III - Modelo Categórico de Linguagens Livres de Contexto

3.1. Introdução

O objetivo deste capítulo é apresentar um modelo categórico para gramáticas e linguagens livres de contexto. Estes modelos são essencialmente os apresentados por Walters em [Walters, 1989] e [Walters, 1991]. As poucas diferenças são que trabalhamos com categorias monoidais, em lugar de categorias com produtos e, além disso, desenvolvemos alguns aspectos com mais detalhes, como o teorema 3.30.

Os modelos apresentados neste capítulo são o ponto de partida da extensão de semântica para derivações da linguagem (capítulo IV) e da modelagem de tradução (capítulo VI). Existem na literatura, outros modelos categóricos de gramáticas livres de contexto como [Wells e Barr, 1988, pp. 518-521] e [Barr e Wells 1995, pag. 133].

O restante deste capítulo está organizado da seguinte forma:

- A seção 3.2 revisa os conceitos tradicionais de gramáticas e linguagens.
- A seção 3.3 apresenta o conceito categórico de gramáticas livres de contexto.
- A seção 3.4 apresenta o conceito categórico de linguagens livres de contexto.
- A seção 3.5 relaciona os conceitos tradicionais aos modelos categóricos.
- A seção 3.6 resume as idéias expostas neste capítulo.

3.2. Noções usuais de linguagens e gramáticas

Entendemos intuitivamente uma “linguagem” como sendo constituída de *símbolos* que, estruturados de acordo com determinadas *regras* (sintaxe) possuem um determinado *significado* (semântica). O estudo formal de linguagens iniciado por lingüistas na década de 50, ao ser assimilado pela comunidade de computação, provocou um impulso na área de linguagens de programação, sugerindo uma metodologia de especificação formal da sintaxe de uma linguagem e uma metodologia para a compilação desta.

O conceito formal de linguagem captura apenas o primeiro aspecto supracitado, i.e., que esta é constituída de *símbolos*. Posteriormente a introdução do conceito de *gramática* formaliza a noção de *regra*. Nesta seção, apresentamos um breve resumo das definições formais destes conceitos, que são amplamente difundidos nos dia de hoje. A referência clássica para o assunto é [Hopcroft e Ullman, 1969], em português destacamos [Menezes, 1997]. A notação adotada segue [Aho e Ullman, 1972].

Definição 3.1. Um **alfabeto** Σ é um conjunto finito de símbolos.

Definição 3.2. Uma **sentença** sobre Σ é uma seqüência finita de símbolos de Σ.

Definição 3.3. $\boldsymbol{\varepsilon}$ é a sentença vazia sobre Σ.

Definição 3.4. $\boldsymbol{\Sigma^*}$ é o conjunto de todas as sentenças sobre Σ, ou seja, o conjunto suporte do monóide livremente gerado a partir de Σ. O conjuntos das sentenças não vazias é chamado de $\boldsymbol{\Sigma^+}$.

Definição 3.5. Uma **linguagem** L sobre Σ é um subconjunto de Σ^*.

Portanto, uma linguagem inclui conjuntos que definitivamente não têm uma *estrutura* (ou sintaxe), como os conjuntos não-recursivos. A noção de estrutura é formalizada através do conceito de *gramática*.

Definição 3.6. Uma **gramática** G é uma quádrupla, $G = (N,\Sigma,P,S)$, onde Σ é um alfabeto, N é um conjunto de "símbolos não-terminais", P é um conjunto finito de regras da forma $\alpha\rightarrow\beta$, $\alpha\in(\Sigma\cup N)^+$, $\beta \in (\Sigma\cup N)^*$ e $S \in N$ é o "símbolo inicial".

Definição 3.7. Uma gramática $G = (N,\Sigma,P,S)$ é uma **gramática livre de contexto** (GLC), se toda regra em P é da forma $A \rightarrow \beta$, $A \in N$.

Exemplo 3.8. A gramática $G_{EXP} = (\{id,+,*\},\{E,ID,S,P\},\{E\to ID,\ E\to ESE,\ E\to EPE,\ ID\to id,\ S\to +,\ P\to \times\},E)$ é livre de contexto.

Definição 3.9. Dada uma gramática $G = (N,\Sigma,P,S)$, definimos a relação $\Rightarrow_{\mathbf{G}}$ ("deriva em um passo"), tal que $\forall w_1,w_2 \in (\Sigma\cup N)^*$, $w_1 \Rightarrow_{\mathbf{G}} w_2$ se e somente se $w_2 = w_1 [\beta/\alpha]$ (onde exatamente uma ocorrência de α é substituída por β, $\alpha\to\beta \in P$). Usaremos simplesmente $\Rightarrow$, quando não for necessário especificar em que gramática ocorre a derivação.

