AFD
AFND
ER

Carlos Timóteo, M. Sc. CAPM

Motivação

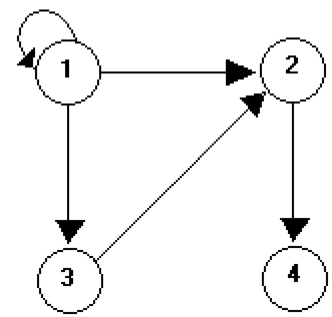
- Como determinar se uma *substring* aparece num determinado texto;
- Como determinar se uma expressão foi formada corretamente

  (5+1) – 8   int main(int argc, char* argv){ return 0; }

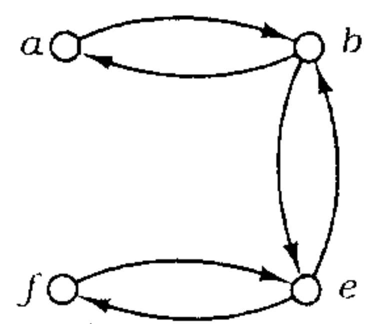
- Como procurar por uma *substring* em arquivos de texto no computador;

Definições Iniciais

- Grafos dirigidos: composto por arcos e vértices.
  - G = {(1,1),(1,2),(1,3),(2,4),(3,2)}

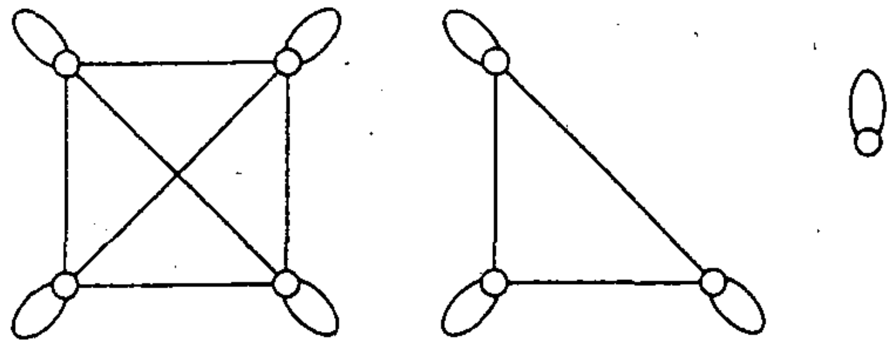


- Relação reflexiva (laços): (n,n)
- Relação simétrica (círculos):
  - (a,b) (b,a)
- Relação anti-simétrica



Definições Iniciais

- Relação de Equivalência: Simétrica e reflexiva



- Caminho e comprimento de um caminho:
  - (a, ..., d) = (a,b,c,d)   comp=4
- Conjuntos finitos, infinitos e equinumerosos.

Alfabetos e Linguagens

- **Estrela de Kleene**: Denotada por L* é o conjunto de todas as *strings* obtidas pela concatenação de zero ou mais *strings* de L.
  - Se L={01,0,100}, então 01001100000 pertence a L*
- **Linguagem**: Conjunto de todas as strings sobre um alfabeto.

Expressões Regulares

- **Expressão regular**: Descreve uma linguagem finita ou infinita de elementos exclusivamente por meio de símbolos únicos e *. Todas as linguagens regulares sobre um alfabeto podem ser descritas por expressões regulares.

AFD – Definição Formal

Um Autômato Finito Determinístico (AFD) M é uma 5-upla:

$$M = (Q, \Sigma, \delta, q_0, F), \text{ onde}$$

Q: conjunto finito de estados do autômato;

