismo  $i_k = (f_k, g_k) : R^k \rightarrow \mathcal{U}_R^k(R)$  entre a rede de ocorrência  $R^k$  e o *unfolding* finito da rede  $R$ ,  $\mathcal{U}_R^k(R)$ .*

*Demonstração:*

- 1: Caso base:  $k = 0$  (trivial)
- 2: Hipótese da indução: existe um isomorfismo  $i_k = (f_k, g_k) : R^k \rightarrow \mathcal{U}_R^k(R)$ .
- 3: Passo da indução: é preciso provar que, para cada regra em  $ApplRules^{k+1}$ , sua transição equivalente em  $R^{k+1}$  tem uma equivalente em  $\mathcal{U}^{k+1}(R)$ , e vice-versa, para cada transição em  $T_{k+1}^U - T_k^U$ , existe uma regra equivalente em  $ApplRules^{k+1}$  (se  $ApplRules^{k+1} = \emptyset$ ,  $i_{k+1} = (f_k, g_k)$ ).

$$\begin{array}{ccccc}
GG & \xrightarrow{\mathbb{R}} & R & & \\
\downarrow u_G^k & & \downarrow u_R^k & & \\
GG^k & \xrightarrow{\mathbb{U}R} & R^k & \xrightarrow{i_k} & \mathcal{U}_R^k(R) \\
\downarrow in_k^G & & \downarrow in_k & & \downarrow in_k^R \\
GG^{k+1} & \xrightarrow{\mathbb{U}R} & R^{k+1} & \xrightarrow{i_{k+1}} & \mathcal{U}_R^{k+1}(R)
\end{array}$$

Sejam:

- $GG = (T, I^T, N, n)$ ,
- $\mathbb{R}(GG) = R = (p_R, d_R : (T_R, O_T) \rightarrow S_R^\oplus, u_R)$ ,
- $GG^k = (C_k^T, I_k, N_k, n_k)$ ,
- $R^k = \mathbb{U}R(GG^k) = (p_k, d_k : (T_k, O_T) \rightarrow S_k^\oplus, u_k)$ ,
- $\mathcal{U}_R^k(R) = (p'_k, d'_k : (T'_k, O_T) \rightarrow S'_k^\oplus, u'_k)$ .

Pela definição de  $\mathbb{U}R$ , o morfismo  $in_k$ , tal como  $in_k^G$ , também é uma inclusão. Portanto, é suficiente definir os valores de  $i_{k+1}$  apenas para os elementos que não pertencem à imagem de  $in_k$ , i.e., os elementos do conjunto de lugares (transições) de  $R^{k+1}$  que pertencem a  $S_{k+1} - S_k$  ( $T_{k+1} - T_k$ ).

Seja  $y = (n_r, r^T, m^T) \in \text{App}Rules^{k+1}$  tal que  $(n_r, a) \in N_{k+1}$ . Por definição,  $G = (V_G, \emptyset, \emptyset, \emptyset) = \text{img}(m^T)$  é um subgrafo concorrente de  $C_k^T$ . Portanto,  $Co(V_G)$  é verdadeiro em  $R^k$  (proposição 9) e, por conseguinte,  $Co(B)$  é verdadeiro em  $\mathcal{U}_R^k(R)$ , onde  $B = \{b \in S'_k | g_k(a) = b \text{ e } a \in V_G\}$  (a imagem de  $V_G$  através de  $i_k$ ).

**Mapeamento das transições:** Os elementos de  $B$  são da forma  $((A, b), n)$ . Seja  $\mu = \bigoplus b | ((A, b), n) \in B, b \in S_R \text{ e } n \in \mathbb{N}$ . Suponha que  $\nexists t' \in T_R | p_R(t') = \mu$ , ou seja,  $\exists b \in S_R | p_R(t')(b) \neq \mu(b)$ . Sejam  $d = p_R(t')(b)$  e  $c = \mu(b)$ . Pela definição de  $\mathbb{R}$ ,  $t'$  é uma regra de  $T_R$  e  $(b, 1), \dots, (b, d)$  são vértices do lado esquerdo de  $t'$ . Como há apenas  $c$  elementos do tipo  $b$  em  $V_G$ , e  $d \neq c$ , não há uma função bijetora do lado direito de  $t'$  para  $G$ , o que é uma contradição, uma vez que  $m^T$  é uma *match* para a regra  $r^T$ . Portanto,  $\exists t' \in T_R | p_R(t') = \mu$ , i.e.,  $(B, t') \in T_{k+1}$ . Conclui-se então que  $t' = n_r$ . A função  $f_{k+1}$  é então dada por:

$$\forall (n_r, a) \in T_{k+1} - T_k, f^{k+1}(n_r, a) = (B', n_r)$$

onde  $B'$  é a imagem de  $B$  através de  $i_k$  e  $\text{img}(pre_a) = (B, \emptyset, \emptyset, \emptyset)$ .

**Mapeamento dos lugares:** Como as regras não preservam itens, serão definidos os valores de  $i_{k+1}$  para os itens (vértices) que pertencem às pós-condições das regras (os itens das pré-condições já estão mapeados por  $i_k$ ).

