FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

-- OPERAÇÕES REGULARES --

União, Concatenação e Estrela

Operações Regulares

- Sejam A e B linguagens. São três as operações regulares:
  - União: $A \cup B = \{\omega \mid \omega \in A \text{ ou } \omega \in B\}$
  - Concatenação: $AB = \{\omega\gamma \mid \omega \in A \text{ e } \gamma \in B\}$
  - Estrela: $A^* = \{\omega_1\omega_2 \cdots \omega_k \mid \omega_i \in A \text{ e } k \geq 0\}$
- Considere $L = \{1,01,100\}$ e $M = \{\varepsilon, aa, bba\}$
  - $L \cup M = \{1,01,100, \varepsilon, aa, bba\}$
  - $LM = \{1,01,100,1aa, 01aa, 100aa, 1bba, 01bba, 100bba\}$
  - $ML = \{1, aa1, bba1,01, aa01, bba01,100, aa100, bba100\}$
  - $L^* = \{\varepsilon, 1,01,100,11,101,1100,011,0101,01100, \dots\}$
  - $M^* = \{\varepsilon, aa, bba, aabba, bbabbabba, bbaaa, bbaaabba, \dots\}$

Propriedades das operações regulares

1. $A \cup B = B \cup A$ (comutativa)
2. $A \cup (B \cup C) = (A \cup B) \cup C$ (associativa)
3. $A(BC) = (AB)C$ (associativa)
4. $A(B \cup C) = AB \cup AC$ (distributiva)
5. $(A \cup B)C = AC \cup BC$ (distributiva)
6. $\emptyset A = A\emptyset = \emptyset$ (aniquilador)
7. $\{\varepsilon\}A = A\{\varepsilon\} = A$ (identidade)
8. $A \cup \emptyset = \emptyset \cup A = A$ (identidade)
9. $A \cup A = A$ (idempotência)
10. $(A^*)^* = A^*$ (fechamento)
11. $(\emptyset)^* = \{\varepsilon\}$ (fechamento)
12. $\{\varepsilon\}^* = \{\varepsilon\}$ (fechamento)

Fechamento

- Uma coleção de objetos é fechada sob uma operação se, aplicando-se essa operação a membros da coleção, recebe-se um objeto ainda da coleção
- Exemplo
  - $\mathbb{N}$ é fechado sob a soma
  - $\mathbb{N}$ não é fechado sob a divisão

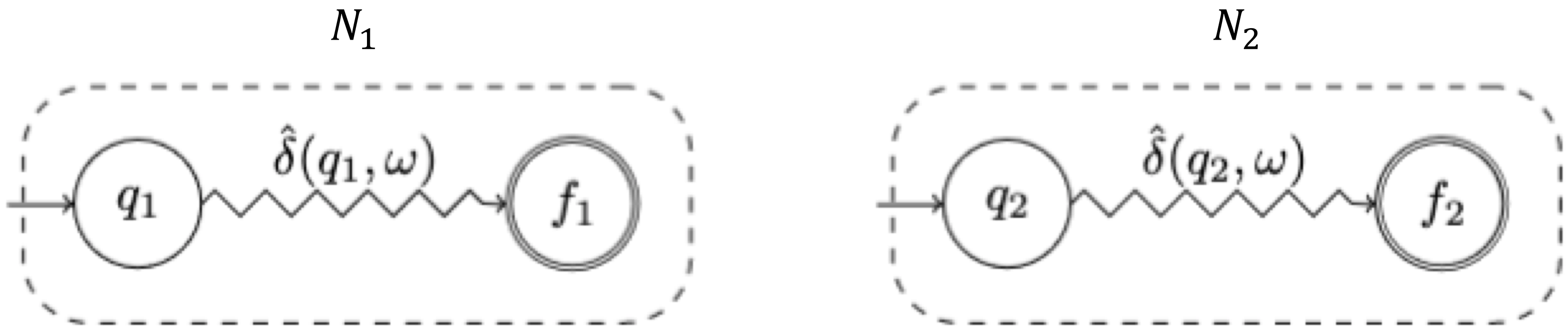
Propriedades de fechamento

- As linguagens regulares são fechadas sob as operações:
  - União
  - Concatenação
  - Estrela
  - Intersecção
  - Complemento ($\bar{A} = \Sigma^{*} \backslash \mathrm{A}$)
  - Diferença ($A \backslash \mathrm{B} = \{\omega \in A \land \omega \notin B\}$)
  - Reverso ($A^{R} = \{\omega^{R} \mid \omega \in A\}$)
- Se tais linguagens são regulares, e uma outra linguagem L é formada apartir de algumas operações, então L também é regular