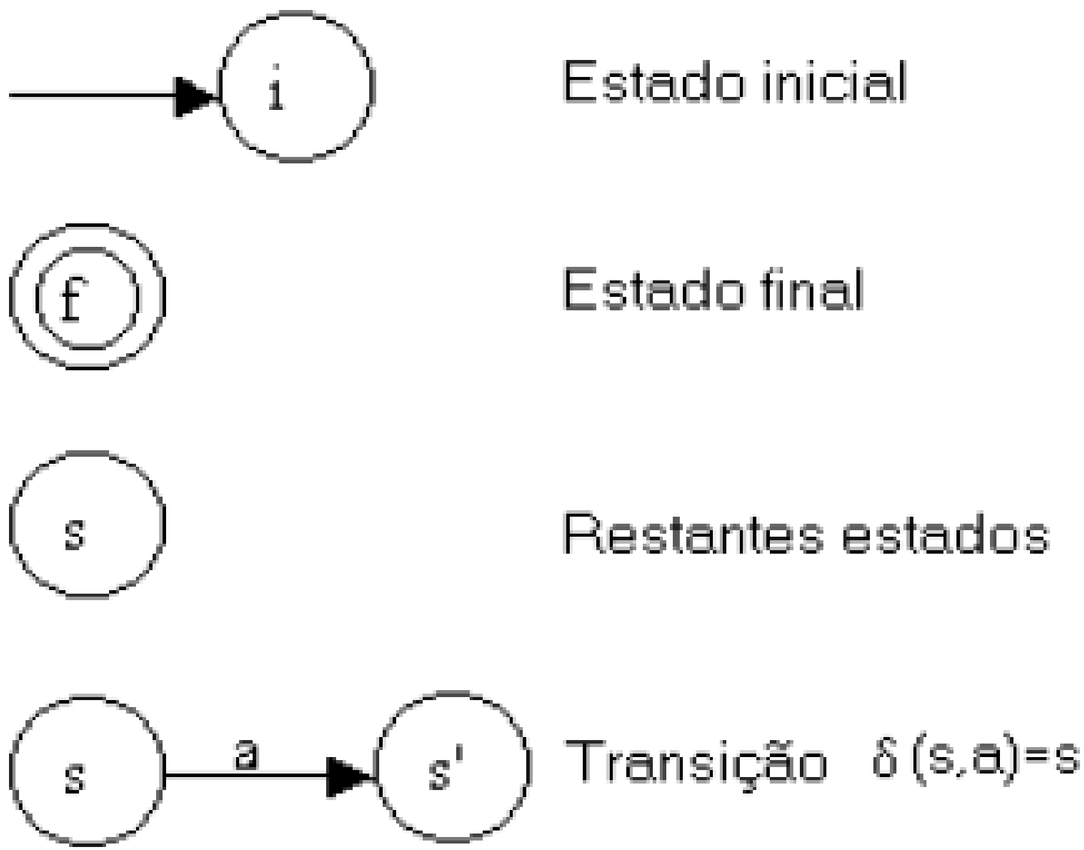
Σ: alfabeto de símbolos de entrada;

δ: função programa ou função de transição (parcial)
$\delta: Q \times \Sigma \rightarrow Q$. Significa dizer que permanecendo em um estado e lendo um símbolo do alfabeto faz o autômato passar para outro estado ou mesmo ficar no mesmo

q_0: estado inicial ($q_0 \in Q$)

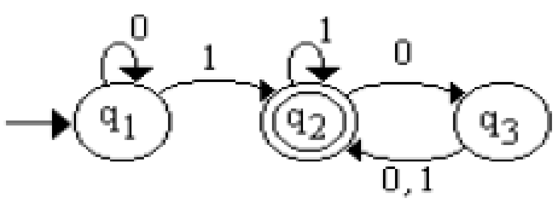
F: conjunto de estados finais ou estados de aceitação ($F \subseteq Q$)

AFD – Representação Gráfica



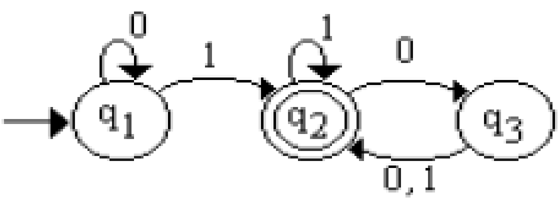
AFD – Representação Gráfica

Um autômato finito M_1: (diagrama de estados)



M_1 tem **3 estados**, q_1, q_2, q_3 ;
M_1 **inicia** no estado q_1 ;
M_1 tem um **estado final**, q_2 ;
Os arcos que vão de um estado p/ outro chamam-se **<u>transições</u>**.