Definição 3.10. A relação $\Rightarrow^{*}_{\mathbf{G}}$ ("deriva em zero ou mais passos") é o fecho transitivo-reflexivo da relação $\Rightarrow_G$.

Finalmente, a próxima definição relaciona o conceito de gramática com o conceito de linguagem, o que nos permite falar da "estrutura sintática" de uma linguagem.

Definição 3.11. A **linguagem gerada por uma gramática** $G = (N,\Sigma,P,S)$ é $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^*_G w\}$.

Definição 3.12.. Uma **linguagem livre de contexto** (LLC) é uma linguagem L tal que $L = L(G)$ para alguma gramática livre de contexto G.

Definição 3.13. Seja $G = (N,\Sigma,P,S)$, $A \in N$, $w \in (\Sigma\cup N)^*$, $A \Rightarrow^*_G w$. Uma **derivação de w a partir de A**, é a seqüência das regras de P aplicadas a 'A' para se obter w. A derivação é chamada de **mais à esquerda** se em cada passo as regras forem aplicadas ao não-terminal mais à esquerda.

Exemplo 3.14. A derivação abaixo é uma derivação mais à esquerda de "id × E" a partir de "E" na gramática G_{EXP} (exemplo 3.8):

$$E \Rightarrow E\ P\ E \Rightarrow ID\ P\ E \Rightarrow id\ P\ E \Rightarrow id \times E.$$

Definição 3.15. Seja $G = (N,\Sigma,P,S)$, $A \in N$, $w \in (\Sigma\cup N)^*$, $A \Rightarrow^*_G w$. Uma **árvore de derivação de w a partir de A** é uma árvore onde:

- Todo nó possui um nome que é um símbolo de $(\Sigma \cup N)$.
- Se os nós de nome $B_1, B_2, \ldots, B_n$ são, nesta ordem, os descendentes diretos de um nó B, então $B \rightarrow B_1 B_2 \ldots B_n$ é uma regra de produção em P.
- O nome do nó-raiz é A.
- A concatenação do nome das folhas, tomadas da esquerda para a direita, é w.

Existe uma correspondência biunívoca entre as árvore de derivação e as derivações mais à esquerda [Aho e Ullman, 1972, pag. 142]. Da mesma forma, podemos associar tuplas de árvores a trechos de derivação cujo lado esquerdo é uma tupla de símbolos não-terminais.

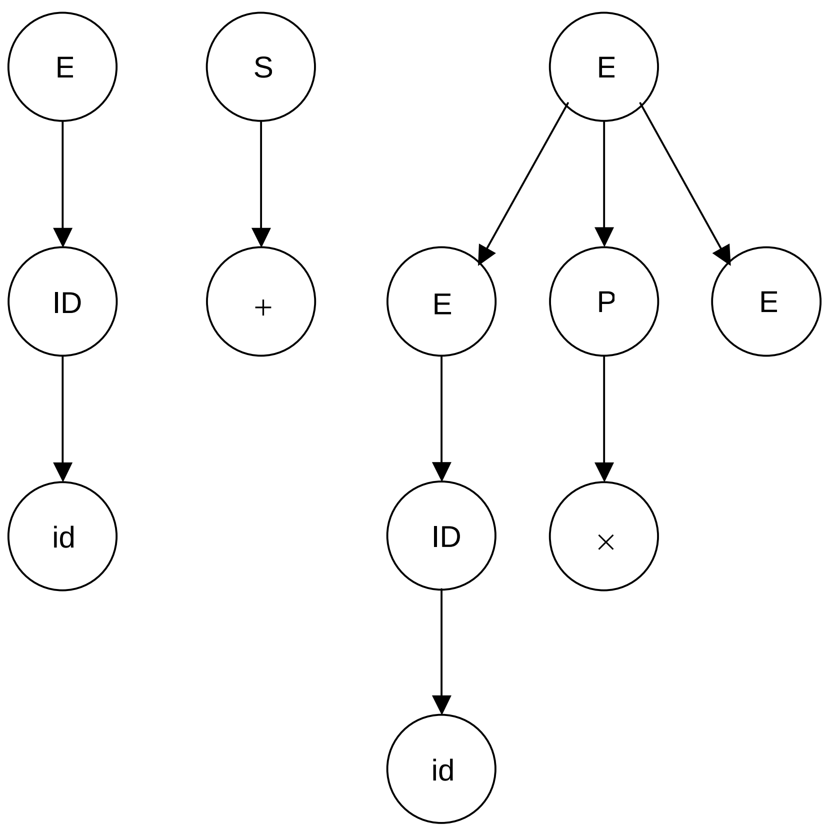

Figura 3.1 Tupla de árvores representando um trecho de derivação.