Sejam uma ação  $(n_r, a) \in N_{k+1} - N_k$ , os grafos  $\text{img}(post_a) = (V, \emptyset, \emptyset, \emptyset)$  e  $\text{img}(pre_a) = (B, \emptyset, \emptyset, \emptyset)$  e o multiconjunto  $\mu = d_R(B', n_r)$ , onde  $B'$  é a imagem de  $B$  através de  $i_k$ . Para cada  $x \in T$ , seja  $i_x$  um isomorfismo entre os conjuntos  $A = \{y \in V | t(y) = x\}$  e  $A' = \{((D, x), n) | ((D, x), n) \in B\}$ , onde  $t$  é o morfismo de tipo do lado esquerdo da ação  $(n_r, a)$ . O morfismo  $i_{k+1}$  para o conjunto  $V$  é dado por:

$$\forall y \in V_G, i_{k+1}(y) = ((D, x), n) \Leftrightarrow t(y) = x \text{ e } i_x(y) = ((D, x), n)$$

Por esta construção, pode-se concluir diretamente que o morfismo  $i_{k+1}$  é total e injetor.

4: O morfismo  $i_{k+1}$  é sobrejetor:

**Transições:** Seja  $(B, t) \in T_{k+1}^U - T_k^U$  e  $\mu = p_{k+1}(B, t)$ . Por definição,  $Co(\llbracket \mu \rrbracket)$  é verdadeiro. Seja  $V$  a imagem de  $\llbracket \mu \rrbracket$  através de  $i_k^{-1}$ .  $Co(V)$  também é verdadeiro em  $R^k$  e, pela proposição 9,  $G = (V, \emptyset, \emptyset, \emptyset)$  é um subgrafo con-corrente de  $C_k^T$  (lembre-se que numa rede de ocorrência, a multiplicidade dos lugares é unitária). Será provado que  $G$  é imagem de algum *match* de alguma regra de  $GG$ .

Seja  $a \in \llbracket p_R(t) \rrbracket$  e  $l = p_R(t)(a)$ . Pela definição de  $\mathcal{U}$ ,  $Cd(A) = l$ , onde  $A = \{((A, x), m) \mid ((A, x), m) \in \llbracket \mu \rrbracket, x = a \text{ e } m \in \mathbb{N}\}$ . Pela definição de  $\mathbb{R}$ ,  $t \in N$  e  $n(t) = r : L^T \rightarrow R^T$  e  $(a, 1), \dots, (a, l)$  são vértices de  $L^T$  e possuem o tipo  $a$ . Como os elementos de  $A$  pertencem a  $S_k^l$ , eles são mapeados por  $i_k^{-1}$  e, portanto, sua imagem em  $S_k$  tem tipo  $a$  (lembre-se que  $S_k$  é o conjunto de vértices do grafo-núcleo  $C_k$ ). Isto implica que cada vértice do grafo  $L^T$  possui uma (única) imagem em  $C_k$  e, portanto, a regra  $r$  pode ser aplicada no *match* formado por estas imagens.

**Lugares:** Com um raciocínio análogo ao mapeamento dos lugares, pode-se mostrar que  $g_{k+1}$  é sobrejetor.

⊙

De modo análogo, por indução, pode-se mostrar que há um morfismo  $h_k$  entre a gramática de grafos  $\mathbb{UG}(\mathcal{U}_R^k(R))$  e o *unfolding* finito  $\mathcal{U}_G^k(\mathbb{G}(R))$ .

**PROPOSIÇÃO 21.** *Sejam  $R$  uma rede  $L/T$  e  $R^k = \mathcal{U}_R^k(R)$  o seu  $k$ -ésimo *unfolding* e  $GG$  e  $GG^k$  duas gramáticas de grafos tais que  $GG = \mathbb{G}(R)$  e  $GG^k = \mathbb{UG}(R^k)$ . Há um morfismo  $h_k = (f_T^k, f_N^k) : \mathcal{U}_G^k(GG) \rightarrow GG^k$  entre a gramática de grafos de ocorrência  $GG^k$  e  $\mathcal{U}_G^k(GG)$ , o *unfolding* de profundidade  $k$  da gramática  $GG$ , onde o morfismo  $f_N^k$  é total e sobrejetor e  $f_N^k$  é a identidade  $id_{N_k}$ .*

1: Caso base:  $k = 0$  (trivial)

2: Hipótese da indução: existe um morfismo total e sobrejetor  $h_k = (f_T^k, f_N^k) : \mathcal{U}_G^k(GG) \rightarrow GG^k$ , onde o morfismo  $f_N^k$  é total e sobrejetor e  $f_N^k$  é a identidade  $id_{N_k}$ .

3: Passo da indução: seja o diagrama abaixo, onde os morfismos  $i_R$  e  $i_G$  são inclusões. Será então definido o morfismo  $h_{k+1} = (f_T^{k+1}, f_N^{k+1}) : \mathcal{U}_G^{k+1}(GG) \rightarrow GG^{k+1}$  e mostrar que onde o morfismo  $f_N^{k+1}$  é total e sobrejetor e  $f_N^{k+1}$  é a identidade entre os conjuntos de nomes de ações das duas gramáticas.

