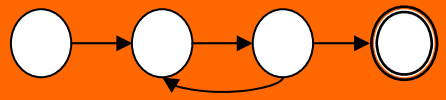
Automata e Linguagens Formais

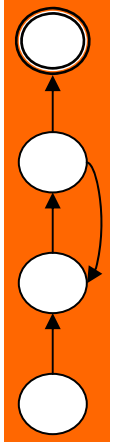
CTC 34



2

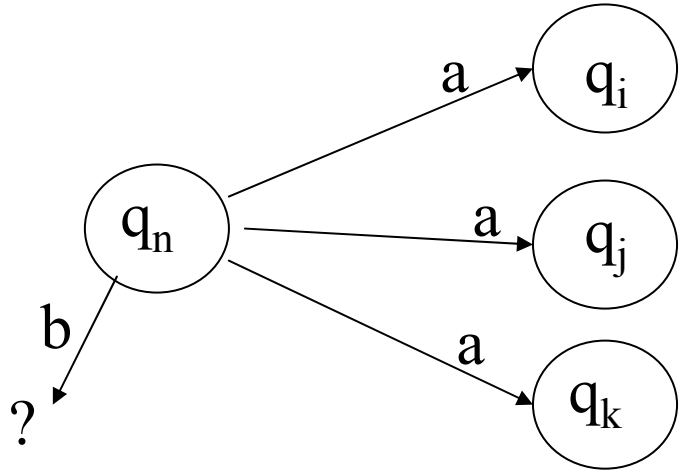
AFs não-determinísticos
AFNDs com transições ε
AFs e expressões regulares
Teorema de Kleene

Prof. Carlos H. C. Ribeiro
carlos@ita.br
Ramal: 5895 Sala: 106



AFs Não-Determinísticos

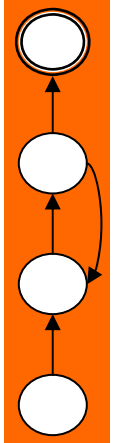
- **Zero, um ou mais** próximos estados podem resultar de computação.



$\delta(q_n, a) = \{q_i, q_j, q_k\}$

$\delta(q_n, b) = \emptyset$

- Um dado AFN **equivale** a algum AF (embora mais "complicado").
- Vantagens:
  - simplifica prova de teoremas relacionados com AFs.
  - facilita projeto de autômatos reconhecedores de linguagens.



AFs Não-Determinísticos: Definição

Um AFND é uma **quíntupla** $M = (Q, \Sigma, \delta, q_0, F)$

- Q = conjunto finito de estados
- Σ = alfabeto
- δ = relação de transição $\subset$ Q x Σ x Q
- q_0 = estado inicial
- F $\subseteq$ Q = conjunto de estados finais

M: Q = $\{q_o, q_1, q_2\}$

$\Sigma = \{a, b\}$

F = $\{q_2\}$

δ	**a**	**b**
$\mathbf{q_0}$	$\{q_0\}$	$\{q_0,q_1\}$
$\mathbf{q_1}$	$\emptyset$	$\{q_2\}$
$\mathbf{q_2}$	$\emptyset$	$\emptyset$

$[q_0, ababb] \vdash [q_0, babb] \vdash [q_0, abb] \vdash [q_0, bb] \vdash [q_0, b] \vdash [q_0, \varepsilon]$

$[q_0, ababb] \vdash [q_0, babb] \vdash [q_1, abb]$

$[q_0, ababb] \vdash [q_0, babb] \vdash [q_0, abb] \vdash [q_0, bb] \vdash [q_1, b] \vdash [q_2, \varepsilon]$