Exemplo 3.16. A figura 3.1 ilustra uma tupla de árvores representando o seguinte trecho de derivação em G_{EXP}:

$$E\ S\ E \Rightarrow ID\ S\ E \Rightarrow id\ S\ E \Rightarrow id + E \Rightarrow id + E\ P\ E \Rightarrow id + ID\ P\ E \Rightarrow id + id\ P\ E \Rightarrow id + id \times E$$

Durante o restante deste trabalho estaremos manipulando morfismos de uma categoria que representarão tuplas de árvores de derivação. Muitas vezes faremos referência a esses morfismos como "derivações", quando o mais preciso seria "tuplas de derivações mais à esquerda".

3.3. Noção categórica de gramáticas livres de contexto

3.3.1 Multigrafos

O ponto de partida para a construção de um modelo categórico de gramáticas livres de contexto é o conceito de multigrafo. Um multigrafo é um grafo cujas arestas possuem mais de um domínio. A figura 3.2 ilustra este conceito.

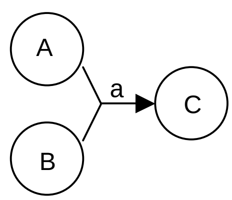

Figura 3.2 Exemplo de Multigrafo.

O multigrafo acima tem três nós {A,B,C} e uma aresta 2-ária a com domínios A e B e codomínio C. Se pensamos em modelar regras de uma gramática como arestas de um multigrafo, é fundamental que os domínios sejam ordenados. Isso não é claro a partir da representação gráfica do multigrafo. No exemplo acima, podemos dizer que o primeiro domínio de a é A e o segundo domínio é B.

Podemos ver um multigrafo como um conjunto $*$ de nós e, para cada $n \in \mathbb{N}$, um conjunto A_n de arestas n-árias com uma função c: $A_n \rightarrow *$ que associa cada aresta a seu codomínio e, para cada i, $1 \leq i \leq n$, uma função d_i: $A_n \rightarrow *$ que determina o i-ésimo domínio de cada aresta n-ária. A sobrecarga de nomes para as funções d_i (que representam a função "i-ésimo domínio" para arestas de diferentes aridades) não acarreta problemas, pois seus domínios são diferentes e, em geral, estão claros pelo contexto. No caso do exemplo acima, teríamos $* = \{A,B,C\}$, $A_n = \emptyset$ para $n \neq 2$, $A_2 = \{a\}$, $c(a)=C$, $d_1(a)=A$, $d_2(a)=B$. A modelagem descrita acima equivale a um funtor da categoria **D** (descrita na figura 3.3) em **SETS**. Na figura omitimos as identidades e sobrecarregamos os nomes dos morfismos para não sobrecarregar a notação. Os morfismos podem ser diferenciados pelo seu domínio.

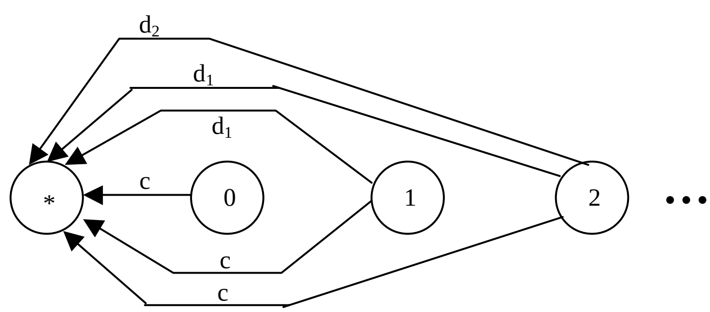

Figura 3.3 Categoria **D**.

Definição 3.17. A categoria **Mgph** dos multigrafos é $\mathbf{SETS}^{\mathbf{D}}$. Um multigrafo é um elemento de **Mgph** e um morfismo entre multigrafos é um morfismo de **Mgph**.