União de AFNε

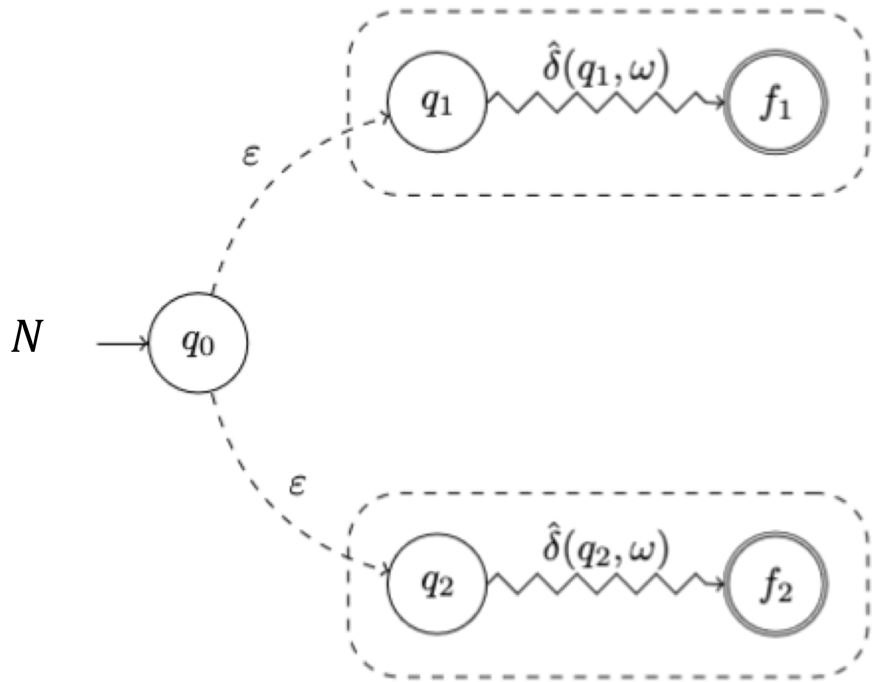
Fecho da União

- Se A_1 e A_2 são linguagen regulares a união $A_1 \cup A_2 = \{x | x \in A_1 \text{ ou } x \in A_2\}$
- Sejam N_1 e N_2 os AFN's que reconhecem as linguagens A_1 e A_2



Fecho da União

O AF N, resultante da união de N_1 e N_2, tem um novo estado inicial q_0 que ramifica para os estados iniciais de N_1 e N_2, de forma não determinística, com arcos ε