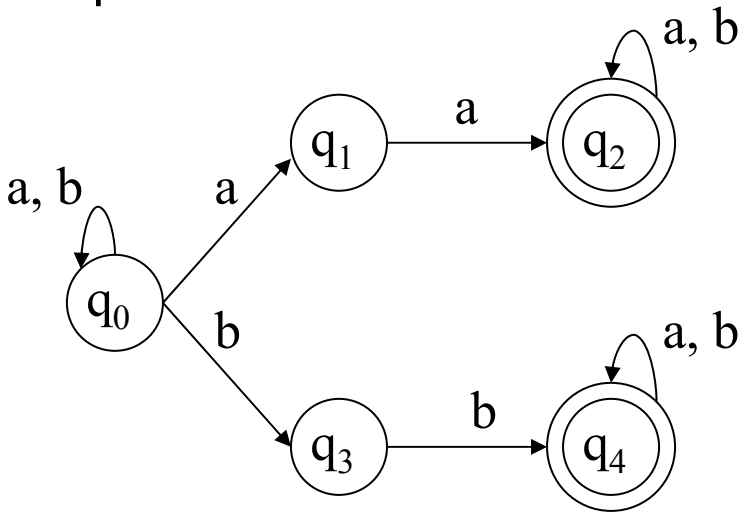
AFNDs: Outras Definições

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFND.

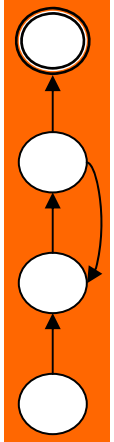
- Uma cadeia *w* é **aceita** por *M* se existir **ao menos uma** computação que processa a cadeia e termina em estado de F.

  No exemplo anterior: ababb é aceita por *M*.
- A **linguagem de *M***, denotada L(*M*), é o conjunto de cadeias em Σ^* aceitas por *M.*

Exemplo:



aceita cadeias contendo subcadeias <u>aa</u> ou <u>bb</u>



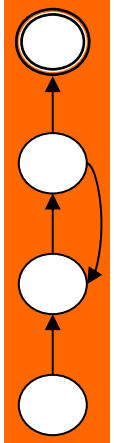
Equivalência entre AFs e AFNDs

Teorema: Seja L uma linguagem aceita por um AFND. Então existe um AF que aceita L.

Prova:

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFND.

Idéia: Construir um AF $M' = (Q', \Sigma, \delta', q_0', F')$ que simula computações realizadas por M. O AF manterá como "estado" o conjunto de todos os estados que podem resultar após uma dada computação de M.



Definamos:

a) Q' = Conjunto de todos os subconjuntos de Q ($Q' = 2^Q$). Note portanto que os estados de M' são então **conjuntos** de estados de M.

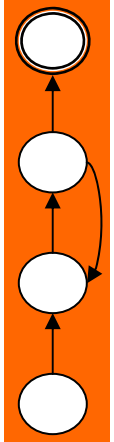
Notação para um elemento de Q': $[q_1, q_2, \ldots q_i]$, onde $q_1, q_2, \ldots q_i \in Q$.

b) $q_0' = [q_0]$

c) F' = estados de Q' contendo um estado de F.

d) $\delta'([q_1, q_2, \ldots q_i], a) = [p_1, p_2, \ldots p_j] \Leftrightarrow \delta(\{q_1, \ldots, q_i\}, a) = \{p_1, \ldots p_j\}$,

onde $\delta(\{q_1, \ldots, q_i\}, a)$ representa a aplicação de δ a cada q_k seguida da união dos conjuntos resultantes. Uma transição em M' produz portanto um conjunto que contém todos os possíveis sucessores para cada possível estado de M.



Provemos agora (por indução) que uma computação de cadeia em M' "imita" o comportamento de M, isto é: $\hat{\delta}'(q_0', \mathbf{v}) = [p_1, \ldots p_j] \Leftrightarrow \hat{\delta}(q_0, \mathbf{v}) = \{p_1, \ldots p_j\}$

i) para $|\mathbf{v}| = 0$: $\mathbf{v} = \varepsilon$ e $\hat{\delta}'(q_0', \varepsilon) = q_0' = [q_0]$

ii) suponha válido para $|\mathbf{v}| \leq m$: $\hat{\delta}'(q_0', \mathbf{v}) = [p_1, \ldots p_j] \Leftrightarrow \hat{\delta}(q_0, \mathbf{v}) = \{p_1, \ldots p_j\}$

iii) prova para $|\mathbf{u}| = m+1$, onde $\mathbf{u} = \mathbf{v}a$ e $|\mathbf{v}| = m$:

$\hat{\delta}'(q_0', \mathbf{u}) = \hat{\delta}'(q_0', \mathbf{v}a) = \delta'(\hat{\delta}'(q_0', \mathbf{v}), a)$

Mas $\hat{\delta}'(q_0', \mathbf{v}) = [p_1, \ldots, p_j] \Leftrightarrow \hat{\delta}(q_0, \mathbf{v}) = \{p_1, \ldots p_j\}$ (pela hipótese **ii**)

e $\delta'([p_1, \ldots p_j], a) = [r_1, r_2, \ldots r_k] \Leftrightarrow \delta(\{p_1, \ldots, p_j\}, a) = \{r_1, \ldots r_k\}$ (pela definição **d**)

$\therefore \delta'(\hat{\delta}'(q_0', \mathbf{v}), a) = [r_1, r_2, \ldots r_k] \Leftrightarrow \delta(\hat{\delta}(q_0, \mathbf{v}), a) = \{r_1, \ldots r_k\}$

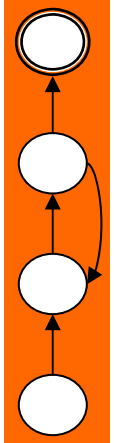
$\therefore \hat{\delta}'(q_0', \mathbf{v}a) = [r_1, r_2, \ldots r_k] \Leftrightarrow \hat{\delta}(q_0, \mathbf{v}a) = \{r_1, \ldots r_k\}$

Finalmente, observe que $\hat{\delta}'(q_0', \mathbf{v}) = [p_1, \ldots p_j] \Leftrightarrow \hat{\delta}(q_0, \mathbf{v}) = \{p_1, \ldots p_j\}$

implica $\hat{\delta}'(q_0', \mathbf{v}) = [p_1, \ldots p_j] \in F' \Leftrightarrow \hat{\delta}(q_0, \mathbf{v}) = \{p_1, \ldots p_j\}$ contém um estado de F.

Portanto, L(M) = L(M').

A volta é trivial.

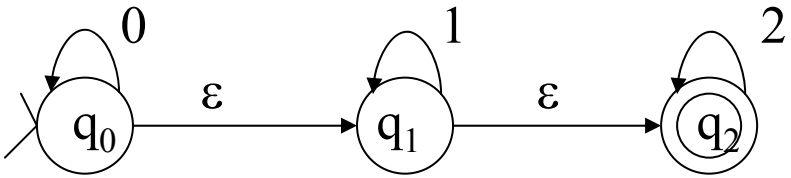


AFNDs com Transições ε (AFND-ε)

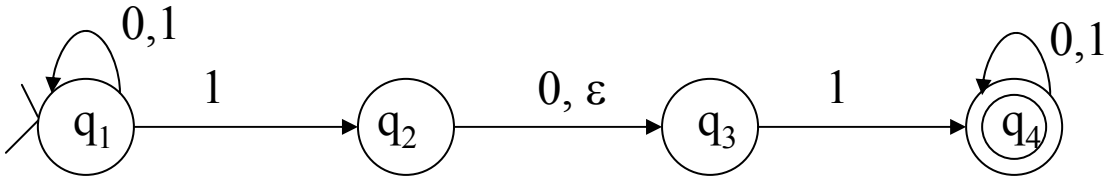
Um AFND com transições ε é um AFND em que
δ = relação de transição ⊂ Q x (Σ∪ε) x Q

Ou seja: um AFND-ε aceita transições "não-forçadas".

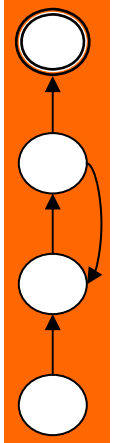
Exemplo 1. Um AFND- ε que aceita um número de zeros (inclusive 0), seguidos por 1´s e 2´s.



Exemplo 2. Q={q1,q2,q3,q4}, Σ={0,1}, q_0 = q1, F={q4}.



δ = ?



Fechamento- ε (ε-*CLOSURE*)

Def.: O **fechamento-ε** (*ε-CLOSURE*) de um estado q é o conjunto de estados p tais que existe um caminho de q a p apenas com rótulos ε.

