Linguagens Formais e Autômatos

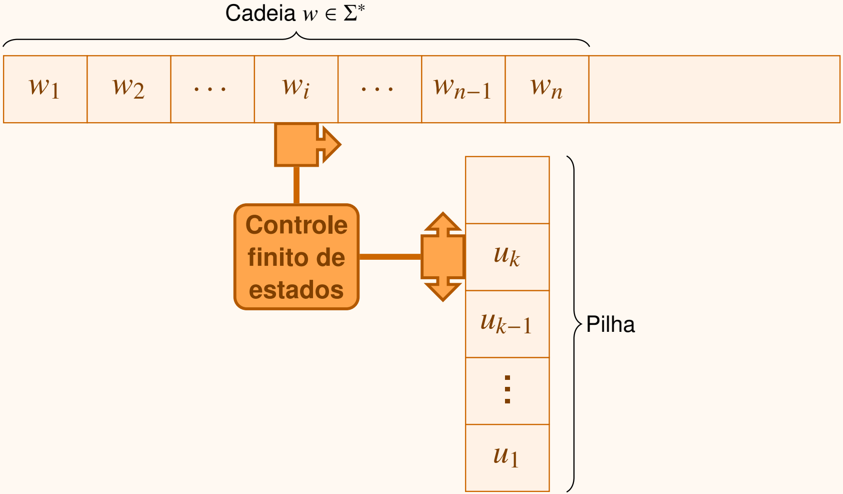
Humberto Longo

Instituto de Informática
Universidade Federal de Goiás

Bacharelado em Ciência da Computação, 2018/2

Pushdown Automata

Esquema básico



Definição

Definição 1.1

- Um Autômato com Pilha (*PDA – Pushdown Automaton*) é uma sextupla $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$, onde:
  - Σ : alfabeto de entrada;
  - Γ : alfabeto da pilha;
  - $S \neq \emptyset$: conjunto finito de estados;
  - $s_0 \in S$: estado inicial;
  - $\delta : S \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \to \mathcal{P}(S \times (\Gamma \cup \{\varepsilon\}))$: função de transição de estados; e
  - $F \subseteq S$: conjunto de estados finais (ou de aceitação).

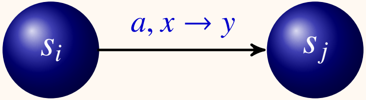
Processamento de um PDA

- $\delta(s_i, a, x) = \{(s_j, y), (s_k, z)\}$.
  - Duas transições possíveis quando o autômato está no estado s_i, lendo o símbolo a de entrada e com x no topo da pilha.
- A transição $(s_j, y) \in \delta(s_i, a, x)$ força o autômato a:
  1. Mudar o estado corrente de s_i para s_j;
  2. Processar o símbolo a (avançar a cabeça de leitura da fita);
  3. Remover o símbolo x do topo da pilha; e
  4. Colocar o símbolo y no topo da pilha.

Representação gráfica

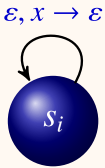
- $a, b \to c$:
  - $a = \varepsilon$: transição sem ler símbolo de entrada.
  - $b = \varepsilon$: transição sem ler símbolo da pilha.
  - $c = \varepsilon$: transição sem escrever na pilha.

- $\delta(s_i, a, x) = \{(s_j, y)\}$.
  - O PDA muda do estado s_i para o s_j, lê a da fita de entrada, desempilha x e empilha y.



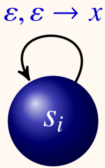
Representação gráfica

- $\delta(s_i, \varepsilon, x) = \{(s_i, \varepsilon)\}$.
  - Se a posição de entrada é ε, a transição não processa um símbolo de entrada, mas desempilha o x.



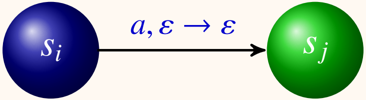
Representação gráfica

- $\delta(s_i, \varepsilon, \varepsilon) = \{(s_i, x)\}$.
  - Se a posição de entrada é ε, a transição não processa um símbolo de entrada, mas empilha o x.



Representação gráfica

- $\delta(s_i, a, \varepsilon) = \{(s_j, \varepsilon)\}$.
  - Transição equivalente a transição de um DFA.
  - Efeito determinado somente pelo estado corrente e pelo símbolo de entrada.
  - Transição não consulta e não altera a pilha.



Processamento de um PDA

- $P = \langle \Sigma, \Gamma, S, s_{ini}, \delta, F \rangle$.
- $w = w_1 w_2 \ldots w_m$, com $w_i \in \Sigma,\ i = 1, \ldots, m$: cadeia de entrada.
- $s_0, s_1, \ldots, s_m \in Q$: seqüência de estados.
- $u_0, u_1, \ldots, u_m \in \Gamma^*$: seqüência de conteúdos da pilha.

Processamento de um PDA

- P aceita a cadeia w se:
  1. $s_0 = s_{ini}$ e $u_0 = \varepsilon$.
     - P começa no estado inicial e com a pilha vazia.
  2. $(s_{i+1}, u_{i+1}) \in \delta(s_i, w_{i+1}, u_i),\ i = 0, \ldots, m-1$, onde $u_i = av$ e $u_{i+1} = bv'$ para $a, b \in \Gamma \cup \{\varepsilon\}$ e $v, v' \in \Gamma^*$.
     - P move-se de acordo com o estado atual, a pilha e o próximo símbolo da cadeia.
  3. $s_m \in F$.
     - Um estado final ocorre no final da cadeia.

Configuração de um PDA

Definição 1.2

- Tripla $[s_i, w, \alpha]$, onde s_i é o estado corrente, $w \in \Sigma^*$ é o conjunto de símbolos ainda não processados e α é o conteúdo da pilha.

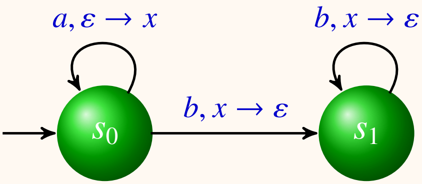
Notação

- $\underset{M}{\longmapsto}$: define uma função de $S \times \Sigma^* \times \Gamma^*$ em $S \times \Sigma^* \times \Gamma^*$.
- $[s_i, w, \alpha] \underset{M}{\longmapsto} [s_j, v, \beta]$: configuração $[s_j, v, \beta]$ é obtida a partir de $[s_i, w, \alpha]$ com apenas uma transição de estados.
- $\overset{*}{\underset{M}{\longmapsto}}$: representa uma seqüência de transições.

Exemplo de autômato com pilha

Exemplo 1.3

- $P = \langle \Sigma = \{a, b\}, \Gamma = \{x\}, S = \{s_0, s_1\}, s_0, \delta, F = \{s_0, s_1\} \rangle$, onde:
  - $\delta(s_0, a, \varepsilon) = \{(s_0, x)\}$
  - $\delta(s_0, b, x) = \{(s_1, \varepsilon)\}$
  - $\delta(s_1, b, x) = \{(s_1, \varepsilon)\}$
- $\mathcal{L}(P) = \{a^i b^i \mid i \geq 0\}$
- $[s_0, aabb, \varepsilon] \longmapsto [s_0, abb, x] \longmapsto [s_0, bb, xx] \longmapsto [s_1, b, x] \longmapsto [s_1, \varepsilon, \varepsilon]$



Linguagem aceita por um PDA

Definição 1.4

Seja $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$ um PDA. Uma cadeia $w \in \Sigma^*$ é aceita por P se existe um processamento

$$[s_0, w, \varepsilon] \overset{*}{\longmapsto} [s_i, \varepsilon, \varepsilon],$$

onde $s_i \in F$. A linguagem de P, denotada $\mathcal{L}(P)$, é o conjunto de cadeias aceitas por P.