3.3.2 Gramáticas Livres de Contexto

Para usarmos multigrafos como modelos de gramáticas livres de contexto há alguns problemas a serem resolvidos. Em primeiro lugar, os multigrafos têm um poder de expressividade muito maior que as gramáticas livres de contexto, pois podem ter infinitas arestas (que serão o modelo de regras), podendo gerar até mesmo linguagens não-recursivas. Evidentemente não há utilidade em se utilizar $\mathbf{FINSETS}^{\mathbf{D}}$, pois o multigrafo ainda poderia ter uma quantidade infinita de arestas. Nesse caso seguiremos a definição de

Walters e simplesmente trabalharemos com “multigrafos com uma quantidade finita de objetos e arestas”[†].

Sabemos também que os multigrafos não têm um símbolo inicial; seus nós são indistinguíveis. A construção de Walters ignora este fato, o que leva ele a definir a linguagem gerada por uma gramática em função de um determinado símbolo não terminal. No presente trabalho, preferimos explicitar os símbolos iniciais no modelo, acrescentando um objeto na categoria D.

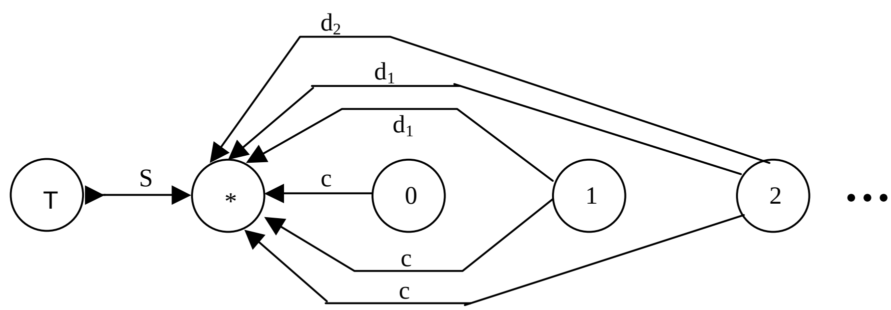

Figura 3.4 Categoria **D’** com símbolo inicial.

Definição 3.18. Uma **gramática livre de contexto categórica** (GLCC) sobre um alfabeto Σ é uma tripla **G** = (G,Σ,ϕ), onde **D’** é a categoria exibida na figura 3.4, G:**D’**$\rightarrow$**SETS** é um funtor tal que $G(\top)$ é um objeto terminal e $\cup G(i)$ é finita e $\phi:G(0)\rightarrow \Sigma^*$ é uma função que atribui rótulos aos morfismos 0-ários. Usaremos simplesmente “gramática livre de contexto” quando for claro que falamos do conceito categórico. Na prática, podemos pensar em GLCCs como multigrafos finitos com um “símbolo inicial”, nos quais as arestas 0-árias possuem rótulos. Essa idéia é ilustrada no exemplo 3.19.

[†] Uma especificação diagramática desta condição encontra dificuldades impostas pelo teorema de Birkhoff: “Uma classe K de álgebras é equacional sss é uma variedade”. O que não é o caso, pois a classe definida por esta condição não é fechada em relação ao produto infinito.

Σ^* e ϕ são deixados de fora da categoria **D’** porque, posteriormente, ao se introduzir o conceito de tradução, a componente Σ^* de uma tradução não será componente de uma transformação natural.

Exemplo 3.19. A figura 3.5 ilustra a gramática de expressões infixas e a representação gráfica de seu modelo categórico, $\mathbf{G_{EXP}}$. Aproveitamos para introduzir a convenção de marcar $G_{EXP}(S)(G_{EXP}(\top))$ com um círculo duplo. Observe que a representação gráfica não contempla toda a informação do modelo categórico, falta dizer a “ordem” dos domínios das arestas com aridade 3.

$E \to E\ S\ E$
$E \to E\ P\ E$
$E \to ID$
$S \to +$
$P \to \times$
$ID \to id$

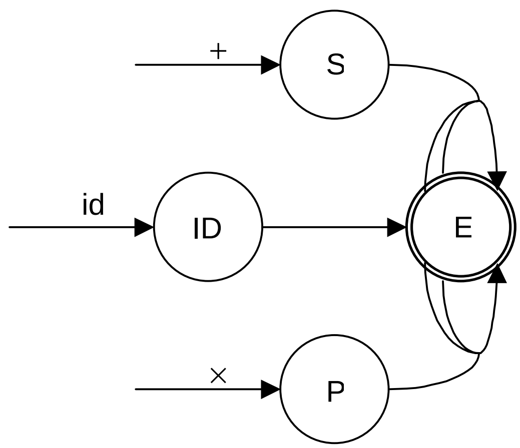

Figura 3.5 Gramática G_{EXP}.

