Gramáticas e Autómatos de Pilha

Linguagens Formais e Autómatos

Francisco Coelho
fc@di.uevora.pt

Departamento de Informática
Escola de Ciências e Tecnologia
Universidade de Évora

UNIVERSIDADE DE ÉVORA

Gramáticas Independentes do Contexto

Autómatos de Pilha

Exemplo

$$
\begin{array}{rcl}
<\textit{frase}> & \rightarrow & <\textit{sujeito}>\ <\textit{predicado}> \\
<\textit{sujeito}> & \rightarrow & <\textit{artigo}>\ <\textit{substantivo}> \\
<\textit{predicado}> & \rightarrow & <\textit{verbo}>\ <\textit{advérbio}> \\
<\textit{artigo}> & \rightarrow & \texttt{a} \\
 & \rightarrow & \texttt{os} \\
<\textit{substantivo}> & \rightarrow & \texttt{água} \\
 & \rightarrow & \texttt{golfinhos} \\
<\textit{verbo}> & \rightarrow & \texttt{evapora-se} \\
 & \rightarrow & \texttt{mergulham} \\
<\textit{advérbio}> & \rightarrow & \texttt{lentamente} \\
 & \rightarrow & \texttt{profundamente}
\end{array}
$$

- Os termos <*assim*> são símbolos não terminais;
- Os termos `neste tipo` são símbolos terminais;

Exemplo — Aplicado

$$
\begin{aligned}
<\!\mathit{frase}\!> \quad & \rightarrow \quad <\!\mathit{sujeito}\!> \; <\!\mathit{predicado}\!> \\
& \rightarrow \quad <\!\mathit{artigo}\!> \; <\!\mathit{substantivo}\!> \; <\!\mathit{predicado}\!> \\
& \rightarrow \quad <\!\mathit{artigo}\!> \; <\!\mathit{substantivo}\!> \; <\!\mathit{verbo}\!> \; <\!\mathit{adv\acute{e}rbio}\!> \\
& \rightarrow \quad \mathsf{a} \; <\!\mathit{substantivo}\!> \; <\!\mathit{verbo}\!> \; <\!\mathit{adv\acute{e}rbio}\!> \\
& \rightarrow \quad \mathsf{a} \; \texttt{golfinhos} \; <\!\mathit{verbo}\!> \; <\!\mathit{adv\acute{e}rbio}\!> \\
& \rightarrow \quad \mathsf{a} \; \texttt{golfinhos evapora-se} \; <\!\mathit{adv\acute{e}rbio}\!> \\
& \rightarrow \quad \mathsf{a} \; \texttt{golfinhos evapora-se profundamente}
\end{aligned}
$$

Uma gramática independente do contexto (abrev. GIC) é um tuplo $G = (V, \Sigma, P, S)$ onde

- não terminais (ou variáveis) V é um conjunto dos símbolos;
- terminais Σ é um conjunto dos símbolos, disjunto de V $(V \cap \Sigma = \emptyset)$;
- produções (ou regras) $P \subseteq V \times (V \cup \Sigma)^*$. A produção $(A, w) \in P$ é escrita na forma $A \to w$;
- símbolo inicial $S \in V$;

Derivação

Seja $G = (V, \Sigma, P, S)$ uma gramática independente do contexto.

- se existe uma produção $A \to w$ em P então uAv deriva directamente uwv: $uAv \Rightarrow_G uwv$;
- se existem $w_0, \ldots, w_n \in (V \cup \Sigma)^*, n \geq 0$ tais que

$$w = w_0 \Rightarrow_G w_1 \Rightarrow_G \cdots \Rightarrow_G w_n = v$$

  então w deriva v em n passos: $w \Rightarrow_G^n v$;
- se existe $n \geq 0$ tal que $w \Rightarrow_G^n v$ então w deriva v: $w \Rightarrow_G^* v$;

UNIVERSIDADE DE ÉVORA

Seja $G = (V, \Sigma, P, S)$ uma gramática independente do contexto.

O conjunto das palavras deriváveis a partir de $v \in (V \cup \Sigma)^*$ é

$$\mathcal{D}(v) = \{w \ : \ v \Rightarrow^* w\}$$

A linguagem gerada por G é o conjunto das palavras de Σ^* deriváveis a partir de S:

$$\mathcal{L}(G) = \{w \in \Sigma^* \ : \ S \Rightarrow^* w\}$$

Nesse caso $\mathcal{L}(G)$ é uma linguagem independente do contexto.

Duas gramáticas são equivalentes se geram a mesma linguagem.

Exemplo

Qual é a linguagem gerada por
$G_1 = (\{S\}, \{0,1\}, \{S \to \lambda|0S1\}, S)$?