Exemplo de autômato com pilha

Exemplo 1.5

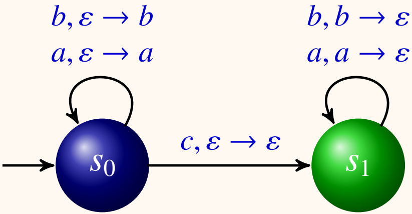
- $L = \{wcw^R \mid w \in \{a, b\}^*\}$.
- $P = \langle \Sigma = \{a, b, c\}, \Gamma = \{a, b\}, S = \{s_0, s_1\}, s_0, \delta, F = \{s_1\}\rangle$, onde:

$$
\begin{aligned}
\delta(s_0, a, \varepsilon) &= \{(s_0, a)\}\\
\delta(s_0, b, \varepsilon) &= \{(s_0, b)\}\\
\delta(s_0, c, \varepsilon) &= \{(s_1, \varepsilon)\}\\
\delta(s_1, a, a) &= \{(s_1, \varepsilon)\}\\
\delta(s_1, b, b) &= \{(s_1, \varepsilon)\}
\end{aligned}
$$

Exemplo de autômato com pilha

Exemplo 1.5

- Processamento da cadeia $abcba$:
  $[s_0, abcba, \varepsilon] \longmapsto [s_0, bcba, a] \longmapsto [s_0, cba, ba]$
  $\longmapsto [s_1, ba, ba] \longmapsto [s_1, a, a] \longmapsto [s_1, \varepsilon, \varepsilon]$



Transições compatíveis

Definição 1.6

- Duas transições $(s_j, c) \in \delta(s_i, u, a)$ e $(s_k, d) \in \delta(s_i, v, b)$ são compatíveis se alguma das condições a seguir é satisfeita:
  1. $u = v$ e $a = b$;
  2. $u = v$ e $a = \varepsilon$ ou $b = \varepsilon$;
  3. $a = b$ e $u = \varepsilon$ ou $v = \varepsilon$;
  4. $u = \varepsilon$ ou $v = \varepsilon$ e $a = \varepsilon$ ou $b = \varepsilon$;
- Transições compatíveis podem ser aplicadas para a mesma configuração.

PDA determinístico

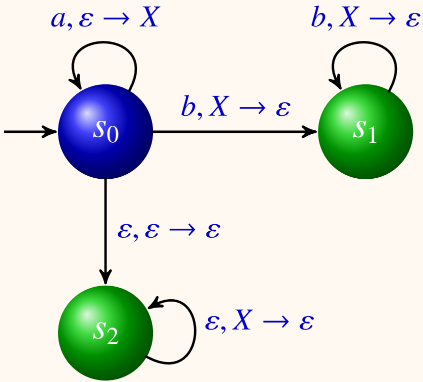
Definição 1.7

- Um PDA é determinístico se existe no máximo uma transição para cada combinação de estado, símbolo de entrada e símbolo no topo da pilha.
- Um PDA é determinístico se não contém transições compatíveis distintas.

Exemplo de PDA não determinístico

Exemplo 1.8

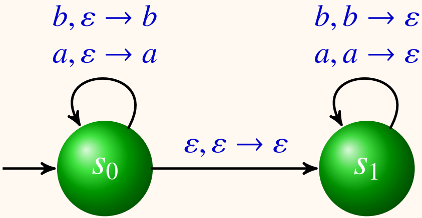
- $L = \{a^i \mid i \geq 0\} \cup \{a^i b^i \mid i \geq 0\}$.
  - A transição ε a partir de s_0 permite chegar a s_2 depois de processar toda a cadeia de entrada.
  - Esta transição introduz o não determinismo ao PDA.



Exemplo de PDA não determinístico

Exemplo 1.9

- $L = \{ww^R \mid w \in \{a, b\}^*\}$.
  - O não determinismo permite ao PDA "adivinhar" quando a metade da cadeia de entrada foi processada.



Processamento de um PDA

- Definição formal não contém mecanismos para testar pilha vazia.
- Um PDA pode simular esse mecanismo com um símbolo particular (por exemplo, $\$$):
  - $\$$ é o primeiro símbolo a ser colocado na pilha.
  - Quando o PDA ler novamente o $\$$, então a pilha está vazia.

Exemplo de PDA não determinístico

Exemplo 1.10

- $L = \{0^n 1^n \mid n \geq 0\}$.
- $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$, onde:
  - $S = \{s_0, s_1, s_2, s_3\}$.
  - $\Sigma = \{0, 1\}$.
  - $\Gamma = \{0, \$\}$.
  - $F = \{s_0, s_3\}$.

Exemplo de PDA não determinístico

Exemplo 1.10

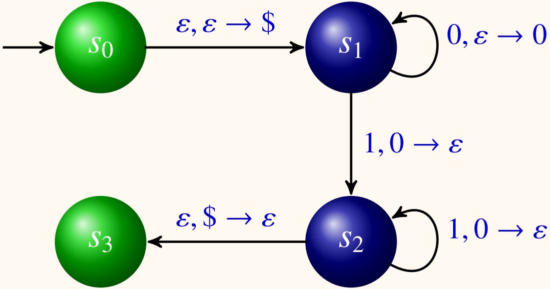
- δ é definida na tabela a seguir, onde entradas em branco significam $\emptyset$:

Entrada	0			1			ε		
Pilha	0	$\$$	ε	0	$\$$	ε	0	$\$$	ε
s_0									$\{(s_1, \$)\}$
s_1			$\{(s_1, 0)\}$	$\{(s_2, \varepsilon)\}$					
s_2				$\{(s_2, \varepsilon)\}$				$\{(s_3, \varepsilon)\}$	
s_3									

Exemplo de PDA não determinístico

Exemplo 1.10

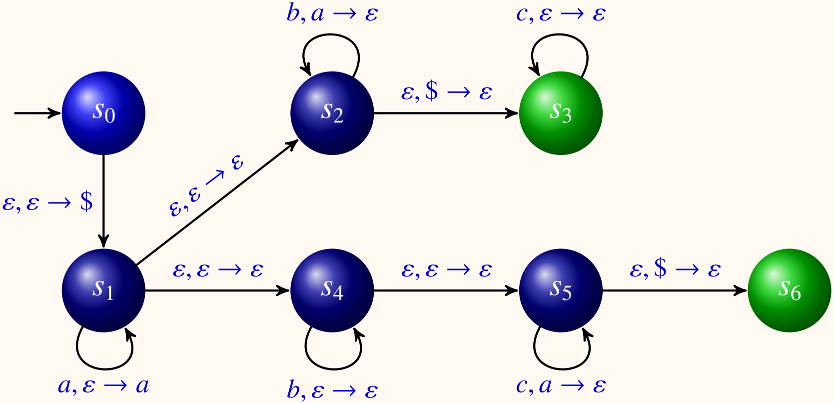
- Representação gráfica:



Exemplo de PDA não determinístico

Exemplo 1.11

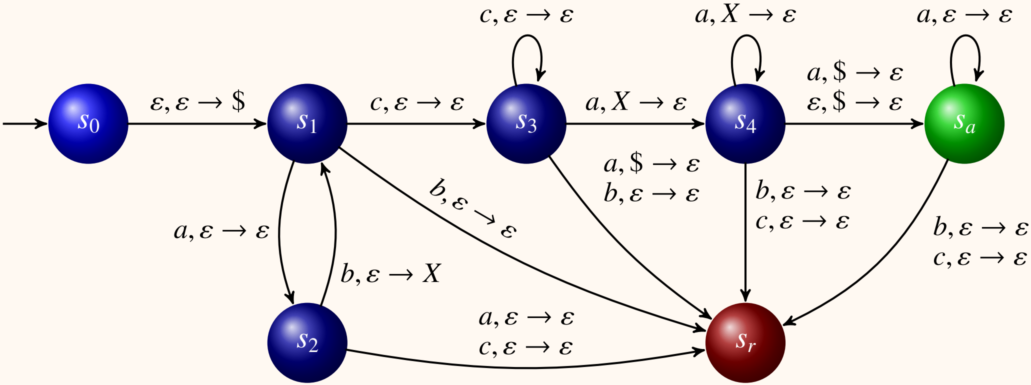
- $\mathcal{L} = \{a^i b^j c^k \in \{a, b, c\}^* \mid i, j, k \geq 0$ e $i = j$ ou $i = k\}$.
  - PDA lê e empilha todos os a's.
  - Os símbolos a's devem ser 'casados' com b's ou c's?
  - Não determinismo é essencial para reconhecer $\mathcal{L}$!



Exemplo de PDA determinístico

Exemplo 1.12

- $\mathcal{L} = \{(ab)^i c^k a^j \in \{a, b, c\}^* \mid j \geqslant i \geqslant 1, k \geqslant 1\}$.



