

2 FORMALISMO ADAPTATIVO E SEU USO EM LINGUAGEM NATURAL

Este capítulo tem por objetivo apresentar, dentre os diversos aspectos do formalismo adaptativo, em particular da gramática e dos autômatos de pilha, apenas aqueles cujos desdobramentos foram identificados como primordiais no tratamento de linguagens dependentes de contexto.

Também relatam-se aqui as conclusões obtidas do estudo detalhado de alguns exemplos apresentados em (NETO, 1993). Tais exemplos referem-se a problemas encontrados em Linguagens de Programação, mas o que se verificou é que podem ser empregados também no processamento de aspectos dependentes de contexto das Linguagens Naturais, por exemplo, em tarefas como a análise de concordâncias e a delimitação de estruturas coordenadas e subordinadas.

2.1 Autômatos de Pilha Adaptativos

Autômatos de pilha adaptativos são dispositivos reconhecedores que exibem a estrutura básica dos autômatos de pilha, mas que incorporam em adição mecanismos capazes de efetuar alterações dinâmicas em sua própria estrutura, em resposta a cada particular texto de entrada analisado. Tais aceitadores podem ser empregados no reconhecimento de linguagens dependentes de contexto.

Este modelo incorpora três mecanismos de reconhecimento, a saber:

- Máquinas de estados finitos, para o tratamento de formas sintáticas regulares. São representadas pelas sub-máquinas dos autômatos de pilha estruturados que servem como formalismo subjacente do autômato adaptativo;
- Pilhas de controle, que usadas pelas sub-máquinas no controle da análise das formas sintáticas mutuamente recursivas, são fundamentais para o tratamento de formas sintáticas livres de contexto;
- Mecanismos que promovem a auto-modificação dinâmica do autômato adaptativo, destinados primordialmente ao tratamento de dependências de contexto.

Assim, no caso geral todos os recursos do autômato são utilizados no reconhecimento; no caso livre de contexto não-regular o autômato somente aciona a utilização dos recursos de sub-máquinas e pilha e naturalmente no caso de linguagens regulares, o autômato pode também dispensar o uso da pilha.

Um autômato de pilha adaptativo pode ser representado por um conjunto de regras de transição da forma $(\gamma \mathbf{g}, \mathbf{e}, \mathbf{s} \alpha) : A, \rightarrow (\gamma \mathbf{g}', \mathbf{e}', \mathbf{s}' \alpha), B$, onde:

- $(\gamma \mathbf{g}, \mathbf{e}, \mathbf{s} \alpha)$ representa a situação do autômato antes da aplicação da regra:
 - \mathbf{g} (opcional) indica o conteúdo exigido do topo da pilha antes da execução da regra de transição. Este elemento é desempilhado pela aplicação da regra de transição. No caso de omissão de \mathbf{g} , a aplicação da regra de transição não será dependente do conteúdo do topo da pilha.
 - \mathbf{e} representa o estado-origem da transição descrita pela regra em questão.
 - \mathbf{s} (opcional) representa o símbolo da cadeia de entrada a ser consumido pela aplicação da regra. Este elemento pode ser um dos símbolos básicos de que se compõe o texto de entrada, ou então qualquer dos símbolos nele empilhados como consequência da execução das regras de transição do autômato adaptativo.

• $(\gamma \mathbf{g}' , \mathbf{e}' , \mathbf{s}' \alpha)$ representa a situação do autômato depois da aplicação da regra de transição:

- \mathbf{g}' (opcional) representa o elemento a ser empilhado como resultado da aplicação da regra de transição. Caso seja omitido, nenhum empilhamento será efetuado como decorrência da execução dessa regra.
- \mathbf{e}' representa o estado-destino da transição que a regra de transição descreve.
- \mathbf{s}' (opcional) corresponde a um símbolo, a ser incluído na cadeia de entrada como resultado da execução da transição descrita pela regra de transição:

$$(\gamma \mathbf{g} , \mathbf{e} , \mathbf{s} \alpha) : \mathbf{A} , \rightarrow (\gamma \mathbf{g}' , \mathbf{e}' , \mathbf{s}' \alpha) , \mathbf{B}$$

Considere-se a situação em que $\mathbf{g} = \mathbf{g}'$. O mapeamento destas regras de transição para a gramática adaptativa correspondente, emprega o meta-símbolo " \leftarrow " introduzido por Iwai, conforme será apresentado adiante.

Cumpra observar que caso $\mathbf{s} = \mathbf{s}'$, tais elementos deverão aparecer explicitamente na regra de transição, o que então representa uma transição que não efetua alterações sobre a cadeia de entrada, mas que exige, para ser aplicada, que o próximo símbolo de entrada a ser consumido seja $\mathbf{s} = \mathbf{s}'$, implementando assim uma operação de "look-ahead".

Os elementos $\mathbf{g} , \mathbf{g}' , \mathbf{s}$ e \mathbf{s}' explicitam apenas a parte da pilha ou da cadeia de entrada que são relevantes à aplicação da regra de transição.

- \mathbf{A} (opcional) representa uma ação adaptativa a ser executada antes de ser efetuada a transição de estado.
- \mathbf{B} (opcional) simboliza uma ação adaptativa a ser executada após a realização da transição de estado.

Há alguns casos a considerar a respeito da interpretação do símbolo \mathbf{s} nas regras de transição, a saber:

- s representa uma cadeia vazia: neste caso, obviamente uma transição não é condicionada ao conteúdo da cadeia de entrada;
- s corresponde a um símbolo ASCII denotado entre aspas: isto indica que tal símbolo deve ser utilizado diretamente pela transição. (NETO, 1993)] observa que:

“[...]a bem da clareza, convém confinar o uso deste tipo de regras de transição apenas à parte do autômato que executa a função de análise léxica, onde é extensivamente utilizado para a extração de átomos a partir do texto-fonte[...]”

- s pode corresponder a qualquer símbolo que não seja um dos símbolos ASCII, e neste caso, deve-se subentender que este símbolo esteja representando um não-terminal, ao qual deverá estar associada uma sub-máquina que, a partir de algum dos seus estados finais, empilhe na cadeia de entrada o símbolo s correspondente. Esclarece o autor a respeito do consumo do símbolo s e observa que nas partes de autômato encarregadas da análise sintática, não devem figurar regras de transição que representem transições diretas com caracteres ASCII:

[...]Supondo que I seja o estado inicial de tal sub-máquina, a regra de transição acima $((e, s) : A \rightarrow e', B)$ deverá ser interpretada como sendo a abreviatura do seguinte par de regras de transição:

$$(\gamma, e, \alpha) : A, \rightarrow (\gamma e, I, \alpha)$$

$$(\gamma, e, s, \alpha) : \rightarrow (\gamma, e', \alpha), B$$

desde que se convençione que todos os retornos de sub-máquinas empilhem na cadeia de entrada uma informação, associada ao estado final por onde se deu o retorno, correspondente ao símbolo não-terminal s esperado. Note-se que A e B são opcionais, e que a sub-máquina mencionada efetua normalmente a função de um analisador léxico.

Assim, uma regra de transição da forma $(e, s) : A, \rightarrow e', B$ representa uma transição de um autômato encarregado da análise sintática. Esta regra supõe que seja realizada uma transição para uma máquina M , analisador léxico, cujo estado inicial é I . O analisador léxico é um transdutor, que poderá mediante o consumo de símbolos presentes na cadeia de entrada,

inserir subseqüentemente um símbolo s . Quando o estado e for desempilhado, e uma vez que se apresente na cadeia de entrada o símbolo s , inserido pelo analisador léxico, o analisador sintático poderá prosseguir com o reconhecimento a partir do estado e' .

- Um outro caso pertinente ao símbolo s diz respeito às transições envolvendo símbolos formados de caracteres ASCII isolados na parte sintática do autômato. Neste caso, o símbolo s representa um símbolo ASCII σ , e este não está denotado entre aspas ou, se apresenta denotado na forma $\nabla \text{"}\sigma\text{"}$, ou seja, as regras de transição são:

$$(e, \sigma) : A \rightarrow e', B$$

$$\text{ou } (e, \nabla \text{"}\sigma\text{"}) : A \rightarrow e', B$$

Deve-se entender que tal regra de transição abrevia o seguinte par:

$$(\gamma, e, \text{"}\sigma\text{" } \alpha) : A \rightarrow (\gamma e, I, \text{"}\sigma\text{" } \alpha)$$

$$\text{e } (\gamma, e, s\alpha) \rightarrow (\gamma, e', \alpha), B$$

com s representando, portanto, para efeito de análise sintática, um meta-símbolo que simboliza o código ASCII correspondente.

O uso de transições envolvendo símbolos formados de caracteres ASCII isolados na parte sintática do autômato pode ser indicada utilizando-se este tipo de regras de transição em lugar de transições diretas com caracteres ASCII.

Enquanto no caso anterior construções sintáticas da linguagem supõem a ocorrência de uma seqüência de símbolos na cadeia de entrada, neste caso o formalismo prediz a ocorrência de um símbolo terminal isolado. Adianta-se portanto, que um reconhecedor adaptativo poderá apresentar dois modos de operação: aquele em que se consome uma seqüência de símbolos da cadeia de entrada e a subseqüente inserção de um símbolo não-terminal na cadeia de entrada, e aquele em que se consome um único símbolo terminal $\text{"}\sigma\text{"}$ e a subseqüente inserção do símbolo σ na cadeia de entrada.

2.2 Gramática adaptativa

A gramática adaptativa é um formalismo generativo capaz de representar linguagens sensíveis ao contexto através de um conjunto dinâmica e espontaneamente modificável de regras de substituição. O que distingue este formalismo das gramáticas livres de contexto convencionais é a capacidade de se auto-modificar à medida que uma sentença da linguagem vai sendo derivada.