AFD – Exemplo Funcionamento



Exemplo: entrada 1101

1. Inicia no estado q_1.
2. Lê 1, segue transição de q_1 p/ q_2.
3. Lê 1, segue transição de q_2 p/ q_2.
4. Lê 0, segue transição de q_2 p/ q_3.
5. Lê 1, segue transição de q_3 p/ q_2.
6. Pára c/ saída reconhece.

AFD - Propriedades

- Um AFD nunca entra em loop infinito
- Novos símbolos da entrada são lidos a cada aplicação da função programa, o processo de reconhecimento de qualquer cadeia pára de duas maneiras: **aceitando** ou **rejeitando** uma entrada.

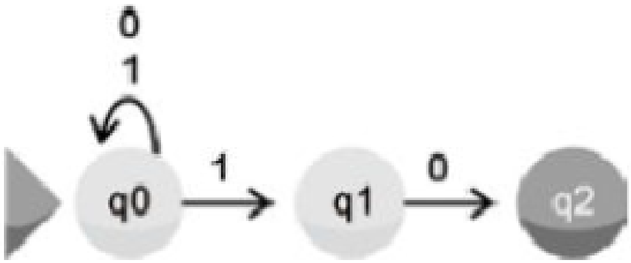
Autômatos Finitos Não-Determinísticos

- Adiciona o não-determinismo aos autômatos finitos.
  - Capacidade de mudar de estado de forma apenas parcialmente determinada pelo estado corrente e pelo símbolo de entrada.
  - Podemos ter zero, uma ou mais transições de estado com o mesmo símbolo de entrada.

AFD vs AFND

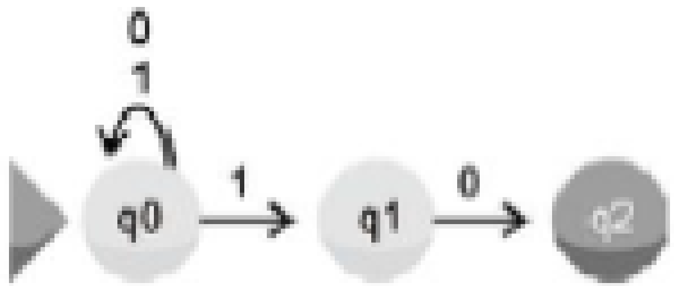
- Determinístico
  - Transições bem definidas
  - Função de transição leva a um único estado
  - Sequência de estados é única para cada palavra
- Não-determinístico
  - Transições ambíguas
  - Função de transição leva a vários estados alternativos
  - Várias sequência possíveis

Autômatos Finitos Não Determinísticos



- Um estado pode ter zero, um ou mais arcos "saindo" para cada símbolo do alfabeto;
- zero, um ou mais arcos podem sair de cada estado rotulados com λ.

Representações de um AFND



δ	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\varnothing$
$^* q_2$	$\varnothing$	$\varnothing$

AFND – Definição Formal

Um Autômato Finito Não-Determinístico (AFND) é uma 5-tupla onde:

1. Q é um conjunto finito de estados;
2. Σ é um alfabeto finito;
3. $\delta : Q \times \Sigma_\lambda \rightarrow P(Q)$ é a função de transição;
4. $q_0 \in Q$ é o estado inicial; e
5. $F \subseteq Q$ é o conjunto de estados de aceitação.

Uma sequência de entrada a,b,c...d é aceita por um AFND se existe uma sequência de transições, correspondendo a sequencia de entrada, que leva do estado inicial a algum dos estados finais.

Equivalência AFND/AFD

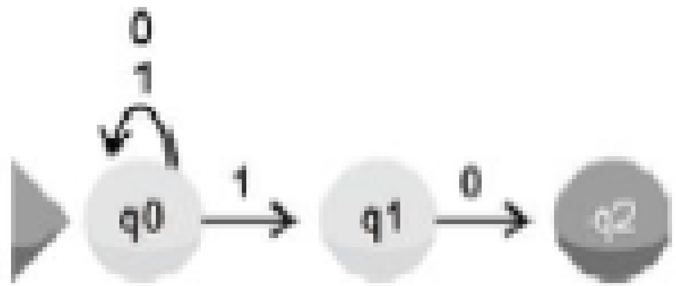
- De um AFD é possível criar um AFND equivalente?
  - Trivial de mostrar
  - Basta criar um AFND cuja função leva a conjuntos unitários

- De um AFND é possível criar um AFD equivalente?
  - Dado **M = (T, Q, δ, q_0, F)** não-determinístico, construir **M'** determinístico
  - Veremos como fazer...