No exemplo 1:

ε-*CLOSURE* (q_0) = $\{q_0,q_1,q_2\}$

ε-*CLOSURE* (q_0) = $\{q_1,q_2\}$

ε-*CLOSURE* (q_0) = $\{q_2\}$

No exemplo 2:

ε-*CLOSURE*(q_1) = $\{q_1\}$

ε-CLOSURE(q_2)= $\{q_2, q_3\}$

ε-CLOSURE(q_3)= $\{q_3\}$

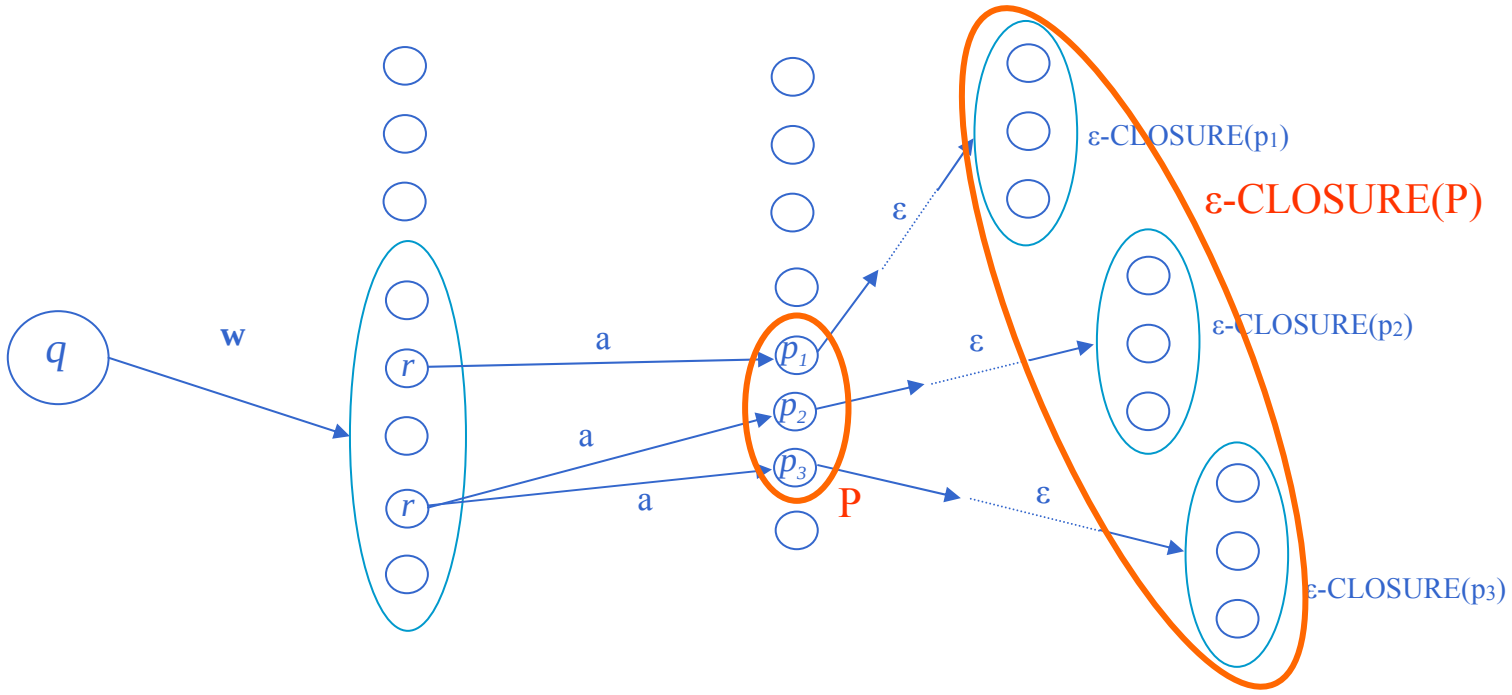
ε-CLOSURE(q_4)= $\{q_4\}$

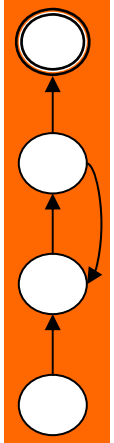
Função de transição estendida em AFND-ε

- Para AFs, tínhamos: $\hat{\delta}(q, \varepsilon) = q$ $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$
- Para AFNDs com transições ε:

$$\hat{\delta}(q, \varepsilon) = \varepsilon - CLOSURE(q) \qquad \hat{\delta}(q, wa) = \varepsilon - CLOSURE(P)$$

onde P={p | para algum r em $\hat{\delta}(q, w)$, p está em $\delta(r, a)$}





Equivalência entre AFNDs e AFNDs com Transições ε

Teorema: Uma linguagem L é aceita por um AFN-ε se e somente se L é aceita por um AFN

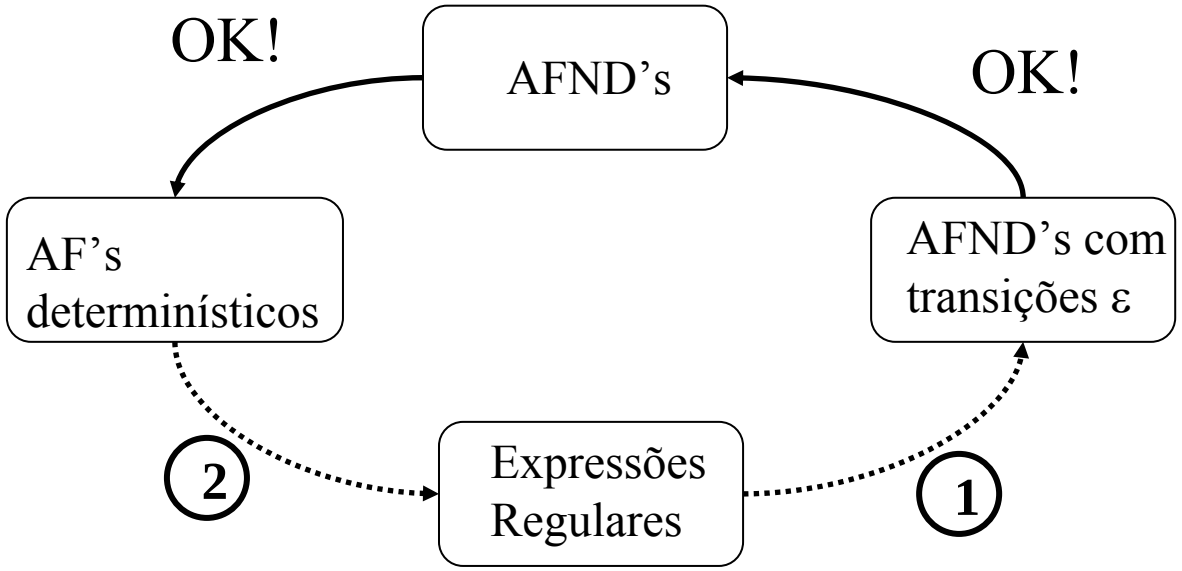
Prova: Hopcroft/Ullman, pág. 26.

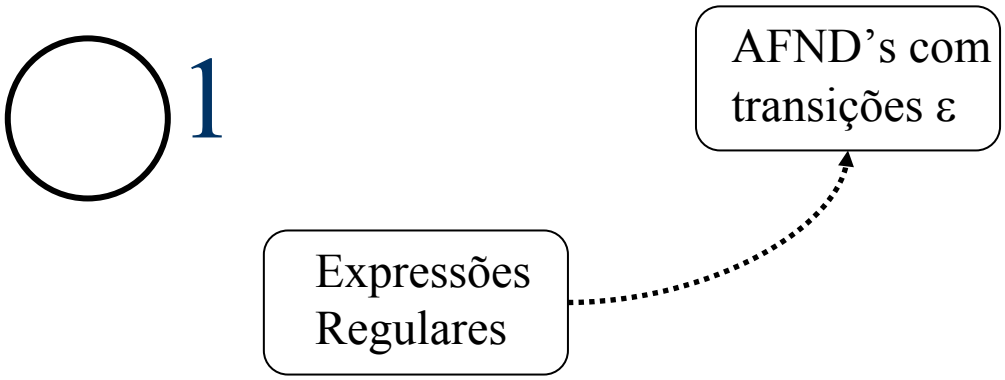
Observe que a volta também vale, trivialmente.