As regras da gramática adaptativa podem, opcionalmente, estar associadas a ações adaptativas e assim, quando presentes e ativadas, promovem uma sucessão de gramáticas intermediárias. Se uma sentença ω , pertence à linguagem representada pela gramática adaptativa, sua derivação será finalizada em alguma versão G_f da gramática adaptativa G .

Uma gramática adaptativa G pode ser definida como uma tripla ordenada (G_0, T, R_0) , onde:

- T é um conjunto finito, possivelmente vazio, de funções adaptativas
- $G_0 = (VN^0, VT, VC, PL^0, PD^0, S)$ é uma *gramática inicial*, onde:
 - VN^0 é um conjunto finito e não vazio de símbolos não terminais;
 - VT é um conjunto finito e não vazio de símbolos terminais;
 - $VN^0 \cap VT = \emptyset$.
 - VC é um conjunto finito de *símbolos de contexto*.
 - $V^0 = VN^0 \cup VT \cup VC$
 - VN^0, VT e VC são conjuntos disjuntos dois a dois.
 - $S \in VN^0$ é o símbolo inicial da gramática.
 - PL^0 é o conjunto de regras de produção aplicáveis às situações livres de contexto.
 - PD^0 é o conjunto de regras de produção aplicáveis às situações dependentes de contexto.

As regras de produção da Gramática adaptativa consistem de expressões com os seguintes formatos, sendo i um indicador do número de alterações adaptativas já sofridas pela gramática inicial:

Tipo 1: pertencentes ao conjunto PL^i , com $i \in \mathbf{N}$:

$$N \rightarrow \{A\} \alpha$$

onde $\alpha \in (VT \cup VN)^*$, $N \in VN^i$ e A é uma ação adaptativa opcional associada à regra de produção.

Tipo 2: pertencentes ao conjunto PL^i , com $i \in \mathbf{N}$:

$$N \rightarrow \emptyset$$

onde \emptyset é um meta-símbolo que indica o conjunto vazio. Esta regra de transição indica que, embora o símbolo não-terminal N esteja definido, deriva um conjunto vazio, ou seja, não há qualquer substituição prevista para esse não-terminal. Isto significa que, se esta regra for aplicada em alguma derivação, a gramática não gerará qualquer sentença. Esta regra é utilizada para o caso em que na gramática existirem regras que referenciem não-terminais que deverão ser dinamicamente definidos, como resultado da aplicação de alguma ação adaptativa.

Tipo 3: pertencentes ao conjunto PD^i , com $i \in \mathbf{N}$:

$$\alpha N \leftarrow \{A\} \beta M, \text{ onde } \alpha \in VC \cup \{\epsilon\} \text{ e } \beta \in VC, \text{ ou}$$

$$\alpha N \rightarrow \{A\} \beta M \text{ onde } \alpha \in VC, \beta \in VT \cup \{\epsilon\}, N \text{ e } M \in VN^i, \text{ com } A \text{ opcional.}$$

As regras de produção da primeira forma, que apresentam o meta-símbolo " \leftarrow ", indicam que β está sendo injetado na cadeia de entrada. Esta regra substitui αN por βM , inserindo assim na cadeia de entrada a informação de contexto representada por β .

As regras de produção do segundo formato exibem uma seta " \rightarrow " no sentido convencional, mas possuem no seu lado esquerdo um símbolo de contexto seguido por um

símbolo não-terminal. Isto indica que αN é substituído por βM , sendo a informação de contexto β inserida na cadeia de entrada.

$R^0 \subseteq P^0 \times (T \cup \{\epsilon\})$ é uma relação do tipo (r, A) , onde $r \in (PL^0 \cup PD^0)$ e $A \in T$.

Para cada regra de produção r , a relação R^0 associa uma ação adaptativa A correspondente.

Uma regra de produção à qual está associada uma ação adaptativa é denominada *regra de transição adaptativa*.

A expressão que define uma regra de produção do tipo 1 da gramática adaptativa é do tipo livre de contexto a menos da ação adaptativa. Esta, em conjunto com as regras de produção em PD^0 , responde pela representação das dependências de contexto nesta gramática.

A equivalência entre autômatos adaptativos e o das gramáticas adaptativas é demonstrado em (IWAI, 2000) através dos seguintes teoremas:

- *O resultado da execução de qualquer ação adaptativa é equivalente a uma seqüência não-vazia de ações adaptativas elementares*
- *Se uma linguagem é gerada por uma gramática adaptativa, então ela é aceita por algum autômato adaptativo.*
- *Se uma linguagem gerada por um não-terminal corresponde a uma submáquina, então essa linguagem pode ser definida por uma gramática cuja raiz é o não-terminal em questão.*
- *Se uma linguagem é aceita por um autômato adaptativo, então ela pode ser gerada por alguma gramática adaptativa.*

A autora também apresenta um roteiro para o mapeamento dos conjuntos de regras de produção, normais e de contexto, da gramática adaptativa para um conjunto equivalente de regras de transição do autômato adaptativo que reconhece a mesma linguagem, e descreve os seguintes algoritmos:

- conversão canônica da gramática adaptativa para o autômato adaptativo;
- conversão canônica do autômato adaptativo para a forma da gramática adaptativa;
- um algoritmo eficiente para a obtenção de um analisador sintático a partir de gramáticas adaptativas;
- um algoritmo para a construção do autômato.

Foram relevantes ao estudo do formalismo adaptativo as seguintes observações:

- Os símbolos de contexto de uma gramática adaptativa estão associados a um particular símbolo não-terminal da gramática;
- Um único terminal da gramática pode estar associado a um símbolo de contexto e portanto a um particular símbolo não-terminal da gramática;
- Uma sentença inicialmente constituída de símbolos terminais é *modificada* ao longo de sua derivação através da inserção de símbolos de contexto, passíveis de serem utilizados, ao longo da derivação, da mesma forma que os símbolos terminais.

Na próxima seção são apresentados mecanismos adaptativos para o tratamento adequado de linguagens dependentes de contexto. Tais exemplos refletem o uso intenso do que foi exposto até o momento.

2.3 Mecanismos adaptativos para o tratamento de linguagens dependentes de contexto

Este trabalho procurou investigar a potencialidade do formalismo adaptativo para a representação e tratamento de dependências sintáticas em Linguagem Natural. Do estudo dos formalismos mencionados anteriormente, constatou-se serem apropriados para o desenvolvimento de diversos dos elementos necessários ao processamento de dependências de contexto, características das linguagens naturais.

Com o intuito de determinar precisamente uma forma adequada de utilização das técnicas anteriormente desenvolvidas, para o processamento de linguagens naturais, procedeu-se a um estudo aprofundado dos exemplos aduzidos em (NETO, 1993), originalmente projetados contemplando a análise de Linguagens de Programação, e do estudo dos mecanismos empregados nesses exemplos, identificaram-se procedimentos, que foram adotados como modelos para o tratamento de Linguagens Dependentes de Contexto quaisquer, e no particular caso desta tese, de Linguagens Naturais.

No texto que se segue tecem-se considerações a respeito destes procedimentos e sua origem.

2.3.1 Simulador de uma Pilha

Os autômatos adaptativos podem apresentar transições que geram uma cadeia de elementos *de um alfabeto de saída pré-determinado*. Trata-se do conceito de transdutor adaptativo. Tais transições podem ser aquelas já existentes na configuração inicial do autômato ou podem ser aquelas resultantes das auto-modificações dos utômatos ocorridas ao longo do reconhecimento de uma linguagem.

Diferentemente dos formalismos Máquina de Mealy ou Máquina de Moore, os elementos pertencentes *ao alfabeto de saída* e gerados pelo transdutor adaptativo podem ser armazenados na cadeia de entrada.

A gramática adaptativa como formalismo dual ao dos autômatos adaptativos, exhibe em sua definição um conjunto de símbolos de contexto e tipos de regras que permitem a inserção ou extração destes símbolos ao longo da derivação de uma sentença de entrada.

O exemplo que se segue corresponde àquele com o mesmo sub-título apresentado em (NETO, 1993), mapeado para o formalismo gramatical. Acrescentaram-se ainda ao exemplo dois símbolos de contexto, a saber: *beta* e ∇_beta .

Este exemplo, apesar de simples, mostra a derivação de uma sentença pertencente a uma linguagem *livre de contexto*. O seguinte conjunto de regras gera, sobre o alfabeto $V_T = \{ "(", ")", "\beta" \}$, a linguagem $L(w) = \{ w \mid w = ({}^n \beta)^n, n \geq 0 \}$:

$$\begin{aligned} N_0 &\rightarrow beta\ N_1 \\ N_1 &\rightarrow "\beta"\ N_4 \\ N_1 &\rightarrow \{A(N_2, N_3, N_1)\} "("\ N_2 \\ N_2 &\rightarrow \beta\ N_3 \\ N_3 &\rightarrow ")"\ N_4 \\ N_4 &\leftarrow \nabla_beta\ N_5 \\ N_5 &\rightarrow \varepsilon \end{aligned}$$

A declaração da ação adaptativa *A* é apresentada a seguir:

$$\begin{aligned} A(I_N, J_N, N_n) = & \quad \{ K_n^*, M_n^*: \\ & +[K_n \rightarrow "\beta"\ M_n] \\ & +[M_n \rightarrow ")"\ J_N] \\ & +[I_N \rightarrow \{A(K_N, M_N, I_N)\} "("\ K_N] \\ & -[N_N \rightarrow \{A(I_N, J_N, N_N)\} "("\ I_N] \\ & +[N_N \rightarrow "("\ I_N] \end{aligned}$$

Observe-se que o símbolo "(" é um terminal da linguagem que delimita um não-determinismo. As regras de produção são especificadas de tal forma que, quando da ocorrência de *n* símbolos "(", a gramática se auto-modifica permitindo que a derivação prossiga normalmente, tanto em uma subsequente ocorrência do símbolo β , como em uma

subseqüente ocorrência de outro símbolo "("). Isto é efetuado graças à execução da ação adaptativa A , que por sua vez cria dinamicamente uma regra adaptativa para o eventual consumo para o símbolo "(" e uma regra não-adaptativa para o consumo do símbolo β . A essas regras estão associados dois novos não-terminais, contornando-se assim, incremental e dinamicamente, o não-determinismo original, segundo as necessidades decorrentes da ocorrência dos símbolos da sentença em análise, presentes na cadeia de entrada.

