

Linguagens Formais

UNICAP

Eduardo Araújo Oliveira
<http://sites.google.com/site/eaoufpe>



Estrutura

- Introdução
- Definição Formal
- Aceita / Rejeita / Loop
- Conversão

Linguagens Livres de Contexto

- Vimos um formalismo gerador
 - Gramáticas Livres de Contexto
- Veremos hoje um formalismo reconhecedor
 - Autômato com Pilha

3

Introdução

- **autômato de pilha (AP)**
 - mais poderoso do que os autômatos finitos
 - capaz de reconhecer a classe das **linguagens livres de contexto** (que inclui toda a classe das linguagens regulares)

slide 4

Introdução

O autômato de pilha é como um modelo de poder **equivalente** ao das gramáticas livres de contexto

Porém, o princípio de funcionamento do autômato de pilha é mais parecido com o dos autômatos finitos

slide 5

Funcionamento

Os autômatos com pilha (APs) são como os autômatos finitos com transições vazias (AFND-e), mas têm um componente extra chamado **pilha**

A pilha serve como uma memória adicional infinita (por isso não podemos chamar um autômato de pilha de autômato "finito")

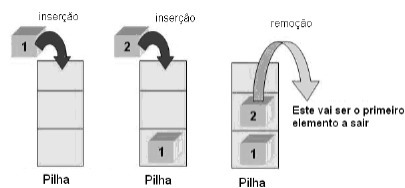
slide 6

Funcionamento

o autômato pode escrever símbolos na pilha para ler mais tarde e isso dá ao autômato um poder adicional em relação aos AFND-e

slide 7

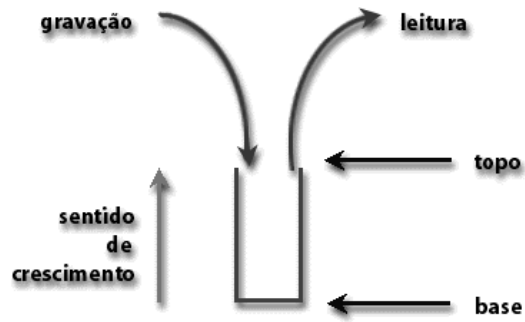
Pilha



Os AP usam uma pilha para guardar símbolos especiais, que chamaremos **símbolos da pilha**

slide 8

Pilha



slide 9

Funcionamento

A pilha dos APs não inicia vazia – ela começa automaticamente com um **símbolo de início**

Nos APs, o primeiro símbolo da pilha **serve para decidir para qual estado o autômato mudará**

slide 10

Funcionamento

No AP, as mudanças de estado vão depender de um **par de condições**:

- o próximo símbolo da entrada
- o próximo símbolo da pilha

Semelhante aos AFND-e, também podemos usar o símbolo especial para indicar que não precisa ler símbolo (da entrada ou da pilha) para mudar de estado

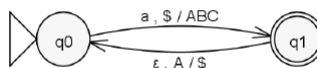
slide 11

Funcionamento

tripas chamadas de **descrições instantâneas ou IDs**
(do inglês *instantaneous description*) do autômato

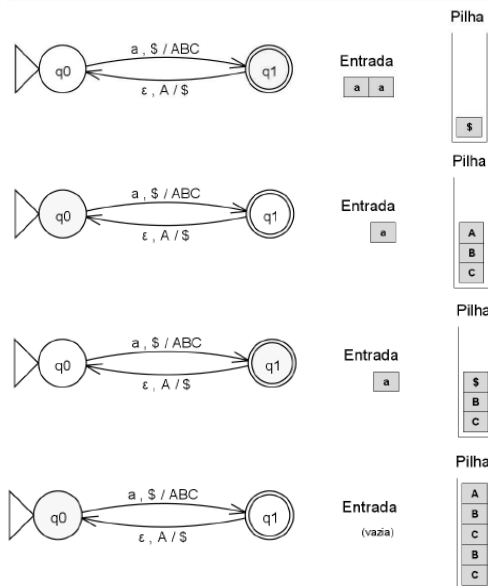
$s, X / Y_1 \dots Y_k$, onde:

- **s** é o símbolo que será lido da palavra (ou ϵ para não ler nada)
- **X** é o símbolo que será lido da pilha (ou ϵ para não ler nada)
- **$Y_1 \dots Y_k$** é a cadeia de símbolos que será escrita na pilha (ou ϵ para não escrever nada), considerando que os símbolos serão empilhados na ordem inversa (de Y_k para Y_1), deixando Y_1 no topo



slide 12

Computação e Aceitação no AP



slide 13

Computação e Aceitação no AP

O modelo Autômato com Pilha possui duas definições universalmente aceitas que diferem no critério de parada do autômato, como segue:

- o valor inicial da pilha é vazio e o autômato pára aceitando ao atingir um estado final;
- a pilha contém, inicialmente, um símbolo especial denominado símbolo inicial da pilha. Não existem estados finais e o autômato pára aceitando quando a pilha estiver vazia.

slide 14

Definição Formal

7-tuplas (ou seja, estruturas contendo uma sequência de 7 elementos) na forma:

$P = (Q, \Sigma, \Gamma, \delta, s, \$, F)$, onde:

- Q é conjunto finito de estados, como nos autômatos finitos
- Σ (sigma) é o conjunto dos símbolos de entrada, como nos autômatos finitos
- Γ (gama) é o conjunto dos símbolos da pilha, que são os únicos símbolos que podem ser escritos na pilha
- δ (delta) é a função de transição, parecida com a dos autômatos finitos
- s é o elemento de Q que serve de estado inicial, como também ocorre nos autômatos finitos
- $\$$ é um elemento usado como símbolo inicial da pilha
- F é o conjunto de estados de aceitação ou estados finais, como nos autômatos finitos

slide 15

Definição Formal

δ (delta) – função de transição

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \rightarrow 2^Q \times \Gamma^*$$