Para Hoje

- Expressões Regulares
- Transformações
  - AFD -> AFND
  - AFND -> AFD
  - AFND -> AFD -> ER
  - ER -> AFD -> AFND

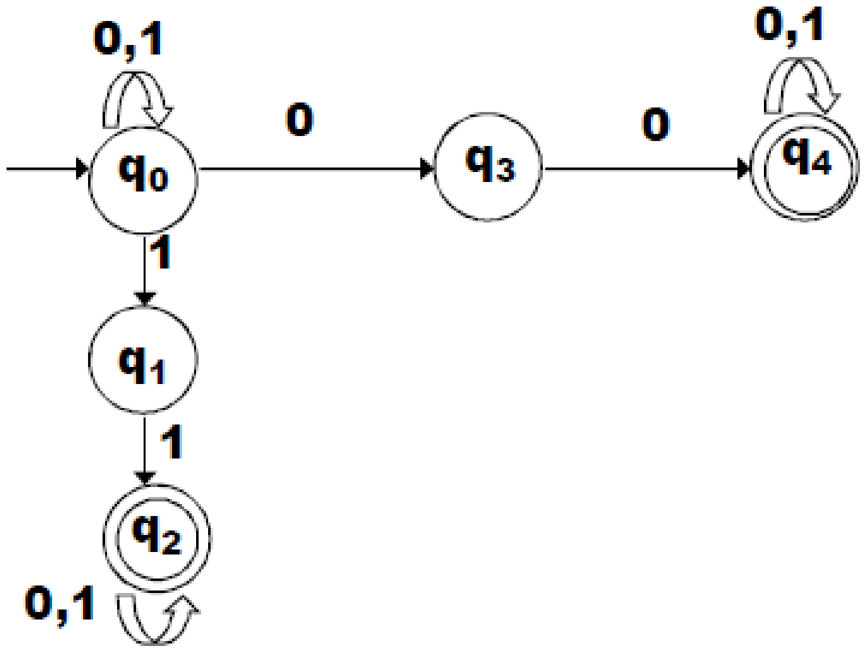
AFND -> AFD
AFD -> AFND



δ	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\varnothing$
$* q_2$	$\varnothing$	$\varnothing$

AFND -> AFD

Transformar de AFND para AFD



AFNDe

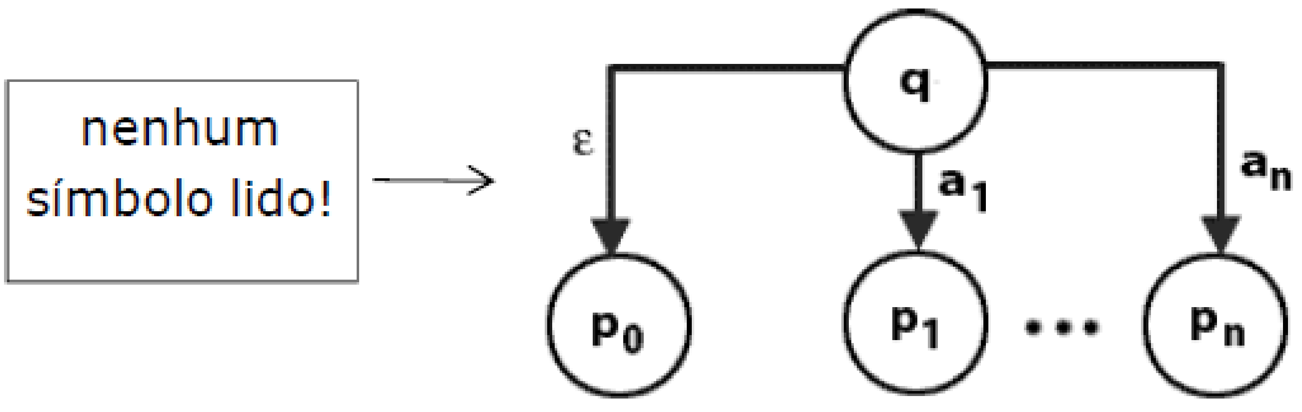
- Uma extensão do formalismo AFND
  - A diferença é que permite movimentos vazios