Relação entre AFs e Expressões Regulares

Vamos agora provar que as linguagens aceitas pelos automata finitos são aquelas definidas pelas expressões regulares (ou seja, as linguagens regulares).

Idéia:

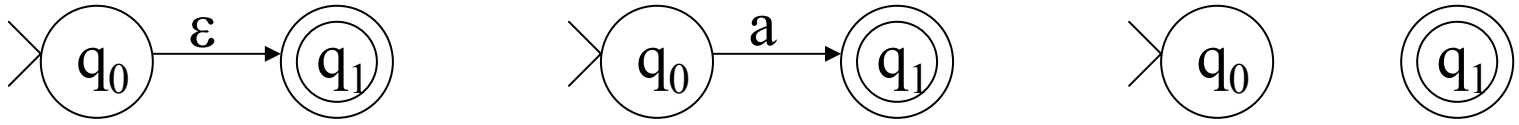


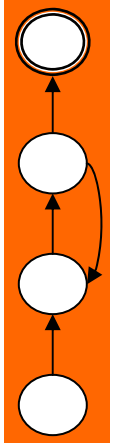


Teorema: Seja r uma expressão regular. Então existe um AFND-ε que aceita L(r).

Prova: Usaremos PIF sobre o número de operadores na expressão r.

1. Base (nenhum operador). Neste caso, r é ε (cadeia vazia), um símbolo a $\in\Sigma$ ou ∅ (nenhuma cadeia aceita). Os seguintes AFND-ε servem, respectivamente:



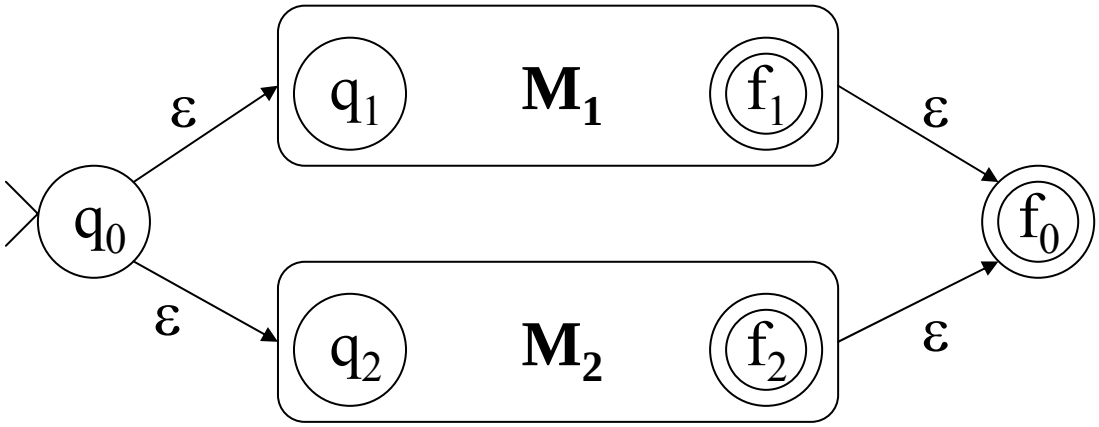


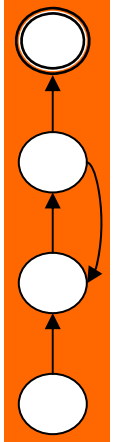
2. Indução. Suponha que o teorema seja válido para expressões regulares r com i-1 operadores.

3. Provemos a validade para i operadores. Por definição, expressões regulares são formadas via $\cup$, concatenação ou estrela de Kleene. Temos portanto três casos a analisar:

3.1 $r = r_1 \cup r_2$, onde r_1 e r_2 tem até i-1 operadores.