Definição 3.20. Definimos a categoria $\mathbf{Gram}^{\Sigma}$ como a categoria cujos objetos são triplas $\mathbf{G} = (G,\Sigma,\phi)$, onde G:**D’**$\to$**SETS** é um funtor tal que $G(\top)$ é um objeto terminal, Σ é um alfabeto e ϕ é uma função $\phi{:}G(0)\to\Sigma^*$. Os morfismos entre (G_1,Σ,ϕ_1) e (G_2,Σ,ϕ_2) são transformações naturais $\eta{:}G_1 \xrightarrow{\bullet} G_2$. Entre os objetos de $\mathbf{Gram}^{\Sigma}$, aqueles que satisfazem a condição de finitude da definição 3.18 são gramáticas sobre Σ.

3.4 Noção categórica de linguagens livres de contexto

3.4.1 Linguagens livres de contexto

Uma vez que estabelecemos um modelo para GLCs, falta-nos criar um mecanismo para compor as regras de produção de forma a gerar uma linguagem. Walters faz isso através de uma adjunção entre **Mgph** e $\mathbf{Cat}_\times$ (categoria das categorias pequenas com produtos finitos estritamente associativos). Esta abordagem apresenta o inconveniente de apresentar morfismos não necessários na linguagem (as projeções). Em lugar de $\mathbf{Cat}_\times$, iremos trabalhar com a categoria das categorias pequenas estritamente monoidais [Mac Lane, 1971, pp. 157-158], nas quais um dos objetos é marcado como "símbolo inicial". Iniciamos apresentando o conceito de "floresta" e em seguida mostraremos como construir a linguagem gerada por uma gramática livre de contexto categórica.

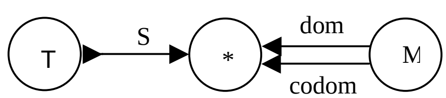

Figura 3.6 Categoria **A**.

Definição 3.21. Uma **floresta de derivações (ou floresta)** sobre um alfabeto Σ é uma tripla **F** = (F, Σ, ϕ, **C**), onde:

- F é um funtor F: **A** → **SETS**, tal que:
  - F(T) é um objeto terminal em **Sets**.
  - F(*) e F(M) são respectivamente os objetos e os morfismos de uma categoria estritamente monoidal **C** = (C, $\square$, **1**).
  - F(dom) e F(codom) são as funções que retornam, respectivamente, o domínio e o codomínio de um dado morfismo em **C**.

- ϕ é uma função dos "morfismos 0-ários" de C em Σ^*. Isto é seja Z = { m ∈ F(M) | m: **1** →A, A ≠ **1** } então ϕ é uma função ϕ:Z→Σ^*.

Ou seja, uma floresta é uma categoria estritamente monoidal pequena pontuada, cujos morfismos "0-ários" possuem um rótulo em Σ^*. O motivo do nome "floresta" ficará mais claro posteriormente, pois cada morfismo em C representará uma árvore de derivação em uma determinada gramática. O exemplo 3.22 ilustra esta relação.

Exemplo 3.22. A figura 3.7 ilustra a representação gráfica da floresta de derivações para a linguagem gerada pela gramática G_a com o conjunto de regras P = {S→AS, S→A, A→a}. Da mesma forma que no caso das GLCCs convencionamos marcar F_a(S)(F_a(T)) com um círculo duplo e indicamos ϕ através de rótulos nas arestas apropriadas. Note que a representação da figura 3.7 é parcial, pois a linguagem na verdade possui infinitas derivações.

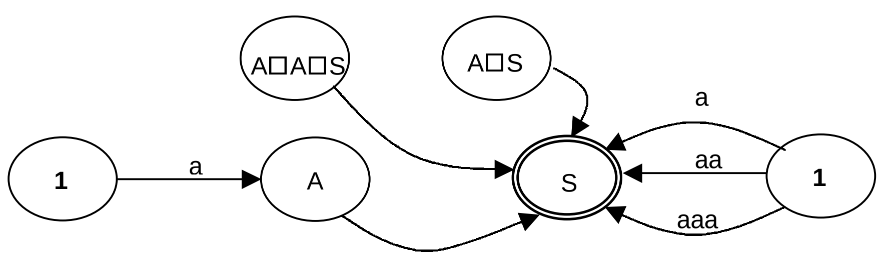

Figura 3.7 Floresta associada à linguagem gerada por G_a.

