

Relatório técnico de implementação de um Autômato a Pilha

Raphael Franco de Lima¹, Maicon Henrique Rodrigues do Nascimento¹

¹Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brasil

[ra98336, ra102504]@uem.br

Abstract. *This report presents the definition and concepts of a pushdown automata and the questions and considerations relative to its implementation using the C++ programming language.*

Resumo. *O presente relatório apresenta a definição e conceito de um Autômato a Pilha e as questões e considerações inerentes a sua implementação utilizando a linguagem de programação C++.*

1. Introdução

Na teoria da computação um Autômato a Pilha (AP) é uma máquina que possui como entrada uma “palavra” e ao final uma resposta de se está pertence ou não a uma linguagem definida. A diferença entre um AFD e um AP é que neste último é necessário uma estrutura de dados do tipo pilha como memória auxiliar. Afim de se aprofundar sobre o funcionamento de tal máquina, chamadas por definição de máquina de estados finita, foi realizada sua implementação utilizando-se de uma linguagem de programação e posteriormente a análise de todas as implicações referentes ao projeto.

2. Objetivos/Justificativas

O presente relatório objetiva por explicitar a estruturação e construção de um Autômato a Pilha utilizando uma linguagem de programação. Tal ação é feita com o intuito de exercitar na prática o conhecimento adquirido sobre o respectivo assunto e assim validar sua correta implementação e aprendizado.

3. Autômato a Pilha

Um Autômato a Pilha é uma máquina de estados finita que dado uma “palavra” avalia cada símbolo desta e indica se está é aceita pela linguagem. Uma palavra é dita aceita quando esta é lida por completo e a pilha usada como memória auxiliar se encontra vazia. Para entender melhor tal funcionamento é necessário a explicação de alguns termos.

Um alfabeto é um conjunto finito de símbolos com o qual se podem formar linguagens, sendo estas a concatenação de múltiplos símbolos.

Desta forma para realizar a análise da palavra tem-se que a cada símbolo lido será removido um símbolo auxiliar do topo da pilha e inserido zero, um ou mais.

O Autômato pode ser definido de duas formas, a primeira delas é chamada de definição formal e consiste de uma tupla de seis valores sendo eles:

E é o conjunto finito de estados

Σ é o conjunto finito de símbolos alfabeto

δ é a função de transição, tal que $\delta: E \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow E \times \Gamma^*$

i é o estado inicial ($i \in E$)

B é a base da pilha ($B \in \Gamma$)

Γ são os símbolos auxiliares da pilha

Sua segunda representação dá-se de maneira gráfica onde:

- Vértices representam os estados
- Arcos com legenda representam uma transição
 - A legenda é no formato (x, Y, Z) onde x representa um símbolo do alfabeto, Y o símbolo auxiliar no topo da pilha e Z o que será inserido na pilha.
- O estado inicial é representado com um arco de entrada sendo sua origem vazia

4. Exemplos

Para clarificar a definição e funcionamento de um AP podemos utilizar alguns exemplos que exibem tanto a representação gráfica quanto formal.

Exemplo 1:

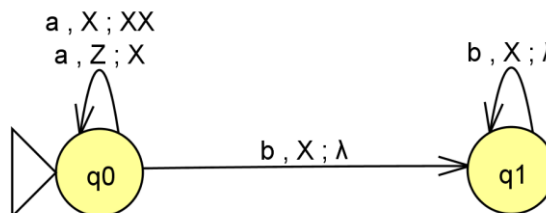


Figura 1- Exemplo AP

Tal exemplo representa a linguagem $L = \{a^n b^n \mid n > 0\}$ e tem como descrição formal:

$E = \{q0, q1\}$

$\Sigma = \{a, b\}$

$i = q0$

$\Gamma = \{Z, X\}$

Base = Z

Função de transição

- $\delta(q0, a, Z) = (q0, X)$

- $\delta(q_0, a, X) = (q_0, XX)$
- $\delta(q_0, b, X) = (q_1, \lambda)$
- $\delta(q_1, b, X) = (q_1, \lambda)$

Para esta linguagem podemos demonstrar alguns exemplos de palavras aceitas ou não por ela.

Palavra 1: abc – Rejeita

Palavra 2: aaabbb – Aceita

Palavra 3: aab – Rejeita

Palavra 4: aaaaabbbb – Aceita

A linguagem acima descrita pode ser testada através do programa implementado e descrito por este relatório. Para isso basta usar o arquivo de exemplo ‘ap.txt’ que consta no projeto disponibilizado.

Exemplo 2:

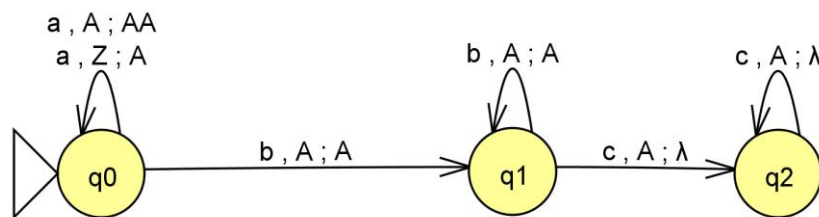


Figura 2 - Exemplo AP

Tal exemplo representa a linguagem $L = \{a^n b^m c^n \mid n > 0, m > 0\}$ e tem como descrição formal:

$E = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$

$i = q_0$

$\Gamma = \{Z, A\}$

Base = Z

Função de transição

- $\delta(q_0, a, Z) = (q_0, A)$
- $\delta(q_0, a, A) = (q_0, AA)$
- $\delta(q_0, b, A) = (q_1, A)$
- $\delta(q_1, b, A) = (q_1, A)$
- $\delta(q_1, c, A) = (q_2, \lambda)$
- $\delta(q_2, c, A) = (q_2, \lambda)$

Para esta linguagem podemos demonstrar alguns exemplos de palavras aceitas ou não por ela.