Fecho da União

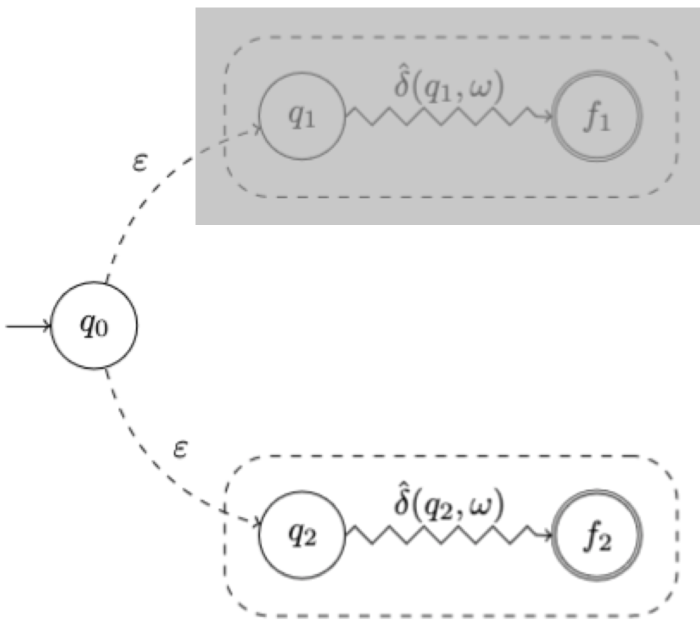
Formalizando

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconhece A_1 e $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconhece A_2. Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$, de tal forma que $Q = \{q_0\} \cup Q_1 \cup Q_2$, onde q_0 é o novo estado inicial, $F = F_1 \cup F_2$ e a função de transição é:

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & q \in Q_1 \\ \delta_2(q,a), & q \in Q_2 \\ \{q_1\} \cup \{q_2\}, & q = q_0 \wedge a = \varepsilon \\ \emptyset, & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

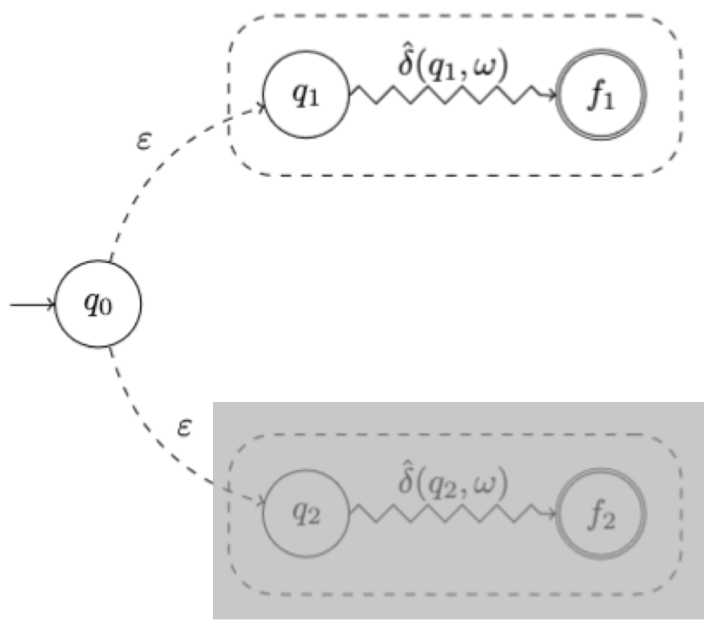
Fecho da União

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \\ \delta_2(q,a), & \text{se } q \in Q_2 \\ \{q_1\} \cup \{q_2\}, & \text{se } q = q_0 \wedge a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \wedge a \neq \varepsilon \end{cases}$$



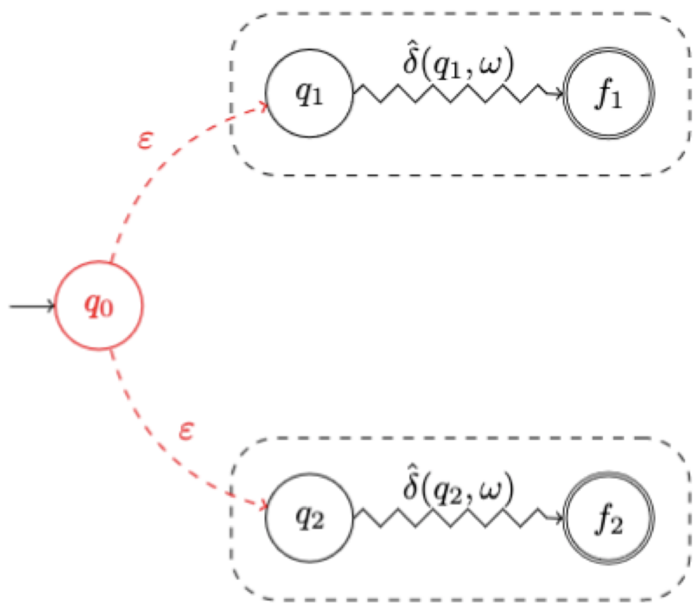
Fecho da União

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \\ \delta_2(q,a), & \text{se } q \in Q_2 \\ \{q_1\} \cup \{q_2\}, & \text{se } q = q_0 \wedge a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \wedge a \neq \varepsilon \end{cases}$$