Definição 3.23. **Cat**$^{\Sigma}_{\square}$ é a categoria cujos objetos são florestas de derivações e cujos morfismos entre (F$_1$, Σ, ϕ_1, **C$_1$**) e (F$_2$, Σ, ϕ_2, **C$_2$**) são transformações naturais η:F$_1 \stackrel{\bullet}{\to}$ F$_2$, tais que η_* e η_M são respectivamente as componentes objeto e morfismo de um funtor entre **C$_1$** e **C$_2$**, que preserva o produto.

3.4.2 Linguagem gerada por uma gramática.

Para definir a linguagem gerada por uma gramática livre de contexto, precisamos de um meio para compor as regras de produção, formando derivações. Para isso, definiremos um funtor $U_\square$: $\mathbf{Cat}^{\Sigma}_{\square} \to \mathbf{Gram}^{\Sigma}$, que "esquece" o produto "□" da categoria estritamente monoidal, codificando este produto usando as setas n-árias em $\mathbf{Gram}^{\Sigma}$. O seu adjunto à esquerda é o funtor que procuramos, o qual compõe as regras de produção.

Definição 3.24. $U_\square$: $\mathbf{Cat}^{\Sigma}_{\square} \to \mathbf{Gram}^{\Sigma}$ é um funtor definido como:

- A componente-objeto de $U_\square$ leva uma floresta (F, Σ, ϕ_{F}, (C,□,**1**)) a uma gramática (G, Σ, ϕ_G) tal que:
  - ❑ $G(\mathsf{T}) = \mathsf{F}(\mathsf{T})$.
  - ❑ $G(*) = \mathsf{F}(*)$.
  - ❑ $\mathsf{F}(S) = G(S)$.
  - ❑ Para cada aresta $a \in \mathsf{F}(M)$, $a{:}X_1\square X_2\square\ldots\square X_n \to Y$, tal que X_i, $Y \in G(*)$, existe $a' \in G(n)$ tal que $c(a')=Y$ e $d_i(a')=X_i$. O caso em que a: $\mathbf{1}\to Y$ é um caso particular da construção anterior onde n = 0.
  - ❑ $\phi_G(a')=\phi_{\mathsf{F}}(a)$, para $a' \in G(0)$. Observe que ambas as funções tem o mesmo codomínio, Σ^*.
- A componente-morfismo de $U_\square$ leva um morfismo $\eta_\square$ de $\mathbf{Cat}^{\Sigma}_{\square}$ em um morfismo η_G em $\mathbf{Gram}^{\Sigma}$ tal que:
  - ❑ $\eta_{G_{\mathsf{T}}} = \eta_{\square_{\mathsf{T}}}$.
  - ❑ $\eta_{G*} = \eta_{\square *}$.
  - ❑ $\eta_{G_i}(a')=b' \Leftrightarrow \eta_{\square_M}(a)=b$, para a' e b' correspondentes à definição da componente-objeto.

Observe que, em geral, $U_\square$ leva florestas de derivações em objetos de **Gram**$^\Sigma$ que não são gramáticas, pois possuem infinitas arestas. Ainda assim, este objeto de **Gram**$^\Sigma$, quando da forma $U_\square L_\square(G, \Sigma, \phi_G)$, gera a mesma linguagem que (G, Σ, ϕ_G).

Exemplo 3.25. A figura 3.8 ilustra a componente objeto do funtor $U_\square$. Os morfismos em **F** correspondem a algumas das derivações produzidas pela gramática $S\rightarrow AS$, $S\rightarrow A$, $A\rightarrow a$. O processo inverso, de se obter **F** a partir da gramática, é feito através do funtor $L_\square$ introduzido em 3.26 e ilustrado no exemplo 3.27.