Palavra 1: aaba – Rejeita

Palavra 2: abc – Aceita

Palavra 3: aabc – Rejeita

Palavra 4: aaaaaabbbccccc – Aceita

Para realizar o teste da linguagem acima utilizando-se do programa desenvolvido basta utilizar o arquivo ‘ap2.txt’ que consta na no projeto disponibilizado.

Exemplo 3:

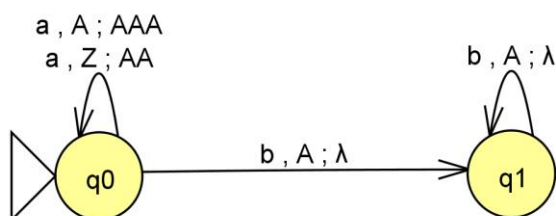


Figura 3 - Exemplo AP

Tal exemplo representa a linguagem $L = \{a^n b^{2n} \mid n > 0\}$ e tem como descrição formal:

$E = \{q0, q1\}$

$\Sigma = \{a, b\}$

$i = q0$

$\Gamma = \{Z, A\}$

Base = Z

Função de transição

- $\delta(q0, a, Z) = (q0, AA)$
- $\delta(q0, a, A) = (q0, AAA)$
- $\delta(q0, b, A) = (q1, \lambda)$
- $\delta(q1, b, A) = (q1, \lambda)$

Para esta linguagem podemos demonstrar alguns exemplos de palavras aceitas ou não por ela.

Palavra 1: ac – Rejeita

Palavra 2: abb – Aceita

Palavra 3: aabbb – Rejeita

Palavra 4: aabbbb – Aceita

Para realizar o teste da linguagem acima utilizando-se do programa desenvolvido basta

utilizar o arquivo ‘ap3.txt’ que consta na no projeto disponibilizado.

5. Decisões de projeto

Para realizar a implementação primeiramente foi analisado qual linguagem seria utilizada. A decisão tomada foi a de utilizar a linguagem C++ devido a familiaridade com ela, seu bom desempenho e boa abstração para manipulação de arquivos. Em seguida foi considerado qual estrutura de dados seria utilizada para armazenar o autômato. Neste quesito optou-se por utilizar o conceito de orientação a objetos e modelar tanto a pilha como o autômato como classes. A pilha possui método para inserir, remover, verificar se está vazia e retornar o topo. Ela foi desenvolvida com o conceito de nós e alocação dinâmica, sendo que cada nó armazena a referência do próximo item da pilha.

Já autômato possui como atributos um vetor de estados (o tipo estado possui uma string com seu nome e o seu tipo), o alfabeto (um vetor de caracteres), o estado inicial, a base da pilha, um vetor de caracteres com os símbolos auxiliares, a pilha mencionada anteriormente e um vetor para armazenar transições. O tipo transição é uma classe com os atributos, estado inicial, estado final, topo da pilha, símbolos a serem inseridos e símbolo do alfabeto.

Em relação a entrada dos dados optou-se por utilizar a leitura de um arquivo sendo que que sua estrutura deve seguir rigidamente o seguinte formato:

- 1ª linha: quantidade de estado
- 2ª linha: conjunto de todos os estados separados por um espaço em branco
- 3ª linha: quantidade de símbolos do alfabeto
- 4ª linha: conjunto de todos os símbolos do alfabeto separados por um espaço em branco
- 5ª linha: estado inicial
- 6ª linha: quantidade de símbolos auxiliares
- 7ª linha: símbolos auxiliares separados por um espaço em branco
- 8ª linha: base da pilha
- 9ª linha: quantidade de transições
- 10ª linha e subsequentes: representação de uma transição no formato “x y z w v” onde:
 - x = estado inicial
 - y = símbolo do alfabeto
 - z = topo da pilha
 - w = estado final
 - v = símbolo(s) a ser(em) inseridos na pilha

Já a palavra a ser analisada é lida através de inserção manual do usuário através do teclado.

O algoritmo em si pode ser visto como um *loop* que percorre todos os símbolos da palavra

e verifica se o símbolo lido é válido e caso seja, se existe uma transição entre ele e o estado atual.

Para executar o programa é necessário o compilador *gcc*. Após isso, os seguintes comandos são necessários:

- `gcc -o ap main.cpp ap.cpp Pilha.cpp`
- `./ap`

6. Conclusão

Através do exposto pode-se concluir que o objetivo de estudar o funcionamento de um Autômato a Pilha através de sua implementação utilizando uma linguagem de programação foi alcançado dado o sucesso desta tarefa comprovado pelo teste de múltiplas linguagens e palavras.

7. Referências

Pushdown automaton. (17 de janeiro de 2020). Fonte: Wikipedia:
https://en.wikipedia.org/wiki/Pushdown_automaton

Introduction of Pushdown Automata. (17 de janeiro de 2020). Fonte:
GeeksforGeeks: <https://www.geeksforgeeks.org/introduction-of-pushdown-automata/>