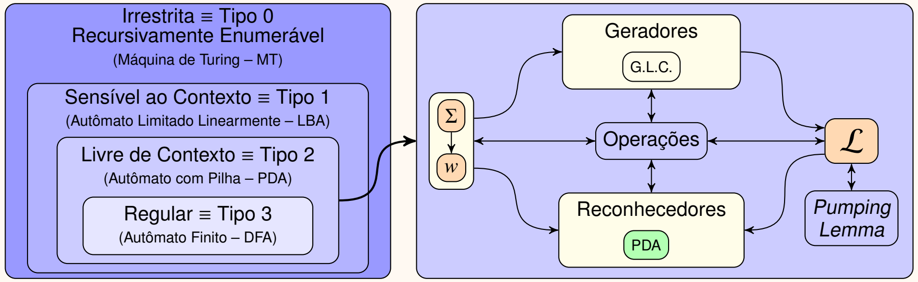
Linguagens Formais e Autômatos

Humberto Longo

Instituto de Informática
Universidade Federal de Goiás

Bacharelado em Ciência da Computação, 2021/1

Roteiro



PDA atômico

Definição 1.14

- 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 1.15

- *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 1.15

- *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.

□

Transição estendida

Definição 1.16

- 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 1.17

- *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 em P deve ser trocada por uma sequência de transições.
  - Por exemplo, dada a transição $(s_j, bcd) \in \delta(s_i, u, a)$ de P, em 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 1.18

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

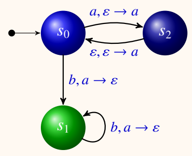
PDA	PDA atômico	PDA estendido
$S = \{s_0, s_1, s_2\}$	$S = \{s_0, s_1, s_2, s_3, s_4\}$	$S = \{s_0, s_1\}$
$\delta(s_0, a, \varepsilon) = \{(s_2, a)\}$	$\delta(s_0, a, \varepsilon) = \{(s_3, \varepsilon)\}$	$\delta(s_0, a, \varepsilon) = \{(s_0, aa)\}$
$\delta(s_2, \varepsilon, \varepsilon) = \{(s_0,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

Transição estendida

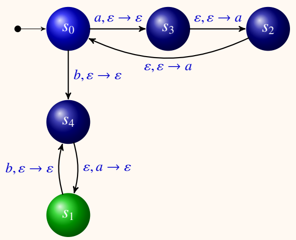
Exemplo 1.18

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

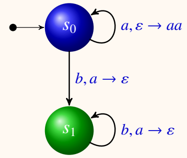
- PDA:



- PDA atômico:



- PDA estendido:



Aceitação por estado final

Definição 1.19

- 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 1.20

- *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 1.20

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

$$[s_0, w, \varepsilon] \overset{*}{\underset{P}{\longmapsto}} [s_i, \varepsilon, \alpha] \underset{P'}{\longmapsto} [s_f, \varepsilon, \alpha] \overset{*}{\underset{P'}{\longmapsto}} [s_f, \varepsilon, \varepsilon].$$

□

Aceitação por estado final

Lema 1.20

- *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 1.21

- 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 1.22

- *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'.

□

Aceitação por pilha vazia

Lema 1.22

- *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, $\mathcal{L}(P') = \mathcal{L}_E(P)$.

□

Linguagens aceitas por PDA's

Teorema 1.23

- *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 $\mathcal{L}_E(P_2) = \mathcal{L}(P)$.*

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 Ã 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.