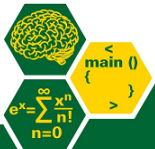




Universidade Federal do ABC

CMCC

Centro de Matemática, Computação e Cognição



Teoria da Computação

Autômatos de Pilha

Mirtha Lina Fernández Venero
mirtha.lina@ufabc.edu.br

outubro 2017



Sumário

Autômatos de Pilha

Equivalência de GLCs e APs

Linguagens livres e não livres de contexto

Bibliografia

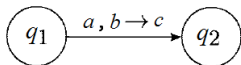
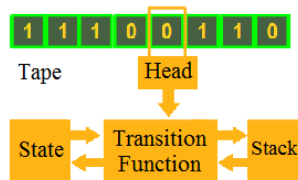
Autômatos de Pilha

Inventado por Anthony Gervin Oettinger in 1961



Definição: $\mathbf{A} = (\mathbf{Q}, \Sigma, \Gamma, q_0, \delta, F)$

- ▶ Q conjunto *finito* de estados
- ▶ Σ alfabeto de entrada
- ▶ Γ alfabeto da pilha
- ▶ $q_0 \in Q$ estado inicial
- ▶ $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow \mathcal{P}(Q \times \Gamma_\epsilon)$ função de transição de estados



- ▶ $F \subseteq Q$ conjunto de estados finais ou de aceitação



Exemplo de autômato de pilha determinístico

$$\mathbf{A1} = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$$

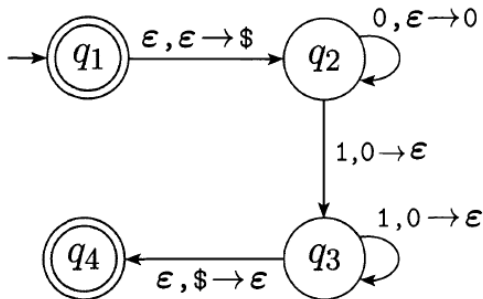
$$\delta(q_1, \epsilon, \epsilon) = \{(q_2, \$)\}$$

$$\delta(q_2, 0, \epsilon) = \{(q_2, 0)\}$$

$$\delta(q_2, 1, 0) = \{(q_3, \epsilon)\}$$

$$\delta(q_3, 1, 0) = \{(q_3, \epsilon)\}$$

$$\delta(q_3, \epsilon, \$) = \{(q_4, \epsilon)\}$$





Autômatos de Pilha

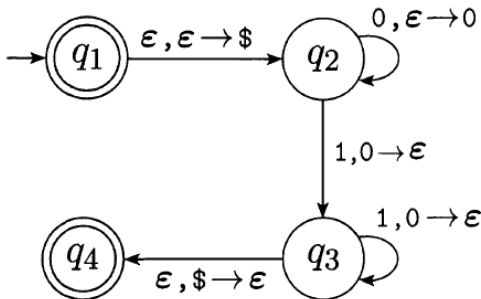
Características

- ▶ A memória do autômato é caracterizada por seus estados e uma pilha infinita
- ▶ A configuração é representada pelo estado atual, a porção não lida da entrada e o conteúdo da pilha
- ▶ Em cada passo, o autômato pode ler ou não **um** símbolo da entrada e empilhar/desempilhar ou não **um** símbolo da pilha

Processo de Reconhecimento

- ▶ Começar pelo **estado inicial** com a pilha vazia
- ▶ Ler cada símbolo da cadeia e mudar a configuração dependendo de δ até chegar no final da cadeia ou não ter como avançar. A cadeia é reconhecida se após ler todos seus símbolos, o último estado é final

Exemplo de reconhecimento da cadeia 000111

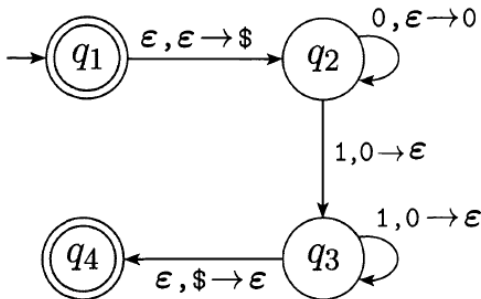


$(q_1, 000111, \epsilon) \rightarrow (q_2, 000111, \$) \rightarrow (q_2, 00111, 0\$) \rightarrow$

