



UNIVERSIDADE FEDERAL DO MARANHÃO

Conversor AFN para AFD

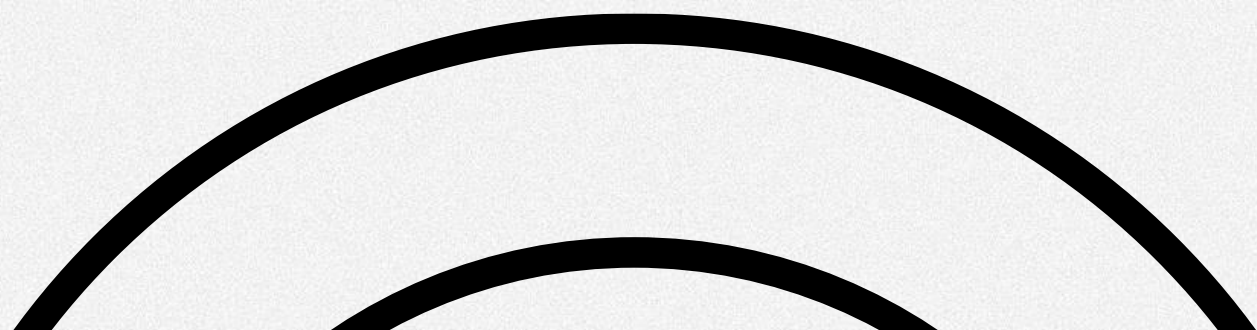
ENGENHARIA DA COMPUTAÇÃO

Alunos: Victor Ferreira e William Valther

Professor: Dr^o Thales Levi

Índice

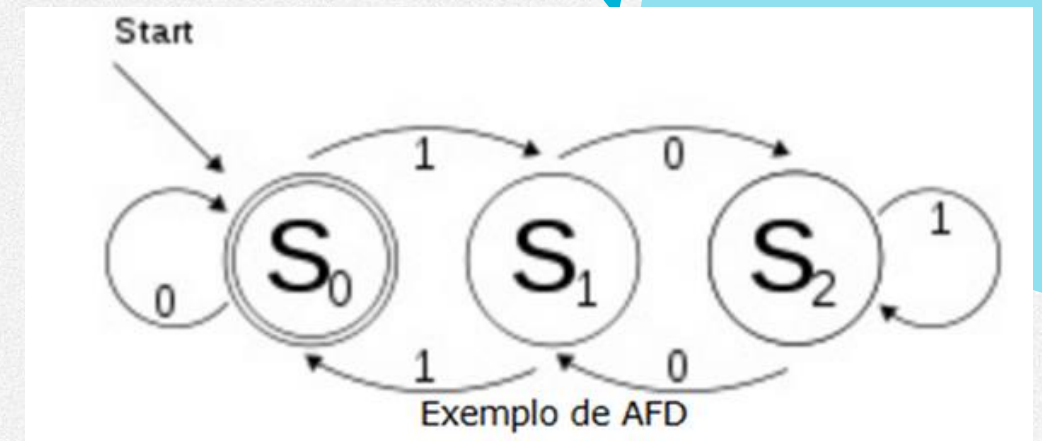
- 03** INTRODUÇÃO
- 04** EQUIVALÊNCIA ENTRE AUTÔMATOS
- 05** ALGORITMO DE CONVERSÃO
- 08** FERRAMENTAS E FUNÇÕES
- 09** SIMULAÇÕES
- 11** CONCLUSÃO



Introdução

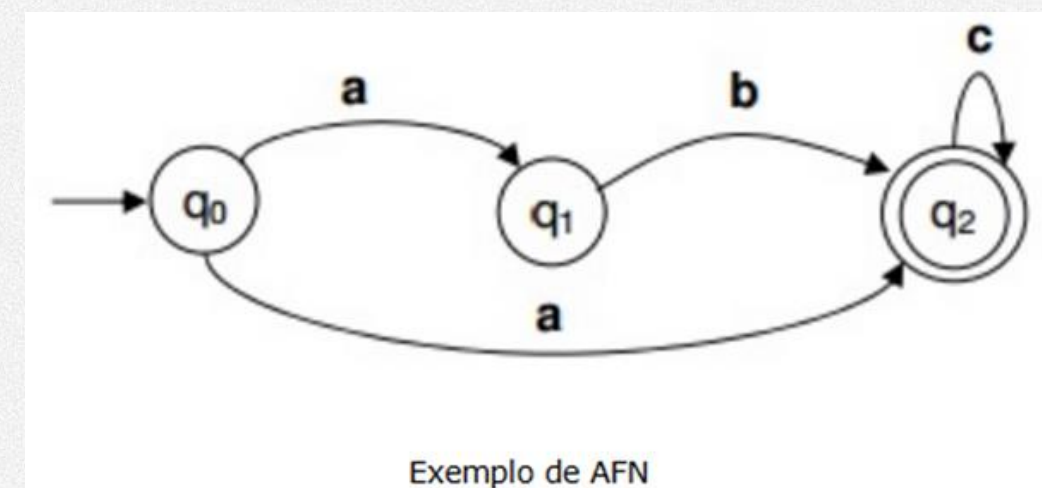
Autômatos Finitos Determinísticos (AFD)

