Linguagens livres de contexto e autômatos de pilha

José Lucas Rangel, maio 1999

6.1 - Introdução.

Os aceitadores, ou reconhecedores, das linguagens livres de contexto são os chamados *autômatos de pilha* ou *ap*'s. Usaremos aqui o modelo mais geral de ap, o *ap não determinístico*, ou *apnd*, que consiste basicamente de um autômato finito não determinístico, com uma memória adicional, em forma de pilha. Numa pilha, símbolos novos só podem ser acrescentados no topo da pilha; apenas o último símbolo armazenado, o símbolo que se encontra no topo da pilha pode ser consultado; esse símbolo deve ser retirado para que os demais possam ser alcançados.

Neste capítulo vamos provar que a classe de linguagens reconhecidas pelos apnd's é exatamente a classe das llc.

6.2 - Autômatos de pilha não determinísticos

Definição. Definimos um autômato de pilha não determinístico (apnd) como uma construção (tupla) $A = \langle K, \Sigma, \Gamma, \delta, i, I, F \rangle$, formada pelos seguintes elementos:

K - conjunto (finito) de estados

Σ - alfabeto de entrada

Γ - alfabeto da pilha

δ - função de transição

$\delta: K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow P(K \times \Gamma^*)$

i - estado inicial

$i \in K$

I - símbolo inicial da pilha

$I \in \Gamma$

F - conjunto de estados finais

$F \subseteq K$

O alfabeto Γ contém os símbolos que podem ser armazenados na pilha, ou seja, *empilhados*. Para simplificar a especificação da função de transição δ, consideramos que a cada passo, um símbolo é lido (e retirado) do topo da pilha, e, no mesmo passo, uma seqüência de comprimento qualquer pode ser empilhada. Assim, se queremos retirar um símbolo do topo da pilha, basta que a sequência a ser empilhada seja a seqüência vazia ε. Por outro lado, se quisermos manter o símbolo do topo da pilha, ele deve ser re-empilhado.

Escolher de forma não determinística, uma opção $(p, \alpha) \in \delta(q, a, Z)$ quer dizer que a partir do estado q, lendo a da entrada, e lendo Z do topo da pilha, uma transição possível para A terá como próximo estado p, e a seqüência α será empilhada, depois da remoção de Z da pilha.

Da mesma forma que num autômato finito não determinístico (afnd), temos a possibilidade de executar uma transição sem avançar na leitura dos símbolos da entrada, e essa ação é representada pela leitura da cadeia vazia ε. Por essa razão, o segundo argumento de δ pode ser ε, além de poder ser um símbolo de Σ.

Usamos $P(K \times \Gamma^*)$ para representar o conjunto potência de $K \times \Gamma^*$, ou seja o conjunto de todos os subconjuntos de $K \times \Gamma^*$. Como se trata aqui de um conjunto infinito, é necessário especificar que, em qualquer caso, $\delta(q, a, Z)$ será sempre um conjunto finito, de forma a manter finita a descrição do apnd A.

Para descrever os passos de uma computação realizada por um apnd, utilizamos *configurações* $[q, y, \gamma] \in K \times \Sigma^* \times \Gamma^*$. Os três elementos de uma configuração são o estado corrente $q \in K$, a parte da entrada ainda a ser lida, $y \in \Sigma^*$, e o conteúdo $\gamma \in \Gamma^*$ da pilha. Por convenção, *o topo da pilha fica à esquerda*, isto é, o primeiro símbolo de γ é considerado como sendo o símbolo do topo da pilha. A configuração inicial correspondente à entrada x é $[i, x, I]$.

