

AUTÔMATOS DE PILHA

Linguagens Formais e Autômatos

Willian Cavalheiro
Gabriel Schramm
Lucas da Silva Bettio
Henrique Boelter

2023

1

Introdução

2

Objetivo

3

Formalismo

4

Utilidade e
exemplos

5

Projeto prático

Introdução

Grupo 2 | Autômato de Pilha

Este trabalho tem como objetivo apresentar os autômatos de pilha, uma extensão dos autômatos finitos que são capazes de reconhecer linguagens mais complexas. Exploraremos o formalismo dos autômatos de pilha, sua descrição e forneceremos exemplos práticos, além de discutir uma aplicação real.

Formalismo:

Os autômatos de pilha são descritos por meio de conjuntos de estados, alfabeto de entrada, alfabeto da pilha, estados inicial e de aceitação, e transições que determinam a mudança de estado e operações na pilha.

Descrição:

Explicaremos detalhadamente como os autômatos de pilha funcionam, incluindo a importância da pilha como estrutura de dados, as transições entre os estados e a diferença entre autômatos de pilha determinísticos e não determinísticos.

Exemplos:

Apresentaremos exemplos práticos de autômatos de pilha, como reconhecimento de palíndromos e parênteses bem formados, para ilustrar seu funcionamento e capacidades.

Aplicação Real:

Destacaremos a aplicação real dos autômatos de pilha no desenvolvimento de compiladores e interpretadores de linguagens de programação. Explicaremos como são usados na análise sintática para validar a estrutura das construções da linguagem.

Objetivo

Grupo 2 | Autômato de Pilha

Definição

O autômato de pilha é um modelo análogo ao autômato finito, Semelhante a um AFND porém com um componente adicional para representar uma memória auxiliar na forma de uma pilha, seguindo o conceito de LIFO (Last In, First Out).

A pilha é independente da fita de entrada e os estados do autômato em si são finitos mas a capacidade de armazenamento da pilha é infinita, entretanto só pode ler e escrever o elemento que estiver no topo da pilha.

Compreendendo a Transição:

Os autômatos de pilha são diferentes do restante, por conta de dois pontos:

Com o autômato de pilhas, é possível fazer uso da informação armazenada no topo da pilha para assim definir qual transição será feita. O outro ponto é que por consequência eles acabam por manipular a pilha ao efetuar essa transição.

A transição é feita a partir da análise do símbolo atual da cadeia de entrada, o estado atual e o topo da pilha.

A pilha é uma ferramenta extra, utilizada também para auxiliar no processo de transição.

Uma diferença visível é que máquinas de estados finitos apenas escolhem como resultado da transição um novo estado e as de pilhas podem manipular a pilha em si.

Um autômato finito que possui acesso à duas pilhas têm poder e capacidade computacional equivalente à uma máquina de Turing.

Movimentação

O que determinará se haverá movimentação pelos estados, é o número de elementos dentro da Função " δ ", composta pela tripla (q, σ, γ) , onde " q " é o estado corrente, " σ " é o símbolo lido, e " γ " é o símbolo da pilha. que pode conter nenhum, um ou mais de um elementos:

- No caso de não haver elementos, não há troca de estado;
- Havendo um elemento, a movimentação será determinística;
- Para mais de um elemento, a transição é não determinística (explosão de estados)

Formalismo

Grupo 2 | Autômato de Pilha

Os autômatos de pilha são definidos pela seguinte 7-upla

$$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

Onde temos como definições para cada elemento da tupla:

Q

Conjunto finito de estados

Σ

Alfabeto (símbolos de entrada)

Γ

Alfabeto finito da pilha

δ

Função de transição – $\delta: Q \times (\Sigma \cup \lambda) \times \Gamma^* \rightarrow Q \times \Gamma^*$

q_0

Estado inicial

Z_0

Símbolo de início da pilha (opcional)

F

Conjunto de estados finais.