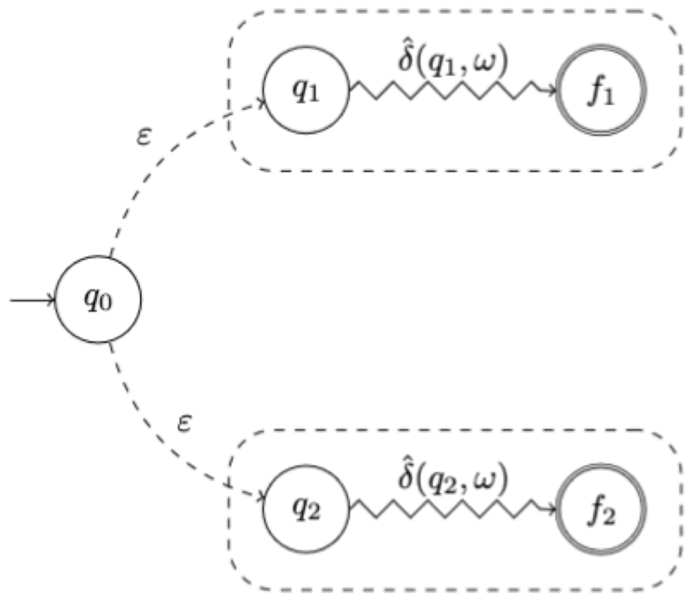
Fecho da União

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \\ \delta_2(q,a), & \text{se } q \in Q_2 \\ \{q_1\} \cup \{q_2\}, & \text{se } q = q_0 \land a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \land a \neq \varepsilon \end{cases}$$



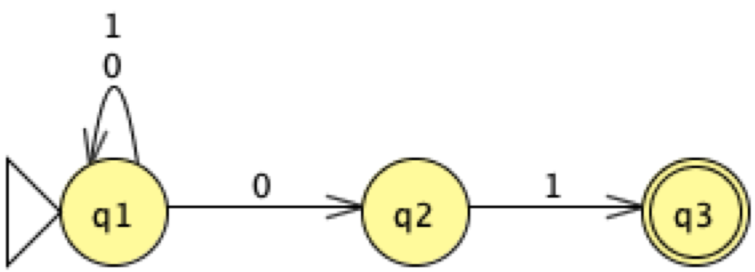
Fecho da União

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \\ \delta_2(q,a), & \text{se } q \in Q_2 \\ \{q_1\} \cup \{q_2\}, & \text{se } q = q_0 \wedge a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \wedge a \neq \varepsilon \end{cases}$$