PDA atômico

Definição 1.13

- A transição de um PDA acarreta três ações: processar um símbolo da cadeia, retirar um símbolo da pilha e colocar outro símbolo na pilha.
- Um PDA é chamado de atômico se cada transição causa apenas uma dessas ações.
- Transições em um PDA atômico têm a forma:

$$(s_j, \varepsilon) \in \delta(s_i, a, \varepsilon)$$
$$(s_j, \varepsilon) \in \delta(s_i, \varepsilon, a)$$
$$(s_j, a) \in \delta(s_i, \varepsilon, \varepsilon)$$

PDA atômico

Teorema 1.14

- *Se P é um PDA, então existe um PDA atômico P' com $\mathcal{L}(P') = \mathcal{L}(P)$.*

Demonstração.

- Para construir P', cada transição não atômica de P deve ser trocada por uma sequência de transições atômicas.
  - Dada a transição $(s_j, b) \in \delta(s_i, a, a)$ de P, são necessários dois novos estados s_1 e s_2 e as transições:

$$
\begin{aligned}
(s_1, \varepsilon) &\in \delta(s_i, a, \varepsilon), \\
\delta(s_1, \varepsilon, a) &= \{(s_2, \varepsilon)\}, \\
\delta(s_2, \varepsilon, \varepsilon) &= \{(s_j, b)\}.
\end{aligned}
$$

□

PDA atômico

Teorema 1.14

- *Se P é um PDA, então existe um PDA atômico P' com $\mathcal{L}(P') = \mathcal{L}(P)$.*

Demonstração.

- De forma similar, uma transição que consiste na mudança de estado e que acarreta apenas duas ações, pode ser trocada por uma sequência de duas transições atômicas.
- A remoção de todas transições não atômicas produz um PDA atômico equivalente.

$\square$

Transição estendida

Definição 1.15

- Uma transição estendida, em um PDA, empilha uma cadeia de caracteres e não apenas um único símbolo.
  - Ex.: a transição $(s_j, bcd) \in \delta(s_i, u, a)$ empilha bcd, com b ficando no topo da pilha.
- Um PDA estendido é aquele que contém transições estendidas.

Transição estendida

Teorema 1.16

- *Se P é um PDA estendido, então existe um PDA P' com $\mathcal{L}(P') = \mathcal{L}(P)$.*

Demonstração.

- Para construir P', cada transição estendida P deve ser trocada por uma sequência de transições.
  - Dada a transição $(s_j, bcd) \in \delta(s_i, u, a)$ de P, são necessários dois novos estados s_1 e s_2 e as transições:

$$
\begin{aligned}
(s_1, d) &\in \delta(s_i, u, a), \\
\delta(s_1, \varepsilon, \varepsilon) &= \{(s_2, c)\}, \\
\delta(s_2, \varepsilon, \varepsilon) &= \{(s_j, b)\}.
\end{aligned}
$$

□

Transição estendida

Exemplo 1.17

- $L = \{a^i b^{2i} \mid i \geq 1\}$.

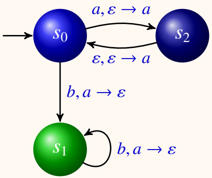
PDA	PDA atômico	PDA estendido
$S = \{s_0, s_1, s_2\}$	$S = \{s_0, s_1, s_2, s_3, s_4\}$	$S = \{s_0, s_1\}$
$\delta(s_0, a, \varepsilon) = \{(s_2, a)\}$	$\delta(s_0, a, \varepsilon) = \{(s_3, \varepsilon)\}$	$\delta(s_0, a, \varepsilon) = \{(s_0, aa)\}$
$\delta(s_2, \varepsilon, \varepsilon) = \{(s_0, a)\}$	$\delta(s_3, \varepsilon, \varepsilon) = \{(s_2, a)\}$	$\delta(s_0, b, a) = \{(s_1, \varepsilon)\}$
$\delta(s_0, b, a) = \{(s_1, \varepsilon)\}$	$\delta(s_2, \varepsilon, \varepsilon) = \{(s_0, a)\}$	$\delta(s_1, b, a) = \{(s_1, \varepsilon)\}$
$\delta(s_1, b, a) = \{(s_1, \varepsilon)\}$	$\delta(s_0, b, \varepsilon) = \{(s_4, \varepsilon)\}$	
	$\delta(s_4, \varepsilon, a) = \{(s_1, \varepsilon)\}$	
	$\delta(s_1, b, \varepsilon) = \{(s_4, \varepsilon)\}$	

Transição estendida

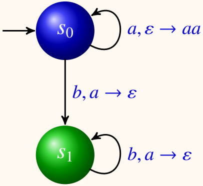
Exemplo 1.17

- $L = \{a^i b^{2i} \mid i \geq 1\}$.

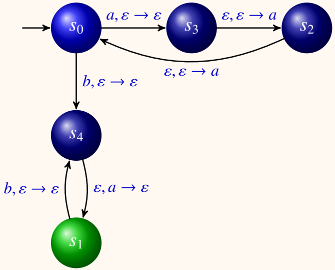
- PDA:



- PDA estendido:



- PDA atômico:



Aceitação por estado final

Definição 1.18

- Seja o PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$. A aceitação da cadeia $w \in \Sigma^*$ é definida por estado final se existe um processamento

$$[s_0, w, \varepsilon] \overset{*}{\longmapsto} [s_i, \varepsilon, \alpha],$$

  onde $s_i \in F$ e $\alpha \in \Gamma^*$.
  - Definir aceitação em termos do estado final ou da configuração da pilha não altera o conjunto de linguagens reconhecidas pelos autômatos finitos.
- A linguagem aceita por estado final é denotada $\mathcal{L}_F$.

Aceitação por estado final

Lema 1.19

- *Se $\mathcal{L}(P)$ é a linguagem aceita pelo PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$, com aceitação definida por estado final, então existe um PDA P' que aceita $\mathcal{L}(P)$, com aceitação definida por estado final e pilha vazia.*

Demonstração.

- $P' = \langle \Sigma, \Gamma, S \cup \{s_f\}, s_0, \delta', \{s_f\} \rangle$.
- a função δ' é igual à função δ acrescida das transições:

$$\delta'(s_i, \varepsilon, \varepsilon) = \{(s_f, \varepsilon)\}, \quad \forall\, s_i \in F;$$
$$\delta'(s_f, \varepsilon, a) = \{(s_f, \varepsilon)\}, \quad \forall\, a \in \Gamma.$$

□

Aceitação por estado final

Lema 1.19

- *Se $\mathcal{L}(P)$ é a linguagem aceita pelo PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$, com aceitação definida por estado final, então existe um PDA P' que aceita $\mathcal{L}(P)$, com aceitação definida por estado final e pilha vazia.*

Demonstração.

- Seja o processamento $[s_0, w, \varepsilon] \underset{P}{\overset{*}{\longmapsto}} [s_i, \varepsilon, \alpha]$ que aceita w por estado final.
- O equivalente em P' é:
$$[s_0, w, \varepsilon] \underset{P}{\overset{*}{\longmapsto}} [s_i, \varepsilon, \alpha] \underset{P'}{\longmapsto} [s_f, \varepsilon, \alpha] \underset{P'}{\overset{*}{\longmapsto}} [s_f, \varepsilon, \varepsilon].$$

□

Aceitação por estado final

Lema 1.19

- *Se $\mathcal{L}(P)$ é a linguagem aceita pelo PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta, F \rangle$, com aceitação definida por estado final, então existe um PDA P' que aceita $\mathcal{L}(P)$, com aceitação definida por estado final e pilha vazia.*

Demonstração.

- As novas transições não levam P' a aceitar cadeias que não pertençam à $\mathcal{L}(P)$:
  - O único estado final de P' é s_f, o qual é alcançável a partir de qualquer estado final de P.
  - As transições a partir de s_f desempilham símbolos, mas não processam a cadeia de entrada.

□

Aceitação por pilha vazia

Definição 1.20

- Seja o PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta \rangle$. A aceitação da cadeia $w \in \Sigma^*$ é definida por pilha vazia se existe um processamento

$$[s_0, w, \varepsilon] \overset{+}{\longmapsto} [s_i, \varepsilon, \varepsilon],$$

  onde não há restrição quanto ao estado s_i de parada do processamento.
  - É necessário pelo menos uma transição para permitir a aceitação de linguagens que não contenham a cadeia vazia.
- A linguagem aceita por pilha vazia é denotada $\mathcal{L}_E$.