Utilidade e exemplos

Grupo 2 | Autômato de Pilha

Os autômatos de pilha são amplamente utilizados no processamento de linguagens formais, como em compiladores, processamento de linguagens naturais e análise de expressões matemáticas. Eles desempenham um papel crucial na validação da estrutura sintática das construções da linguagem, garantindo que elas sigam as regras definidas. Isso proporciona maior confiabilidade e eficiência no processamento de informações em várias áreas da computação. . Aqui estão alguns exemplos de sua utilização.

Compiladores e Interpretadores

Autômatos de pilha são amplamente utilizados na construção de compiladores e interpretadores de linguagens de programação.

Na fase de análise sintática, o autômato de pilha é usado para verificar a estrutura hierárquica das construções da linguagem, como expressões matemáticas e comandos de controle de fluxo. Ele garante que a sintaxe do programa esteja correta antes de prosseguir para a geração de código ou execução.

Processamento de Linguagens Naturais

Autômatos de pilha são empregados no processamento de linguagens naturais, como análise de sentenças gramaticais e construção de chatbots.

Ao lidar com gramáticas formais que descrevem a estrutura de sentenças, um autômato de pilha pode ser usado para validar e analisar a correteza sintática de sentenças em linguagem natural. Isso permite a construção de sistemas de diálogo inteligentes e a interpretação de comandos na linguagem humana.

Análise de Expressões Matemáticas

Autômatos de pilha são aplicados na análise de expressões matemáticas para verificar a validade e a correta ordem de operações.

O autômato de pilha pode ser usado para validar a sintaxe de expressões matemáticas, garantindo que os parênteses, colchetes e chaves estejam corretamente balanceados. Isso é útil em calculadoras e sistemas de avaliação de fórmulas matemáticas.

Verificação de Linguagens de Programação

Autômatos de pilha são usados para verificar se um programa em uma linguagem de programação obedece à sua gramática definida.

O autômato de pilha pode ser utilizado para realizar a análise léxica e sintática de um programa, verificando se ele está escrito corretamente e segue as regras gramaticais da linguagem. Isso ajuda a evitar erros de programação e garante a correta interpretação e execução do código.

PROJETO CÓDIGO

Grupo 2 | Autômato de Pilha

Implementação do Código: O código desenvolvido consiste em uma calculadora simples em JavaScript, onde os números e operadores são inseridos por meio de botões. O destaque principal é a função `expressaoValida`, que utiliza uma estrutura de pilha para verificar se a expressão contém parênteses corretamente balanceados. Essa função percorre a expressão caractere por caractere, empilhando os parênteses de abertura e desempilhando um parêntese de fechamento a cada ocorrência. Caso a pilha esteja vazia ao encontrar um parêntese de fechamento, a expressão é considerada inválida.

Além disso, o código inclui funções para realizar as operações matemáticas básicas, como adição, subtração, multiplicação e divisão. Essas funções são acionadas pela função `calcular`, que recebe a expressão matemática válida (Com todos parênteses testados) e utiliza a função `eval` para avaliar a expressão e retornar o resultado final.

LINK GITHUB PARA O CÓDIGO COMPLETO:

https://github.com/OmegaTopo/exemplo_automato_pilha

```
class Pilha {
  constructor() {
    this.pilha = [];
  }

  push(item) {
    this.pilha.push(item);
  }

  pop() {
    if (!this.vazia()) {
      return this.pilha.pop();
    } else {
      return null;
    }
  }

  vazia() {
    return this.pilha.length === 0;
  }
}

function expressaoValida(expressao) {
  const pilha = new Pilha();

  for (let i = 0; i < expressao.length; i++) {
    const caractere = expressao[i];
    if (caractere === '(') {
      pilha.push(caractere);
    } else if (caractere === ')') {
      if (pilha.pop() === null) {
        return false;
      }
    }
  }

  return pilha.vazia();
}
```

Referências

Grupo 2 | Autômato de Pilha

<https://www.geeksforgeeks.org/introduction-of-pushdown-automata/>

https://www.tutorialspoint.com/automata_theory/pushdown_automata_introduction.htm#

<http://wiki.icmc.usp.br/images/d/d1/LLC%26APJoao.pdf>