Assim, a presença do símbolo de contexto *beta* na cadeia de entrada dá início à derivação da parte da cadeia de entrada que lhe sucede. Ao final da derivação, o analisador sintático deverá substituir o símbolo de contexto *beta* e a sentença $w \in L(w)$ pelo mesmo símbolo de contexto, porém denotado como $\nabla_beta.$, indicando que um trecho da cadeia de entrada foi analisada sintaticamente e classificada como do tipo $\nabla_beta.$

Avaliação de complexidade:

Sendo n o número de ocorrências do símbolo "(" em uma sentença original em análise, a sua derivação, segundo a gramática apresentada, exige a execução de $2*n + 3$ regras gramaticais elementares e implica na execução de $5*n$ operações de criação de regras, ficando assim a gramática original ampliada em $3*n$ regras.

Este exemplo ilustra o desempenho com que o formalismo adaptativo pode reconhecer linguagens livre de contexto, sem aumentar a sua complexidade no tempo e no espaço, quando comparado com o formalismo do autômato de pilha.

Como será discutido adiante, o mesmo princípio identificado neste exemplo norteia a criação de novas instâncias da gramática, uma vez que β pode representar, em vez de um único símbolo, toda uma sub-cadeia.

Tem-se também que *beta* é um símbolo de contexto associado à gramática original. Uma vez presente na cadeia de entrada, o seu consumo poderá promover o empilhamento do

estado do reconhecedor que o ativou. Por outro lado, a substituição na cadeia de entrada por *beta* e a palavra $w \in L(w)$ pelo *token* ∇_beta pode ser realizada com a recuperação do estado do reconhecimento anterior á ocorrência do símbolo *beta*, pela remoção da pilha explícita onde fora interrompido o reconhecimento.

2.3.2 Alteração dinâmica dos atributos de símbolos não-terminais de uma gramática.

Um dos mecanismos adaptativos a serem empregados na descrição e tratamento das linguagens dependentes de contexto é a alteração dinâmica dos atributos associados aos símbolos não-terminais da gramática.

Para executar essa tarefa, ao longo do processo de derivação de uma sentença em que se apresenta um símbolo de contexto representando um determinado atributo, esse símbolo pode figurar isoladamente na cadeia de entrada e ser memorizado, por exemplo, como argumento de uma ação adaptativa, o que permitirá que, posteriormente, essa informação seja recuperada e utilizada.

Considere-se abaixo o trecho de gramática que descreve simplificadaamente a categoria gramatical NP (grupo nominal), cujo núcleo é um substantivo.

Admita-se que o não-terminal $Y1$ se refira a um transdutor que seja capaz de extrair da sentença os símbolos terminais da gramática, e, portanto, de substituí-los por símbolos de contexto, tais como σ_sa se se tratar de um sintagma adjetivo que antecede um substantivo, ou α_subst_pm , se se tratar de um substantivo plural masculino, α_subst_pf se for plural feminino, e assim sucessivamente.

Iguamente, suponha-se que o não-terminal $Y2$ corresponda a um transdutor capaz de extrair da sentença de entrada os símbolos terminais da linguagem e de substituí-los por

símbolos de contexto, tais como σ_{sa} , ou seja, um sintagma adjetivo que sucede a ocorrência de um substantivo.

$$\begin{aligned} NP0 &\rightarrow \{ C(NP_Void) \} Y1 NP1 \\ \sigma_{sa} NP1 &\rightarrow NP0 \\ \alpha_{subst_pm} NP1 &\rightarrow \{ C(NP_PM) \} NP2 \\ \alpha_{subst_pf} NP2 &\rightarrow \{ C(NP_PF) \} NP2 \\ \alpha_{subst_sm} NP1 &\rightarrow \{ C(NP_SM) \} NP2 \\ \alpha_{subst_sf} NP1 &\rightarrow \{ C(NP_SF) \} NP2 \\ \alpha_x NP1 &\leftarrow \{ C(NP_Vazio) \} \alpha_x NP2 \\ NP2 &\rightarrow Y2 NP3 \\ \sigma_{sa} NP3 &\rightarrow NP2 \end{aligned}$$

Uma vez que, ao longo da derivação, são memorizados os atributos do substantivo, no argumento da ação adaptativa C , esta é executada para alterar a regra que define o tipo da classe gramatical NP. Para tanto, a definição de C supõe a utilização de um não-terminal, denominado, por exemplo, $50_TipoAtual$, que indica o tipo da construção sintática que está sendo derivada – no caso, o grupo nominal NP.

$$\begin{aligned} C(Tipo) = \quad \{X: \\ \quad -[50_TipoAtual \rightarrow X] \\ \quad +[50_TipoAtual \rightarrow Tipo]\} \end{aligned}$$

O símbolo de contexto α_x indica qualquer símbolo na cadeia de entrada, que não seja um substantivo, e sua utilização permite interpretar a ausência de um substantivo como a ocorrência, pela omissão associada a essa ausência, de uma construção sintática da *categoria vazia*.

Este exemplo ilustra uma forma como se pode transferir os atributos de um símbolo de uma determinada hierarquia na gramática para outro de hierarquia superior (neste caso, a

transferência de um atributo associado a um símbolo pertencente a uma categoria lexical para a categoria gramatical da qual ele é o núcleo) e também uma possível forma de se detectar situações em que se deva interpretar como sendo a ocorrência sintática legitimamente classificável como sendo da categoria vazia.

Note-se que esta técnica é bastante geral, e independe da hierarquia dos símbolos envolvidos na derivação, de sua natureza, bem como dos atributos a eles associados.

Avaliação de complexidade:

A atribuição do atributo de gênero e número da classe lexical ao grupo nominal, depende apenas de sua detecção na cadeia de entrada, visto que sua determinação a partir do texto da sentença em análise é da competência do analisador léxico, encarregado de determinar a categoria das palavras e de verificar suas flexões, destas extraindo atributos tais como o gênero e o número a elas associados. Assim, o número de regras da gramática não é alterado pela execução da ação adaptativa C.

2.3.3 Concordância de Atributos entre elementos imediatos

A solução aqui aduzida implementa a verificação da concordância entre os atributos de dois símbolos imediatos presentes na sentença em análise, e é inspirada no exemplo 4.4.6 de (NETO, 1993).

Para ilustrar o mecanismo, apresenta-se um conjunto de regras que verifica a concordância entre a categoria gramatical DP do sujeito de uma oração e a categoria gramatical VP da mesma oração, em gênero, número e pessoa.

Pelas convenções anteriormente discutidas, a ocorrência de uma sub-cadeia, classificada como pertencente à categoria gramatical DP, está associada à presença de um símbolo de contexto na cadeia de entrada.

O mesmo se dá com uma sub-cadeia, marcada como sendo da categoria VP. Assim, uma vez que ocorra na sentença de entrada o correspondente símbolo de contexto, os atributos associados podem ser memorizados como argumentos de uma função adaptativa, neste caso a função FCV.

A execução da função FCV promoverá a alteração do conjunto de regras de produção, de forma que sejam passíveis de derivação apenas o sintagma verbal ou o grupo determinante que apresentem atributos concordantes com a categoria primeiramente presente na oração.

Após a execução da ação adaptativa FCV, é executada outra ação adaptativa, que efetua a restauração do conjunto inicial das regras, preparando a gramática para receber uma nova construção sintática similar.

Uma concordância verbal dessa natureza pode ser gramaticalmente especificada, por exemplo, conforme o conjunto de regras abaixo, as quais contemplam uma eventual inversão do sujeito.

Os símbolos de contexto que se apresentarem na cadeia de entrada referem-se ao grupo funcional Determinante ($\sigma_{dp_pessoa_número}$) e ao grupo gramatical Verbal ($\sigma_{vp_pessoa_número}$). Como as regras impõem que haja concordância entre pessoa e o número das construções sintáticas já classificadas como tais, o símbolo de contexto $\sigma_?$, presente em algumas regras, age como uma chave, que permite que a derivação prossiga ou não.

$$CV1 \rightarrow \{ FCV (\sigma_{vp_1s}) \} \sigma_{dp_1s} CV2$$

$$CV1 \rightarrow \{ FCV (\sigma_{vp_2s}) \} \sigma_{dp_2s} CV2$$

$$CV1 \rightarrow \{ FCV (\sigma_{vp_3s}) \} \sigma_{dp_3s} CV2$$

$$\begin{aligned}
CV1 &\rightarrow \{ FCV (\sigma_{vp_1p}) \} \sigma_{dp_1p} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{vp_2p}) \} \sigma_{dp_2p} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{vp_3p}) \} \sigma_{dp_3p} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{dp_1s}) \} \sigma_{vp_1s} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{dp_2s}) \} \sigma_{vp_2s} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{dp_3s}) \} \sigma_{vp_3s} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{dp_1p}) \} \sigma_{vp_1p} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{dp_2p}) \} \sigma_{vp_2p} CV2 \\
CV1 &\rightarrow \{ FCV (\sigma_{dp_3p}) \} \sigma_{vp_3p} CV2 \\
CV2 &\rightarrow \sigma_? CV3 \\
\theta CV3 &\leftarrow \theta CV4 \quad \forall \theta \in \Sigma \\
CV4 &\leftarrow \sigma_CV CV5 \\
CV5 &\rightarrow \varepsilon
\end{aligned}$$