Um autômato finito determinístico — também chamado máquina de estados finita determinística (AFD) — é uma Máquina de estados finita que aceita ou rejeita cadeias de símbolos gerando um único ramo de computação para cada cadeia de entrada.



Autômatos Finitos Não-determinísticos (AFN)

Um AFN, similarmente a um AFD, lê uma cadeia de símbolos de entrada. Para cada símbolo da entrada há uma transição para um novo estado, até que todos os símbolos de entrada sejam lidos, porém existe pelo menos um estado tal que ao ler um mesmo símbolo há mais de uma possibilidade de estado destino. Assim, o próximo estado é um elemento do conjunto das partes dos estados.



Equivalência entre autômatos

- **Teorema:** A classe de autômatos finitos determinísticos é equivalente a classe dos autômatos finitos não determinísticos.
- Para todo AFN existe um AFD equivalente



ALGORITMO DE CONVERSÃO

1º IDENTIFICAR OS COMPONENTES DO AFN

**2º CALCULAR O FECHO-E (SE HOUVER E-
TRANSIÇÕES)**

3º CONSTRUIR OS ESTADOS DO AFD

4. DEFINIR AS TRANSIÇÕES DO AFD

**5. DETERMINAR OS ESTADOS FINAIS DO
AFD**

6. REMOVER REDUNDÂNCIAS

Ferramentas

LINGUAGEM

PYTHON



BIBLIOTECAS

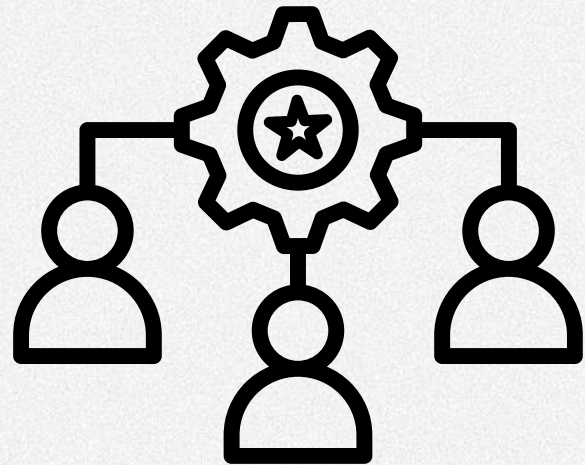
SYS
GRAPHVIZ



EDITOR

VSCODE





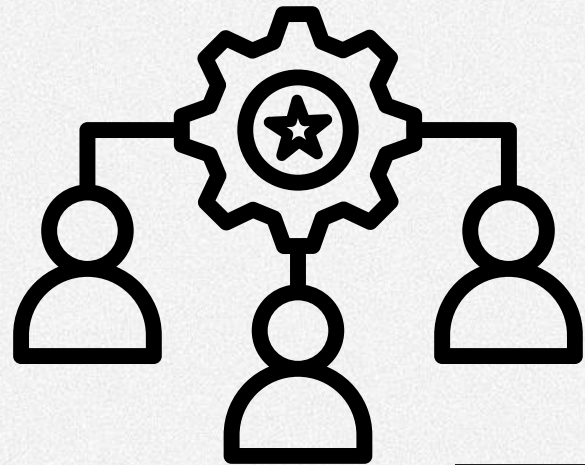
ADICIONARTRANSICAO

FUNÇÕES

```
# Função para adicionar transições ao autômato
# Recebe o autômato, o estado de origem, a letra da transição e o estado de destino
def adicionar_transicao(automato, origem, letra, destino):
    # Inicializa o dicionário para o estado de origem, se ainda não existir
    if origem not in automato["função_programa"]:
        automato["função_programa"][origem] = {}

    # Inicializa o conjunto de destinos para a letra, se ainda não existir
    if letra not in automato["função_programa"][origem]:
        automato["função_programa"][origem][letra] = set()

    # Adiciona o estado de destino à transição
    automato["função_programa"][origem][letra].add(destino)
```