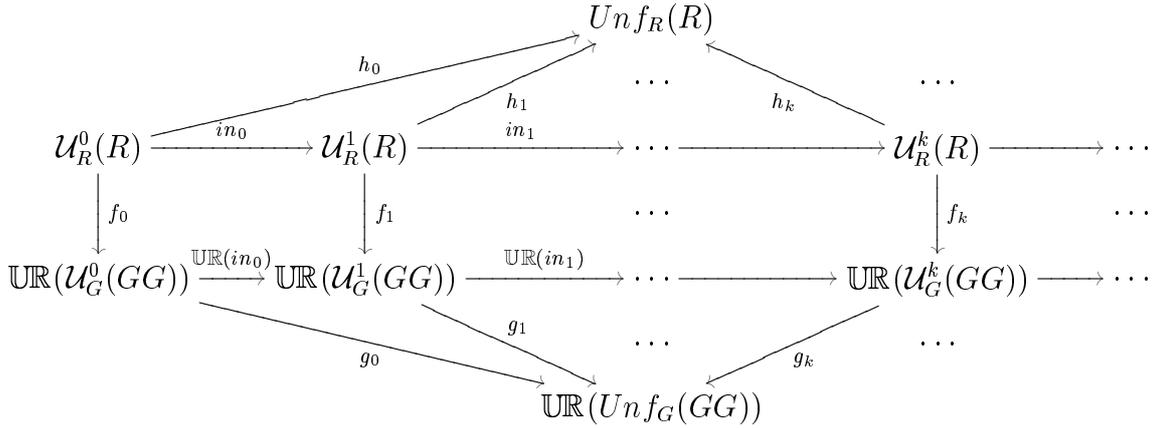
$$\begin{array}{ccccc}
 R & \xrightarrow{\mathbb{G}} & GG & & \\
 \downarrow u_R^k & & \downarrow u_G^k & & \\
 R^k & \xrightarrow{\mathbb{UG}} & GG^k & \xleftarrow{h_k} & \mathcal{U}_G^k(GG) \\
 \downarrow in_k^R & & \downarrow in_k & & \downarrow in_k^G \\
 R^{k+1} & \xrightarrow{\mathbb{UG}} & GG^{k+1} & \xleftarrow{h_{k+1}} & \mathcal{U}_G^{k+1}(GG)
 \end{array}$$

Pela definição de  $\mathbb{UG}$ , o morfismo  $in_k$ , tal como  $in_k^G$ , também é uma inclusão. Portanto, é suficiente definir os valores de  $i_{k+1}$  apenas para os elementos que não pertencem à imagem de  $in_k$ . Como se pode notar, a prova desta

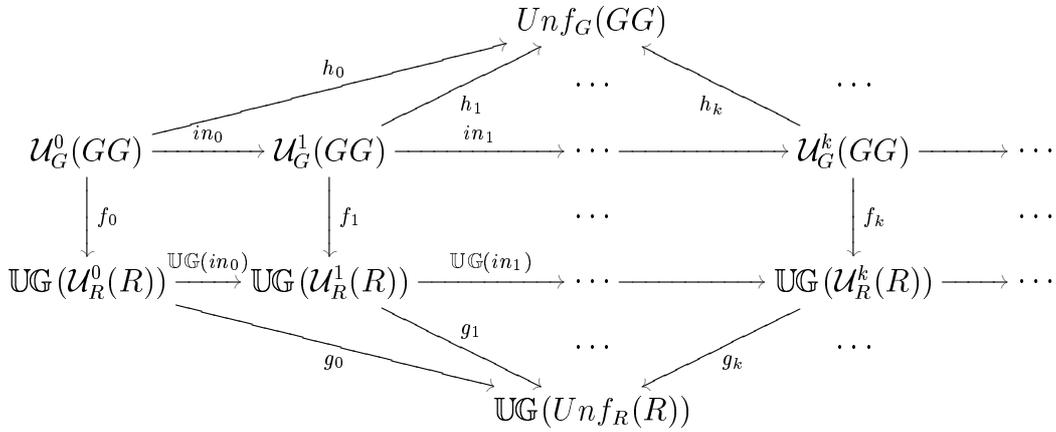
proposição segue um raciocínio análogo à prova da proposição anterior e será omitida.  $\odot$

### Relacionamento entre Colimites Especiais:

Seja  $R$  uma rede L/T tal que  $GG = \mathbb{G}(R)$ . Qual o relacionamento entre as semânticas de  $R$  e de  $GG$ ? Foi mostrado que cada  $f_k$  do diagrama a seguir é um isomorfismo. Como  $(Unf_R(R), \{h_0, h_1, \dots, h_k, \dots\})$  e  $(\mathbb{U}R(Unf_R(R)), \{g_0, g_1, \dots, g_k, \dots\})$  são colimites, pode-se concluir que  $Unf_R(R)$  e  $\mathbb{U}R(Unf_R(R))$  são também isomórficos.