A execução da Função Adaptativa FCV – função Concordância Verbal – habilita a inserção do símbolo σ_CV , indicador de concordância verbal na cadeia de entrada, mediante a alteração das regras $CV2 \rightarrow \sigma_? CV3$ e $\theta CV3 \leftarrow \theta CV4$, conforme a descrição seguinte:

$$\begin{aligned}
FCV(x) = & \quad \{ GCV_aux*: \\
& -[CV2 \rightarrow \sigma_? CV3] \\
& +[CV2 \rightarrow x CV3] \\
& -\{[\theta CV3 \leftarrow \theta CV4] \forall \theta \in \Sigma\} \\
& +\{[\theta CV3 \leftarrow \{Restaura_CV(x)\} GCV_Aux] \forall \theta \in \Sigma\} \\
& +[GCV_Aux \leftarrow CV4] \quad \}
\end{aligned}$$

A função adaptativa *Restaura_CV* promove a restauração da gramática para a sua configuração inicial, no que diz respeito à concordância verbal, mantendo a gramática receptiva a novas instâncias de construções sintáticas da mesma natureza, nos escopos onde isso se faça necessário.

$$\begin{aligned}
 \text{Restaura_CV}(x) = \{ & \\
 & -[\text{CV2} \rightarrow x \text{CV3}] \\
 & +[\text{CV2} \rightarrow \sigma_? \text{CV3}] \\
 & -\{[\theta \text{CV3} \leftarrow \theta \text{CV4}] \forall \theta \in \Sigma\} \\
 & -\{[\theta \text{CV3} \leftarrow \{\text{Restaura_CV}(x)\} \text{GCV_Aux}] \forall \theta \in \Sigma\} \\
 & +\{[\theta \text{CV3} \leftarrow \theta \text{CV4}] \forall \theta \in \Sigma\} \quad \}
 \end{aligned}$$

Observe-se que, após a execução da função FCV, seguida da execução de Restaura_CV, o número de regras inicialmente presentes na gramática fica inalterado.

Por tratar-se de meras manipulações simbólicas, as alterações e restaurações do conjunto de produções presentes na gramática, em cada um desses passos, não serão aqui desenvolvidas por extenso.

2.3.4 Mecanismo Adaptativo para tratamento de escopos aninhados

Por analogia com o procedimento inspirado no exemplo 4.4.7 de (NETO, 1993), intitulado “Analisador Léxico para uma linguagem estruturada em blocos”, pode-se, através de uma série de interpretações e adaptações à realidade das linguagens naturais, inferir um método de análise para a delimitação dinâmica de escopos aninhados.

Também com base na mesma fonte pode-se conceituar, para o analisador sintático, dois modos de operação, a saber: o modo de “treinamento” e o modo de “uso”. Um escopo é iniciado em modo “treinamento”, e a partir de então os elementos da cadeia de entrada vão sendo extraídos e memorizados.

Na ocorrência de um símbolo, presente na cadeia de entrada, mas que não esteja previsto na sub-gramática da linguagem associada àquele escopo, o modo de operação do analisador sintático é alterado para o modo “uso”.

Neste modo, a gramaticalidade de construções sintáticas constituídas dos símbolos recém-memorizados, podem ser verificadas, analisadas e eventualmente um escopo subordinado ao primeiro pode ser identificado e sua análise, iniciada.

O modo “uso” é finalizado quando todas as estruturas subordinadas ao escopo corrente tiverem sido tratadas, e a gramaticalidade da construção, analisada. Nesta situação, o escopo é finalizado. Em caso contrário, detecta-se um erro de sintaxe..

A seguir, detalham-se alguns aspectos desse mecanismo.

2.3.4.1 Delimitação dinâmica de escopos aninhados.

Admita-se a sub-gramática inicial indicada em (1).

- (1) $IO \rightarrow VO$
 $SP \rightarrow V$
 $SP \rightarrow \{Z(SP)\} SP$

A Fig.1 apresenta a configuração inicial de um autômato adaptativo correspondente às regras elementares em (1).

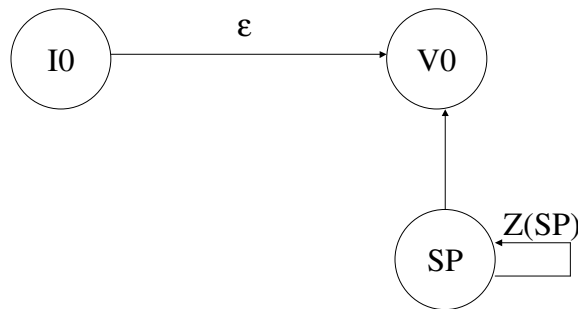


Fig. 1 – Máquina Inicial de uma estrutura auxiliar para armazenar informações em escopos aninhados

IO deve apontar sucessivamente para os estados, I_1, I_2, \dots, I_n , estados iniciais para a memorização e análise da informação E_1, E_2, \dots, E_n .

A i -ésima informação E_i é armazenada entre os estados I_i e N_i . A ação adaptativa Z ., memoriza o topo da pilha assim simulada.

O estado direito da informação N_i deverá apontar dinamicamente para o estado I_{i-1} .

O ponteiro da pilha SP, aponta sempre para o estado final N_i e é apontado por uma lista de estados N_{i-1} .

Tal lista, dessa forma, mantém a informação de quais escopos estão ainda sendo analisados pelo autômato.

Assim, na ocasião em que for detectado o início de um novo escopo sintático na sentença em análise, a seguinte função adaptativa de inserção deverá ser executada:

$$\begin{aligned}
 \text{Inserção (IO, NO, SP)} = & \{i_anterior, n_anterior, m_anterior, I_n^*, N_n^*, M_n^*, \\
 & -[(IO \rightarrow i_anterior)] \\
 & +[(IO \rightarrow I_n)] \\
 & +[(N_n \rightarrow i_anterior)] \\
 & -[(SP \rightarrow n_anterior)] \\
 & +[(SP \rightarrow N_n)] \\
 & -[m_anterior \rightarrow \{Z(n_anterior)\} SP]
 \end{aligned}$$

$$\begin{aligned}
 &+[m_anterior \rightarrow \{Z(n_anterior)\} M_n] \\
 &+[M_n \rightarrow \{Z(n_anterior)\} SP] \quad \}
 \end{aligned}$$

Avaliação de complexidade:

Observe-se que para cada informação E_n a ser armazenada, são criadas cinco novas regras e simultaneamente, são removidas três regras elementares, uma vez que a ação adaptativa $Z(ultimo)$ nada realiza além de memorizar o último não-terminal do escopo ancestral.

Em outras palavras, para tratar q informações E_n , o número de regras na gramática cresce em $2*q$ regras, e ainda incorpora $3*q$ novos não-terminais.

A Fig. 2 apresenta a configuração do autômato para a inserção da informação E_1 :

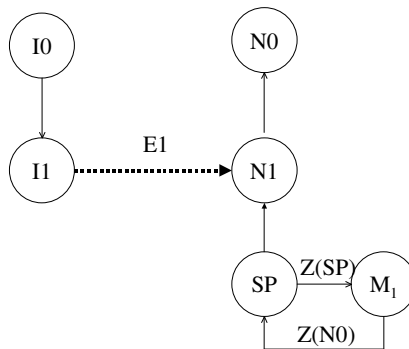


Fig. 2 - Configuração do autômato para inserção da informação E_1

Analogamente, para se remover uma informação E_n , deve-se executar a função adaptativa **Remoção**, descrita a seguir:

$$\begin{aligned}
 \text{Remoção (I0, SP)} = \quad &\{i_atual, n_atual, i_anterior, m_atual, m_anterior: \\
 &-[(I0 \rightarrow i_atual)] \\
 &-[(SP \rightarrow n_atual)] \\
 &-[(n_atual \rightarrow i_anterior)]
 \end{aligned}$$

$+[(IO \rightarrow i_anterior)]$
 $-[(m_atual \rightarrow \{ Z(n_anterior) SP \})]$
 $+[(SP \rightarrow n_anterior)]$
 $-[(m_anterior \rightarrow \{ Z(n_anterior) \} m_atual)]$
 $+[(m_anterior \rightarrow Z(n_anterior) SP)]$

Avaliação de complexidade:

Para cada informação que seja desconsiderada pelo analisador em algum instante da derivação, duas regras da gramática são removidas. O tempo gasto para isso é portanto constante, e igual ao tempo de execução de oito ações elementares.

2.3.4.2 Extração e Memorização de uma seqüência de símbolos

Neste item, comentam-se os mecanismos adaptativos que podem ser empregados na construção de um analisador léxico para linguagens dependentes de contexto, em particular, para linguagens naturais.

A partir de uma sentença de entrada, originalmente constituída de terminais da gramática, os mesmos são extraídos, memorizados, classificados e simultaneamente, uma gramática inicial com um conjunto de regras P^0 é alterada através da execução de particulares e adequadas ações adaptativas.

O conjunto de regras assim obtido deverá ser usado para a posterior e efetiva análise da gramaticalidade da sentença.

De imediato percebe-se a necessidade de que a gramática dinamicamente modificada apresente duas regras mutuamente exclusivas, as quais permitem que ora a mesma seja empregada em modo “treinamento”, com a extração e memorização de símbolos, ora no modo “uso”. Trata-se das seguintes regras:

97_Modo \rightarrow 98_Treino

97_Modo → 99_Uso

Um conjunto inicial de regras de produção que se propõe é o seguinte:

$$P^0 = \{ \begin{array}{l} 1^0. I \rightarrow \{A_0(I, J, K, \theta, N)\} \theta J \quad \forall \theta \in V_T \\ 2^0. J \rightarrow \{A_1(I, J, K, \theta, N)\} K \\ 3^0. N \rightarrow \emptyset \end{array} \}$$

V_T é conjunto de todas as entradas lexicais pertencentes a uma determinada linguagem;

A_0 e A_1 (descritas mais adiante) são as ações adaptativas que efetivamente possibilitam a memorização das entradas lexicais e a alteração do conjunto de regras, respectivamente.

