# Manual de Usuário: Conversão de AFϵ para AFN

## Introdução

Este documento fornece uma explicação detalhada de um programa em Python que converte um Autômato Finito Não-determinístico com transições ε (AFϵ) em um Autômato Finito Não-determinístico (AFN). A documentação descreve a funcionalidade do programa, explica cada passo do código e apresenta exemplos e resultados.

## Objetivo

O programa serve para converter um AFϵ em um AFN, eliminando as transições ε e recalculando os estados de destino para cada transição. Isso é útil para simplificar a análise e implementação de autômatos finitos em sistemas computacionais.

## Passo a Passo do Código

### 1. Importação de Bibliotecas

As bibliotecas *‘defaultdict’* e *‘Digraph’* são importadas. *‘defaultdict’* é utilizada para criar dicionários com valores padrão, e *‘Digraph’* da biblioteca *‘graphviz’* é usada para visualização gráfica dos autômatos.

### 2. Função ‘fecho\_epsilon’

Esta função calcula o fecho-ε de um estado, que é o conjunto de todos os estados alcançáveis a partir de um estado dado através de transições ε. Ela utiliza uma pilha para realizar uma busca em profundidade.

### 3. Função ‘afe\_afn’

A função ‘afe\_afn’ converte um AFϵ em um AFN. Ela realiza os seguintes passos:

- Calcula o fecho-ε para todos os estados (fecho\_epsilon\_todos).

- Cria novas transições sem ε.

- Determina os novos estados finais considerando os fechos-ε.

- Retorna o novo AFN.

### 4. Função ‘imprimir\_automato’

Esta função imprime de forma legível a definição de um AFN, mostrando seus estados, alfabeto, estado inicial, estados finais e transições.

### 5. Função ‘graficos’

Esta função gera um gráfico visual de um autômato usando a biblioteca *‘graphviz’*. Ela cria um diagrama de estados com transições rotuladas.

### 6. Utilização do Programa

Usuário deve passar um Autômato Finito com movimentos vazios (AFε) para o programa no formato do exemplo abaixo. Obrigatoriamente as movimentações vazias devem ser representadas pelo símbolo epsilon (ε).

ex1\_afn\_epsilon = {

'estados': {'q0', 'q1'},

'alfabeto': {'a', 'b'},

'transicoes': {

'q0': {'a': {'q0'}, 'ϵ': {'q1'}},

'q1': {'b': {'q1'}}

},

'inicial': 'q0',

'finais': {'q1'}

}

Este autômato deverá ser passado a função ‘afe\_afn’ que irá realizar os passos 1 a 3 listados anteriormente realizando a conversão para o Autômato Finito Não-determinístico (AFN).

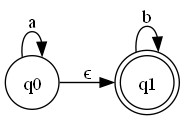
Para a apresentação da conversão serão utilizadas as funções dos passos 4 e 5, ‘imprimir\_automato’ e ‘graficos’. Como o nome já fala, a função ‘imprimir\_automato’ irá receber o autômato e imprimir sua quíntupla, mas de forma organizada para melhor entendimento do usuário. A função ‘graficos’ deverá receber o autômato e gerar a representação do Grafo.

## Resultados

Seguem resultados dos três exemplos utilizados no trabalho.

### Exemplo 1

#### AFϵ

Estados: {'q0', 'q1'}

Alfabeto: {'a', 'b'}

Estado Inicial: q0

Estados Finais: {'q1'}

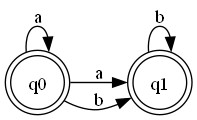
Transições:

δ(q0, a) -> q0

δ(q0, ϵ) -> q1

δ(q1, b) -> q1

#### AFN Resultante

Estados: {'q0', 'q1'}

Alfabeto: {'a', 'b'}

Estado Inicial: q0

Estados Finais: {'q1'}

Transições:

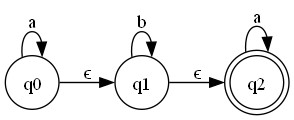
δ(q0, a) -> q0

δ(q0, b) -> q1

δ(q1, b) -> q1

### Exemplo 2

#### AFϵ

Estados: {'q2', 'q1', 'q0'}

Alfabeto: {'a', 'b'}

Estado Inicial: q0

Estados Finais: {'q2'}

Transições:

δ(q0, a) -> q0

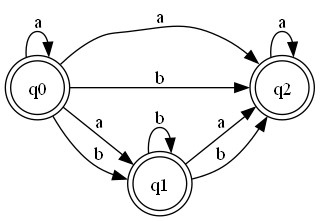
δ(q0, ϵ) -> q1

δ(q1, b) -> q1

δ(q1, ϵ) -> q2

δ(q2, a) -> q2

#### AFN Resultante

Estados: {'q2', 'q1', 'q0'}

Alfabeto: {'a', 'b'}

Estado Inicial: q0

Estados Finais: {'q2', 'q1', 'q0'}

Transições:

δ(q2, a) -> q2

δ(q1, a) -> q2

δ(q1, b) -> q2

δ(q1, b) -> q1

δ(q0, a) -> q2

δ(q0, a) -> q1

δ(q0, a) -> q0

δ(q0, b) -> q2

δ(q0, b) -> q1

### Exemplo 3

#### AFϵ

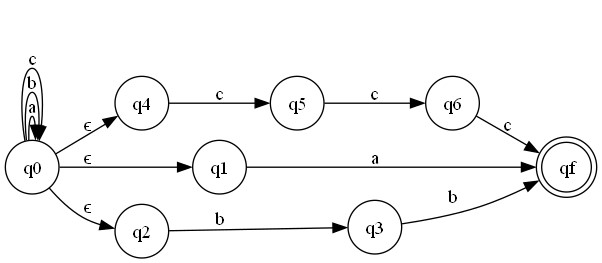
Estados: {'q4', 'q5', 'q0', 'q3', 'q1', 'q6', 'q2', 'qf'}

Alfabeto: {'b', 'c', 'a'}

Estado Inicial: q0

Estados Finais: {'qf'}

Transições:

δ(q0, a) -> q0

δ(q0, b) -> q0

δ(q0, c) -> q0

δ(q0, ϵ) -> q4

δ(q0, ϵ) -> q1

δ(q0, ϵ) -> q2

δ(q1, a) -> qf

δ(q2, b) -> q3

δ(q3, b) -> qf

δ(q4, c) -> q5

δ(q5, c) -> q6

δ(q6, c) -> qf

#### AFN Resultante

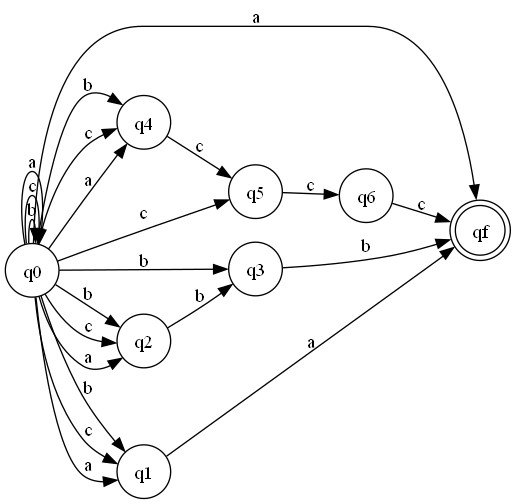
Estados: {'q4', 'q5', 'q0', 'q3', 'q1', 'q6', 'q2', 'qf'}

Alfabeto: {'b', 'c', 'a'}

Estado Inicial: q0

Estados Finais: {'qf'}

Transições:

δ(q4, c) -> q5

δ(q5, c) -> q6

δ(q0, b) -> q3

δ(q0, b) -> q1

δ(q0, b) -> q4

δ(q0, b) -> q0

δ(q0, b) -> q2

δ(q0, c) -> q4

δ(q0, c) -> q1

δ(q0, c) -> q5

δ(q0, c) -> q0

δ(q0, c) -> q2

δ(q0, a) -> q1

δ(q0, a) -> q4

δ(q0, a) -> q0

δ(q0, a) -> qf

δ(q0, a) -> q2

δ(q3, b) -> qf

δ(q1, a) -> qf

δ(q6, c) -> qf

δ(q2, b) -> q3

## Conclusão

Este programa facilita a conversão de AFϵ para AFN, eliminando as transições ε e recalculando os estados de destino. Ele é útil para simplificar a análise de autômatos finitos e pode ser adaptado para diferentes definições de AFϵ. As funções de visualização ajudam a entender melhor a estrutura e o comportamento dos autômatos.