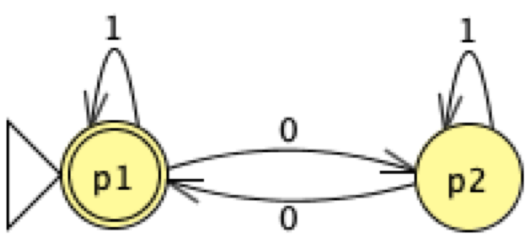


A partir de q_0 não pode processar nenhum símbolo do alfabeto

Exemplo de união de AFNs

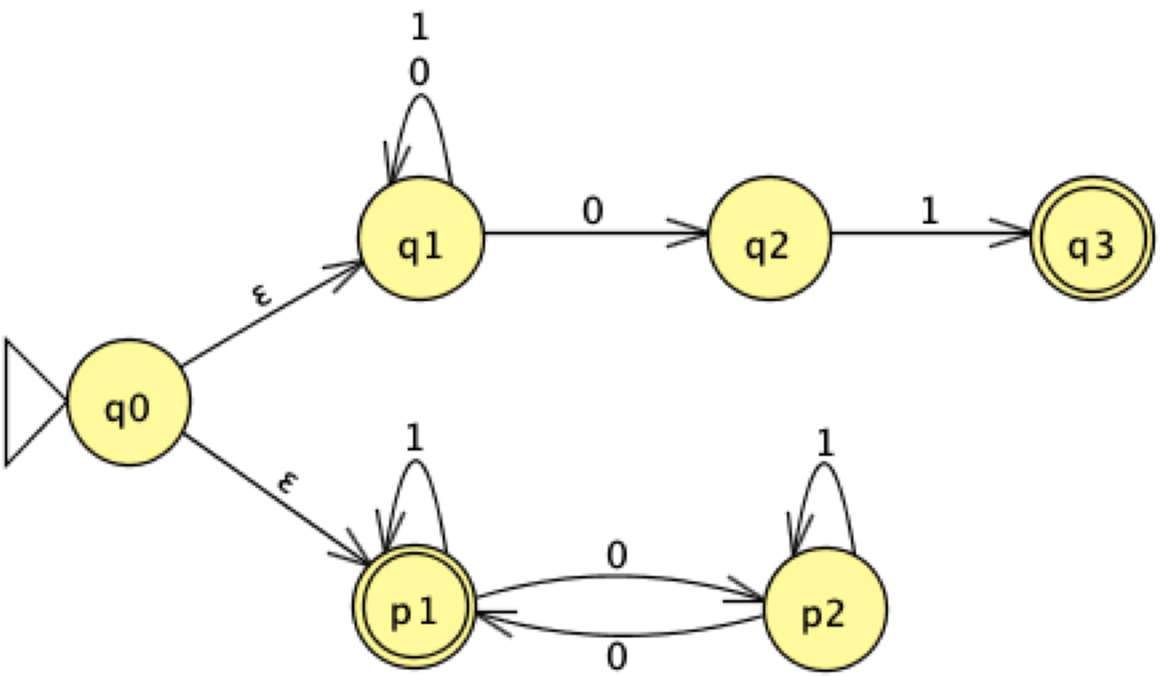


A$_1$ $= \{w \in \{0,1\}^*$ | w sempre termina com “01”}



A$_2$ $= \{w \in \{0,1\}^*$ | a quantidade de zeros é par}

Exemplo de união de AFNs



$A_1 \cup A_2 = \{w \in \{0,1\}^* \mid w$ termina em "01" **ou** a qtde de zeros é par$\}$