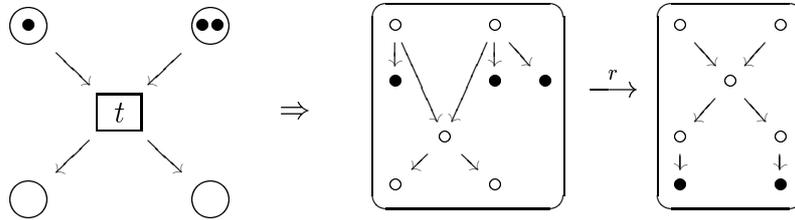


Sejam  $GG$  uma gramática de grafos discreta e  $R = \mathbb{U}R(GG)$ . Seguindo um raciocínio análogo, pode-se concluir que existe um morfismo  $f : Unf_G(GG) \rightarrow \mathbb{U}G(Unf_R(R))$  com as mesmas propriedades de  $f_k$ :



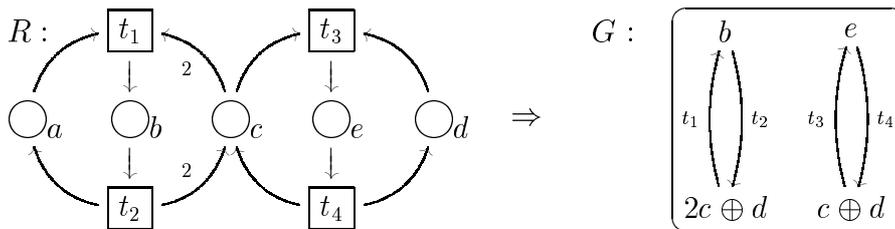
## 7 Trabalhos Relacionados

O interrelacionamento entre redes de Petri e gramáticas de grafos já vem sendo pesquisado há algum tempo. Em [KRE 80] é descrita uma maneira de simular uma rede de Petri por uma gramática de grafos, onde cada transição é transformada numa regra: um lugar com  $n$  tokens é representado por um vértice com  $n$  arcos, cada arco com um único vértice de destino. Por exemplo, a transição  $t$  abaixo é representada pela regra  $r$ :



Uma desvantagem deste método é que as marcações são representadas duas vezes: para  $n$  marcas, há  $n$  vértices e  $n$  arcos. Note que, neste caso, os arcos são necessários, pois caso contrário, não haveria como identificar em qual lugar estão as marcas. Veja que estes grafos são mais complexos do que os grafos discretos, usados neste trabalho. Este método é usado também em [KRE 86], onde é estendido para redes com lugares de capacidade finita. É estabelecido também uma equivalência entre o comportamento seqüencial de uma rede e o comportamento seqüencial da gramática de grafos simuladora da rede. Sistemas condição/evento (redes onde os lugares possuem capacidade unitária e arcos valorados unitários, i.e., redes seguras) são usados para estabelecer uma equivalência entre comportamento não-seqüencial. Para uma comparação desta semântica com outros modelos, veja [RIB 96b] e [MES 94b].

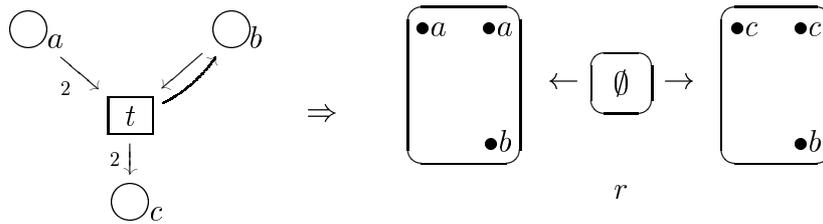
Em [MES 90], uma rede de Petri é vista como um grafo dirigido (e não como uma gramática de grafos) equipado com duas operações algébricas (composição paralela e seqüencial de transições): uma rede  $R$  é considerada como um grafo onde o conjunto de vértices é um subconjunto de  $S^\oplus$  ( $S$  é o conjunto de lugares de  $R$ ) e os arcos do grafo são, portanto, rotulados com as transições de  $R$ . A rede  $R$  abaixo é representada pelo grafo  $G$ :



Veja que, neste método, os elementos dos conjuntos de vértices dos grafos resultantes possuem uma estrutura de monóide. Obviamente, este método não de

adequada a este trabalho, uma vez que o modelo semântico utilizado não leva em consideração a estrutura dos vértices.

Em [COR 95], gramáticas de grafos são consideradas como uma generalização de redes de Petri, onde o estado de um sistema é descrito por um grafo, no lugar de uma coleção de *tokens*. Uma rede de Petri é considerada como uma gramática de grafos onde os grafos são discretos. As transições são transformadas em regras utilizando a abordagem algébrica DPO (nesta abordagem, uma derivação direta é definida por meio de dois *pushouts*, ao invés de um único, como em SPO), onde o grafo-interface é vazio. Por exemplo, a transição  $t$  abaixo é representada pela regra  $r$ :



O uso de grafos discretos na representação de redes traz uma grande simplificação. Este método é usado também em [KOR 96] porém, usando a abordagem algébrica SPO. Como foi visto no capítulo 4, este é o método de transformação seguido neste trabalho. Nesta referência é também definida uma relação semântica entre redes e gramáticas de grafos usando árvores como domínio semântico.