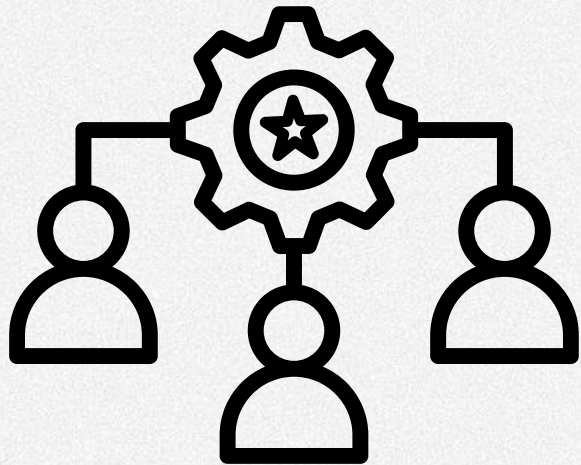
FECHO

FUNÇÕES

```
# Função para calcular o fecho vazio ( $\epsilon$ -fecho) de um conjunto de estados
# Permite alcançar estados via transições vazias ( $\epsilon$ -transições)
def fecho_vazio(automato, estados):
    fecho = set(estados) # Inicializa o fecho com os estados fornecidos
    fila = list(estados) # Fila para processamento iterativo

    while fila:
        estado = fila.pop(0) # Remove o próximo estado da fila
        # Verifica se o estado tem transições vazias ( $\epsilon$ )
        if estado in automato["função_programa"] and "" in automato["função_programa"][estado]:
            estados_novos = automato["função_programa"][estado][""]
            for estado_novo in estados_novos:
                # Adiciona novos estados ao fecho, se ainda não estiverem presentes
                if estado_novo not in fecho:
                    fecho.add(estado_novo)
                    fila.append(estado_novo)

    return fecho
```

AFN_PARA_AFD

FUNÇÕES

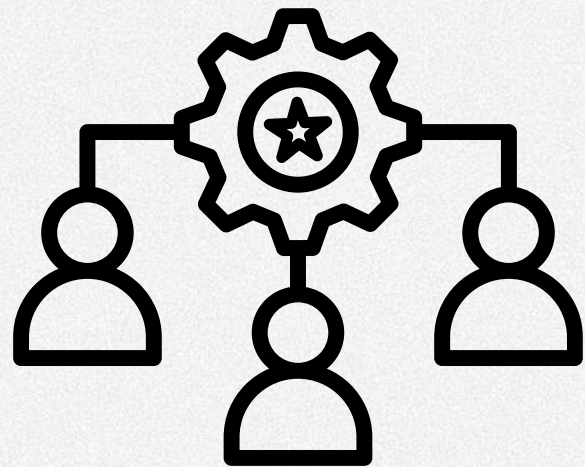
```
# Função para converter um Autômato Finito Não-determinístico (AFN) para Determinístico (AFD)
def afn_para_afd(automato):
    # Inicializa a estrutura do AFD
    afd = {
        "alfabeto": automato["alfabeto"],
        "estados": set(),
        "função_programa": {},
        "estado_inicial": None,
        "estados_finais": set(),
    }

    # Calcula o estado inicial do AFD a partir do fecho vazio do estado inicial do AFN
    estado_inicial = ",".join(sorted(fecho_vazio(automato, {automato["estado_inicial"]})))
    afd["estado_inicial"] = estado_inicial
    afd["estados"].add(estado_inicial)

    fila = [estado_inicial] # Fila para processar estados do AFD
    visitados = set() # Conjunto de estados já processados
```