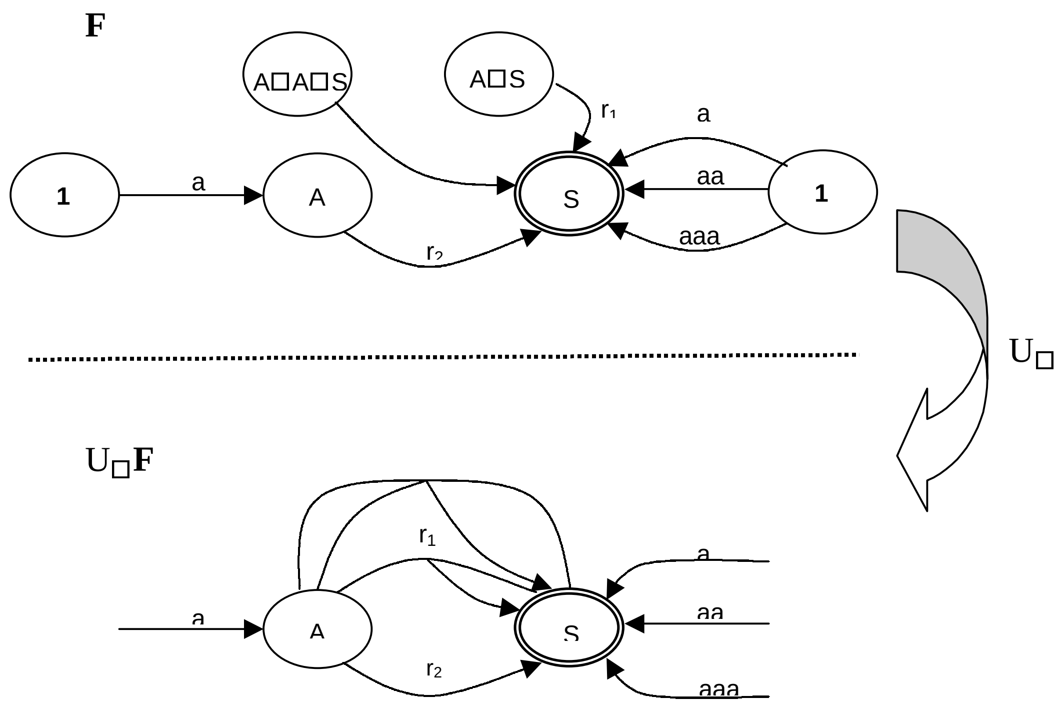

Figura 3.8 Exemplo de $U_\square$F.

Teorema 3.26. $U_\square$ possui um adjunto à esquerda $L_\square$:**Gram**$^\Sigma\rightarrow$**Cat**$^\Sigma_\square$ tal que:

- A componente-objeto de $L_\square$ leva uma gramática (G,Σ,ϕ_G) a uma floresta $(F,\Sigma,\phi_F,$**C**$)$:
  - $F(T) = G(T)$, enquanto, $F(S) = G(S) \circ i$, onde i é a inclusão $i:G(*) \hookrightarrow F(*)$.
  - $F(*) = \{ x \mid x$ é produto finito de elementos de $G(*) \}$, portanto $F(*) \supseteq G(*)$.

    - Para cada aresta n-ária a’ $\in$ G(n) tal que c(a’)=Y e d_i(a’)=X_i, existe uma aresta a:$X_1 \square X_2 \square \ldots \square X_n \to$ Y em $L_\square$(G).
    - As arestas de $L_\square$(G) são as correspondentes ao item acima e as obtidas através de composição ou produto destas.
    - $\phi_F(a) = \phi_G(a')$, para a:**1**$\to$Y.
- A componente-morfismo de $L_\square$ leva um morfismo η_G de $\mathbf{Gram}^\Sigma$ em um morfismo $\eta_\square$ em $\mathbf{Cat}^\Sigma_\square$ tal que:
    - $\eta_{\square T}=\eta_{GT}$, enquanto, $\eta_{\square *}$ estende η_{G*} para F(*) preservando produtos.
    - $\eta_{\square M}(a)=b \Leftrightarrow \eta_{G_i}(a')=b'$, para a’ e b’ correspondentes à definição da componente-objeto.

A demonstração deste e dos demais teoremas enunciados neste trabalho é apresentada no Apêndice A, no intuito de melhorar a legibilidade do texto.

Exemplo 3.27. A figura 3.9 ilustra **parte** da floresta $F_{EXP} = L_\square \mathbf{G_{EXP}}$. Onde $\mathbf{G_{EXP}}$ é a gramática do exemplo 3.19. Em particular, a figura ilustra o morfismo g associado à derivação: E $\Rightarrow$ E S E $\Rightarrow$ ID S E $\Rightarrow$ id S E $\Rightarrow$ id + E $\Rightarrow$ id + ID $\Rightarrow$ id + id.