Aceitação por pilha vazia

Lema 1.21

- *Se $\mathcal{L}(P)$ é a linguagem aceita pelo PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta \rangle$, com aceitação definida por pilha vazia, então existe um PDA P' que aceita $\mathcal{L}(P)$, com aceitação definida por estado final e pilha vazia.*

Demonstração.

- $P' = \langle \Sigma, \Gamma, S \cup \{s'_0\}, s'_0, \delta', S \rangle$, onde:
$$\delta'(s_i, a, x) = \delta(s_i, a, x) \text{ e } \delta'(s'_0, a, x) = \delta(s_0, a, x),$$
$$\forall\ s_i \in S, a \in \Sigma \cup \{\varepsilon\} \text{ e } x \in \Gamma \cup \{\varepsilon\}.$$
  - Os processamentos de P e P' são idênticos, exceto que o estado inicial de P é s_0 e o inicial de P' é s'_0.

$\square$

Aceitação por pilha vazia

Lema 1.21

- *Se $\mathcal{L}(P)$ é a linguagem aceita pelo PDA $P = \langle \Sigma, \Gamma, S, s_0, \delta \rangle$, com aceitação definida por pilha vazia, então existe um PDA P' que aceita $\mathcal{L}(P)$, com aceitação definida por estado final e pilha vazia.*

Demonstração.

- Todo processamento em P', de comprimento um ou maior, que para com pilha vazia também para em um estado final.
- Como s_0' não é final, ε é aceito por P' só se é aceita por P.
- Portanto, $\mathcal{L}(P') = \mathcal{L}_E(P)$.

$\square$

Linguagens aceitas por PDA's

Teorema 1.22

- *As três condições a seguir são equivalentes:*
  1. *a linguagem $\mathcal{L}(P)$ é aceita pelo PDA P;*
  2. *existe um PDA P_1 tal que $\mathcal{L}_F(P_1) = \mathcal{L}(P)$; e*
  3. *existe um PDA P_2 tal que $\mathcal{L}_E(P_2) = \mathcal{L}(P)$.*

Linguagens livres de contexto e PDA's

Conversão de GLC em PDA

- Toda LLC é aceita por um PDA estendido.
  - As regras de derivação podem ser usadas para gerar as transições do PDA.
- Seja $\mathcal{L}$ uma LLC e G uma gramática na forma normal de Greibach com $\mathcal{L}(G) = \mathcal{L}$.
  - As regras de G, exceto $S \to \varepsilon$, tem a forma $A \to aA_1A_2 \ldots A_n$.
  - Em uma derivação à esquerda, as variáveis A_i são substituídas na seqüência $A_1, A_2, \ldots, A_n$.
- Empilhar $A_1, A_2, \ldots, A_n$, com A_1 no topo da pilha, armazena as variáveis na ordem requerida pela derivação.

Exemplo 1.23

- $\mathcal{L} = \{a^i b^i \mid i > 0\}$.
- Gramática G na forma normal de Greibach que aceita $\mathcal{L}$:

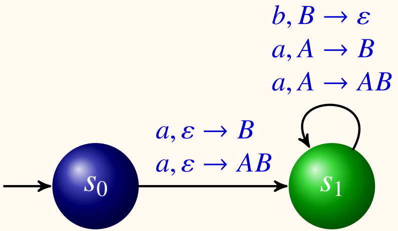
$$G : \left\{ \begin{array}{l} S \rightarrow aAB \mid aB, \\ A \rightarrow aAB \mid aB, \\ B \rightarrow b \end{array} \right\}.$$

Linguagens livres de contexto e PDA's

Exemplo 1.23

- PDA $P = \langle \Sigma = \{a, b\}, \Gamma = \{A, B\}, S = \{s_0, s_1\}, s_0, \delta, F = \{s_1\} \rangle$, onde:

$$\delta(s_0, a, \varepsilon) = \{(s_1, AB), (s_1, B)\},$$
$$\delta(s_1, a, A) = \{(s_1, AB), (s_1, B)\},$$
$$\delta(s_1, b, B) = \{(s_1, \varepsilon)\}.$$



- Uma regra da forma $S \to aA_1A_2 \ldots A_n$ gera uma transição que processa a, empilha $A_1, A_2, \ldots, A_n$ e entra no estado s_1.

Exemplo 1.23

- Derivação da cadeia $aaabbb$ por G e processamento por P:

$$
\begin{aligned}
S &\Rightarrow aAB & [s_0, aaabbb, \varepsilon] &\longmapsto [s_1, aabbb, AB] \\
&\Rightarrow aaABB & &\longmapsto [s_1, abbb, ABB] \\
&\Rightarrow aaaBBB & &\longmapsto [s_1, bbb, BBB] \\
&\Rightarrow aaabBB & &\longmapsto [s_1, bb, BB] \\
&\Rightarrow aaabbB & &\longmapsto [s_1, b, B] \\
&\Rightarrow aaabbb. & &\longmapsto [s_1, \varepsilon, \varepsilon].
\end{aligned}
$$

Conversão de GLC em PDA – Alternativa

- $\mathcal{L}$: LLC.
- G : GLC que gera $\mathcal{L}$.
- Conversão de G em um PDA P.
  - Se G gera w, então P aceita w.
  - P determina se existe uma derivação para w.
- Quais produções devem ser utilizadas na derivação?
  - P deve ser não determinístico.

Equivalência de GLC e PDA

Funcionamento do PDA P

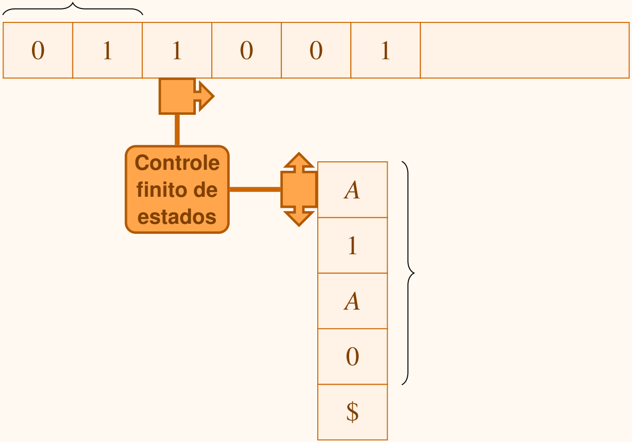
1. Variável inicial na pilha.
2. Série de cadeias intermediárias: substituições uma a uma.
    - Pode chegar a uma cadeia que só contém símbolos terminais.
    - P aceita essa cadeia se é igual à cadeia w.

- Tratamento das cadeias intermediárias:
    - P tem acesso somente ao topo da pilha, que pode ser um terminal ou uma variável.
    - Retirar parte da cadeia intermediária (primeira variável) da pilha.
    - 'Casar' qualquer terminal anterior com os símbolos da cadeia de entrada.

Equivalência de GLC e PDA

Exemplo 1.24

- PDA P com a cadeia intermediária $A1A0$ a pilha:



Equivalência de GLC e PDA

Funcionamento do PDA P

1. Inserir símbolo $\$$ na pilha.
2. Inserir variável inicial na pilha.
3. Repetir os passos:
   3.1 Se topo da pilha é uma variável A, escolher (não determinístico) uma produção para A e substituir A pelo lado direito da produção.
   3.2 Se topo da pilha é um terminal a, ler próximo símbolo da cadeia de entrada e comparar com a. Se iguais, repetir, senão rejeitar.
   3.3 Se topo da pilha é o símbolo $\$$, ir para estado final. Se cadeia de entrada foi toda lida, então foi aceita.

Equivalência de GLC e PDA

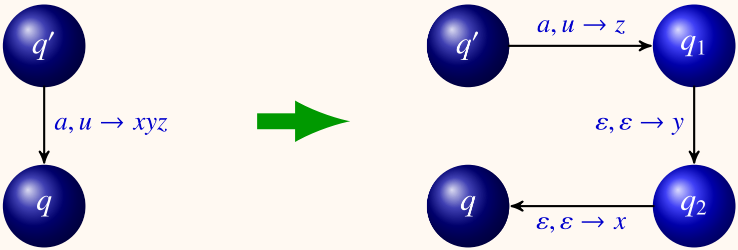
Esquema de construção do PDA

- Construção do PDA $P = \langle \Sigma, \Gamma, Q, q_0, \delta, F \rangle$:
  - $q', q \in Q$,
  - $a \in \Sigma$,
  - $u \in \Gamma$,
  - P passa do estado q' para o $q \Rightarrow P$ lê a e desempilha u.