$(q_2, 0111, 00\$) \rightarrow (q_2, 111, 000\$) \rightarrow (q_3, 11, 00\$) \rightarrow$

$(q_3, 1, 0\$) \rightarrow (q_3, \epsilon, \$) \rightarrow (q_4, \epsilon, \epsilon)$ **A1 aceita a cadeia!**

Exemplo de reconhecimento da cadeia 000101



$(q_1, 000101, \epsilon) \rightarrow (q_2, 000101, \$) \rightarrow (q_2, 00101, 0\$) \rightarrow$

$(q_2, 0101, 00\$) \rightarrow (q_2, 101, 000\$) \rightarrow (q_3, 01, 00\$)$

A1 não aceita a cadeia!



Exemplo de autômato de pilha não determinístico

$$A2 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, \$\}, \delta, q_1, \{q_1, q_4\})$$

$$\delta(q_1, \epsilon, \epsilon) = \{(q_2, \$)\}$$

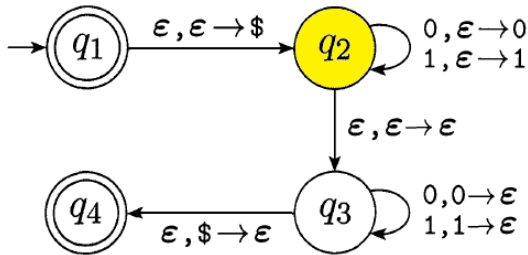
$$\delta(q_2, 0, \epsilon) = \{(q_2, 0)\}$$

$$\delta(q_2, 1, \epsilon) = \{(q_2, \epsilon)\}$$

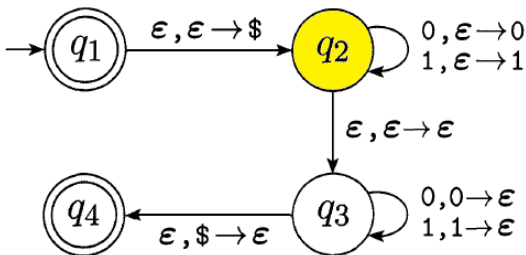
$$\delta(q_2, \epsilon, \epsilon) = \{(q_3, \epsilon)\}$$

$$\delta(q_3, 0, 0) = \{(q_3, \epsilon)\}$$

$$\delta(q_3, 1, 1) = \{(q_3, \epsilon)\} \quad \delta(q_3, \epsilon, \$) = \{(q_4, \epsilon)\}$$



Exemplo de reconhecimento da cadeia 00



$(q_1, 00, \epsilon) \rightarrow (q_2, 00, \$) \rightarrow$

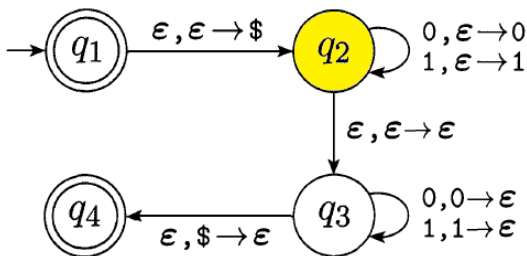
▶ $(q_3, 00, \$) \rightarrow (q_4, 00, \epsilon)$ **Não!**

▶ $(q_2, 0, 0\$) \rightarrow$

▶ $(q_2, \epsilon, 00\$) \rightarrow (q_3, \epsilon, 00\$)$ **Não!**

▶ $(q_3, 0, 0\$) \rightarrow (q_3, \epsilon, \$) \rightarrow (q_4, \epsilon, \epsilon)$ **OK! A2 aceita a cadeia**

Exemplo de reconhecimento da cadeia 01



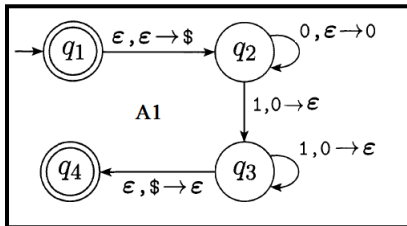
$(q_1, 01, \epsilon) \rightarrow (q_2, 01, \$) \rightarrow$

- ▶ $(q_3, 01, \$) \rightarrow (q_4, 01, \epsilon)$ **Não!**
- ▶ $(q_2, 1, 0\$) \rightarrow$
 - ▶ $(q_2, \epsilon, 10\$) \rightarrow (q_3, \epsilon, 10\$)$ **Não!**
 - ▶ $(q_3, 1, 0\$)$ **Não!**