Vamos definir a relação *mudança de configuração*, representada por $\vdash$ ou por $\vdash_A$, se quisermos explicitar o apnd A considerado. Para isso utilizamos a função de transição δ. Suponha uma configuração $[q, ax, Z\gamma]$, com $(p, \alpha) \in \delta(q, a, Z)$. Escolhida esta opção, o próximo estado será p, e a cadeia α deve ser empilhada, após a leitura de Z. Assim, podemos passar da configuração $[q, ax, Z\gamma]$para a configuração $[p, x, \alpha\gamma]$, em que o estado passou a ser p, o "símbolo" a da entrada e o símbolo Z da pilha foram lidos, e a seqüência α foi empilhada. Ou seja,

se $\quad (p, \alpha) \in \delta(q, a, Z)$

então $\quad [q, ax, Z\gamma] \vdash [p, x, \alpha\gamma]$

Temos duas situações em que se pode dizer que um apnd aceita sua entrada, que correspondem a duas definições independentes do que se entende por linguagem reconhecida por um apnd. A primeira é semelhante à usada nos afnd: o apnd aceita se, após ler sua entrada, *o estado é um estado final*; a segunda é mais característica dos apnd: o apnd aceita se, após ler sua entrada, *a pilha está vazia*. Assim, podemos definir a linguagem de um apnd A (a linguagem aceita, ou reconhecida por A) de duas maneiras diferentes:

- aceitação por estado final:

  L_{ef} (A) = { x $\in \Sigma^*$ | [i, x, I] $|\!\!—^*$ [f, ε, γ], com f∈F e γ∈Γ* qualquer }

- aceitação por pilha vazia:

  L_{pv} (A) = { x $\in \Sigma^*$ | [i, x, I] $|\!\!—^*$ [f, ε, ε], com q∈K qualquer }

  São, portanto, duas as definições de configuração final:

  [q, ε, γ] é uma configuração final para aceitação por estado final se q ∈ F;

  [q, ε, γ] é uma configuração final para aceitação por pilha vazia, se γ = ε.

  No primeiro caso, o conteúdo γ da pilha é irrelevante; no segundo caso, o estado q é irrelevante. Note-se que o conjunto de estados finais F de um apnd A não é utilizado na definição da linguagem que A aceita por pilha vazia. Por essa razão costuma-se, neste caso, fazer F = ∅, sem perda de generalidade.

Um ponto importante a observar é que, no caso geral, as linguagens L_{ef}(A) e L_{pv}(A) podem ser distintas.

Exemplo 6.1: Seja o apnd A = < K, Σ, Γ, δ, i, I, F >, com

K = {0, 1, 2}   i = 0

Σ = {a, b}   I = X

Γ = {X, A}   F = {2}

A função δ:{0,1,2}×{a,b,ε}×{X,A} → P({0,1,2}×{X,A}*) é dada por

δ(0, a, X) = {(0, AX)}   δ(1, b, A) = {(1, ε)}

δ(0, a, A) = {(0, AA)}   δ(1, ε, X) = {(2, X)}

δ(0, b, A) = {(1, ε)}

- o primeiro símbolo deve ser um a
- o número de a's deve ser igual ao de b's.
- após o primeiro b, nenhum outro a pode ser aceito.
- após o último b, uma transição-ε leva A para uma configuração em que o estado é 2, e nenhuma transição adicional é possível.

Portanto, L_{ef} (A) = { $a^j\ b^j$ | j≥1 }. Por outro lado, nenhuma das transições prevê a retirada de X do fundo da pilha, de forma que L_{pv}(A)=∅.

Exemplo 6.2: Seja o apnd $A = < K, \Sigma, \Gamma, \delta, i, I, F >$, com

$K = \{0, 1\}$ $\quad$ $i = 0$

$\Sigma = \{a, b\}$ $\quad$ $I = X$

$\Gamma = \{X, A\}$ $\quad$ $F = \varnothing$

sendo a função $\delta: \{0,1\} \times \{a,b,\varepsilon\} \times \{X,A\} \to P(\{0,1\} \times \{X,A\}^*)$ dada por:

$\delta(0, a, X) = \{(0, A)\}$ $\quad$ $\delta(0, b, A) = \{(1, \varepsilon)\}$

$\delta(0, a, A) = \{(0, AA)\}$ $\quad$ $\delta(1, b, A) = \{(1, \varepsilon)\}$

Com entrada aaabbb, os seguintes são passos possíveis:

$$[0, aaabbb, X] \vdash [0, aabbb, A] \vdash [0, abbb, AA] \vdash [0, bbb, AAA]$$
$$\vdash [1, bb, AA] \vdash [1, b, A] \vdash [1, \varepsilon, \varepsilon]$$

sendo δ dada por:

$\delta(q, \varepsilon, E) = \{(q, E+T), (q, T)\}$

$\delta(q, \varepsilon, T) = \{(q, T*F), (q, F)\}$

$\delta(q, \varepsilon, F) = \{(q, (E)), (q, a)\}$

$\delta(q, +, +) = \delta(q, *, *) = \delta(q, (, () = \delta(q,),)) = \delta(q, a, a) = \{(q, \varepsilon)\}$

De acordo com a especificação da função δ acima, vemos que há dois tipos de transições possíveis neste apnd, conforme o símbolo de $N \cup \Sigma$ que aparece no topo da pilha:

- se o símbolo do topo da pilha é um nãoterminal de G_0, nenhum símbolo da entrada é lido, e há uma possibilidade para cada regra do nãoterminal.

  O apnd A depende do não-determinismo para escolher entre as diversas possibilidades aquela que corresponde à regra de G_0 usada no passo correspondente numa derivação esquerda da entrada x.

- se o símbolo do topo da pilha é um terminal de G_0, e o mesmo símbolo é lido da entrada.

  Esta transição só pode ser feita se a escolha (não-determinística) das transições do primeiro tipo foi feita de forma correta para a entrada.

Como toda a entrada foi lida e a pilha está vazia, temos aaabbb $\in L_{pv}$ (A).

Examinando o funcionamento de A, podemos verificar que $L_{pv}(A) = \{ a^j b^j \mid j \geq 1 \}$, ou seja, que a linguagem aceita por pilha vazia por este autômato de pilha é idêntica à linguagem aceita por estado final pelo autômato do exemplo anterior. Por outro lado, $F=\emptyset$ faz com que $L_{ef}(A)=\emptyset$.

Exercício 6.1: Construa um autômato de pilha A que aceita a linguagem dos dois exemplos anteriores, simultaneamente, por pilha vazia e por estado final, ou seja, construa um apnd A tal que $L_{ef}(A) = L_{pv}(A) = \{ a^i b^i \mid i \geq 1 \}$

Exemplo 6.3: Seja a glc $G_0 = \langle N, \Sigma, P, S \rangle$, com $N = \langle \{E, T, F\}$,

$\Sigma = \{+, *, (,), a\}, P, E \rangle$, S=E e P composto pelas regras:

$$E \to E + T \mid T$$

$$T \to T * F \mid F$$

$$F \to (E) \mid a$$

O apnd A, a seguir, aceita $L(G_0)$, por pilha vazia:

$$A = \langle \{q\}, \Sigma, N \cup \Sigma, \delta, q, E, \emptyset \rangle,$$

Por exemplo, para a entrada (a+a)*a, temos a derivação esquerda

$$E \Rightarrow T \Rightarrow T*F \Rightarrow F*F \Rightarrow (E)*F \Rightarrow (E+T)*F \Rightarrow (T+T)*F \Rightarrow (F+T)*F$$
$$\Rightarrow (a+T)*F \Rightarrow (a+F)*F \Rightarrow (a+a)*F \Rightarrow (a+a)*a$$

A mesma seqüência de regras é usada, e passos dos dois tipos são intercalados, de acordo com a natureza do símbolo que aparece no topo da pilha:

$$[q, (a+a)*a, E] \vdash [q, (a+a)*a, T] \vdash [q, (a+a)*a, T*F] \vdash [q, (a+a)*a, F*F]$$
$$\vdash [q, (a+a)*a, (E)*F] \vdash [q, a+a)*a, E)*F] \vdash [q, a+a)*a, E+T)*F]$$
$$\vdash [q, a+a)*a, T+T)*F] \vdash [q, a+a)*a, F+T)*F]$$
$$\vdash [q, a+a)*a, a+T)*F] \vdash [q, +a)*a, +T)*F] \vdash [q, a)*a, T)*F]$$
$$\vdash [q, a)*a, F)*F] \vdash [q, a)*a, a)*F] \vdash [q,)*a,)*F] \vdash [q, *a, *F]$$
$$\vdash [q, a, F] \vdash [q, a, a] \vdash [q, \varepsilon, \varepsilon]$$

(Usamos aqui colchetes para representar as configurações, para evitar confusão com os parênteses de Σ.)