Pontifícia Universidade Católica de Goiás

Goiânia, 10 de junho de 2025

PUC
GOIÁS

AED - Linguagens Formais e Autômatos

Aluno: Caio da Silveira Leal Granja

- **OBJETIVOS DO SISTEMA**

Este programa simula o comportamento de um compilador por meio da construção de um Autômato **Finito Não Determinístico com transições ε (AFN-ε)**. O sistema recebe como entrada os elementos que compõem uma gramática formal e, a partir disso, constrói um autômato e testa a aceitação de palavras com base nas transições definidas.

- **DESCRIÇÃO GERAL**

O sistema é desenvolvido em Python e está estruturado com uma classe principal Automato, responsável por:

- Capturar os dados da gramática fornecidos pelo usuário.
- Construir a tabela de transições do autômato.
- Exibir a definição formal do autômato.
- Testar se palavras são aceitas pela linguagem reconhecida.

- **FUNCIONAMENTO DO SISTEMA**

1. **ENTRADA DA GRAMÁTICA**:

O usuário é guiado pelo terminal para fornecer os seguintes dados:

- **Alfabeto (símbolos terminais):** Lista de símbolos que compõem o alfabeto do autômato, separados por espaço.
- **Estados (símbolos não-terminais):** Representações dos estados do autômato.
- **Símbolo Inicial:** Estado que inicia o autômato.
- **Produções:** Devem ser escriatas no formato A -> aB | b | vazio.

2. **CONSTRUÇÃO DO AUTÔMATO**:

Com base nas entradas:

- As produções são transformadas em transições.
- Um estado universal F é adicionado como estado final padrão.
- Estados com produção “ε” (ou “vazio”) são marcados como finais.
- As transições são armazenadas em um dicionário de dicionários com conjuntos (defaultdict), permitindo múltiplos destinos por estado e símbolo.

3. **EXIBIÇÃO DO AUTÔMATO:**

O sistema apresenta:

- A definição formal do autômato no formato:

  L= ({Estados}, {Alfabeto}, δ, Estado_inicial, {Estados_finais}).

- A tabela de transições formatada como:

  δ(estado_origem, símbolo) -> estado_destino.

- Transições ε para estados de aceitação direta também são indicadas.

4. **TESTE DAS PALAVRAS NO AUTÔMATO:**

O algoritmo de simulação percorre símbolo por símbolo da palavra, mantendo um conjunto de **estados ativos**. A cada novo símbolo, os estados ativos são atualizados com os possíveis destinos válidos. A palavra é **aceita** se algum estado final estiver entre os estados ativos após o processamento completo, fornecendo ao usuário, por meio do terminal, se a palavra escrita é aceita ou não.

5. **MENU INTERATIVO:**

O programa exibe um menu com as opções:

1. *Exibir Autômato*
2. *Testar Palavra*
3. *Encerrar Execução*

- **EXEMPLO DE ENTRADA:**

DIGITE OS SÍMBOLOS TERMINAIS (espaçados): 0 1

DIGITE OS ESTADOS NÃO TERMINAIS (espaçados): S B

DIGITE O ESTADO INICIAL: S

DIGITE AS PRODUÇÕES (formato: A -> aB | b | vazio):

S -> 0B

B -> 0B | 1S | 0

- **EXEMPLO DE PALAVRA ACEITA:**

Digite a palavra a ser testada: 0010

'0010' → A PALAVRA É ACEITA PELO AUTÔMATO!

Observações

- O símbolo de ε é representado internamente como “ε”.
- O sistema aceita tanto a produção “vazio” ou o símbolo “ε” explicitamente.
- O estado “F” é reservado para transições que levam a aceitação imediata.