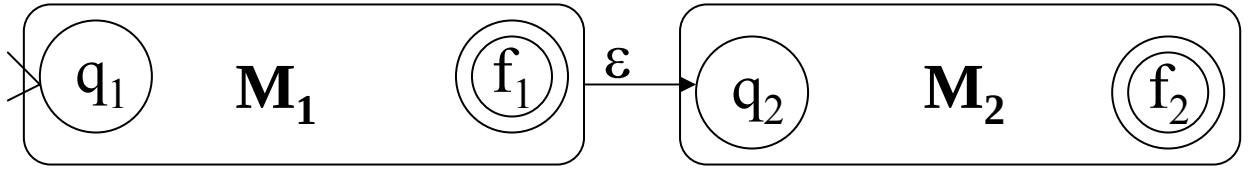
Neste caso, serve o seguinte AFND-ε:





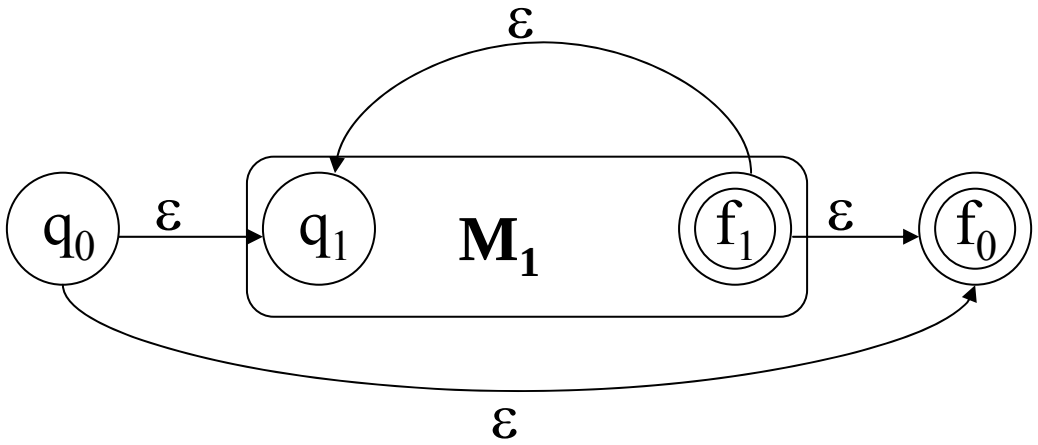
3.2 $r = r_1\ r_2$, onde r_1 e r_2 tem até i-1 operadores.

Neste caso, serve o seguinte AFND-ε:



3.3 $r = r_1^*$, onde r_1 tem até i-1 operadores.

Neste caso, serve o seguinte AFND-ε:



Exercícios

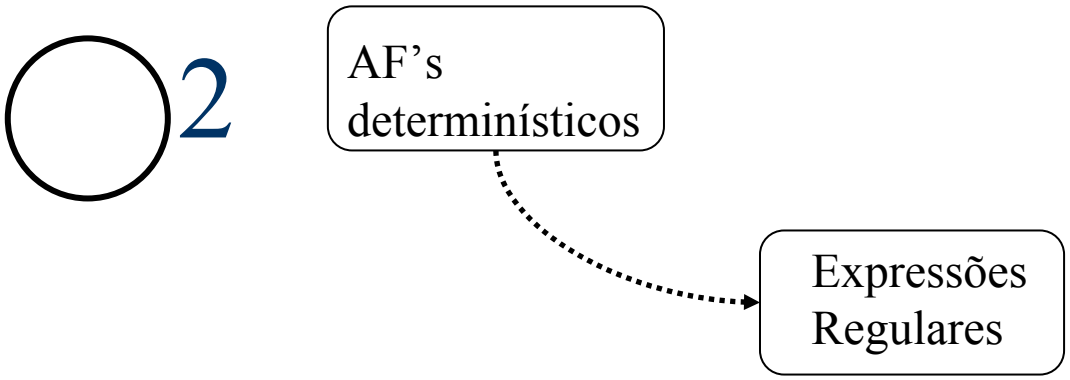
1. Construir AFND-ε's para as seguintes linguagens regulares:

1.1. $\mathbf{01^* \cup 1}$

1.2. $\mathbf{10+(0+11)0^*1}$

2. Construir um AF equiv. ao AFND ({p,q,r,s},{0,1},δ,p,{s}), sendo a função de transição:

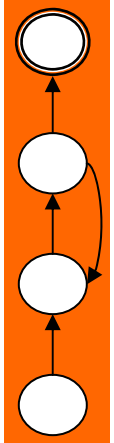
	0	1
p	p,q	p
q	r	r
r	s	-
s	s	s



Teorema: Seja L uma linguagem aceita por um AF. Então L é uma linguagem regular.