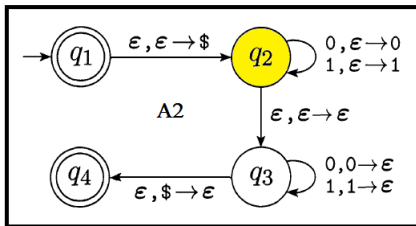
A2 não aceita a cadeia

Linguagem aceita por um autômato de pilha A

O conjunto de todas as cadeias reconhecidas ou aceitas por A



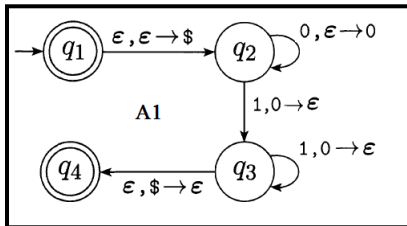
$$L(A1) = \{0^n 1^n \mid n \geq 0\}$$



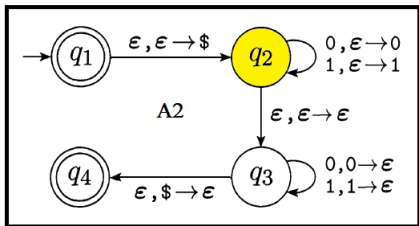
$$L(A2) = \{ww^r \mid w \in (0 + 1)^*\}$$

Linguagem aceita por um autômato de pilha A

O conjunto de todas as cadeias reconhecidas ou aceitas por A



$$L(A1) = \{0^n 1^n \mid n \geq 0\}$$

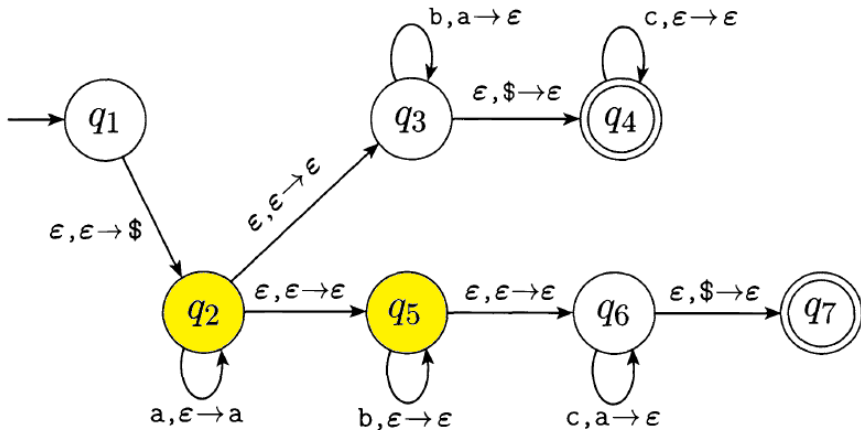


$$L(A2) = \{ww^r \mid w \in (0+1)^*\}$$

Exercício: Escreva AP para reconhecer as seguintes linguagens

- ▶ $L(A3) = \{a^n b^m \mid m > n \geq 0\} \cup \{a^n c^{2n} \mid n \geq 0\}$
- ▶ $L(A4) = \{w \mid w \in (a|b|c)^*, w = w^r\}$

Qual é a linguagem que reconhece o seguinte AP?





Sumário

Autômatos de Pilha

Equivalência de GLCs e APs

Linguagens livres e não livres de contexto

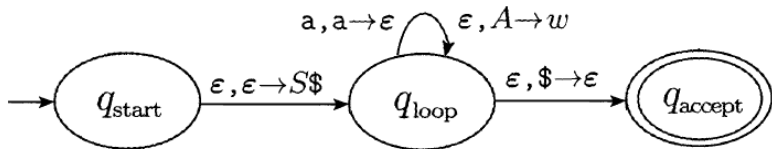
Bibliografia

Equivalência de GLCs e APs

Uma linguagem é livre de contexto se e somente se ela é reconhecida por algum autômato de pilha