## 8 Conclusão

Neste trabalho foi estabelecido um relacionamento entre os formalismos de redes (de Petri) lugar/transição e de gramáticas de grafos. Foi visto que as gramáticas de grafos são mais expressivas do que estas redes, o que resultou na necessidade de encontrar uma classe de gramáticas de grafos que fossem equivalentes às redes. Tecnicamente, foi mostrado que a categoria **PTNets** das redes lugar/transição é equivalente a uma subcategoria completa (**GGD**) das gramáticas de grafos onde os grafos são grafos discretos (possuem apenas vértices) e as regras não preservam itens (i.e., as regras apenas eliminam e/ou criam vértices). Para isto, foram definidos dois funtores  $\mathbb{G}$  e  $\mathbb{R}$ .

Para a semântica, foi visto que no modelo utilizado a semântica de uma rede é menos informativa do que a semântica de uma gramática de grafos. Porém, isto não foi empecilho para estabelecer uma equivalência semântica. Tecnicamente, foram definidos os funtores  $\mathbb{UG}$  e  $\mathbb{UR}$  entre as categorias semânticas de redes L/T (**OccNets**) e de gramáticas de grafos discretas (**OccGGD**). Foi mostrado que estes funtores preservam a semântica e que o diagrama (1) abaixo comuta a menos de isomorfismo, havendo, portanto, uma equivalência entre as semânticas de uma rede L/T e de sua gramática equivalente através de  $\mathbb{UR}$ .

$$\begin{array}{ccc}
 \text{PTNets} & \xrightarrow{Unf^R} & \text{OccNets} \\
 \mathbb{G} \downarrow & (1) & \uparrow \mathbb{UR} \\
 \text{GGD} & \xrightarrow{Unf^G} & \overline{\text{OccGGD}}
 \end{array}
 \qquad
 \begin{array}{ccc}
 \text{PTNets} & \xrightarrow{Unf^R} & \text{OccNets} \\
 \uparrow \mathbb{R} & (2) & \mathbb{UG} \downarrow \\
 \text{GGD} & \xrightarrow{Unf^G} & \overline{\text{OccGGD}}
 \end{array}$$

Foi visto que ocorre perda de informações quando da aplicação do funtor  $\mathbb{UR}$  a uma gramática de ocorrência *Occ*. Porém, esta informação que é perdida não é relevante para a descrição semântica: foi mostrado que a semântica de uma gramática de grafos discreta possui uma estreita relação com a semântica de sua rede L/T equivalente através do funtor  $\mathbb{UG}$ , havendo, neste caso, uma equivalência em termos de comportamento (para cada possível derivação direta de  $GG$ , existe uma transição que pode ser disparada em  $\mathbb{R}(GG)$ , e vice-versa). Isto ocorre devido ao fato de que o funtor  $\mathbb{UR}$  “esquece” informações dos objetos de **OccGGD**: as redes de ocorrência são mais abstratas que as gramáticas de grafos de ocorrência. Este problema pode ser contornado modificando-se a definição das redes de ocorrência para torná-las mais concretas ou modificando-se as gramáticas de grafos de ocorrência para torná-las mais abstratas, o que, obviamente, implica numa mudança da definição do modelo semântico utilizado. Porém, isto foge ao escopo dos objetivos desta dissertação.

Portanto, este trabalho mostrou que, usando a semântica de *unfoldings*, sempre se tem comportamentos compatíveis entre as redes L/T e as gramáticas de grafos discretas, de modo que qualquer uma delas pode ser modelada pela outra, preservando a semântica.

### Perspectivas para Trabalhos Futuros:

1. com este relacionamento entre as duas teorias, analisar quais resultados/conceitos de uma teoria podem ser aplicados na outra;
2. comparar o poder de expressão dos formalismos utilizando as relações e os funtores definidos;
3. modificar as definições da semântica obtendo novas categorias **NOccNets** e **NOccGGD** para que  $\mathbb{UR}$  seja adjunto direito e esquerdo de  $\mathbb{UG}$ , de modo a obter uma co-reflexão nos termos de [SAS 93]. Assim, o diagrama a seguir comutará a menos de isomorfismo.

$$\begin{array}{ccc}
 \text{PTNets} & \xrightarrow{U_n f^R} & \text{NOccNets} \\
 \mathbb{G} \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \mathbb{R} & & \mathbb{UG} \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \mathbb{UR} \\
 \text{GGD} & \xrightarrow{U_n f^G} & \text{NOccGGD}
 \end{array}$$

4. estender os conceitos definidos neste trabalho para outros tipos de gramáticas de grafos/redes de Petri;
5. utilizar outros modelos semânticos para estabelecer relações semânticas.

## A Transformação de Gramáticas de Grafos em Gramáticas de Grafos Discretas

O objetivo deste apêndice é estabelecer relações entre as categorias  $\mathbb{G}\mathbb{G}$  e  $\mathbb{G}\mathbb{G}\mathbb{D}$ . Estas relações serão representadas por funtores entre estas categorias ( $\mathbb{F} : \mathbb{G}\mathbb{G} \rightarrow \mathbb{G}\mathbb{G}\mathbb{D}$  e  $\mathbb{H} : \mathbb{G}\mathbb{G}\mathbb{D} \rightarrow \mathbb{G}\mathbb{G}$ ), sendo discutida a possibilidade de se encontrar definições que permitam uma adjunção entre tais categorias. (É sugerido mapeamento dos objetos. O mapeamento dos morfismos é feito de modo análogo e compatível.)