Quando várias produções têm o mesmo lado esquerdo, como em

$$A \to w_1, A \to w_2, \ldots, A \to w_n$$

abreviamos a escrita $A \to w_1|w_2|\ldots|w_n$.
Neste exemplo $S \to \lambda|0S1$ resulta de abreviar $S \to \lambda, S \to 0S1$.

Como só há um símbolo não terminal em G_1 e a única regra que o mantém é $S \to 0S1$, as derivações de G_1 são da forma

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow \cdots \Rightarrow 0^n S 1^n \Rightarrow 0^n 1^n.$$

Portanto $\mathcal{L}(G) = \{0^n 1^n \ : \ n \geq 0\}$.
Esta linguagem é regular?

Uma produção (directamente) recursiva tem a forma

$$A \to uAv$$

O não terminal A é recursivo se

$$A \Rightarrow^{+} uAv$$

Uma derivação da forma

$$A \Rightarrow w \Rightarrow^{+} uAv$$

onde A não ocorre em w, é indirectamente recursiva (com $u, v, w \in (V \cup \Sigma)^*$).

Sejam $G = (V, \Sigma, P, S)$ uma GIC e $v \Rightarrow^n w$ uma derivação em G onde

$$v = w_1 A_1 w_2 A_2 \cdots w_k A_k w_{k+1}$$

com os $w_i \in \Sigma^*$.

Então existem palavras $p_i \in (V \cup \Sigma)^*$ tais que

- $A_i \Rightarrow^{t_i} p_i$
- $w = w_1 p_1 w_2 p_2 \cdots w_k p_k w_{k+1}$
- $\sum_{i=1}^{k} t_i = n$

Derivação Esquerda e Direita

Numa derivação esquerda ($\Rightarrow_L$) em todos os passos é reescrito o não terminal mais à esquerda.

Numa derivação direita ($\Rightarrow_R$) em todos os passos é reescrito o não terminal mais à direita.

Teorema (existência de derivação esquerda)

Seja $G = (V, \Sigma, P, S)$ uma GIC. Para cada $w \in \Sigma^*$

$$w \in \mathcal{L}(G) \text{ sse } S \Rightarrow_L^* w \text{ sse } S \Rightarrow_R^* w$$

Seja $G = (V, \Sigma, P, S)$ uma GIC.

A árvore de derivação de $S \Rightarrow^* w$ é formada por

1. a raiz é o símbolo inicial S;
2. se $A \to x_1 x_2 \cdots x_n$ (com $x_i \in V \cup \Sigma$) é a produção usada para reescrever o símbolo A então o nó A na árvore tem filhos $x_1, x_2, \ldots, x_n$ por esta ordem;
3. se $A \to \lambda$ é a produção usada pra reescrever o símbolo A então o nó A na árvore tem λ como único filho;

Uma palavra w tem árvore de derivação T (e T é uma árvore de derivação de w) se w for a concatenação das folhas de T.

Ambiguidade

UNIVERSIDADE DE ÉVORA

Uma gramática G é ambígua se alguma palavra de $\mathcal{L}(G)$ tem pelo menos

- duas árvores de derivação distintas *ou*
- duas derivações esquerdas distintas *ou*
- duas derivações direitas distintas.

Uma linguagem é inerentemente ambígua se não existir uma gramática não ambígua que a gere.

$$\left\{a^i b^j c^k \ : \ i = j \vee j = k\right\}$$

Em "`if A then if B then C else D`"

- o "`else D`" é de "`if A`"?
- ou de "`if B`"?

Exemplo — Expressões Aritméticas

Seja $G_{EA} = (\{E, \cdots\}, \{n, +, -, \times, \div\}, \cdots, E)$ com produções...

1ª versão ambígua: $n + n \times n$

$$E \rightarrow E + E \mid E - E \mid E \times E \mid E \div E \mid n$$

precedência ainda ambígua: $n + n \times n \checkmark$, mas... $n + n + n$

$$E \rightarrow E + E \mid E - E \mid T$$
$$T \rightarrow T \times T \mid T \div T \mid F$$
$$F \rightarrow n$$

precedência + associatividade não ambígua

$$E \rightarrow E + T \mid E - T \mid T$$
$$T \rightarrow T \times F \mid T \div F \mid F$$
$$F \rightarrow n$$

E para incluir parênteses... por exemplo para $(n + n) \times n$?
Basta acrescentar a produção $F \rightarrow (E)$.

Uma gramática regular é uma GIC (V, Σ, P, S) em que todas as produções têm uma das formas

$$A \to a$$
$$A \to aB$$
$$A \to \lambda$$

com $A, B \in V$ e $a \in \Sigma$.

A linguagem gerada por uma gramática regular é uma linguagem regular.

Uma linguagem regular pode ser gerada por uma gramática não regular.

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFND.