Equivalência de GLC e PDA

Esquema de construção do PDA

- $(q, w) \in \delta(q', a, u) \Rightarrow q'$ é o estado do PDA, a é o próximo símbolo de entrada e u é o topo da pilha.
  - O PDA deve ler a, desempilhar u, empilhar a cadeia w e ir para o estado q.
- Exemplo para $(q, xyz) \in \delta(q', a, u)$:



Equivalência de GLC e PDA

Esquema de construção do PDA

- Empilhar toda a cadeia $w = w_1 \dots w_\ell$ ao mesmo tempo:
  - Novos estados $q_1, \dots, q_{\ell-1}$ e transição $\delta(q, a, u)$ tal que:

$$
\begin{aligned}
(q_1, w_\ell) &\in \delta(q, a, u), \\
\delta(q_1, \varepsilon, \varepsilon) &= \{(q_2, w_{\ell-1})\}, \\
\delta(q_2, \varepsilon, \varepsilon) &= \{(q_3, w_{\ell-2})\}, \\
&\vdots \\
\delta(q_{\ell-1}, \varepsilon, \varepsilon) &= \{(q, w_\ell)\}.
\end{aligned}
$$

Equivalência de GLC e PDA

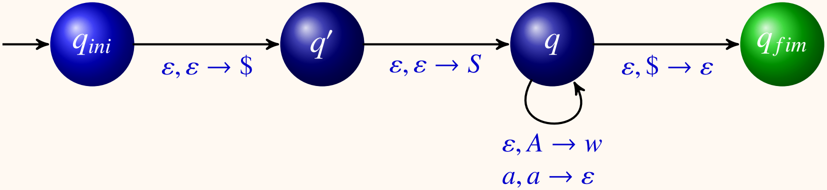
Esquema de construção do PDA

- $PDA\ \ P = \langle \Sigma, \Gamma, Q, q_0, \delta, F \rangle$.
  - $Q = \{q_{ini}, q', q, q_{fim}\} \cup E$.
    - E : novos estados para a notação simplificada para transições.
  - $q_0 = q_{ini}$.
  - $F = \{q_{fim}\}$.
  - $\delta(q_{ini}, \varepsilon, \varepsilon) = \{(q', \$)\}$.
    - A pilha é iniciada com $\$$.
  - $\delta(q', \varepsilon, \varepsilon) = \{(q, S)\}$.
    - A variável inicial S é colocada na pilha.
  - $\delta(q, \varepsilon, A) = \{(q, a)\}$, onde $(A \to a) \in R$.
    - O topo da pilha contém uma variável.
  - $\delta(q, a, a) = \{(q, \varepsilon)\}$.
    - O topo da pilha contém um terminal.
  - $\delta(q, \varepsilon, \$) = \{(q_{fim}, \varepsilon)\}$.
    - Marcador de pilha vazia ($\$$) está no topo.

Equivalência de GLC e PDA

Esquema de construção do PDA

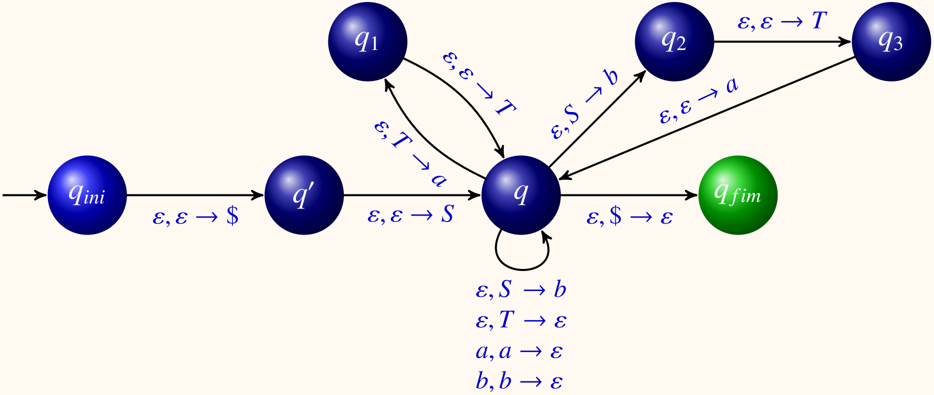
- Diagrama simplificado de estados para o PDA P:



Equivalência de GLC e PDA

Exemplo 1.25

- $GLC\ \ G = (V = \{S, T\}, \Sigma = \{a, b\}, R = \{S \rightarrow aTb \mid b,\ \ T \rightarrow Ta \mid \varepsilon\}, S)$.
- Diagrama de estados do PDA que simula G:



Linguagens livres de contexto e PDA's

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

- Seja $G = (V, \Sigma, P, S)$, na FNG, que aceita $\mathcal{L}$.
- Seja o PDA estendido $M = \langle \Sigma_M = \Sigma, \Gamma_M = V - \{S\}, S_M = \{s_0, s_1\}, s_0, \delta, F_M = \{s_1\} \rangle$, onde:

$$\begin{aligned}
\delta(s_0, a, \varepsilon) &= \{(s_1, w) \mid (S \rightarrow aw) \in P \text{ e } w \in V^*\},\\
\delta(s_1, a, A) &= \{(s_1, w) \mid (A \rightarrow aw) \in P,\ A \in V - \{S\} \text{ e } w \in V^*\},\\
\delta(s_0, \varepsilon, \varepsilon) &= \{(s_1, \varepsilon) \text{ se } (S \rightarrow \varepsilon) \in P\}.
\end{aligned}$$

$\square$

Linguagens livres de contexto e PDA's

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

1. $\mathcal{L} \subseteq \mathcal{L}(M)$.
2. $\mathcal{L}(M) \subseteq \mathcal{L}$.

$\square$

Linguagens livres de contexto e PDA's

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

1. $\mathcal{L} \subseteq \mathcal{L}(M)$.
   - Seja a derivação $S \stackrel{*}{\Longrightarrow} uw$, com $u \in \Sigma^+$ e $w \in V^*$.
   - Existe um processamento $[s_0, u, \varepsilon] \stackrel{*}{\longmapsto} [s_1, \varepsilon, w]$.

   (Indução no comprimento da derivação):
     - Base:
       Derivações $S \Longrightarrow aw$ de comprimento 1. A transição gerada pela regra $S \rightarrow aw$ é o processamento requerido.

$\square$

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

1. $\mathcal{L} \subseteq \mathcal{L}(M)$.
   - Seja a derivação $S \stackrel{*}{\Longrightarrow} uw$, com $u \in \Sigma^+$ e $w \in V^*$.
   - Existe um processamento $[s_0, u, \varepsilon] \stackrel{*}{\longmapsto} [s_1, \varepsilon, w]$.

   (Indução no comprimento da derivação):
     - Hipótese:

       Suponha que para todas cadeias uw geradas por derivações $S \stackrel{n}{\Longrightarrow} uw$ existe em M um processamento $[s_0, u, \varepsilon] \stackrel{*}{\longmapsto} [s_1, \varepsilon, w]$.

$\square$

Linguagens livres de contexto e PDA's

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

1. $\mathcal{L} \subseteq \mathcal{L}(M)$.
   - Passo indutivo:
     - Seja a derivação $S \overset{n+1}{\Longrightarrow} uw$, com $u = va \in \Sigma^+$ e $w \in V^*$.
     - $S \overset{n}{\Longrightarrow} vAw_2 \Longrightarrow uw$, onde $w = w_1 w_2$ e $(A \to aw_1) \in P$.
     - Por HI e $(s_1, w_1) \in \delta(s_1, a, A)$:

$$
\begin{aligned}
[s_0, va, \varepsilon] &\overset{*}{\longmapsto} [s_1, a, Aw_2] \\
&\longmapsto [s_1, \varepsilon, w_1 w_2].
\end{aligned}
$$

□

Linguagens livres de contexto e PDA's

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

1. $\mathcal{L} \subseteq \mathcal{L}(M)$.
   - Passo indutivo:
     - Para toda $u \in \mathcal{L}$, com $|u| > 0$, a aceitação de u por M é mostrada pelo processamento correspondente à derivação $S \overset{*}{\Longrightarrow} u$.
     - Se $\varepsilon \in \mathcal{L}$, então $(S \to \varepsilon) \in P$ e o processamento $[s_0, \varepsilon, \varepsilon] \longmapsto [s_1, \varepsilon, \varepsilon]$ aceita ε.