### A.1 Transformação Direta

Como  $\mathbb{G}\mathbb{G}\mathbb{D}$  é uma subcategoria de  $\mathbb{G}\mathbb{G}$ , uma definição trivial para  $\mathbb{H}$  é uma inclusão. Para uma definição de  $\mathbb{F}$ , foi visto no início da seção 1 do capítulo 4 que os vértices dos grafos não podem ser simplesmente esquecidos, pois isto pode resultar em regras que não são mais regras. Uma possibilidade é simplesmente transformar os arcos em vértices e “esquecer” o morfismo das regras. Neste caso, não é possível recuperar os arcos ou morfismos das regras pela aplicação de  $\mathbb{H}$ , i.e.,  $\mathbb{H}(\mathbb{F}(GG))$  em geral é diferente de  $GG$ , para  $GG$  objeto de  $\mathbb{G}\mathbb{G}$ . Estes funtores não constituem uma adjunção, pois, apesar de todos os vértices dos grafos de  $GG$  estarem presentes em  $\mathbb{H}(\mathbb{F}(GG))$ , quando alguma regra de  $GG$  preservar algum item, a relação de subregra (definição 3.10, página 37) imporá restrições aos morfismos entre  $GG$  e  $\mathbb{H}(\mathbb{F}(GG))$ . Restrições também são impostas se algum grafo de  $GG$  tiver arcos.

Um contra-exemplo para uma possível adjunção constituída por estes funtores é dado na figura A.1, onde  $GG_1$  é objeto de  $\mathbb{G}\mathbb{G}$  e  $GG_2$  é objeto de ambas as categorias: os conjuntos de morfismos  $[GG_1, \mathbb{H}(GG_2)]$  e  $[\mathbb{F}(GG_1), GG_2]$  não são isomórficos.

### A.2 Transformação Estruturada

Fornecendo uma estrutura aos elementos do conjunto de vértices, é possível uma definição para  $\mathbb{F}$  e  $\mathbb{H}$  de modo que  $\mathbb{H}(\mathbb{F}(GG)) = GG$ . Para isto, será considerada a categoria  $\mathbb{G}\mathbb{G}\mathbb{D}\mathbb{E}$ , onde os objetos são grafos discretos em que os vértices possuem uma certa estrutura (em termos de união disjuntiva): o funtor  $\mathbb{F} : \mathbb{G}\mathbb{G} \rightarrow \mathbb{G}\mathbb{G}\mathbb{D}\mathbb{E}$  é definido como:

$$\begin{array}{ccc}
 GG_1 : & I_1 = (\emptyset, \emptyset, \emptyset, \emptyset) & GG_2 : & I_2 = (\emptyset, \emptyset, \emptyset, \emptyset) \\
 \\
 N_1 = \emptyset & & N_2 = \emptyset & \\
 T_1 & \boxed{\begin{array}{c} \curvearrowright \\ \hookrightarrow \end{array}} & T_2 & \boxed{\begin{array}{c} \times \\ \bullet \end{array}}
 \end{array}$$

FIGURA A.1- Gramática de grafos discreta ( $GG_2$ ) e não-discreta ( $GG_1$ ).

- $\mathbb{F}$  esquece o morfismo das regras;
- $\mathbb{F}$  transforma os grafos da gramática da seguinte forma: um grafo  $G = (V, A, p, d)$  é transformado no grafo  $G' = (V \uplus A \uplus p \uplus d, \emptyset, \emptyset, \emptyset)$

Portanto, cada grafo é transformado num grafo discreto onde o conjunto de vértices é dado pela união disjuntiva dos conjuntos  $V$ ,  $A$ ,  $p$  e  $d$  (lembre-se de que as funções  $d$  e  $p$  não são nada mais do que um conjunto de pares ordenados). Deste modo, é possível reconstituir o grafo  $G$  a partir de  $G'$ , com a definição direta do functor  $\mathbb{H} : \text{GGDE} \rightarrow \text{GG}$ , de modo a desfazer a união disjuntiva. É fácil verificar as seguintes propriedades:

- $\mathbb{H}(\mathbb{F}(GG)) = GG$ , qualquer que seja  $GG$  objeto de  $\text{GG}$ .
- as categorias  $\text{GG}$  e  $\text{GGDE}$  são equivalentes (a única diferença é que os grafos usados em  $\text{GGDE}$  possuem vértices “estruturados”).

Devido ao fato de que os vértices impõem restrições aos morfismos, pode ocorrer o caso de que, para objetos  $GG_1$  e  $GG_2$  de  $\text{GGDE}$ , se tenha mais morfismos entre  $GG_1$  e  $GG_2$  do que entre  $\mathbb{H}(GG_1)$  e  $\mathbb{H}(GG_2)$ , i.e.,  $\mathbb{H}$  é sobrejetor em relação aos morfismos.