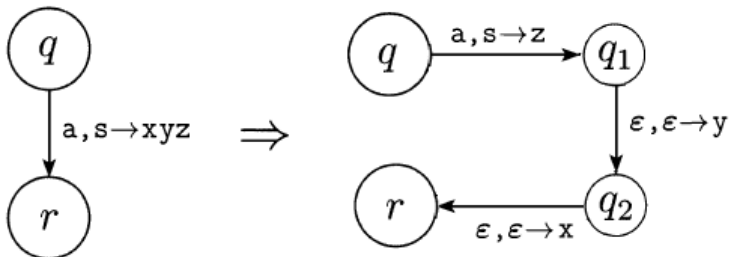
Ideia da Prova: \Rightarrow Se uma linguagem L é livre de contexto então existe uma GLC que a gera. Precisamos construir um AP (não determinístico), a partir da GLC, que reconheça a linguagem.

- ▶ o alfabeto de entrada da GLC e AP coincidem
- ▶ O alfabeto da pilha inclui todos os símbolos terminais e não terminais da GLC além de $\$$
- ▶ Usar um estado com loops para cada regra e símbolo terminal

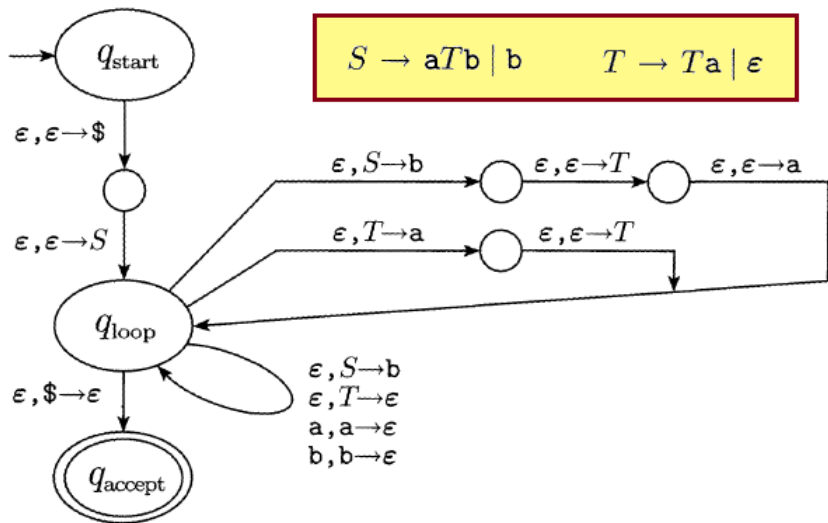


Equivalência de GLCs e APs

- ▶ acrescentar estados intermediários para definir de forma correta δ



Equivalência de GLCs e APs - Exemplo





Sumário

Autômatos de Pilha

Equivalência de GLCs e APs

Linguagens livres e não livres de contexto

Bibliografia



Lema do bombeamento (Pumping Lemma)

Se L é uma linguagem livre de contexto, então existe um número p (comprimento do bombeamento) tal que se s é uma cadeia de tamanho pelo menos p , então ela pode ser dividida em cinco partes $s = uvxyz$ que satisfazem as seguintes condições

- (1) para todo $i \geq 0$, $uv^i xy^i z \in L$
- (2) $|vy| > 0$
- (3) $|vxy| \leq p$



Lema do bombeamento (Pumping Lemma)

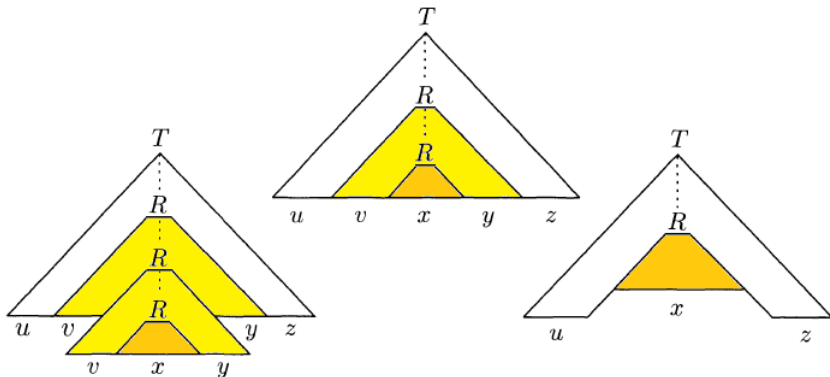
Se L é uma linguagem livre de contexto, então existe um número p (comprimento do bombeamento) tal que se s é uma cadeia de tamanho pelo menos p , então ela pode ser dividida em cinco partes $s = uvxyz$ que satisfazem as seguintes condições