□

Linguagens livres de contexto e PDA's

Teorema 1.26

- *Se $\mathcal{L}$ é uma LLC, então existe um PDA M que aceita $\mathcal{L}$.*

Demonstração.

2. $\mathcal{L}(M) \subseteq \mathcal{L}$.
   - Mostrar que para todo processamento $[s_0, u, \varepsilon] \overset{*}{\longmapsto} [s_1, \varepsilon, w]$ existe a correspondente derivação $S \overset{*}{\Longrightarrow} uw$ em G.
     - Prova por indução.

□

Linguagens livres de contexto e PDA's

- Toda linguagem aceita por um PDA é uma LLC.
  - As regras de derivação da GLC são construídas a partir das transições do PDA.
  - A gramática é construída de modo que a aplicação de uma regra de derivação corresponda a uma transição no PDA.
- Seja o PDA $M = \langle \Sigma_M, \Gamma_M, S_M, s_0, \delta, F_M \rangle$. Um PDA estendido M' é construído a partir de M aumentando-se a função δ com as transições:
  1. $(s_j, \varepsilon) \in \delta(s_i, u, \varepsilon) \Rightarrow (s_j, X) \in \delta'(s_i, u, X), \ \forall\, X \in \Gamma.$
  2. $(s_j, Y) \in \delta(s_i, u, \varepsilon) \Rightarrow (s_j, YX) \in \delta'(s_i, u, X), \ \forall\, X \in \Gamma.$

Linguagens livres de contexto e PDA's

- A gramática $G = (V, \Sigma, P, S)$ é construída a partir das transições de M':
  - $\Sigma = \Sigma_{M'}$.
  - $V = \{S\} \cup \{\langle s_i, X, s_j \rangle\}$, onde $s_i, s_j \in S_{M'}$ e $X \in \Gamma \cup \{\varepsilon\}$.
    - $\langle s_i, X, s_j \rangle$: processamento em M' que inicia em s_i, encerra em s_j e desempilha X.

- A gramática $G = (V, \Sigma, P, S)$ é construída a partir das transições de M':
  - Conjunto P de regras de derivação:
    1. $S \rightarrow \langle s_0, \varepsilon, s_j \rangle, \ \forall \ s_j \in F_{M'}$,
    2. Cada transição $(s_j, Y) \in \delta'(s_i, u, X)$, onde $X, Y \in \Gamma \cup \{\varepsilon\}$, gera
    $$\{\langle s_i, X, s_k \rangle \rightarrow u\langle s_j, Y, s_k \rangle \mid s_k \in S_{M'}\},$$
    3. Cada transição $(s_j, YX) \in \delta'(s_i, u, X)$, onde $X, Y \in \Gamma$, gera
    $$\{\langle s_i, X, s_k \rangle \rightarrow u\langle s_j, Y, s_n \rangle\langle s_n, X, s_k \rangle \mid s_k, s_n \in S_{M'}\},$$
    4. $\langle s_k, \varepsilon, s_k \rangle \rightarrow \varepsilon, \ \forall \ s_k \in S_{M'}$.

Linguagens livres de contexto e PDA’s

- Uma derivação começa com uma regra do tipo 1:
  - O lado direito representa um processamento que começa no estado s_0 e termina em um estado final com pilha vazia.
  - Um processamento de sucesso no PDA M'.
- Regras do tipo 2 e 3 mapeiam as ações do PDA.
  - Regras do tipo 3 correspondem a transições estendidas de M', as quais aumentam o tamanho da pilha. O efeito na derivação é introduzir uma variável adicional.
- Regras do tipo 4 são usadas para terminar a derivação.
  - Representam um processamento a partir de um estado s_k para s_k que não altera a pilha (processamento nulo).

Linguagens livres de contexto e PDA's

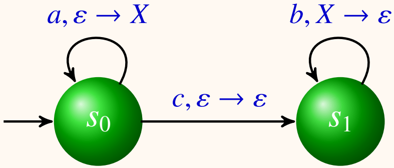
Exemplo 1.27

- $\mathcal{L} = \{a^n c b^n \mid n \geqslant 0\}$.

  - $M = \langle \Sigma_M, \Gamma_M, S_M, s_0, \delta, F_M \rangle$:

    - $\Sigma_M = \{a, b, c\}$;
    - $\Gamma_M = \{X\}$;
    - $S_M = \{s_0, s_1\}$;
    - $F_M = \{s_1\}$;

    - $\delta(s_0, a, \varepsilon) = \{(s_0, X)\}$;
    - $\delta(s_0, c, \varepsilon) = \{(s_1, \varepsilon)\}$;
    - $\delta(s_1, b, X) = \{(s_1, \varepsilon)\}$.



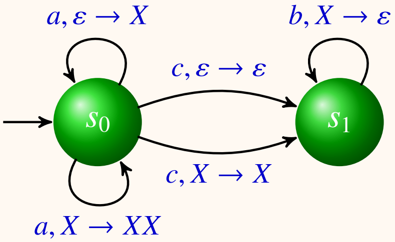
Linguagens livres de contexto e PDA's

Exemplo 1.27

- $\mathcal{L} = \{a^n c b^n \mid n \geqslant 0\}$.

  - $M' = \langle \Sigma_M, \Gamma_M, S_M, s_0, \delta', F_M \rangle$:

    - $\Sigma_M = \{a, b, c\}$;
    - $\Gamma_M = \{X\}$;
    - $S_M = \{s_0, s_1\}$;
    - $F_M = \{s_1\}$;
    - $\delta'(s_0, a, \varepsilon) = \{(s_0, X)\}$;
    - $\delta'(s_0, c, \varepsilon) = \{(s_1, \varepsilon)\}$;
    - $\delta'(s_1, b, X) = \{(s_1, \varepsilon)\}$;
    - $\delta'(s_0, a, X) = \{(s_0, XX)\}$;
    - $\delta'(s_0, c, X) = \{(s_1, X)\}$.



Linguagens livres de contexto e PDA's

Exemplo 1.27

- $\mathcal{L} = \{a^n c b^n \mid n \geqslant 0\}$.
- $G = (V, \Sigma, P, S)$:
  - $\Sigma = \Sigma_M$.
  - $V = \{S\} \cup \{\langle s_i, X, s_j \rangle\}$, onde $s_i, s_j \in S_M$ e $X \in \Gamma \cup \{\varepsilon\}$.

Linguagens livres de contexto e PDA's

Exemplo 1.27

- $\mathcal{L} = \{a^n c b^n \mid n \geqslant 0\}$.

Transições	Regras de derivação
	$S \rightarrow \langle s_0, \varepsilon, s_0 \rangle$
	$S \rightarrow \langle s_0, \varepsilon, s_1 \rangle$
$\delta'(s_0, a, \varepsilon) = \{(s_0, X)\}$	$\langle s_0, \varepsilon, s_0 \rangle \rightarrow a \langle s_0, X, s_0 \rangle$
	$\langle s_0, \varepsilon, s_1 \rangle \rightarrow a \langle s_0, X, s_1 \rangle$
$\delta'(s_0, c, \varepsilon) = \{(s_1, \varepsilon)\}$	$\langle s_0, \varepsilon, s_0 \rangle \rightarrow c \langle s_1, \varepsilon, s_0 \rangle$
	$\langle s_0, \varepsilon, s_1 \rangle \rightarrow c \langle s_1, \varepsilon, s_1 \rangle$
$\delta'(s_0, c, X) = \{(s_1, X)\}$	$\langle s_0, X, s_0 \rangle \rightarrow c \langle s_1, X, s_0 \rangle$
	$\langle s_0, X, s_1 \rangle \rightarrow c \langle s_1, X, s_1 \rangle$
$\delta'(s_1, b, X) = \{(s_1, \varepsilon)\}$	$\langle s_1, X, s_0 \rangle \rightarrow b \langle s_1, \varepsilon, s_0 \rangle$
	$\langle s_1, X, s_1 \rangle \rightarrow b \langle s_1, \varepsilon, s_1 \rangle$
$\delta'(s_0, a, X) = \{(s_0, XX)\}$	$\langle s_0, X, s_0 \rangle \rightarrow a \langle s_0, X, s_0 \rangle \langle s_0, X, s_0 \rangle$
	$\langle s_0, X, s_0 \rangle \rightarrow a \langle s_0, X, s_1 \rangle \langle s_1, X, s_0 \rangle$
	$\langle s_0, X, s_1 \rangle \rightarrow a \langle s_0, X, s_0 \rangle \langle s_0, X, s_1 \rangle$
	$\langle s_0, X, s_1 \rangle \rightarrow a \langle s_0, X, s_1 \rangle \langle s_1, X, s_1 \rangle$
	$\langle s_0, \varepsilon, s_0 \rangle \rightarrow \varepsilon$
	$\langle s_1, \varepsilon, s_1 \rangle \rightarrow \varepsilon$

Linguagens livres de contexto e PDA's

Exemplo 1.27

- $\mathcal{L} = \{a^n c b^n \mid n \geqslant 0\}$.