Pelos mesmos motivos da seção anterior, os funtores  $\mathbb{F}$  e  $\mathbb{H}$  definidos deste modo não constituem adjunção.

### A.3 Adjunção entre $\text{GG}$ e $\text{GGD}$

A dificuldade em se definir uma adjunção entre estas categorias sempre recai nos morfismos: a presença de arcos restringe os morfismos e a presença de regras que preservam itens os generaliza em alguns casos e os restringe em outros. Portanto, é necessário realizar algumas modificações nos morfismos da categoria  $\text{GG}$  para que os funtores definidos anteriormente passem a constituir uma adjunção. Por exemplo, as seguintes modificações possibilitariam a definição de uma adjunção:

1. generalizar os morfismos entre grafos de modo que seja possível mapear arcos sem a necessidade de mapeamento de destino e/ou origem. Para isto, é necessário modificar a definição 3.2, página 31 (comutatividade fraca) e a definição 3.1, página 30 (grafo).
2. restringir a relação de subregra, pois se uma regra  $r_1$  for subregra de  $r_2$ , itens que são preservados em  $r_2$  não precisam existir em  $r_1$ . Para isto, é necessário modificar a definição de subregra (definição 3.10, página 37).

Veja que a primeira modificação pode vir a dar aos arcos status de vértice, i.e., um vértice pode ser mapeado para um arco, e vice-versa. A realização destas modificações fogem ao escopo deste trabalho, uma vez que implicam numa redefinição das categorias sintática e semântica.

Com estas modificações, tem-se uma adjunção entre as as categorias  $\text{GG}$  e  $\text{GGD}$ , porém, em prejuízo da preservação da semântica, uma vez que o functor  $\mathbb{F}$  “esquece” o morfismo das regras que preservam itens. Vale enfatizar que, independentemente de qualquer modificação que possa ser feita para viabilizar uma adjunção, uma gramática de grafos que tenha regras que preservam itens será necessariamente mapeada para uma gramática em que as regras não preservam itens.

#### A.4 Implicação Semântica

Pelo fato das definições anteriores de  $\mathbb{F}$  ignorarem a preservação de itens pelas regras, a relação de conflito (definição 3.29, página 52) entre as regras de uma gramática  $GG$ , quando alguma de suas regras preservar itens, será diferente da relação de conflito entre as regras de  $\mathbb{F}(GG)$ , pois, como foi visto, uma regra que elimina um item não está em conflito com uma outra regra que preserva este mesmo item. Porém, duas regras que eliminam um mesmo item estão em conflito e são, por isso, mutuamente exclusivas.

A implicação semântica é que regras que poderiam ocorrer (i.e., serem aplicadas) em paralelo (veja a definição 3.36, página 58), agora somente podem ocorrer seqüencialmente. Para gramáticas de grafos, isto diminui o grau de paralelismo, uma de suas principais vantagens. Para efeitos de comparação com as redes de Petri, isto não é um problema, desde que se considere apenas os objetos de  $GGD$  na comparação.

Portanto, considerando os funtores supradefinidos, tem-se que a semântica de uma gramática de grafos  $GG$  não é isomórfica à semântica de  $\mathbb{F}(GG)$  (i.e., a semântica não é preservada). Note que isto independe da definição de  $\mathbb{F}$ , uma vez que, qualquer que seja sua definição, regras que preservam itens são mapeadas para regras que apenas eliminam itens. Obviamente, se for usada uma semântica que não distingue “preservar itens” de “eliminar e depois criar itens”, este problema deixa de existir.

Do exposto, pode-se concluir que a definição de uma adjunção as categorias  $GG$  e  $GGD$  não se adequa a este trabalho, que objetiva uma equivalência semântica utilizando o modelo semântico *unfolding*.