A ação adaptativa A_0 memoriza em um de seus argumentos o símbolo presente na sentença em processo de derivação, e modifica os argumentos da ação adaptativa A_1 presente na regra seguinte. Cabe aqui salientarmos a notação “ $\forall \theta \in V_T$ ”, presente na regra 1^0 , pela definição original (NETO, 1993) deve representar um conjunto de regras elementares, uma para cada elemento $\theta \in V_T$.

A ação adaptativa A_1 é executada apenas no modo “treinamento”. Esta ação adaptativa permite que sejam criadas regras de produção específicas a partir de cada terminal presente na sentença original em análise.

Uma vez que se encontre no texto de entrada um símbolo que não pertença ao alfabeto da Linguagem, o autômato se auto-modifica de forma que possa ser empregado no modo “uso”.: a transição associada à ação adaptativa A_1 é substituída por uma transição associada à ação adaptativa A_{39} . O autômato assim modificado é disponibilizado no modo “uso”. Neste modo de operação, a execução da ação adaptativa A_0 alterará os argumentos da ação adaptativa A_{39} , de forma análoga ao que ocorre no modo “treinamento”.

Seendo ϕ um símbolo que não pertence ao conjunto de terminais da linguagem, para um fragmento de sentença de entrada, tal como: *a bela praia* ϕ , as seguintes regras de produção podem ser criadas, como resultado da execução da ação adaptativa A_1 :

$$\begin{aligned} I &\rightarrow "a" \text{ GJ1} \\ \text{GJ1} &\rightarrow "bela" \text{ GJ2} \\ \text{GJ2} &\rightarrow "praia" \text{ GJ3} \end{aligned}$$

Naturalmente, para cada novo símbolo terminal extraído, um novo não-terminal também vai sendo criado. A execução da ação adaptativa A_1 gera um conjunto de regras de produção adaptativas tais que:

- seja possível que as palavras recém-identificadas na cadeia de entrada possam ser classificadas e, por conseguinte, inseridos na cadeia de entrada os símbolos de contexto associados;
- seja memorizado o não-terminal recém-criado, em correspondência ao terminal lido da sentença de entrada;
- seja viável a transição do modo “treinamento” para o modo “uso”;
- um determinado símbolo seja analisado em diversos segmentos da gramática;

Seguem-se as descrições das funções adaptativas que implementam o mecanismo adaptativo proposto:

Função Adaptativa A_0

$A_0(I, J, K, \sigma, N) = \{M, N, P, Q, R, S, T, U, V, W, X, Y:$

/* insere argumentos na chamada de A_1 , se existir */

- $[J \rightarrow \{A_1(M, N, P, Q, R)\} Y]$

$$+[J \rightarrow \{A_1(l, J, K, \sigma, N)\} Y]$$

/* insere argumentos na chamada de A_{39} , se existir */

$$- [X \rightarrow \{A_{39}(S, T, U, V, W)\} K]$$

$$+[X \rightarrow \{A_{39}(l, J, K, \sigma, N)\} K \quad \}$$

Avaliação de complexidade:

A execução da ação adaptativa A_0 não altera o número de regras presentes na gramática, e seu tempo de execução é constante, e corresponde ao de quatro ações adaptativas elementares.

Função Adaptativa A_1

$$A_1(l, J, K, \sigma, N) = \quad \{L, M, P, Q, R, GJ^*, GN^*, GS^*, T^*:$$

/* substitui a regra associada à ação adaptativa A_0 , se existir*/

$$-[l \rightarrow \{A_0(L, M, P, Q, R)\} \sigma J]$$

/* cria uma regra particular para o símbolo terminal recém-encontrado */

$$+[l \rightarrow \sigma GJ]$$

/*cria regras para que novos terminais sejam identificados na sentença de entrada*/

$$\{+[GJ \rightarrow \{A_0(GJ, J, K, \theta, GN)\} \theta J] \forall \theta \in \Sigma_T\}$$

/* A derivação deverá portanto prosseguir a partir da regra recém-criada. No entanto, o processo de derivação se encontra no nó K . Assim sendo, o processo de derivação deverá ser conduzido do nó K para o nó GJ . Esta regra se torna desnecessária tão logo for aplicada, por isso a ela é associada a uma ação de auto-remoção: */

$$+[K \rightarrow \{AutoRemove_1(K, GJ)\} GJ]$$

/* cria regra que permitirá a posterior inserção na cadeia de entrada do símbolo terminal recém-encontrado e memorizado */

$$+[GN \leftarrow \sigma N]$$

/* cria regras que permitem a derivação de símbolos $\theta \notin V_T$ */

$$\{+[\theta GJ \leftarrow \theta GS] \forall \theta \notin V_T\}$$

/* As seguintes ações elementares completam o tratamento, pela ação adaptativa A_1 , dos símbolos não pertencentes ao alfabeto da linguagem, fazendo uso das ações adaptativas A_{42} e A_{43} , adiante descritas */

+ $[GS \rightarrow \{A_{42}(GS, GT, GN)\} G_Aux]$
 + $[G_Aux \rightarrow GT]$
 + $[GT \rightarrow \{A_{43}(GS, GT, GN, GJ)\} VOID_8] \quad \}$

Avaliação de complexidade:

Para n símbolos terminais presentes na sentença, a execução da ação adaptativa A_1 acrescenta $n \cdot (5 + l + m)$ regras na gramática, onde l é o número de símbolos do alfabeto da linguagem em reconhecimento e m é o número de símbolos não pertencentes ao alfabeto, porém passíveis de ocorrerem na cadeia de entrada.

A seguir, comentam-se as funções adaptativas A_{42} , A_{43} e A_{39} , referenciadas nas produções das ações adaptativas A_1 e A_0 .

Função Adaptativa A_{42}

A função adaptativa A_{42} deve ser ativada apenas quando a gramática está sendo usada para a análise de símbolos presentes na sentença de entrada. Quando executada, permite que uma ou mais palavras sejam buscadas na árvore de derivação, para análise em outros contextos.

Segue-se a descrição da função adaptativa A_{42} .

$A_{42}(GS, GT, GN) = \{y:$

/* verifica se a gramática está sendo empregada em modo uso */

? $[(97_Modo \rightarrow y \ 99_Uso]$

/* Se a gramática estiver em modo uso, elimina as seguintes regras da gramática: */

-[GS \rightarrow {A₄₂ (GS, GT, GN)} y G_Aux]

-[G_Aux \rightarrow yGT]

-[GT \rightarrow { A₄₃(GS, GT, GN, GJ) } y VOID_8]

/* Criação de regra, condicionada ao modo de classificação */

+[GS \rightarrow y GN] }

Avaliação de complexidade:

Para cada símbolo da sentença, a execução da ação adaptativa A₄₂ elimina três regras, substituindo-as por uma única regra, portanto sua execução promove uma redução de duas regras no conjunto de produções da gramática. Seu tempo de processamento de um símbolo é constante, e corresponde à execução de cinco ações adaptativas elementares.

Função Adaptativa A₄₃

A função adaptativa A₄₃ deve ser ativada apenas em tempo de extração e memorização de símbolos. Sua execução permite que símbolos terminais na cadeia de entrada, sejam substituídos por símbolos de contexto, desde que haja uma consulta à regra 50_TipoAtual \rightarrow Y, onde Y é o símbolo não-terminal associado a um determinado tipo do terminal recém-identificado na cadeia de entrada. Assume-se aqui que a palavra atributo, abrange também a classificação do terminal .

A₄₃ (GS, GT, GN, GJ) = {X, Y, G_AUX, Y_AUX:

/* verifica se se trata de modo de extração de símbolos*/

```

?[97_Modo → X 98_Treino]

/* verifica qual o não-terminal que deve ser associado à cadeia de entrada */

?[50_TipoAtual → Y]

?[Y_AUX → Y]

/* remove a regra que ativa A43 */

-[GT → {A43(GS, GT, GN, GJ)} VOID_8]

/* remove as regras associadas à ação adaptativa A42 */

-[GS → {A42(GS, GT, GN)} G_Aux]

-[G_Aux → GT]

/* insere regras para classificar cadeia de entrada como sendo do tipo Y */

+[GS → {A_SbAt(GN)} Y_AUX]

+[Y_AUX → Y]

/* marca o não-terminal criado */

A_SbAt(GN) }

```

Avaliação de complexidade:

Para cada símbolo presente na sentença de entrada, três regras da gramática são removidas e outras duas são acrescentadas, resultando em uma redução de uma regra no conjunto de produções da gramática. A execução da ação adaptativa A_SbAt não altera o número de regras na gramática. O tempo de execução desta função adaptativa é constante, e corresponde ao tempo de execução de oito ações adaptativas elementares, acrescido do tempo de execução de A_SbAt , ou seja, do tempo de execução de duas ações adaptativas elementares.

Função Adaptativa A_{39}

A única regra presente em A_{39} implica que o primeiro símbolo que foi detectado na cadeia de entrada, e que ativou o modo de “uso” das regras, é inserido na cadeia de entrada, no escopo cujo início é demarcado pelo não-terminal K. Uma vez empregada esta regra, é imediatamente removida.

$$A_{39}(I, J, K, \sigma, N) = \{+[K \leftarrow \{\text{AutoRemove}_{39}(K, N)\} \sigma N]\}$$

$$\text{AutoRemove}_{39}(X, Y) = \{-[X \leftarrow \{\text{AutoRemove}_{39}(X, Y)\} \sigma Y]\}$$

Avaliação de complexidade:

A execução da ação adaptativa A_{39} , por estar associada a uma ação adaptativa de auto-remoção, não altera globalmente o número de regras da gramática. Naturalmente, sua execução depende apenas do número de alterações do modo de memorização para o modo de uso das regras.