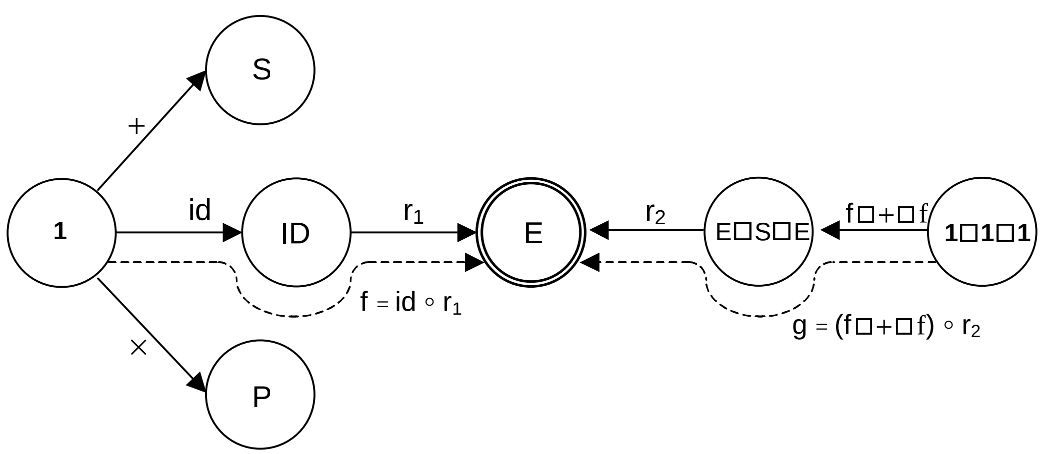

Figura 3.9 Parte da Floresta $F_{EXP} = L_\square(G_{EXP})$.

Na figura acima, **1** é o elemento neutro da operação $\square$, desta forma, $\mathbf{1}\square\mathbf{1}\square\mathbf{1} = \mathbf{1}$. Portanto g é o morfismo desejado de **1** para E.

3.5 Relação entre as noções usuais e categóricas

Informalmente, podemos entender uma gramática livre de contexto categórica como uma representação de uma gramática livre de contexto na qual todas as regras de produção são da forma $A \rightarrow \alpha$ ou $A \rightarrow w$ (onde $A \in N$, $\alpha \in N^*$ e $w \in \Sigma^*$). A fim de formalizar essa relação, introduzimos o conceito de linguagem gerada por uma gramática livre de contexto categórica.

Definição 3.28. Dada uma floresta $(F,\Sigma,\phi,\mathbf{C})$ tal que $(F,\Sigma,\phi,\mathbf{C})= L_{\square}(G,\Sigma,\phi)$ e um morfismo m de F(M) tal que $m \in Hom_{\mathbf{C}}(1, F(T)F(S))$, m pode ser fatorado da seguinte forma: $m = m_A \circ m_B$, onde m_A é a composição de morfismos que não sejam 0-ários e $m_B = m_1 \square m_2 \square \ldots \square m_n$ onde m_i, $i \in \{1,..n\}$, é um morfismo 0-ário. Neste caso, definimos **T**(m), "os terminais de m" como o *string* $T(m) = \phi(m_1) \bullet \phi(m_2) \bullet \ldots \bullet \phi(m_n)$. Onde $\bullet$ é a concatenação de *strings*.

Lembramos que os morfismos em F(M) correspondem a derivações; portanto, T(m) está bem definida e retorna os terminais do lado direito da derivação.

Definição 3.29.. A linguagem gerada por uma gramática livre de contexto categórica **G** é dada por $L(\mathbf{G}) = \{ T(m) \mid m \in Hom_{\mathbf{C}}(1, F(T)F(S)) \}$. Onde $(F,\Sigma,\phi,\mathbf{C}) = L_{\square}\mathbf{G}$.

A relação entre os conceitos categórico e usual de LLCs é que eles possuem o mesmo poder de expressividade. Isso é formalizado através do teorema abaixo.

Teorema 3.30. Para cada gramática livre de contexto G, podemos construir uma gramática livre de contexto categórica **G'**, tal que L(**G'**) = L(G). Reciprocamente, para cada gramática livre de contexto categórica **G'** podemos construir uma gramática livre de contexto G tal que L(G) = L(**G'**).

3.6 Resumo

Iniciamos o capítulo apresentando os conceitos tradicionais de gramáticas e linguagens livres de contexto. Em seguida, apresentamos um modelo categórico para esses conceitos. Esses modelos são o ponto de partida para o modelo de tradução entre linguagens a ser introduzido no capítulo V.

Concluímos o capítulo apresentando a definição de linguagem gerada por GLCs categóricas e provando que estas têm o mesmo poder de expressividade das GLCs usuais. No restante deste trabalho estaremos mais interessados em L□**G’**, pois necessitaremos da estrutura sintática da linguagem, que não está presente em L(**G’**).