- Movimento vazio (transição ε)
  - Uma transição <u>sem leitura de símbolo</u>
  - Transição não obrigatória
  - A fita não se altera

AFNDe

A diferença para os AFNDs é a função de transição

- Além dos símbolos, agora também está definida para ε (ausência de símbolo)

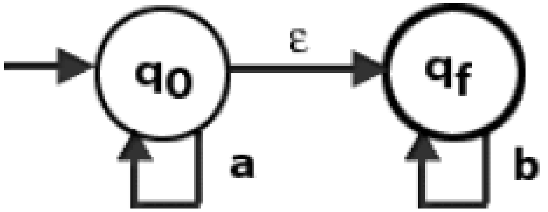


AFNDe

Exemplo

- ACEITA(M) = {w | todo a antecede todo b}

M:



δ:

	a	b	ε
$q0$	{q0}	{ }	{qf}
qf	{ }	{qf}	{ }

Linguagens Regulares

- Todos os formalismos reconhecedores foram vistos
  - Autômatos Finitos Determinísticos
  - Autômatos Finitos Não-determinísticos
  - Autômatos Finitos com Movimentos e

- Expressões Regulares é um formalismo denotacional

Expressões Regulares

- As expressões regulares são utilizadas principalmente como descritores de lingaugens, ou seja, a partir destas expressões podemos identificar uma linguagem regular e dada uma linguagem podemos escrevê-la de forma simplificada usando expressões (se a linguagem for regular).

Expressões Regulares

- Utilização
  - Localizar cadeias em um texto
  - Para criar analisadores léxicos, que são componentes fundamentais dos compiladores
- Assim como uma expressão aritmética representa um número natural: (10 + 5) x 7
- Uma expressão regular representa uma linguagem: (0 + 1) . 0*

Expressões Regulares

- T = {c, d}
- L = {palavra que tem “cc” como subpalavra}

$$(c+d)^*(cc)(c+d)^*$$

ER

Definição formal: Seja Σ um alfabeto

1. Se $a \in \Sigma$, então **a** é uma expressão regular.
2. Se λ é a palavra nula, então $\boldsymbol{\lambda}$ é uma expressão regular.
3. Se $\varnothing$ é o conjunto vazio, então $\boldsymbol{\varnothing}$ é uma expressão regular.
4. Se R_1 e R_2 são expressões regulares, então $(R_1 + R_2)$ e $(R_1 \cdot R_2)$ são expressões regulares.
5. Se R_1 é uma expressão regular, então $(R_1{}^*)$ é uma expressão regular.

ER

- Na expressão $(\mathbf{0} + \mathbf{1}) \cdot \mathbf{0}^*$:
  - $\mathbf{0}$ representa o conjunto $\{0\}$
  - $\mathbf{1}$ representa o conjunto $\{1\}$
  - $(\mathbf{0} + \mathbf{1})$ representa o conjunto $\{0\} \cup \{1\} = \{0, 1\}$
  - $\mathbf{0}^*$ representa $\{0\}^*$

- Então $(\mathbf{0} + \mathbf{1}) \cdot \mathbf{0}^*$ representa a linguagem:

$$\{uv: u \in \{0, 1\} \text{ e } v = 0^n, n \geq 0\}$$

ER - Operadores

- União
  - L = {001, 110} e M= {e, 11, 110}
  - L ∪ M = {001, 110, e, 11}
  - L(E+F) = L(E) ∪ L(F)

- Concatenação
  - L = {001, 110} e M = {e, 11, 110}
  - L.M (com um ponto) ou LM (sem ponto), onde LM = {001, 00111, 001110, 110, 11011, 110110}
  - L(E.F) = L(E).L(F)

- Fechamento de Kleene
  - L ={00, 11}
  - L* = {e, 0011, 001100, 11110011, ...}.
  - L(E*) = (L(E))*