Função Adaptativa A_{SbAt}

Esta função adaptativa recebe como parâmetro o não-terminal N associado à regra criada a partir do terminal θ :

$$A_{SbAt}(N_\theta) = \{X: \\ -[N_{90} \rightarrow X] \\ +[N_{90} \rightarrow N_\theta] \}$$

Para ilustrar a execução dos mecanismos descritos, A Fig. 3 apresenta um autômato criado para a memorização de uma seqüência de três símbolos na cadeia de entrada.

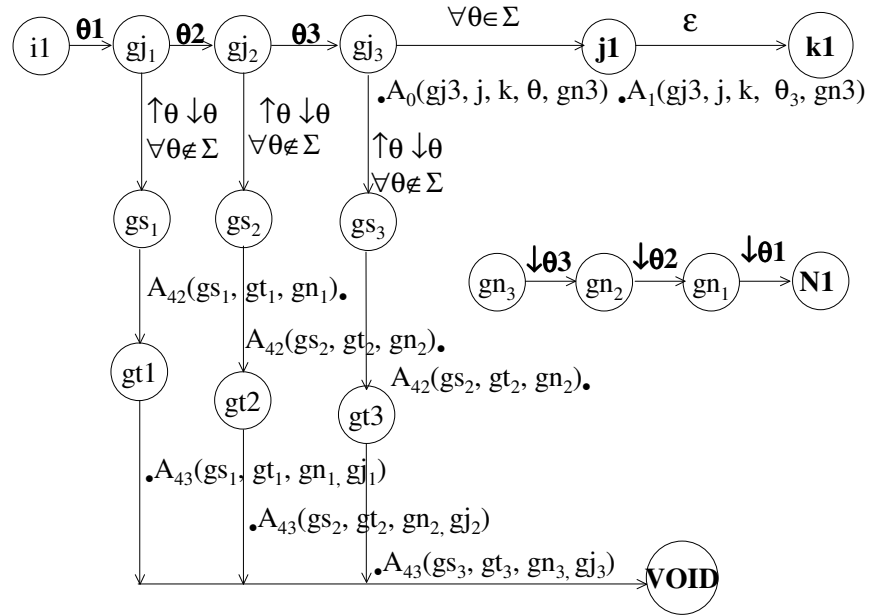


Fig. 3 - Memorização da seqüência $\theta_1\theta_2\theta_3$

Note-se que, as listas parciais são incrementalmente construídas, de tal forma que não-determinismos, eventualmente presentes em uma cadeia de símbolos, possam ser adequadamente tratados.

Por outro lado, considere-se que, no modo “treinamento” após a ocorrência dos símbolos $\theta_1\theta_2\theta_3$ seja encontrado na cadeia de entrada um símbolo não pertencente ao alfabeto, indicando o final da cadeia de entrada que está sendo armazenada. Tal símbolo é devolvido para a cadeia de entrada e a ação adaptativa A_{43} é executada. O autômato resultante é apresentado na Fig. 4.

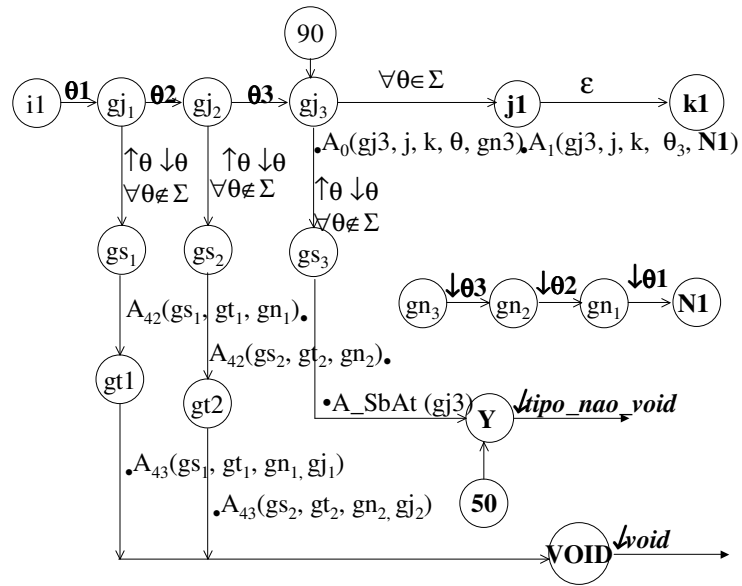


Fig. 4 – Resultado da Execução da ação adaptativa A_{43} . (modo treinamento).

No modo “uso”, a ocorrência da cadeia $\theta_1\theta_2\theta_3$, promoverá a inserção do tipo que lhe representa, no caso `tipo_não_void`.

A ação adaptativa A_{43} ativa a execução da função adaptativa A_{SbAt} ., a qual atualiza o ponteiro 90 para o estado final do último átomo (seqüência de símbolos) extraído (na figura acima, trata-se de gj_3).

Tal técnica não depende da natureza dos símbolos: é válida tanto para símbolos terminais como para símbolos de contexto, ou seja, símbolos correspondentes a não-terminais que são inseridos, extraídos e memorizados pelo transdutor adaptativo.

2.3.4.3 Comutação entre os modos de operação “treinamento” e “uso”

O formalismo adaptativo proposto permite que dinamicamente o modo de operação do analisador sintático seja controladamente alterado do modo “treinamento” para o modo “uso”.

Empregam-se três meta-símbolos associados respectivamente a três ações adaptativas, para o tratamento de escopos aninhados, conforme a tabela seguinte:

Meta-símbolo	Função Adaptativa Associada	Tarefa associada
{	H	Tratamento da detecção do início de um novo escopo.
:	I	Alteração do modo de extração de símbolos (ou treinamento) para o modo uso. Tal meta-símbolo pode estar associado à detecção na cadeia de entrada, de um símbolo que não pertença ao conjunto Σ de símbolos admitidos nesse escopo.
}	J	Tratamento da detecção de término do escopo presente.

Tabela 1 – Meta-símbolos associados a ações adaptativas para delimitação de escopo e alteração do modo de operação

Função adaptativa H:

A função adaptativa H executa as seguintes tarefas:

- inicia um escopo;
- inicia as condições para a ativação do modo “uso”, acertando os parâmetros das ações adaptativas associadas àquele escopo;
- ativa a regra 97_Modo → 98_Treino e desativa a regra 97_Modo → 99_Uso
- marca tipo dos símbolos como Não_Void.

Apresenta-se a seguir a descrição da função adaptativa H:

$$H () = \{w, v, x, y, temp, n_anterior, R, S, I_n^*, N_n^*, E_n^*, J_N^*, K_N^*:$$

$$\begin{aligned} & -[(97_Modo \rightarrow w] \\ & +[(97_Modo \rightarrow 98_Uso] \\ & -[50_TipoAtual \rightarrow v] \\ & +[50_TipoAtual \rightarrow Não_Void] \end{aligned}$$

/* Inicializa um novo escopo - ação adaptativa Inserção () com $l_0 = 3^*/$

$$\begin{aligned} & -[(3 \rightarrow x)] \\ & +[(3 \rightarrow I_n)] \\ & +[(N_n \rightarrow x)] \end{aligned}$$

$$\begin{aligned}
 & -[(SP_{96} \rightarrow y)] \\
 & +[(SP_{96} \rightarrow N_n)] \\
 & -[(temp \rightarrow \{ Z(n_{anterior}) \} SP_{96})] \\
 & +[(temp \rightarrow \{ Z(n_{anterior}) \} E_n)] \\
 & +[(E_n \rightarrow \{ Z(y) SP_{96} \} Z(y)] \\
 & /* altera os parâmetros da ação adaptativa l, associada ao meta-símbolo : */ \\
 & -[B3 \rightarrow \{ l(R, S) \} ":" B4] \\
 & +[B3 \rightarrow \{ l(J_N, K_N) \} ":" B4] \\
 & /* a ação adaptativa A_0, responsável por extrair os símbolos da cadeia de entrada, é associada aos não-terminais l_N e J_N. */ \\
 & \{ +[(l_N \rightarrow \{ A_0(l_N, J_N, K_N, \theta, N_n) \} \theta GJ] \forall \theta \in \Sigma] \\
 & /* a ação adaptativa A_1 responsável por efetivamente memorizar os símbolos da cadeia de entrada, é associada aos não-terminais J_N e K_N. Em outras palavras, o modo “uso” refere-se exatamente aos símbolos anteriormente memorizados */ \\
 & +[(J_N \rightarrow \{ A_1(l_N, J_N, K_N, \varepsilon, N_N) \} K_N] \quad \}
 \end{aligned}$$

A Fig. 5 ilustra o resultado da execução sucessiva de 3 aberturas de escopos. Antes da abertura de qualquer escopo, na gramática inicial, está presente apenas uma a produção :

$$N3 \rightarrow N0$$

Após a abertura de cada escopo, a regra acima é substituída pela regra $N3 \rightarrow l$, onde l é a raiz do escopo mais recente.

Avaliação de complexidade:

Observe-se que a execução da ação adaptativa H implica na remoção de 6 regras e na criação de $9+l$ regras, onde l é o número de símbolos terminais da gramática. Dessa forma, a gramática resultante se apresenta com um número acrescido de $3+l$ produções, 5 novos não-

terminais, para cada abertura de escopo. Ainda, sua execução consome o tempo da ativação de $(15 + 1)$ ações elementares, para cada abertura de escopo.