- (1) para todo $i \geq 0$, $uv^i xy^i z \in L$
- (2) $|vy| > 0$
- (3) $|vxy| \leq p$

Ideia da Prova: Como achar o p ? Se L é livre de contexto, existe uma GLC que gera L . Seja b o número máximo de símbolos na parte direita duma regra da gramática. Se a árvore sintática duma cadeia gerada tem altura h então a árvore tem b^h folhas no máximo. Escolhendo $p = b^{|N|} + 1$ toda cadeia s tal que $|s| \geq p$ terá árvores de altura no mínimo $|N| + 1$

Lema do bombeamento (Pumping Lemma)

Ideia da Prova: Se a **menor** árvore de s tem altura no mínimo $|N| + 1$ então ela tem um caminho onde algum não terminal está repetido, garantindo (1). Escolher a menor árvore garante (2) enquanto escolher as ocorrências mais baixas de R garante (3).





Usando o Lema do bombeamento

Para provar que uma linguagem L não é livre de contexto podemos achar uma cadeia $s \in L$ que não pode ser bombeada.

Exemplo: Prove que $L = \{a^n b^n c^n \mid n \geq 0\}$ não é LC

Prova: Suponhamos que L é livre de contexto. Pelo lema existe um comprimento de bombeamento p . Escolhemos $s = a^p b^p c^p$ e tentamos dividir $s = uvxyz$ para bombear. Por (3), $|t = vxy| \leq p$ e portanto t tem um ou dois tipos de símbolos.

- ▶ Se todos os símbolos de t são iguais então $uv^2xy^2z \notin L$ pois tem um símbolo com quatro ocorrências a mais.
- ▶ Se v (ou y) tem mais dum tipo de símbolo então $uv^2xy^2z \notin a^* b^* c^*$
- ▶ Se x tem mais dum tipo de símbolo então $uv^0xy^0z \notin L$ pois tem um símbolo com duas ocorrências a mais.



Usando do Lema do bombeamento

Para provar que uma linguagem L não é livre de contexto podemos achar uma cadeia $s \in L$ que não pode ser bombeada.

Exemplo: Prove que $L = \{ ww \mid w \in (0 + 1)^* \}$ não é LC

Prova: Suponhamos que L é livre de contexto. Então pelo lema existe um comprimento do bombeamento p . Escolhemos $s = 0^p 1^p 0^p 1^p$ e tentamos dividir $s = uvxyz$ com $|t = vxy| \leq p$.

- ▶ Se todos os símbolos de t são iguais então $uv^2xy^2z \notin L$ pois ao dividir s em duas metades o primeiro ou o último símbolo serão diferentes. O mesmo acontece se t tem símbolos diferentes mas está numa das metades.
- ▶ Se t está no centro, v e y têm símbolos diferentes. Então $uv^0xy^0z = 0^p 1^i 0^j 1^p \notin L$ pois usando (2), podemos concluir que $i \neq p$ ou $j \neq p$.



Algumas propriedades

As linguagens livres de contexto são fechadas para as seguintes operações:

- ▶ união e concatenação; **porém não intersecção**
- ▶ intersecção ou diferença com uma linguagem regular
- ▶ fechamento de Kleene ($*$) e fechamento positivo ($+$)
- ▶ reverso e homomorfismos

Não existe algoritmo para:

- ▶ determinar se uma GLC é ambígua
- ▶ determinar se uma LLC é inerentemente ambígua
- ▶ determinar se duas LLCs são a mesma
- ▶ determinar se a intersecção de duas LLCs é vazia



Sumário

Autômatos de Pilha

Equivalência de GLCs e APs

Linguagens livres e não livres de contexto

Bibliografia

Bibliografia Básica

1. **M. Sipser. Introduction to the Theory of Computation, Cengage Learning 3rd ed, 2012**
2. **J. E. Hopcroft, R. Motwani and J. D. Ullman. Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 2001**
3. **Compilers: Principles, Techniques, and Tools (2nd Edition). Alfred Aho, Monica Lam, Ravi Sethi, and Jeffrey Ullman. Addison-Wesley, 2006**

