

AULA 07 AUTOMATOS PUSHDOWN (PDA)

1. Introducao aos PDAs

- Automatos Pushdown sao como automatos finitos nao deterministicos (NFA), mas com uma estrutura de pilha.
- A pilha serve como memoria adicional e permite que o PDA reconheca linguagens que nao sao regulares, mas sim livres de contexto (CFLs).
- Uma linguagem livre de contexto (CFL) pode ser definida por uma gramatica que obedece a regras do tipo $A \rightarrow e$, onde A e uma variavel e e e uma string de variaveis e terminais.

2. Funcionamento Basico do PDA

- Transicoes: um PDA le um simbolo da entrada, verifica o topo da pilha, executa uma acao (empilha, desempilha, ou nenhuma) e muda de estado.
- Exemplo de linguagem: $L = \{ 0^n 1^n \mid n \geq 0 \}$
- * Para cada 0 lido, empilha-se um simbolo.
- * Ao comecar a ler 1s, desempilha-se um simbolo por 1 lido.
- * A entrada e aceita se a pilha esta vazia ao final da leitura.

3. Verificacao de Pilha Vazia

- Como a pilha nao pode ser "vista" diretamente, insere-se um marcador especial (\$) no inicio.
- Quando o simbolo \$ e encontrado no topo, significa que todos os simbolos foram desempilhados.

4. PDA para linguagem palindroma com marcador central

- Linguagem: $L = \{ w c w^R \mid w \in \{a,b\}^* \}$
- * Empilha-se cada simbolo de w ate encontrar o 'c'.

* Após o 'c', compara-se cada símbolo lido com o topo da pilha.

* Se todos coincidirem e a pilha for esvaziada corretamente, a entrada é aceita.

5. Limitações das Gramáticas Livres de Contexto (CFG)

- Algumas linguagens não podem ser geradas por CFGs, como $L = \{ a^n b^n c^n \mid n \geq 0 \}$.

- Essas linguagens exigem que três contagens estejam balanceadas, o que CFGs não conseguem, pois só lembram de um valor de cada vez.

6. Pumping Lemma para CFLs

- Teorema usado para provar que uma linguagem NÃO é livre de contexto.

- Afirma que se uma linguagem é CFL, então existe um inteiro p (pumping length) tal que qualquer string s com $|s| \geq p$ pode ser dividida como $s = uvwxy$, satisfazendo:

1. $|vwx| \leq p$

2. $vx \neq \epsilon$

3. Para todo $i \geq 0$, a string $u v^i w x^i y \in L$

- Exemplo: provar que $L = \{ a^n b^n c^n \mid n \geq 0 \}$ não é CFL.

* Suponha que seja e aplique o lema para mostrar contradição: ao bombear v e x , perde-se o balanceamento entre os três símbolos.

AULA 09 MAQUINAS DE TURING

1. Definição

- Modelo computacional proposto por Alan Turing em 1936.

- Mais poderoso que AFs e PDAs, podendo simular qualquer algoritmo executável.

2. Componentes da Máquina de Turing (MT)

- Fita: infinita à direita, serve como memória.

- Cabecote: lê/escreve símbolos e move-se para esquerda ou direita.
- Estados: definidos por uma função de transição $(q, a) = (q', b, D)$, onde $D \in \{L, R\}$.

3. Capacidade Computacional

- MTs podem reconhecer linguagens regulares, livres de contexto e sensíveis a contexto.
- Linguagens aceitas por MTs são chamadas recursivamente enumeráveis.
- Se a MT sempre para (aceita ou rejeita), a linguagem é dita decidível (ou recursiva).

4. Church-Turing Thesis

- Toda função que pode ser computada efetivamente pode ser computada por uma MT.
- Formaliza o conceito de "algoritmo efetivo".

5. Problemas Decidíveis vs Indecidíveis

- Decidível: existe uma MT que sempre responde sim ou não corretamente em tempo finito.
- * Exemplo: verificação se uma string pertence a uma linguagem regular.
- Indecidível: não existe MT que resolva o problema para toda entrada.
- * Exemplo clássico: Problema da Parada.

6. Problema da Parada (Halting Problem)

- Dado um programa P e uma entrada x , determine se P termina quando executado com x .
- Prova da indecidibilidade:
- * Assume-se que existe uma MT H que resolve o problema.
- * Usa-se H para construir uma máquina D que leva a uma contradição lógica (paradoxo).
- * Conclusão: H não pode existir.

7. Turing-Decidível vs Turing-Reconhecível

- Turing-Decidível: MT sempre termina (aceita ou rejeita).

- Turing-Reconhecível: MT pode entrar em loop infinito se a string não for da linguagem.

8. Comparação com outros modelos:

* AFs: memória finita, apenas leitura.

* PDAs: memória em pilha, reconhece CFLs.

* MTs: memória infinita, leitura/escrita bidirecional, modelo mais geral.

Este resumo aprofunda os tópicos mais importantes para provas, com exemplos e explicações claras sobre funcionamento, limitações e classificações dos modelos formais.