Prova (informal): Um mecanismo que gera uma linguagem regular a partir de qualquer AF. A idéia é "reduzir" passo a passo o autômato original, mostrando que o que ele produz ao final é uma expressão regular.

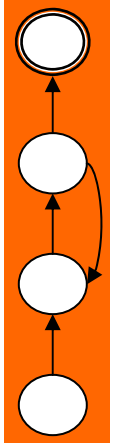
Existe uma prova formal em Hopcroft/Ullman (pág. 33).



Um algoritmo para gerar expressões regulares a partir de um AF

Seja M um automato com $Q=\{q_1,q_2,...,q_n\}$, e seja m o número de estados finais de M (ou seja, card(F)=m). Seja w_{ij} o rótulo (símbolo do alfabeto) associado à transição do estado q_i ao estado q_j.

Inicialmente, faça m cópias $M_1,M_2,...,M_m$ de M, cada uma com um estado final diferente.

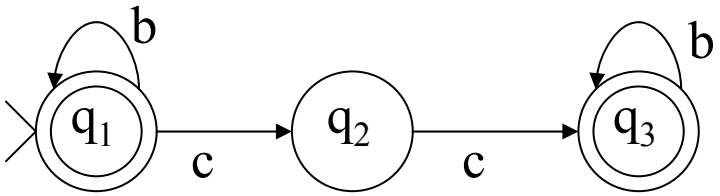


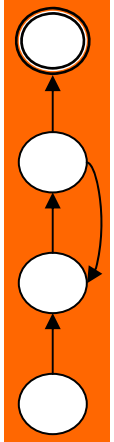
- Para cada M_t faça
  - Escolha um q_i em M_t que não seja inicial ou final em M_t ;
  - Para todo j,k $\neq$ i (mas possivelmente j=k), faça:
    - Se $w_{ji} \neq \varnothing$, $w_{i\,k} \neq \varnothing$, $w_{ii} = \varnothing$ então adicione um arco de q_j a q_k com rótulo $w_{ji}w_{jk}$.
    - Se $w_{ji} \neq \varnothing$, $w_{i\,k} \neq \varnothing$, $w_{ii} \neq \varnothing$ então adicione um arco de q_j a q_k com rótulo $w_{ji}(w_{ii})^* w_{j\,k}$.
    - Se j e k têm arcos w_1, w_2, ..., w_s conectando-os, então substitua todos os arcos por um único $w_1 \cup w_2 \cup \ldots \cup w_s$.
    - Remova o nó q_i e todos os seus arcos incidentes em M_t.

    Até que os únicos nós em M_t sejam o estado inicial e o (único) estado final. Determine então a expressão L_t aceita por M_t

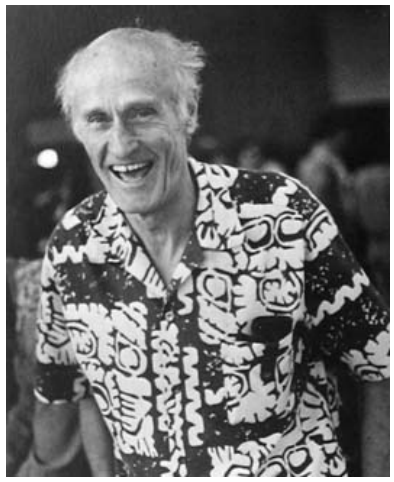
A expressão regular aceita por M é formada pela união: $L_1 \cup L_2 \cup \ldots \cup L_m$

Exemplo:





Terminamos então com um Teorema fundamental da Teoria da Computação:



Stephen Kleene
(1909-1994)

Teorema de Kleene:

Uma linguagem L é aceita por um AF com alfabeto Σ sss L é uma linguagem regular sobre Σ