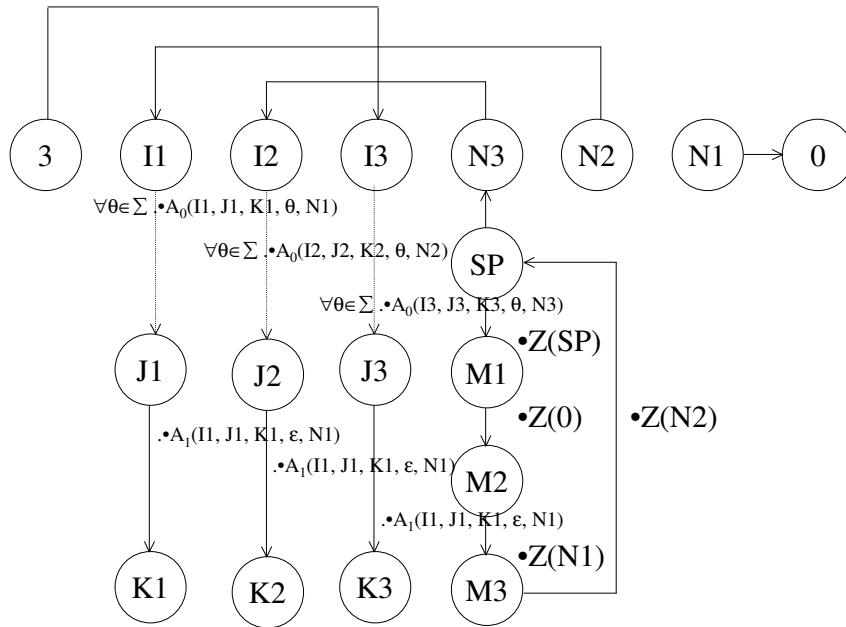


Fig. 5 – Resultado da ativação da função adaptativa $H()$ para três escopos aninhados

Função adaptativa l:

A função adaptativa $l(J_N, K_n)$ executa as seguintes tarefas:

- Ativa a regra $97_Modo \rightarrow 98_Uso$ e desativa a regra $97_Modo \rightarrow 99_Treino$
- Marca tipo dos símbolos como Void_8.
- Desativa a execução da ação adaptativa $A_1(I_N, J_N, K_N, \epsilon, N_N)$ responsável pela memorização dos símbolos da cadeia de entrada e ativa a execução da ação adaptativa $A_{39}(I_N, J_N, K_N, \epsilon, N_N)$, que permite a restauração do símbolo na cadeia de entrada, para que seja buscado nos blocos ancestrais K_N

Segue-se a descrição da função adaptativa l:

$$l(J_N, K_N) = \{I_N, \sigma, N_N, x, y :$$

/* Ativa a regra $97_Modo \rightarrow 98_Uso$ e desativa a regra $97_Modo \rightarrow 99_Treino$ e marca o tipo dos símbolos como Void. */

-[(97_Modo \rightarrow y]
 +[(97_Modo \rightarrow 99_Uso]
 -[50_TipoAtual \rightarrow x]
 +[50_TipoAtual \rightarrow Void]

/* Desativa a execução da função adaptativa $A_1(I_N, J_N, K_N, \varepsilon, N_N)$ responsável pela memorização dos símbolos da cadeia de entrada, e ativa a execução da ação adaptativa $A_{39}(I_N, J_N, K_N, \varepsilon, N_N)$, a qual permite a restauração do símbolo na cadeia de entrada, para que seja buscado nos escopos ancestrais K_N^* */

-[$J_N \rightarrow \{ A_1(I_N, J_N, K_N, \sigma, N_n) \} K_N$]
 +[$J_N \rightarrow \{ A_{39}(I_N, J_N, K_N, \sigma, N_n) \} K_N$] }

Avaliação de complexidade:

Observe-se que a execução da ação adaptativa I não altera o número de regras da gramática e nem o número de não-terminais. Implica, para cada alteração do modo de operação, na execução de seis ações elementares.

A Fig. 6 mostra a substituição da ação adaptativa A_1 pela ação adaptativa A_{39} , na comutação do modo de operação “treinamento” pelo modo “uso”, em um escopo onde foram anteriormente armazenados três símbolos terminais. A Fig. 7 ilustra o resultado da execução da função adaptativa A_{39} , que é ativada quando uma seqüência de símbolos não é encontrada em um determinado escopo. Cria uma transição efêmera, a partir da qual, a cadeia de símbolos é restaurada na cadeia de entrada para ser buscada em outros escopos

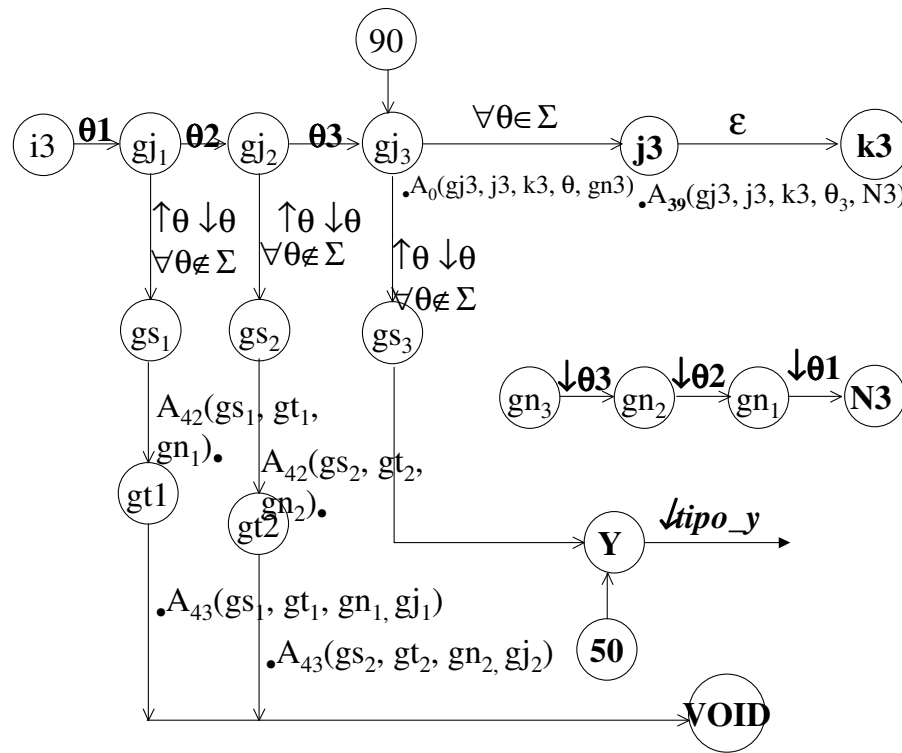


Fig. 6 – Substituição de transição associada à ação adaptativa A_1 por transição associada à ação adaptativa A_{39}

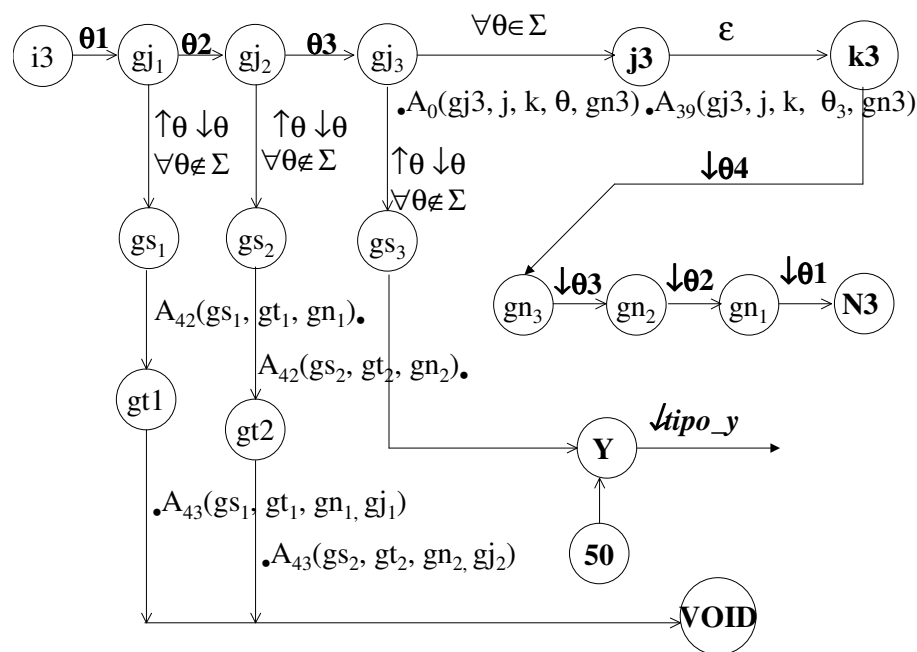


Fig. 7 – Modo “Uso”: Execução da ação adaptativa A_{39} ; no reconhecimento da cadeia $\theta_1\theta_2\theta_3\theta_4$ -

Na ilustração acima é possível constatar que, após a execução de A_{39} , são efetuadas as transições de inserção de símbolos recém-consumidos na cadeia de entrada para serem buscados no escopo ancestral, cujo estado inicial, neste exemplo é I_2 . Confirma-se transição em vazio entre os estados N_3 e I_2 na Fig. 5. Trata-se de uma situação onde a seqüência de símbolos apresenta prefixo comum àquela memorizada no escopo, porém é de comprimento maior.

Avaliação de Complexidade:

A *busca* mal sucedida de um átomo com p símbolos terminais apresenta um custo de p operações de leitura da cadeia de entrada e p operações de escrita do átomo na cadeia de entrada, acrescentado do custo de 2 operações elementares de execução das ações adaptativas A_{39} e $AutoRemove_{39}$

Se a informação assim constituída de p símbolos estiver no q -ésimo escopo mais externo, com relação ao escopo onde se detecta a necessidade de sua busca, esta é efetuada portanto em um tempo proporcional a $q*(2p+3)$ unidades de tempo.

Por outro lado, se a cadeia de símbolos, se apresentar com prefixo comum de comprimento menor, a função adaptativa A_{42} é efetuada, que restaura a seqüência na cadeia de entrada. Observe-se a Fig. 8.

Avaliação de Complexidade:

A *busca* mal sucedida de um átomo com p símbolos terminais apresenta um custo de p operações de leitura da cadeia de entrada e p operações de escrita do átomo na cadeia de entrada, acrescentado do custo de 4 operações elementares de execução da execução da ação adaptativa A_4 . A gramática é reduzida de uma regra em cada escopo percorrido.

