

# Automatos - Implementação

## AFD.cpp (Autômato Finito Determinístico - AFD)

**Objetivo:** Este programa implementa um **Autômato Finito Determinístico (AFD)**, que recebe um conjunto de estados, alfabeto, transições, estado inicial e estados de aceitação como entrada e verifica se uma cadeia de caracteres é aceita ou rejeitada pelo AFD. O programa é interativo, permitindo que o usuário defina as configurações do AFD e teste várias cadeias de entrada.

### Funcionalidades:

- **Definição do AFD:**
  - O programa solicita ao usuário os estados, o alfabeto, as transições, o estado inicial e os estados de aceitação do AFD.
  - As transições são definidas no formato `estado_atual simbolo estado_destino`, permitindo ao programa configurar as transições do AFD.
- **Verificação de Cadeias:**
  - O programa permite ao usuário testar várias cadeias de caracteres para verificar se são aceitas ou rejeitadas pelo AFD.
  - O AFD processa cada símbolo da cadeia, verificando se existe uma transição válida para o símbolo atual a partir do estado atual.
  - O programa aceita a cadeia se, após processar todos os símbolos, o AFD termina em um estado de aceitação. Caso contrário, a cadeia é rejeitada.

### Entrada:

- **Estados:** Conjunto de estados definidos pelo usuário.
- **Alfabeto:** Conjunto de símbolos válidos no autômato.
- **Transições:** Lista de transições no formato `estado_atual simbolo estado_destino`.
- **Estado Inicial:** Estado inicial do AFD.
- **Estados de Aceitação:** Conjunto de estados finais que aceitam a cadeia.
- **Cadeias de Entrada:** Conjunto de cadeias que serão testadas pelo AFD.

### Saída:

- Para cada cadeia fornecida, o programa imprime se a cadeia é **aceita** ou **rejeitada** pelo AFD.

## Input

```
Digite os estados separados por espaço (ex: q0 q1 q2): q0 q1 q2
Digite os símbolos do alfabeto separados por espaço (ex: 0 1): 0 1
Digite as transições no formato 'estado_atual simbolo estado_destino' (uma por linha). Digite 'fim' para encerrar as transições:
q0 0 q1
q1 1 q2
q2 0 q1
q0 1 q0
fim
Digite o estado inicial: q0
Digite os estados de aceitação separados por espaço: q2
```

## Output:

```
Digite uma cadeia para testar (ou 'sair' para encerrar): 011
A cadeia '011' foi aceita pelo AFD.
Digite uma cadeia para testar (ou 'sair' para encerrar): 0011
A cadeia '0011' foi rejeitada pelo AFD.
```

## Pilha.cpp (Autômato a Pilha - AP)

**Objetivo:** Este programa implementa um **Autômato a Pilha (AP)**, que usa uma pilha para simular transições e manipulações de símbolos enquanto processa uma cadeia de caracteres. O programa verifica se o AP aceita ou rejeita uma cadeia de entrada, utilizando a pilha para controlar as transições baseadas na leitura dos símbolos e nas operações de empilhar/desempilhar.

### Funcionalidades:

- **Definição de Estados e Transições:**

- O programa permite ao usuário definir o número de estados, transições e cadeias de entrada.
- Cada transição é composta por um estado de origem, estado de destino, símbolo de entrada, símbolo do topo da pilha e uma ação que descreve como a pilha deve ser manipulada.
- A pilha pode empilhar ou desempilhar símbolos dependendo da ação definida na transição.
- **Simulação do Autômato:**
  - O AP começa no estado inicial com a pilha contendo um símbolo de fundo (  $Z$  ).
  - Para cada símbolo na cadeia de entrada, o programa tenta aplicar uma transição que corresponda ao símbolo atual e ao símbolo no topo da pilha.
  - Caso uma transição válida seja encontrada, a pilha é modificada conforme a ação definida pela transição, e o autômato avança para o próximo estado.
  - Se a cadeia for processada com sucesso até o final, a cadeia é aceita. Caso contrário, é rejeitada.
- **Interatividade e Saída:**
  - O programa permite que o usuário forneça várias cadeias de entrada para serem testadas.
  - Após testar cada cadeia, o programa imprime se a cadeia foi **aceita** ou **rejeitada** pelo autômato.

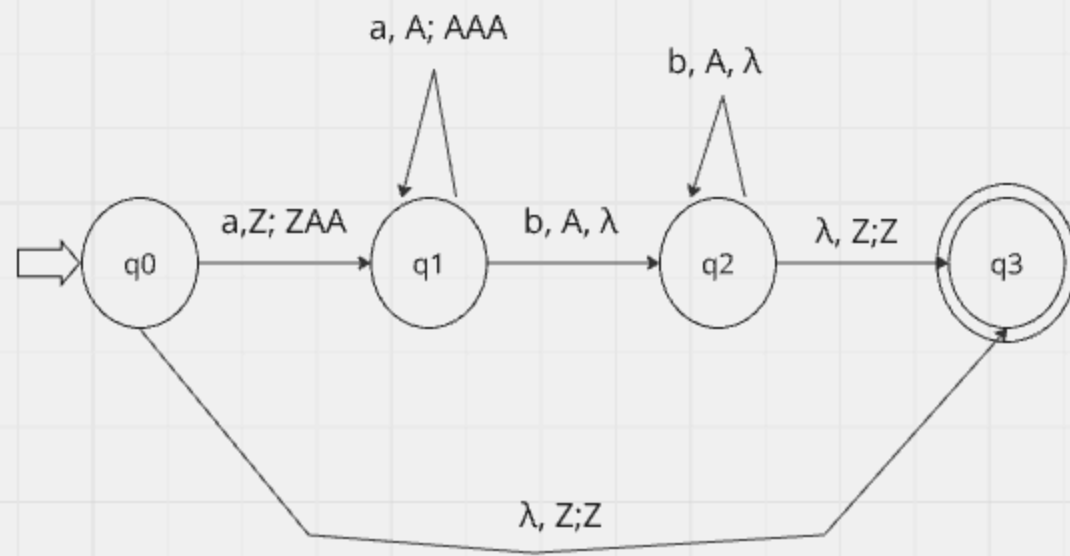
#### Entrada:

- **Número de Cadeias de Entrada:** Quantidade de cadeias a serem testadas.
- **Número de Estados e Transições:** Quantidade de estados e transições a serem definidas.
- **Transições:** Cada transição é definida pelo estado de origem, estado de destino, símbolo de entrada, símbolo no topo da pilha e a ação de manipulação da pilha.
- **Cadeias de Entrada:** As cadeias de caracteres que serão testadas pelo autômato.

#### Saída:

- Para cada cadeia fornecida, o programa imprime se a cadeia foi **aceita** ou **rejeitada** pelo AP, e também imprime o estado da pilha durante o processamento da cadeia.

## Representando o seguinte autômato:



input

```

2
4 6
0 1 a Z ZAA
0 3 ~ Z Z
1 1 a A AAA
1 2 b A ~
2 2 b A ~
2 3 ~ Z Z
abb

```

Output:

Case 1:

Z  
ZAA  
ZA  
Z  
Z  
ACCEPTED

Case 2:

Z  
Z  
ACCEPTED