A GIC equivalente a A é $G = (V, \Sigma, P, S)$ em que

- $V = \{X_i \ : \ i \in Q\}$;
- $S = X_{q_I}$;
- para cada $a \in \Sigma$, se $p \in \delta(q, a)$ então $X_q \to aX_p \in P$;
- se $p \in \delta(q, \lambda)$ então $X_q \to X_p \in P$;
- se $q \in F$ então $X_q \to \lambda \in P$;

Para cada AFND A, a gramática G obtida assim é regular e $\mathcal{L}(A) = \mathcal{L}(G)$.

A aplicação do processo inverso produz um AFND a partir de uma gramática regular (atenção às regras $A \to a$ e $X_q \to X_p$).

Gramáticas Independentes do Contexto

Autómatos de Pilha

Um autómato de pilha (AP) é um tuplo $A = (Q, \Sigma, \Gamma, \delta, q_I, F)$ onde

- Q, Σ, q_I e F são como nos autómatos finitos;
- alfabeto da pilha Γ é um conjunto finito de símbolos;
- transição é uma função com assinatura

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \to \mathcal{P}(Q \times (\Gamma \cup \{\lambda\}))$$

As palavras na pilha (i.e. em Γ^*) são denotadas por letras gregas minúsculas $\alpha, \beta, \gamma, \ldots$

Configuração de um Autómato de Pilha

Uma configuração de um AP é um triplo $[q, w, \alpha] \in Q \times \Sigma^* \times \Gamma^*$

A configuração inicial é $[q_I, w, \lambda]$ e

(AFND) se $(p, \lambda) \in \delta(q, a, \lambda)$

$$[q, aw, \alpha] \vdash [p, w, \alpha]$$

(remove) se $(p, \lambda) \in \delta(q, a, A)$

$$[q, aw, A\alpha] \vdash [p, w, \alpha]$$

(acrescenta) se $(p, B) \in \delta(q, a, \lambda)$

$$[q, aw, \alpha] \vdash [p, w, B\alpha]$$

(troca) se $(p, B) \in \delta(q, a, A)$

$$[q, aw, A\alpha] \vdash [p, w, B\alpha]$$

Linguagem Reconhecida

UNIVERSIDADE DE ÉVORA

Seja $A = (Q, \Sigma, \Gamma, \delta, q_I, F)$ um AP.

Uma palavra $w \in \Sigma^*$ é aceite por A se existe uma computação

$$[q_I, w, \lambda] \vdash_A^* [q_f, \lambda, \lambda]$$

com $q_f \in F$ (critério de aceitação por estado final e pilha vazia);

A linguagem reconhecida por A é o conjunto das palavras aceites por A.

Um autómato de pilha é determinista se para cada combinação de **estado, símbolo de entrada e topo da pilha** existe no máximo **uma** transição aplicável.

Variantes

Um autómato de pilha atómico só tem transições das formas

$$(p, \lambda) \in \delta(q, a, \lambda)$$
$$(p, \lambda) \in \delta(q, \lambda, A)$$
$$(p, A) \in \delta(q, \lambda, \lambda)$$

Um autómato de pilha estendido permite transições em que são empilhados mais do que um símbolo como, por exemplo

$$(p, BCD) \in \delta(q, a, \lambda)$$

- Qualquer linguagem reconhecida por um AP também é reconhecida por um AP atómico
- Qualquer linguagem reconhecida por um AP estendido também é reconhecida por um AP

Converter uma GIC num AP

Seja $G = (V, \Sigma, P, S)$ uma GIC e $A = (\{q\}, \Sigma, \Sigma \cup V, \delta, q, \{q\})$ um AP estendido com transição δ definida por...

- para cada $A \in V$, $\delta(q, \lambda, A) = \{(q, w) \ : \ A \to w \in P\}$;
- para cada $a \in \Sigma$, $\delta(q, a, a) = \{(q, \lambda)\}$

Então a linguagem gerada por G coincide com a linguagem aceite por A se a pilha for iniciada com S.

UNIVERSIDADE DE ÉVORA

Seja L uma linguagem independente do contexto.

Existe um k tal que qualquer palavra $p \in L$ com $|p| > k$ pode ser escrita como

$$p = uvwxy, \qquad \text{com } |vwx| \leq k, |v| + |x| > 0$$

e $uv^n wx^n y \in L$ para cada $n \geq 0$

Hierarquia de Chomsky

Qualquer gramática $G = (V, \Sigma, P, S)$ é de um dos seguintes tipos

tipo 0 ou sem restrições se as produções forem da forma $u \to v$ com $u \in (V \cup \Sigma)^+$ e $v \in (V \cup \Sigma)^*$

tipo 1 ou dependente do contexto se as produções forem da forma $u \to v$ com $u, v \in (V \cup \Sigma)^+$ e $|u| \leq |v|$

tipo 2 ou independente do contexto

tipo 3 ou regular