Variável	Variável original
A	$\langle s_0, \varepsilon, s_0\rangle$
B	$\langle s_0, \varepsilon, s_1\rangle$
C	$\langle s_1, \varepsilon, s_0\rangle$
D	$\langle s_1, \varepsilon, s_1\rangle$
E	$\langle s_0, X, s_0\rangle$
F	$\langle s_0, X, s_1\rangle$
G	$\langle s_1, X, s_0\rangle$
H	$\langle s_1, X, s_1\rangle$

Linguagens livres de contexto e PDA's

Exemplo 1.27

- $\mathcal{L} = \{a^n c b^n \mid n \geqslant 0\}$.

Transições	Regras de derivação
	$S \to A$ $S \to B$
$\delta(s_0, a, \varepsilon) = \{(s_0, X)\}$	$A \to aE$ $B \to aF$
$\delta(s_0, c, \varepsilon) = \{(s_1, \varepsilon)\}$	$A \to cC$ $B \to cD$
$\delta(s_0, c, X) = \{(s_1, X)\}$	$E \to cG$ $F \to cH$
$\delta(s_1, b, X) = \{(s_1, \varepsilon)\}$	$G \to bC$ $H \to bD$
$\delta(s_0, a, X) = \{(s_0, XX)\}$	$E \to aEE$ $E \to aFG$ $F \to aEF$ $F \to aFH$
	$A \to \varepsilon$ $D \to \varepsilon$

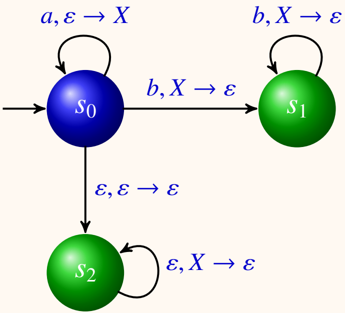
Exemplo 1.27

- $\mathcal{L} = \{a^n cb^n \mid n \geqslant 0\}$.
- $G = (V, \Sigma, P, S)$:
  - $V = \{S, B, D, F, H\} \equiv \{S, F\}$.
  - $\Sigma = \{a, b, c\}$.
  - $P = \left\{ \begin{array}{l} S \to A \mid B, \\ A \to \varepsilon, \\ B \to aF \mid cD, \\ D \to \varepsilon, \\ F \to aFH \mid cH, \\ H \to bD \end{array} \right\} \equiv \left\{ \begin{array}{l} S \to \varepsilon \mid aF \mid c, \\ F \to aFb \mid cb \end{array} \right\}$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.



Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $M = \langle \Sigma_M, \Gamma_M, S_M, s_0, \delta, F_M \rangle$:
  - $\Sigma_M = \{a, b\}$;
  - $\Gamma_M = \{X\}$;
  - $S_M = \{s_0, s_1, s_2\}$;
  - $F_M = \{s_1, s_2\}$;
  - $\delta(s_0, a, \varepsilon) = \{(s_0, X)\}$;
  - $\delta(s_0, b, X) = \{(s_1, \varepsilon)\}$;
  - $\delta(s_0, \varepsilon, \varepsilon) = \{(s_2, \varepsilon)\}$;
  - $\delta(s_1, b, X) = \{(s_1, \varepsilon)\}$;
  - $\delta(s_2, \varepsilon, X) = \{(s_2, \varepsilon)\}$.

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $M' = \langle \Sigma_M, \Gamma_M, S_M, s_0, \delta', F_M \rangle$:
  - $\Sigma_M = \{a, b\}$;
  - $\Gamma_M = \{X\}$;
  - $S_M = \{s_0, s_1, s_2\}$;
  - $F_M = \{s_1, s_2\}$;

  - $\delta'(s_0, a, \varepsilon) = \{(s_0, X)\}$;
  - $\delta'(s_0, b, X) = \{(s_1, \varepsilon)\}$;
  - $\delta'(s_0, \varepsilon, \varepsilon) = \{(s_2, \varepsilon)\}$;
  - $\delta'(s_1, b, X) = \{(s_1, \varepsilon)\}$;
  - $\delta'(s_2, \varepsilon, X) = \{(s_2, \varepsilon)\}$;

  - $\delta'(s_0, a, X) = \{(s_0, XX)\}$;
  - $\delta'(s_0, \varepsilon, X) = \{(s_2, X)\}$.

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $G = (V, \Sigma, P, S)$:
  - $\Sigma = \Sigma_M$.
  - $V = \{S\} \cup \{\langle s_i, X, s_j \rangle\}$, onde $s_i, s_j \in S_M$ e $X \in \Gamma \cup \{\varepsilon\}$.

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Transições	Regras de derivação
	$S \to \langle s_0, \varepsilon, s_1\rangle$
	$S \to \langle s_0, \varepsilon, s_2\rangle$
$\delta'(s_0, a, \varepsilon) = \{(s_0, X)\}$	$\langle s_0, \varepsilon, s_0\rangle \to a\langle s_0, X, s_0\rangle$
	$\langle s_0, \varepsilon, s_1\rangle \to a\langle s_0, X, s_1\rangle$
	$\langle s_0, \varepsilon, s_2\rangle \to a\langle s_0, X, s_2\rangle$
$\delta'(s_0, b, X) = \{(s_1, \varepsilon)\}$	$\langle s_0, X, s_0\rangle \to b\langle s_1, \varepsilon, s_0\rangle$
	$\langle s_0, X, s_1\rangle \to b\langle s_1, \varepsilon, s_1\rangle$
	$\langle s_0, X, s_2\rangle \to b\langle s_1, \varepsilon, s_2\rangle$
$\delta'(s_0, \varepsilon, \varepsilon) = \{(s_2, \varepsilon)\}$	$\langle s_0, \varepsilon, s_0\rangle \to \varepsilon\langle s_2, \varepsilon, s_0\rangle$
	$\langle s_0, \varepsilon, s_1\rangle \to \varepsilon\langle s_2, \varepsilon, s_1\rangle$
	$\langle s_0, \varepsilon, s_2\rangle \to \varepsilon\langle s_2, \varepsilon, s_2\rangle$
$\delta'(s_0, \varepsilon, X) = \{(s_2, X)\}$	$\langle s_0, X, s_0\rangle \to \varepsilon\langle s_2, X, s_0\rangle$
	$\langle s_0, X, s_1\rangle \to \varepsilon\langle s_2, X, s_1\rangle$
	$\langle s_0, X, s_2\rangle \to \varepsilon\langle s_2, X, s_2\rangle$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Transições	Regras de derivação
$\delta'(s_0, a, X) = \{(s_0, XX)\}$	$\langle s_0, X, s_0\rangle \to a\langle s_0, X, s_0\rangle\langle s_0, X, s_0\rangle$
	$\langle s_0, X, s_1\rangle \to a\langle s_0, X, s_0\rangle\langle s_0, X, s_1\rangle$
	$\langle s_0, X, s_2\rangle \to a\langle s_0, X, s_0\rangle\langle s_0, X, s_2\rangle$
	$\langle s_0, X, s_0\rangle \to a\langle s_0, X, s_1\rangle\langle s_1, X, s_0\rangle$
	$\langle s_0, X, s_1\rangle \to a\langle s_0, X, s_1\rangle\langle s_1, X, s_1\rangle$
	$\langle s_0, X, s_2\rangle \to a\langle s_0, X, s_1\rangle\langle s_1, X, s_2\rangle$
	$\langle s_0, X, s_0\rangle \to a\langle s_0, X, s_2\rangle\langle s_2, X, s_0\rangle$
	$\langle s_0, X, s_1\rangle \to a\langle s_0, X, s_2\rangle\langle s_2, X, s_1\rangle$
	$\langle s_0, X, s_2\rangle \to a\langle s_0, X, s_2\rangle\langle s_2, X, s_2\rangle$