## Bibliografia

- [ADA 90] ADAMEK, J.; HERRLICH, H; STRECKER, G. **Abstract and Concrete Categories**. New York: John Wiley and Sons, 1990.
- [AND 96] ANDRIES, M. et al. **Graph Transformation for Specification and Programming**. Berlin: Technical University of Berlin, 1996. (Technical Report).
- [ASP 91] ASPERTI, A.; LONGO, G. **Categories, Types and Structures**. Cambridge: MIT Press, 1991.
- [BES 88] BEST, E; FERNÁNDEZ, C. **Nonsequential Processes - A Petri Net View**. Berlin: Springer-Verlag, 1988.
- [BRA 96] BRAUER, W; GOLD, R. Concurrent Processes and Petri Nets. In: **Proof and Computation**. Berlin: Springer Verlag, 1995. (NATO ASI Series, v. 139).
- [COR 95] CORRADINI, A. Concurrent Computing: from Petri Nets to Graph Grammars. **Theoretical Computer Science**, Amsterdam, n. 2, 1995.
- [COR 96] CORRADINI, A.; MONTANARI, U.; ROSSI, F. **Algebraic Approaches to Graph Transformation**. Part I: Basic Concepts and Double Pushout Approach. Berlin: Technical University of Berlin, 1996. (Technical Report).
- [DAS 96] DASSOW, J.; HABEL, A.; TAUBENBERGER, S. **Chain-code Pictures and Collages Generated by Hyperedge Replacement**. Berlin: Springer Verlag, 1996. (Lecture Notes in Computer Science, v. 1073).
- [EHR 79] EHRIG, H. Introduction to the Algebraic Theory of Graph Grammars. In: GRAPH GRAMMAR WORKSHOP, 1., 1979. **Proceedings...** Berlin: Springer Verlag, 1979. (Lecture Notes in Computer Science, v. 73).
- [EHR 96] EHRIG, H.; HECKEL, R.; KORFF, M.; LÖWE, M.; RIBEIRO, L.; WAGNER, A.; CORRADINI, A. **Algebraic Approaches to Graph Transformation**. Part II: Single Pushout Approach and Comparison with Double Pushout Approach. Berlin: Technical University of Berlin, 1996. (Technical Report 96-20).
- [FRA 96] FRACCHIA, F.D. **Integrating lineage and interaction for the visualization of cellular structures**. Berlin: Springer Verlag, 1996. (Lecture Notes in Computer Science, v. 1073).
- [KAP 92] KAPLAN, S.M.; LOYALL, J.P.; GOERING, S.K. **Specifying Concurrent Languages and Systems with  $\Delta$ -Grammars**. Berlin: Springer Verlag, 1992. (Lecture Notes in Computer Science, v. 532).
- [KOR 93] KORFF, M.; LÖWE, M.; WAGNER, A. An Algebraic framework for the transformation of attributed graphs. In **Term Graph Rewriting: Theory and Practice**. [S.l.]: John Wiley and Sons, 1993.
- [KOR 96] KORFF, M.; Ribeiro, L. **Formal Relationships between Graph Grammars and Petri Nets**. Berlin: Springer Verlag, 1996. (Lecture Notes in Computer Science, v. 1073).
- [KRE 80] KREOWSKI, H. **A Comparison between Petri Nets and Graph Grammars**. Berlin: Springer Verlag, 1980. (Lecture Notes in Computer Science, v. 100).
- [KRE 86] KREOWSKI, H.; WILHARM, Anne. Net Processes Correspond to Derivation Processes in Graph Grammars. **Theoretical Computer Science**, Amsterdam, v. 44, n. 3, 1986.
- [LIN 96] LINS, R.D; MARTINS, P.R.M.; CUNHA, P.R.F. **Introdução às Redes de Petri e Aplicações**. Campinas: Instituto de Computação/UNICAMP, 1996. Trabalho apresentado na Escola de Computação, 10., 1996.

- [LÖW 93] LÖWE, M. Algebraic Approach to Single-pushout Graph Transformation. **Theoretical Computer Science**, Amsterdam, v. 109, 1993.
- [LÜC 96] LÜCK, J.; LÜCK, H.B. **Cellworks with Cell Rewriting and Packing for Plant Morphogenesis**. Berlin: Springer Verlag, 1996. (Lecture Notes in Computer Science, v. 1073).
- [MAR 96] MARTINI, A. **Elements of Basic Category Theory**. Berlin: Technical University of Berlin, 1996. (Technical Report 96-5).
- [MES 90] MESEGUER, J; MONTANARI, U. Petri Nets are Monoids. **Information and Computation**, Orlando, FL, v. 88, n. 2, 1990.
- [MES 94a] MESEGUER, J; MONTANARI, U; SASSONE, V. **On the Semantics of Computation of Place/Transition Petri Nets**. Berlin: Springer Verlag, 1994. (Lecture Notes in Computer Science, v. 815).
- [MES 94b] MESEGUER, J; MONTANARI, U; SASSONE, V. **On the Model of Computation of Place/Transition Petri Nets**. Berlin: Springer Verlag, 1994. (Lecture Notes in Computer Science, v. 815).
- [PET 62] PETRI, C.A. **Kommunikation mit Automaten**. Bonn: Institut für Instrumentelle Mathematik, 1962. PhD Thesis.
- [REI 82] REISIG, W. **Petri Nets**. Berlin: Springer-Verlag, 1982.
- [RIB 96a] RIBEIRO, L. **A Telephone System's Specification Using Graph Grammars**. Berlin: Technical University of Berlin, 1996. (Technical Report 96-23).
- [RIB 96b] RIBEIRO, L. **Parallel Composition and Unfolding Semantics of Graph Grammars**. Berlin: Technical University of Berlin, 1996. PhD. Thesis.
- [SAS 93] SASSONE, V; NIELSON, M; WINSKEL, G. **A Classification of Models for Concurrency**. Berlin: Springer Verlag, 1993. (Lecture Notes in Computer Science, v. 715).
- [SCH 93] SCHNEIDER, H.J. On Categorical Graph Grammars Integrating Structural Transformations and Operations on Labels. **Theoretical Computer Science**, Amsterdam, v. 109, 1993.
- [TAE 96] TAENTZER, G. **Parallel and Distributed Graph Transformation: Formal Description and Application to Communication-based Systems**, Berlin: Technical University of Berlin, 1996. PhD. Thesis.
- [ZÜN 96] ZÜNDORF, A. **Graph Pattern Matching in PROGRES**. Berlin: Springer Verlag, 1996. (Lecture Notes in Computer Science, v. 1073).