Se a informação assim constiuída de p símbolos estiver no q-ésimo escopo mais externo, com relação ao escopo onde se detecta a necessidade de sua busca, esta é efetuada portanto em um tempo proporcional a $q \cdot (2p+4)$ unidades de tempo.

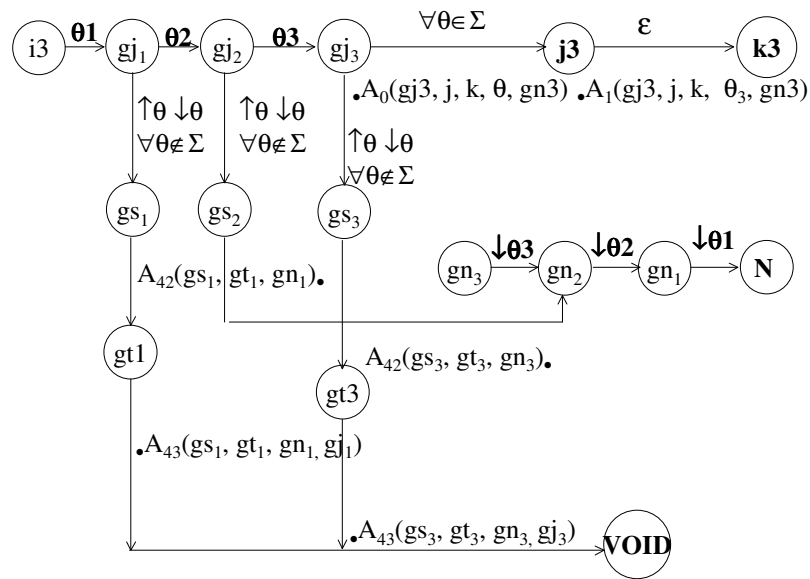


Fig. 8 – Modo “Uso”: Ativação da função adaptativa A_{42} no reconhecimento da cadeia $\theta_1\theta_2$

Função adaptativa J:

A função adaptativa J, corresponde à função adaptativa Remoção, anteriormente apresentada trata a finalização de um escopo, desconectando-o do processamento da derivação, uma vez que a análise do mesmo foi efetuada com sucesso. Renomeou-se o não-terminal I_0 como 3.

$$\begin{aligned}
 J() &= \{x, y, z, n_anterior, temp, m: \\
 &\quad -[(3 \rightarrow x)] \\
 &\quad -[(SP_96 \rightarrow y)] \\
 &\quad +[(y \rightarrow z)] \\
 &\quad +[(3 \rightarrow z)]
 \end{aligned}$$

$$\begin{aligned}
& -[(temp \rightarrow \{Z(n_anterior)\} SP_96)] \\
& +[(SP_96 \rightarrow n_anterior)] \\
& -[(temp \rightarrow \{Z(u)\} m)] \\
& +[(temp \rightarrow \{Z(u)\} SP_96] \quad \}
\end{aligned}$$

Avaliação de complexidade:

Observe-se que a execução da ação adaptativa J não altera o número de regras de produção da gramática adaptativa. No fechamento de um escopo são executadas oito ações elementares e portanto esta função adaptativa tem um tempo de processamento constante.

2.4 Algumas considerações

Neste item são comentados alguns desdobramentos do uso das técnicas expostas anteriormente.

Concordância entre atributos de elementos imediatos

Os mecanismos adaptativos analisados neste capítulo se mostram passíveis de serem empregados na execução de tarefas de tratamento de concordâncias de atributos de símbolos que se apresentam inicialmente justapostos na fita de entrada.

Delimitação de Escopos Aninhados:

O desempenho dos mecanismos de delimitação de escopos aninhados no tempo de execução, bem como o número de não-terminais e produções (ou estados e transições) acrescentadas e até mesmo removidas pela ativação de tais mecanismos, são proporcionais ao produto dos seguintes fatores: comprimento da fita de entrada, número de ocorrências de símbolos na cadeia e em alguns casos, ao número de terminais do alfabeto da Linguagem.

Modo de operação treino e modo de operação uso – aprendizagem e automatização

Uma das características do formalismo adaptativo é o fato de ser ele próprio a realização de uma forma de aprendizado. Assim, as técnicas adaptivas podem fazer uso de dois modos de operação: o modo “treinamento”, onde ocorre o aprendizado ou armazenamento de um determinado conhecimento representado em uma fita de entrada (um arquivo texto não formatado), e o modo “uso”, onde é possível buscar uma determinada informação armazenada (ou não) no modo “treinamento”.

De fato, ainda que o alfabeto da linguagem empregada nos mecanismos apresentados anteriormente ser ASCII, estes mecanismos não dependem da natureza dos símbolos presentes na fita de entrada.

Ao longo do modo treinamento, o conhecimento a ser memorizado é disponível em uma fita de entrada (um arquivo tipo texto não formatado) e delimitado por meta-símbolos “{“ e “}” que *representam* o início e o fim de escopos.

As ações adaptativas apresentadas permitem a designação *automática* dos novos não-terminais (ou estados) para cada escopo representado inicialmente na fita de entrada.

Além dos ponteiros para os modos de operação do dispositivo adaptativo, são empregados dois outros ponteiros no modo treinamento. O ponteiro N90 aponta sempre para o último símbolo memorizado em um escopo. Em tempo de treinamento, este símbolo coincide com o último símbolo lido da cadeia de entrada. O ponteiro 50_TipoAtual permite que ao longo do aprendizado seja processada a associação entre um escopo e o atributo deste mesmo escopo.

De fato, a função adaptativa A₄₃, ativada em tempo de treino, prepara o dispositivo para ser empregado no modo uso. É esta ação adaptativa que captura simultaneamente o estado final (ou último terminal) do escopo e o seu atributo.

À medida que novas informações são inseridas no dispositivo, os ponteiros, N3 para o início do escopo mais recente e o ponteiro SP_96 para a extremidade direita do escopo são *automaticamente* atualizados. O mesmo se dá, ao final do tratamento de um escopo.

A alteração do modo treinamento para o modo uso pode ser efetuado automaticamente mediante a presença do meta-símbolo “:” na informação inicialmente representada na fita de entrada e armazenada (como qualquer outro símbolo).

Representação de informação

Note-se, que se pode representar uma informação e memorizá-la de forma parentetizada e etiquetada.

Considere-se o seguinte exemplo:

$$\{S \{DP1 \{det \ o\} \{N \ mecânico\}\} \{VP \{V \ limpou\} \{DP2 \{det \ o\} \{N \ automóvel\}\}\} \{PP \{PCOM\} \{NP3 \{det \ um\} \{N \ pano\}\}\}\}$$

De fato, como se viu é possível representar no modo treinamento, as seguintes regras:

$$det = \{o\}, \text{ equivalente a: } \{det \ o\}$$

$$dp = \{det \ n\}, \text{ equivalente a } \{dp \ \{det\} \ \{n\}\}$$

Constatou-se que no modo “uso”, a cadeia de símbolos a ser pesquisada é especificada na fita de entrada e sua busca se inicializa em um escopo mais interno e se orienta para o escopo mais externo.

A ação adaptativa A₄₃ preparou o dispositivo adaptativo de forma que se uma informação é encontrada, o ponteiro N₉₀, aponta para o último símbolo extraído da fita de entrada, enquanto que o símbolo de contexto associado ao seu tipo é inserido na fita, naquele escopo específico.

Considere-se a seqüência, presente na fita de entrada:

No modo uso, para o trecho da fita de entrada:

o mecânico

é possível se buscar e encontrar a informação:

$$\{DP1 \{_{det} O\} \{N \text{ mecânico}\}\}$$

equivalente a :

$$\{\{O\}_{det}\{mecânico\}_N\}_{DP1}$$

Tem-se que det e n e DP1 são símbolos de contexto inseridos na fita de entrada, ao longo da análise.

Coordenação e Subordinação

Por outro lado, se uma informação não é encontrada, as transições entre os diversos escopos são realizadas automaticamente pelas ações adaptativas A₃₉. ou A₄₂ O desempenho da busca é proporcional ao produto do comprimento da cadeia a ser pesquisada pela diferença de profundidade existente entre o escopo onde a busca é especificada e onde é (ou não) encontrada.

Em uma árvore, a interligação de cada nó-pai com um conjunto correspondente de nós-filhos imita a maneira como a correspondente regra de produção de uma gramática gramática manda substituir um nó não-terminal (associado ao lado esquerdo da produção) por uma seqüência formada de terminais e não-terminais (relativa ao seu lado direito). Os nós-irmãos referem-se a elementos justapostos entre si e são subordinados ao nó-pai.

Em outras palavras, é possível estabelecer o relacionamento estrutural (coordenação e subordinação) entre elementos no modo treinamento, através da especificação da mesma e empregando-se os dos símbolos “{“ e “}”

Considere-se:

dp, ip, ib e coord, símbolos de contexto, que representam os grupos determinante, flexão, primeira projeção do grupo flexão e uma entrada lexical coordenativa, respectivamente. No modo treino, é possível se especificar facilmente:

$$ip = \{ dp \text{ ib } \}$$

Duas orações coordenadas que se apresentem com grupos determinantes coordenados poderiam ser representados como:

$$ip = \{ \{ \{ dp \} \text{ coord } \{ dp \} \} \{ ib \} \} \text{ coord } \{ \{ \{ dp \} \text{ coord } \{ dp \} \} \{ ib \} \}$$

Analogamente, representa-se a subordinação.

2.5 Conclusão

Este capítulo mostrou como as técnicas adaptativas utilizadas em (NETO, 1993) podem representar e tratar diversos problemas encontrados no processamento de Linguagem Natural.

Tais constatações associadas às que serão apresentadas no próximo capítulo resultaram na proposição de uma arquitetura adaptativa para processamento de linguagem natural, fundamentada no uso dessas técnicas.