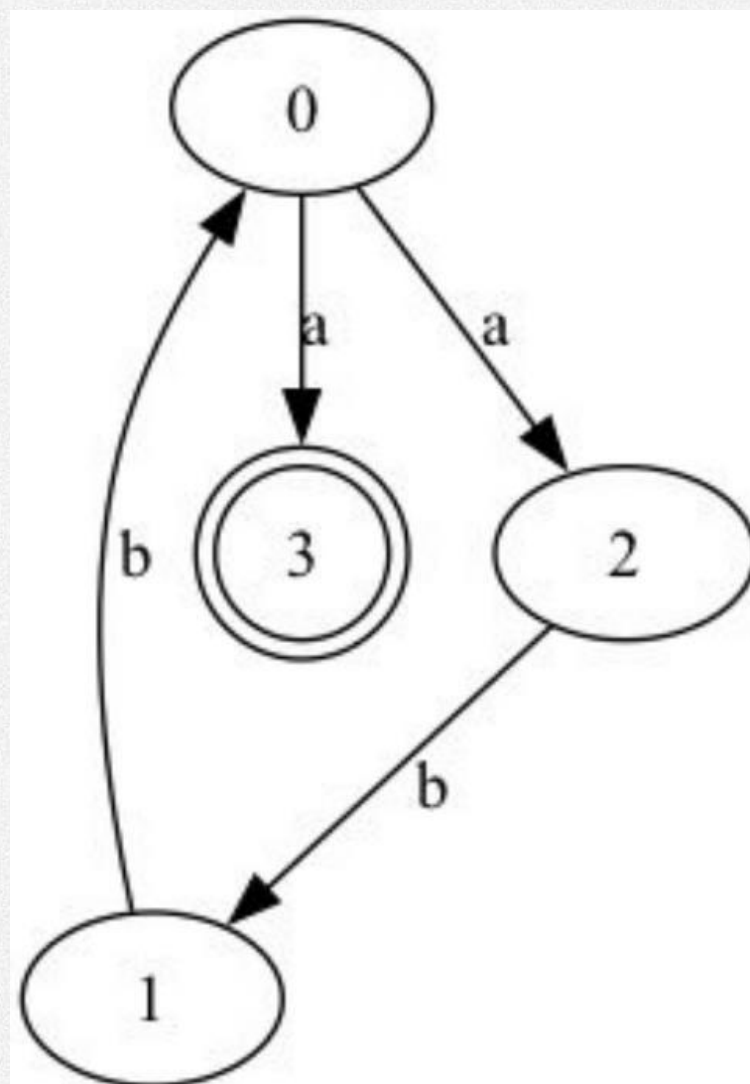
```
while fila:
    estado_atual = fila.pop(0) # Remove o próximo estado da fila
    if estado_atual in visitados:
        continue
    visitados.add(estado_atual) # Marca o estado como visitado
    estados_atuais = estado_atual.split(",") # Divide o estado em múltiplos estados do AFN
    for letra in automato["alfabeto"]:
        novos_estados = set()
        for estado in estados_atuais:
            # Verifica transições para a letra atual
            if estado in automato["função_programa"] and letra in automato["função_programa"][estado]:
                novos_estados.update(automato["função_programa"][estado][letra])
        if novos_estados:
            # Calcula o novo estado do AFD a partir do fecho vazio dos novos estados
            novo_estado = ",".join(sorted(fecho_vazio(automato, novos_estados)))
            if novo_estado not in afd["estados"]:
                fila.append(novo_estado) # Adiciona o novo estado para processamento futuro
                afd["estados"].add(novo_estado)
            # Verifica se o novo estado é um estado final
            if any(est in automato["estados_finais"] for est in novos_estados):
                afd["estados_finais"].add(novo_estado)
            # Adiciona a transição ao AFD
            adicionar_transicao(afd, estado_atual, letra, novo_estado)

return afd
```

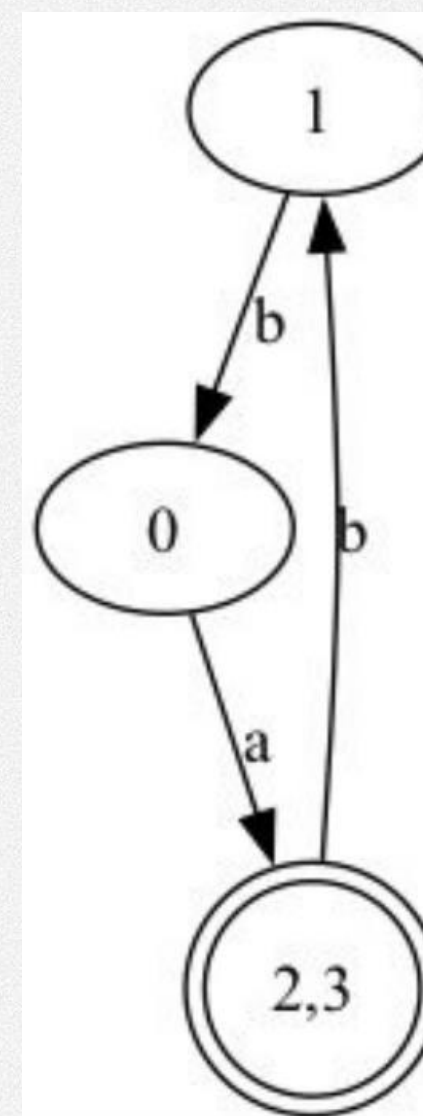



SIMULAÇÕES

AFN



AFD



Conclusão



CONSIDERAÇÕES FINAIS

A conversão de um autômato finito não determinístico (AFN) para um autômato finito determinístico (AFD) é um processo fundamental na teoria da computação, que demonstra a equivalência entre os dois modelos em termos de poder expressivo para reconhecer linguagens regulares. Apesar de sua flexibilidade e simplicidade de definição, o AFN pode ser mais difícil de implementar diretamente em sistemas computacionais devido à sua natureza não determinística. Então faz-se necessário a conversão do AFN para AFD para implementação de determinados modelos.