Trabalho prático de Autômato com Pilha

Henrique Negri Rodrigues RA105480

Departamento de Informática - Universidade Estadual de Maringá (UEM)

Abstract. *This report aims to demonstrate how was the practical work of the subject Formal Languages and Automata of the 3rd year of Informatics, where a Stack Automaton was developed*

Resumo. *Este Relatório visa demonstrar como foi feito o trabalho prático da matéria Linguagens Formais e Autômatos do 3º ano de Informática, onde foi desenvolvido um Autômato Finito Determinístico*

1. Introdução

Autômatos com pilha diferem da definição normal de máquinas de estados finitos de duas maneiras:

Eles podem fazer uso da informação que está no topo da pilha para decidir qual transição deve ser efetuada;

Eles podem manipular a pilha ao efetuar uma transição.

Autômatos com pilha escolhem uma transição analisando o símbolo atual na cadeia de entrada, o estado atual e o topo da pilha. Máquinas de estados finitos convencionais apenas analisam o símbolo na cadeia de entrada e o estado atual. Autômatos com pilha adicionam a pilha como recurso auxiliar, deste modo, dado um símbolo da cadeia de entrada, o estado atual e um símbolo no topo da pilha, uma transição é selecionada.

Máquinas de estados finitos apenas escolhem um novo estado como resultado da sua transição, já os autômatos com pilha também podem manipular a pilha, como resultado de sua transição. A manipulação é feita através do desempilhamento de um símbolo da pilha ou através do empilhamento de um novo símbolo ao topo da mesma. Alternativamente, um autômato com pilha pode ignorar a pilha e deixá-la como está.

2. Objetivos/Justificativa

O trabalho tem como objetivo a implementação de um algoritmo que irá receber a descrição formal de um Autômato com Pilha e em seguida testará palavras providas pelo usuário verificando se a mesma pertence à linguagem descrita pelo autômato.

O algoritmo gerado neste projeto visa manter o objetivo principal com o adicional de acrescentar dinamicidade no recebimento dos dados do Autômato com Pilha e a melhor experiência do usuário possível deixando ele sempre ciente do que está acontecendo em tela e mantendo ele sempre no controle da situação.

3. Autômato com Pilha

Um autômato de pilha é formalmente definido por uma 6-tupla:

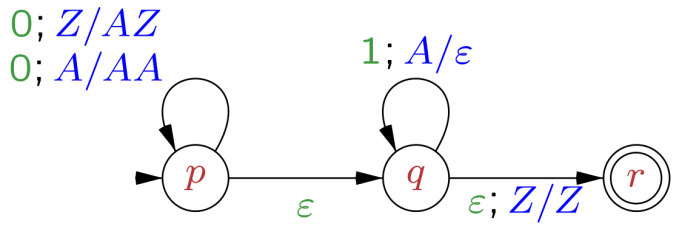
$M = (Q,\ \Sigma,\ \Gamma,\ \Delta,\ q_0,\ F)$

Onde:

- Q é um conjunto finito de estados.
- Σ é um conjunto finito de símbolos, denominado alfabeto de entrada.
- Γ é um conjunto finito de símbolos, denominado alfabeto da pilha.
- $\Delta \subseteq (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Gamma^*)$ é a relação de transição.
- $q_0 \in Q$ é o estado inicial.
- $F \subseteq Q$ é o conjunto de estados finais (ou de aceitação).

Um elemento (p,a,α,q,β) é uma transição de M. Ela significa que M, estando no estado p, com o símbolo a na cadeia de entrada e com o símbolo α no topo da pilha, pode consumir o símbolo a, transitar para o estado q e desempilhar α substituindo-o por β. O Σ^* e o Γ^* denotam o fecho de Kleene do alfabeto de entrada e da pilha, respectivamente. Portanto, estes componentes são utilizados para formalizar que o autômato de pilha pode consumir qualquer quantidade de símbolos da cadeia de entrada e da pilha.

Exemplo:



Neste exemplo podemos ver que o autômato foi definido com a seguinte descrição:

$M = (Q,\ \Sigma,\ \Gamma,\ \delta,\ p,\ Z,\ F)$, onde

$Q = \{p, q, r\}$

$\Sigma = \{0, 1\}$

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

$F = \{r\}$

δ consiste nas seis instruções seguintes:

$(p, 0, Z, p, AZ)$, $(p, 0, A, p, AA)$, (p, ϵ, Z, q, Z), (p, ϵ, A, q, A), $(q, 1, A, q, \epsilon)$, (q, ϵ, Z, r, Z)

Ou seja, no estado p para cada símbolo 0 que for lido, um A é colocado na pilha. Empilhar um símbolo A no topo de um outro A é formalizado como trocar o símbolo A por AA. No estado q, para cada símbolo 1 lido, um A a é desempilhado. Em um outro momento, o autômato se move do estado p para o estado q, enquanto ele pode mover do estado q para o estado de aceitação r somente quando a pilha possuir um único Z.

Representamos a instrução como uma ponte que sai do estado p para o estado q rotulada por $a; A/\alpha$.

4. Decisões de projeto

No atual projeto, resolvi fazer em JavaScript por ser a linguagem que possuo maior conhecimento. Como em JavaScript não há como trabalhar diretamente com ponteiros a pilha foi implementada de uma maneira ligeiramente diferente, porém com o mesmo comportamento.

A pilha foi implementada com um tipo de variável nativo do JS que é o objeto. Um objeto trata-se de um conjunto de dados atrelados por *atributo: valor*, na pilha foi atribuído ao objeto dois atributos, ***valor*** (que irá armazenar o valor daquele campo) e ***proximo*** (que contém o objeto que vem a seguir na pilha), formando um objeto recursivo que contém outros objetos formando a pilha por completo.

Ao adicionar algo na pilha é criado uma nova variável, seguindo o conceito de imutabilidade famoso no JS, contendo um objeto com ***valor*** e o ***proximo*** do novo objeto é a pilha antiga, adicionando sempre acima da pilha. Ao remover da pilha apenas é atribuído a pilha o valor do ***proximo.***

Ao iniciar o programa, é setado a princípio uma base fixa com o valor de **B** então, é pedido para que o usuário defina os outros itens da definição formal do autômato. Durante todo o processo do programa existe uma caixa que mostra ao usuário tudo o que foi setado do autômato até o momento.

Assim que setado o autômato por completo, aparece um campo de texto para que o usuário digite palavras para ser testadas pelo autômato, o usuário deve escrever a palavra e sair do campo, assim que clicar fora do campo será analisado se a palavra é ou não aceita pelo autômato.

5. Conclusão

Esse projeto se baseou na descrição formal do trabalho ditada pelo professor da matéria de Linguagens Formais e Autômatos, as decisões tomadas durante o projeto baseou-se na facilidade de implementação e na experiência do usuário para ficar o mais limpo e específico possível, prevendo diversos erros de entrada de dados do próprio usuário e

tentando deixar a tela o menos poluída possível mantendo todas as informações de forma compacta e direta para que o usuário não se perca nas informações geradas.