Resumindo = tripla $\delta(q, a, X)$

- q é o estado de onde sai a transição
- a é o próximo símbolo da palavra de entrada ou ϵ (para não ler nada)
- X é o próximo símbolo a ser lido da pilha ou ϵ (para não ler nada)

slide 16

Definição Formal

Para cada tripla dada como argumento, a função de transição delta dá como resultado um conjunto de pares $\{ (p_1, a_1), (p_2, a_2), \dots, (p_j, a_j) \}$. Cada par indica o próximo estado p_x e a cadeia que será escrita na pilha a_x

O autômato pode escolher uma das opções:

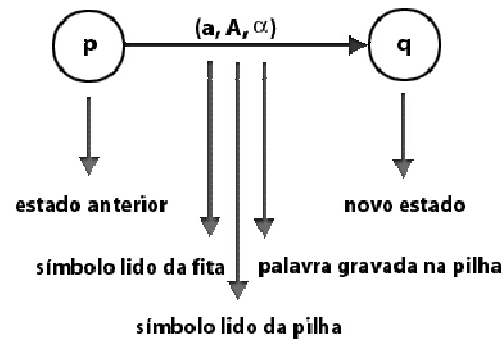
- Mover para p_1 e escrever a_1 na pilha,
- OU mover para p_2 e escrever a_2 na pilha, OU ...,
- OU mover para p_j e escrever a_{1j} na pilha.

slide 17

Transição

se $((p, a, A), (q, B_1 B_2 \dots B_k)) \in \delta$ isto significa intuitivamente que quando a máquina está no estado p lendo o símbolo a (na fita de entrada) e A (no topo da pilha), ela tira A da pilha, coloca $B_1 B_2 \dots B_k$ na pilha (B_k primeiro e B_1 por último), e entra no estado q .

Transição



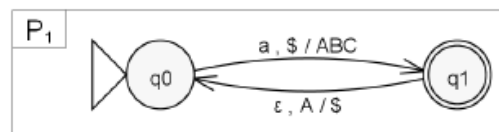
Definição Formal

$$P_1 = (\{q_0, q_1\}, \{a, b\}, \{\$, A, B, C\}, \delta_1, q_0, \$, \{q_1\})$$

$$\delta_1(q_0, a, \$) = \{ (q_1, ABC) \}$$

$$\delta_1(q_1, \varepsilon, A) = \{ (q_0, \$) \}$$

$\delta_1(q_0, b, A)$ ou $\delta_1(q_1, a, B)$ dão o resultado \emptyset .



slide 20

Loop e Rejeição

- *Loop* é quando sempre existe alguma transição válida para uma dada palavra
 - O autômato fica em ciclo e nunca pára!
 - Similar a idéia de um loop infinito em programação
- Não é a mesma coisa que rejeitar
 - Para rejeitar, tem que parar

21

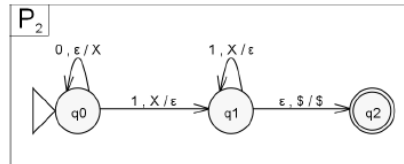
Autômatos com Pilha

- Quando uma palavra permite caminhos de reconhecimento em situações distintas, a palavra é classificada na seguinte ordem:
 - Se um caminho aceita, a palavra é dita **aceita**
 - Senão, mas existe um caminho em loop, a palavra é dita **em loop**
 - Senão, a palavra é dita **rejeitada**

22

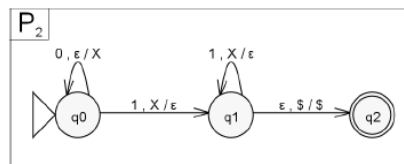
Prática

- **Questão 1:** Mostre a computação do AP da questão anterior quando ele recebe a cadeia 0011. Diga se esta cadeia é aceita e justifique



Prática

- **Questão 2:** Dê a representação formal para o autômato P2



Estados (Q): {q0, q1, q2}

Transição?

Inicial (Q0): q0

Final (F): {q2}

Alfabeto (Σ): {0, 1}

Simb. Pilha (Γ): {\$, X}

inicial: \$

$P_2 = (\{q_0, q_1, q_2\}, \{0, 1\}, \{\$, X\}, \delta_2, q_0, \$, \{q_2\})$, onde δ_2 é dada por:

$\delta_2(q_0, 0, \varepsilon) = \{(q_0, X)\}$

$\delta_2(q_0, 1, X) = \{(q_1, \varepsilon)\}$

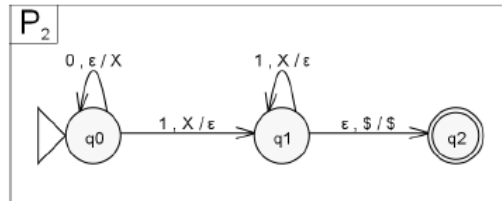
$\delta_2(q_1, 1, X) = \{(q_1, \varepsilon)\}$

$\delta_2(q_1, \varepsilon, \$) = \{(q_2, \$)\}$

slide 24

Prática

- **Questão 3:** a linguagem das palavras na forma 0^n1^n , para todo $n > 1$. Crie agora um AP de nome P2 para representá-la.



slide 25

Linguagens Formais

UNICAP

Eduardo Araújo Oliveira
<http://sites.google.com/site/eaoufpe>