



INSTITUTO TECNOLÓGICO DE AERONÁUTICA

DIVISÃO DE ENGENHARIA DE COMPUTAÇÃO

CTC-34 AUTOMATA E LINGUAGENS FORMAIS

3º Projeto

Simulação não-determinística para autômatos de
pilha (pushdown automata)

Aluno: Luiz A. R. RAFAEL

Prof. responsável:
Carlos H. Q. FORSTER

12 de novembro de 2015

1 Metodologia

Seja Q o conjunto dos estados do autômato a ser simulado; Σ o alfabeto; δ o conjunto de transições do tipo $(ea, symbol, popped, pushed, ep)$, em que ea é o estado atual, $symbol$ é o símbolo do alfabeto a ser lido da cadeia de entrada, $popped$ é o elemento a ser retirado da pilha, $pushed$ é o elemento a ser colocado na pilha e ep é o próximo estado; p_0 o estado inicial e F o conjunto dos estados de aceitação. Na implementação, o símbolo $*$ foi reservado para indicar ϵ . O algoritmo utilizado para a simulação dos autômatos de pilha está descrito em 1.

2 Material e resultado das simulações

Todo o projeto foi desenvolvido na linguagem Java utilizando a IDE Eclipse Mars. Foi utilizada uma classe demo para testes.

Como consulta de implementação, utilizou-se dois principais recursos: o slide de aula 7 de CTC-34 e esta referência [1].

O esquema a seguir mostra a estrutura de pastas na raiz do projeto.

```
src
├── simulator
│   ├── Automata.java
│   ├── Reader.java
│   ├── Simulator.java
│   └── State.java
└── tests
    ├── demos
    └── StackAutomataDemos.java
```

A classe *Reader* lê um *.txt* como *input* com os dados do autômato, armazena esses dados em uma instância da classe *Automata* e *Simulator* realiza a simulação e imprime as transições. Um exemplo de *input* está na figura 1 (os textos após *%* servem apenas para indicar o que cada linha representa, mas eles não estão presentes no *input* real). Os resultados das simulações estão na figura 2.

Referências

- [1] Pushdown automaton. [Online]. Available: http://en.wikipedia.org/wiki/Pushdown_automaton

3 Apêndice

O repositório *Git* que possui todo o código do projeto pode ser encontrado em

<http://github.com/Luizangel50/StackAutomata>

a b c	%alfabeto
q1 q2 q3 q4 q5 q6 q7	%estados
q1	%estado inicial
q4 q7	%estados finais
11	%quantidade de transições
q1 * * \$ q2	%transições
q2 a * a q2	% -
q2 * * * q3	% -
q2 * * * q5	% -
q3 b a * q3	% -
q3 * \$ * q4	% -
q4 c * * q4	% -
q5 b * * q5	% -
q5 * * * q6	% -
q6 c a * q6	% -
q6 * \$ * q7	% -
a a b c c	%cadeia a ser testada

Figura 1: *Input para o exemplo a)*

```

Item a)
[q1, q2, q3, q4]
[q1, q2, q5, q6, q7]
[q1, q2, q2, q3]
[q1, q2, q2, q5, q6]
[q1, q2, q2, q2, q3]
[q1, q2, q2, q2, q3, q3]
[q1, q2, q2, q2, q5, q6]
[q1, q2, q2, q2, q5, q5, q6]
[q1, q2, q2, q2, q5, q5, q6, q6]
[q1, q2, q2, q2, q5, q5, q6, q6, q7] : String accepted
-----
Item b)
[q1, q2, q3, q4]
[q1, q2, q3]
[q1, q2, q2, q3]
[q1, q2, q2, q3, q3, q4]
[q1, q2, q2, q3, q3]
[q1, q2, q2, q2, q3]
[q1, q2, q2, q2, q3, q3]
[q1, q2, q2, q2, q3, q3]
[q1, q2, q2, q2, q2, q3]
[q1, q2, q2, q2, q2, q3]
[q1, q2, q2, q2, q2, q2, q3]
[q1, q2, q2, q2, q2, q2, q3, q3]
[q1, q2, q2, q2, q2, q2, q3, q3]
[q1, q2, q2, q2, q2, q2, q2, q3]
[q1, q2, q2, q2, q2, q2, q2, q3, q3]
[q1, q2, q2, q2, q2, q2, q2, q2, q3]
-----
Item c)
[q1, q2, q3, q4]
-----

```

Figura 2: *Impressões das transições dos autômatos requeridos*

Input: autômato $M = \{Q, \Sigma, \delta, p_0, F\}$ e cadeia str

Output: transições até que a cadeia seja aceita ou todas as transições possíveis se a cadeia não for aceita

$ea = p_0$: estado atual;

s : pilha

$pos = 0$: posição do símbolo de str a ser lido

```
while  $pos \leq str.size()$  do
  if  $ea$  possui transições  $\epsilon$  then
    for cada transição  $\epsilon$  do
      if  $popped == s.top()$  then
         $s.pop()$ ;
         $s.push(pushed)$ ;
         $ea \leftarrow ep$ ;
        recursão
      end
    else
      | imprimir transição;
    end
  end
end
if  $pos < str.size()$  AND  $ea$  possui transições non- $\epsilon$  then
  for cada transição non- $\epsilon$  do
    if  $symbol == str[pos]$  then
      if  $popped == s.top()$  then
         $s.pop()$ ;
         $s.push(pushed)$ ;
         $ea \leftarrow ep$ 
         $pos \leftarrow pos + 1$ 
        recursão
      end
    else
      | imprimir transição;
    end
  end
end
if  $pos == str.size()$  AND  $F.contains(ea)$  AND  $s.isEmpty()$  then
  | imprimir transição;
  | break;
end
end
end
```

Algoritmo 1: Simulação do autômato e impressão das transições