Universidade São Judas Tadeu
Ciências da Computação

Bruno Godoy Dias

Henryk Bagdanovicius Roza

João Victor Varella Siqueira

Matheus Henrique O. Chuang

Raí Joia Miquilino valencio

Autômato de Pilha

São Paulo

2024

Bruno Godoy Dias

Henryk Bagdanovicius Roza

João Victor Varella Siqueira

Matheus Henrique O. Chuang

Raí Joia Miquilino

Autômato de Pilha

Projeto A3 ao Curso de ciências da computação da Universidade São Judas Tadeu, orientado pelo Prof.º Robson Calvetti,.

São Paulo

2024

Resumo

O autômato de pilha faz parte de um grupo de estruturas de dados, dentre eles temos as principais: Pilha, Fila, Lista, Árvore, Binária, Grafo e Tabela de Hashing. O autômato de pilha é do tipo finito. É um modelo computacional importante na teoria da computação e nas linguagens formais. É usado principalmente para analisar e processar linguagens que requerem estruturas hierárquicas, como linguagens de programação.

Em um autômato de pilha, são estruturas de dados do tipo lifo (last in,first out), que podem ser empregadas em autômatos, onde basicamente temos a estruturação na qual o último elemento a ser inserido será o primeiro a ser retirado. Assim, uma pilha permite acesso a apenas um item de dados, isto é, ao último inserido. Para processar o penúltimo item inserido, deve-se remover o último.

Exemplos de entradas

ACEITOS	NÃO ACEITOS
c	cc
aca	aababababa
bcb	abcabcabc
abbc bba	bbb
ababcbaba	aaa
	abcd
	accaa
	ababcaccaa

Códigos

Tela do test de fita

```
public void fecharJanela() {
        this.dispose();
    }

    public void analisarFita(String fita) {
        try {
            Fila fila = new Fila();
            Pilha pilha = new Pilha();
            String[] splitFita = fita.split("");
            int estado = 0;
            boolean aceito = true;

            for (int i = 0; i < fita.length(); i++) {
                fila.inserir(splitFita[i]);
            }

            for (int i = 0; i < fita.length(); i++) {
                if(estado == 0 && pilha.mostrarAtual() == null &&
fila.mostrarPrimeiro().equals("a") ||
                    estado == 0 && pilha.mostrarAtual() == null &&
fila.mostrarPrimeiro().equals("b")){
                    pilha.inserir(fila.mostrarPrimeiro());
                    fila.remover();
                    System.out.println("aceito");
                } else if (estado == 0 && pilha.mostrarAtual() == null &&
fila.mostrarPrimeiro().equals("c")) {
                    fila.remover();
                    estado = 1;
                    System.out.println("aceito");
                } else if (estado == 0 && pilha.mostrarAtual().equals("a") &&
fila.mostrarPrimeiro().equals("a") ||
                    estado == 0 && pilha.mostrarAtual().equals("b") &&
fila.mostrarPrimeiro().equals("a") ||
                    estado == 0 && pilha.mostrarAtual().equals("a") &&
fila.mostrarPrimeiro().equals("b") ||
                    estado == 0 && pilha.mostrarAtual().equals("b") &&
```

```
fila.mostrarPrimeiro().equals("b")){
                    pilha.inserir(fila.mostrarPrimeiro());
                    fila.remover();
                    System.out.println("aceito");
                } else if(estado == 0 && pilha.mostrarAtual().equals("a") &&
fila.mostrarPrimeiro().equals("c")) {
                    estado = 1;
                    pilha.inserir("a");
                    fila.remover();
                    System.out.println("aceito");
                } else if(estado == 0 && pilha.mostrarAtual().equals("b") &&
fila.mostrarPrimeiro().equals("c")) {
                    estado = 1;
                    pilha.inserir("b");
                    fila.remover();
                    System.out.println("aceito");
                } else if(estado == 1 && pilha.mostrarAtual().equals("a") &&
fila.mostrarPrimeiro().equals("a") ||
                            estado == 1 && pilha.mostrarAtual().equals("b") &&
fila.mostrarPrimeiro().equals("b")) {
                    pilha.remover();
                    System.out.println("aceito");
                } else {
                    System.out.println("Nao aceito fila: " +
fila.mostrarPrimeiro() + " pilha: " + pilha.mostrarAtual() + " estado: " +
estado);
                    notFita.setVisible(true);
                    revalidate();
                    repaint();
                    aceito = false;
                    Timer timer = new Timer();

                    TimerTask task = new TimerTask() {
                        @Override
                        public void run() {
                            fecharJanela();
                        }
                    };

                    timer.schedule(task, 3000);
                }
            }
```

```
            if (aceito) {
                notFita.setText("Fita aceita");
                notFita.setVisible(true);
                revalidate();
                repaint();

                Timer timer = new Timer();

                TimerTask task = new TimerTask() {
                    @Override
                    public void run() {
                        fecharJanela();
                    }
                };

                timer.schedule(task, 3000);
            }
        } catch (Exception e) {
            notFita.setVisible(true);
        }

    }
```

Código da fila

```
public class Fila {
  private NoLista inicio;
  private NoLista fim;
  public Fila() {
    this.inicio = null;
    this.fim = null;
  }

public void inserir(String item) {
    NoLista novoNo = new NoLista(item);
    if (inicio == null) {
      inicio = novoNo;
      fim = novoNo;
    } else {
      fim.proximo = novoNo;
      novoNo.anterior = fim;
      fim = novoNo;
    }
  }
  public void remover() {
    if (inicio != null) {
      if (inicio.proximo != null) {
        inicio = inicio.proximo;
        inicio.anterior = null;
      } else {
        inicio = null;
        fim = null;
      }
    }
  }
  public String mostrarPrimeiro() {
    if (inicio != null) {
        return inicio.item;
    } else {
        return null;
    }
  }
  public void mostrarUltimo() {
    if (fim != null) {
      System.out.println("Conteúdo do último nó: " + fim.item);
```

```
    } else {
      System.out.println("Fila vazia");
    }
  }
}
```

Código do nó

```
public class NoLista {
  String item;
  NoLista anterior;
  NoLista proximo;

  public NoLista(String item) {
    this.item = item;
    this.anterior = null;
    this.proximo = null;
  }
}
```

código da pilha

```
public class Pilha {
  private NoLista inicio;
  private NoLista atual;

  public Pilha() {
    this.inicio = null;
    this.atual = null;
  }

  public void inserir(String item) {
    NoLista novoNo = new NoLista(item);
    if (inicio == null) {
      inicio = novoNo;
      atual = novoNo;
    } else {
```

```
      novoNo.anterior = atual;
      atual.proximo = novoNo;
      atual = novoNo;
    }
  }

  public void remover() {
    if (atual != null) {
      if (atual.anterior != null) {
        atual.anterior.proximo = null;
        atual = atual.anterior;
      } else {
        inicio = null;
        atual = null;
      }
    }
  }

  public String mostrarAtual() {
    if (atual != null) {
        return atual.item;
    } else {
        return null;
    }
  }

  public void mostrarProximo() {
    if (atual != null && atual.proximo != null) {
      System.out.println("Conteúdo do próximo nó: " + atual.proximo.item);
    } else {
      System.out.println("Lista Encerrada");
    }
  }

  public void mostrarAnterior() {
    if (atual != null && atual.anterior != null) {
      System.out.println("Conteúdo do nó anterior: " + atual.anterior.item);
    } else {
      System.out.println("Lista Encerrada");
    }
  }
}
```