Linguagens livres de contexto e PDA’s

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Transições	Regras de derivação
$\delta'(s_1, b, X) = \{(s_1, \varepsilon)\}$	$\langle s_1, X, s_0\rangle \to b\langle s_1, \varepsilon, s_0\rangle$
	$\langle s_1, X, s_1\rangle \to b\langle s_1, \varepsilon, s_1\rangle$
	$\langle s_1, X, s_2\rangle \to b\langle s_1, \varepsilon, s_2\rangle$
$\delta'(s_2, \varepsilon, X) = \{(s_2, \varepsilon)\}$	$\langle s_2, X, s_0\rangle \to \varepsilon\langle s_2, \varepsilon, s_0\rangle$
	$\langle s_2, X, s_1\rangle \to \varepsilon\langle s_2, \varepsilon, s_1\rangle$
	$\langle s_2, X, s_2\rangle \to \varepsilon\langle s_2, \varepsilon, s_2\rangle$
	$\langle s_0, \varepsilon, s_0\rangle \to \varepsilon$
	$\langle s_1, \varepsilon, s_1\rangle \to \varepsilon$
	$\langle s_2, \varepsilon, s_2\rangle \to \varepsilon$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Variável	Variável original
A	$\langle s_0, \varepsilon, s_0 \rangle$
B	$\langle s_0, \varepsilon, s_1 \rangle$
C	$\langle s_0, \varepsilon, s_2 \rangle$
D	$\langle s_1, \varepsilon, s_0 \rangle$
E	$\langle s_1, \varepsilon, s_1 \rangle$
F	$\langle s_1, \varepsilon, s_2 \rangle$
G	$\langle s_2, \varepsilon, s_0 \rangle$
H	$\langle s_2, \varepsilon, s_1 \rangle$
I	$\langle s_2, \varepsilon, s_2 \rangle$

Variável	Variável original
J	$\langle s_0, X, s_0 \rangle$
K	$\langle s_0, X, s_1 \rangle$
L	$\langle s_0, X, s_2 \rangle$
M	$\langle s_1, X, s_0 \rangle$
N	$\langle s_1, X, s_1 \rangle$
O	$\langle s_1, X, s_2 \rangle$
P	$\langle s_2, X, s_0 \rangle$
Q	$\langle s_2, X, s_1 \rangle$
R	$\langle s_2, X, s_2 \rangle$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Transições	Regras de derivação
	$S \to B$
	$S \to C$
$\delta(s_0, a, \varepsilon) = \{(s_0, X)\}$	$A \to aJ$
	$B \to aK$
	$C \to aL$
$\delta(s_0, b, X) = \{(s_1, \varepsilon)\}$	$J \to bD$
	$K \to bE$
	$L \to bF$
$\delta(s_0, \varepsilon, \varepsilon) = \{(s_2, \varepsilon)\}$	$A \to G$
	$B \to H$
	$C \to I$
$\delta(s_0, \varepsilon, X) = \{(s_2, X)\}$	$J \to P$
	$K \to Q$
	$L \to R$

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Transições	Regras de derivação
$\delta(s_0, a, X) = \{(s_0, XX)\}$	$J \to aJJ$
	$K \to aJK$
	$L \to aJL$
	$J \to aKM$
	$K \to aKN$
	$L \to aKO$
	$J \to aLP$
	$K \to aLQ$
	$L \to aLR$

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.

Transições	Regras de derivação
$\delta(s_1, b, X) = \{(s_1, \varepsilon)\}$	$M \rightarrow bD$
	$N \rightarrow bE$
	$O \rightarrow bF$
$\delta(s_2, \varepsilon, X) = \{(s_2, \varepsilon)\}$	$P \rightarrow G$
	$Q \rightarrow H$
	$R \rightarrow I$
	$A \rightarrow \varepsilon$
	$E \rightarrow \varepsilon$
	$I \rightarrow \varepsilon$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $G = (V = \{S, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R\}, \Sigma = \{a, b\}, P, S)$:

$$P = \left\{ \begin{array}{l} S \to B \mid C, \\ A \to aJ \mid G \mid \varepsilon, \\ B \to aK \mid H, \\ C \to aL \mid I, \\ E \to \varepsilon, \\ I \to \varepsilon, \\ J \to bD \mid P \mid aJJ \mid aKM \mid aLP, \\ K \to bE \mid Q \mid aJK \mid aKN \mid aLQ, \\ L \to bF \mid R \mid aJL \mid aKO \mid aLR, \\ M \to bD, \\ N \to bE, \\ O \to bF, \\ P \to G, \\ Q \to H, \\ R \to I \end{array} \right\}$$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $G_1 = (V_1 = \{S, B, C, E, I, J, K, L, N, R\}, \Sigma = \{a, b\}, P_1, S)$:

$$P_1 = \left\{ \begin{array}{l} S \rightarrow B \mid C, \\ B \rightarrow aK, \\ C \rightarrow aL \mid I, \\ E \rightarrow \varepsilon, \\ I \rightarrow \varepsilon, \\ J \rightarrow aJJ, \\ K \rightarrow bE \mid aJK \mid aKN, \\ L \rightarrow R \mid aJL \mid aLR, \\ N \rightarrow bE, \\ R \rightarrow bI \end{array} \right\}$$

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $G_2 = (V_2 = \{S, B, C, E, I, K, L, N, R\}, \Sigma = \{a, b\}, P_2, S)$:

$$P_2 = \left\{ \begin{array}{l} S \rightarrow B \mid C, \\ B \rightarrow aK, \\ C \rightarrow aL \mid I, \\ E \rightarrow \varepsilon, \\ I \rightarrow \varepsilon, \\ K \rightarrow bE \mid aKN, \\ L \rightarrow R \mid aLR, \\ N \rightarrow bE, \\ R \rightarrow I \end{array} \right\}$$

Linguagens livres de contexto e PDA's

Exemplo 1.28

- $\mathcal{L} = \{a^i \mid i \geqslant 0\} \cup \{a^i b^i \mid i \geqslant 0\}$.
- $G_3 = (V_3 = \{S, K, L\}, \Sigma = \{a, b\}, P_3, S)$:

$$P_3 = \left\{ \begin{array}{l} S \rightarrow aK \mid aL \mid \varepsilon, \\ K \rightarrow b \mid aKb, \\ L \rightarrow \varepsilon \mid aL \end{array} \right\}$$

Equivalência com *GLC*

Corolário 1.29

- *Toda linguagem regular é livre de contexto.*

Demonstração.

- Toda linguagem regular é reconhecida por um autômato finito.
- Todo autômato finito é um autômato com pilha que simplesmente ignora a sua pilha.
- Toda linguagem regular é também livre de contexto.

Linguagens **L**ivres de **C**ontexto
(*Autômato com Pilha – PDA*)

Linguagens **R**egulares
(*Autômato Finito – DFA*)

□

Livros texto

R. P. Grimaldi
Discrete and Combinatorial Mathematics – An Applied Introduction.
Addison Wesley, 1994.

D. J. Velleman
How To Prove It – A Structured Approach.
Cambridge University Press, 1996.

J. E. Hopcroft; J. Ullman.
Introdução à Teoria de Autômatos, Linguagens e Computação.
Ed. Campus.

T. A. Sudkamp.
Languages and Machines – An Introduction to the Theory of Computer Science.
Addison Wesley Longman, Inc. 1998.

J. Carroll; D. Long.
Theory of Finite Automata – With an Introduction to Formal Languages.
Prentice-Hall, 1989.

M. Sipser.
Introduction to the Theory of Computation.
PWS Publishing Company, 1997.

H. R. Lewis; C. H. Papadimitriou
Elementos de Teoria da Computação.
Bookman, 2000.