Exemplo de união de AFNs

Editor | Multiple Run

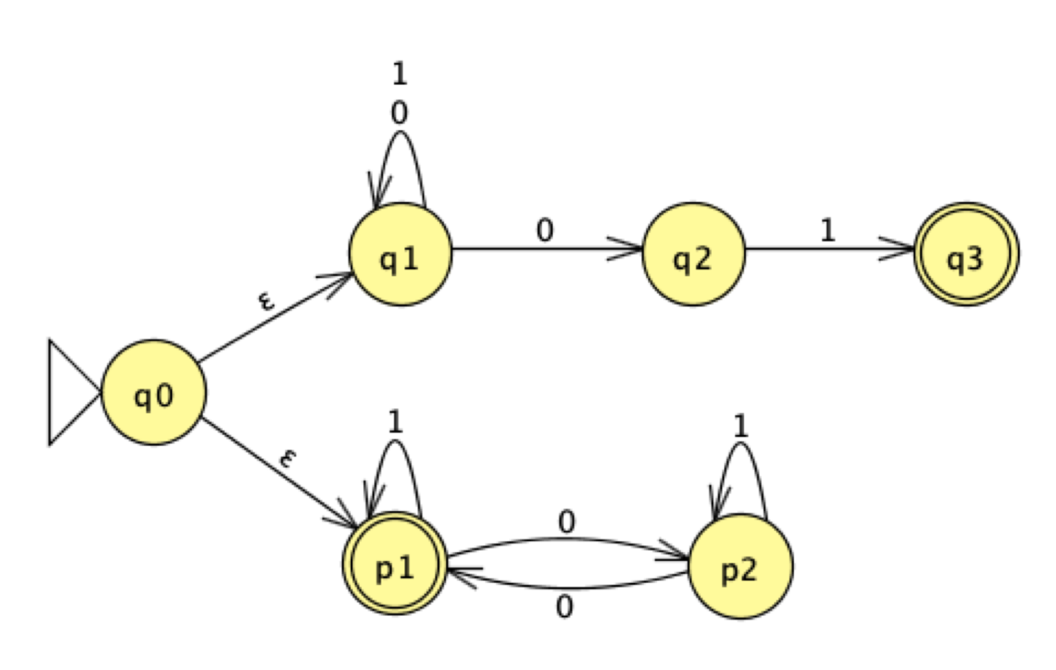


Table Text Size

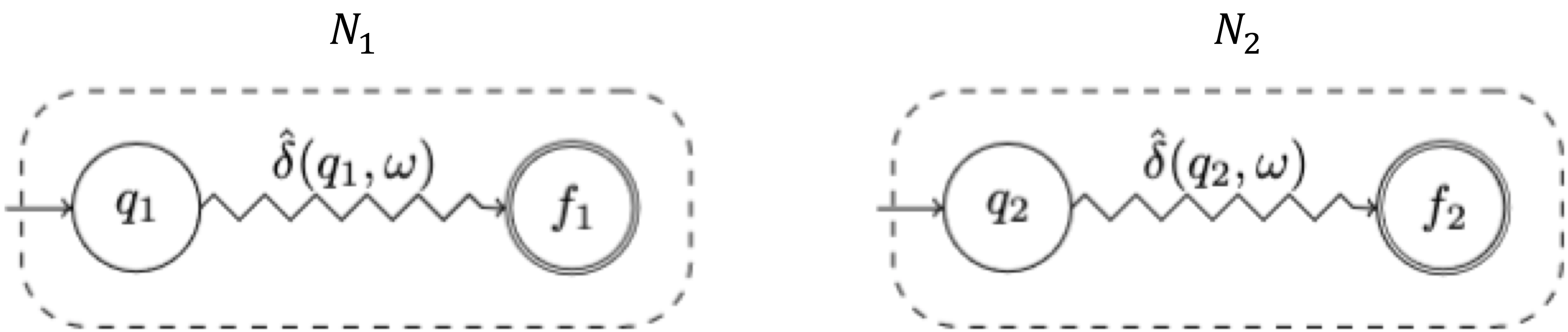
Input	Result
01	Accept
001	Accept
0000	Accept
0110	Accept
0001	Accept
0111	Reject
	Accept

Load Inputs | Run Inputs | Clear | Enter Epsilon | View Trace

Concatenação

Fecho da Concatenação

Sejam N_1 e N_2 os AFN's que reconhecem as linguagens A_1 e A_2

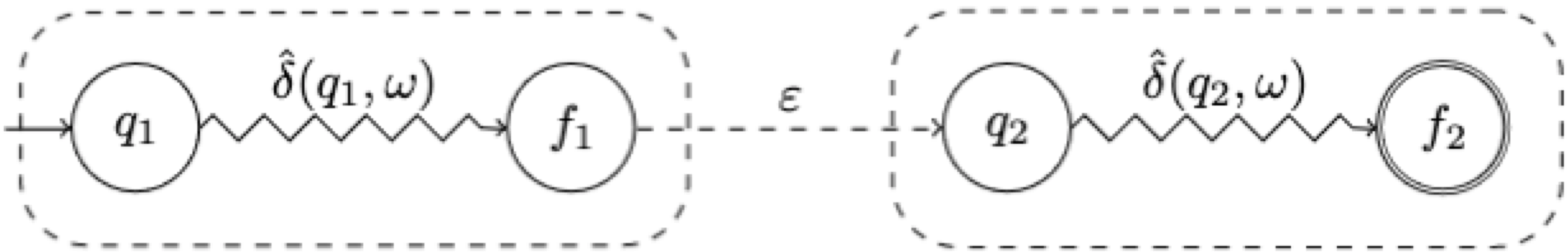


Vamos construir um novo AFN N que aceita ω se a primeira parte (prefixo) for aceita por N_1 e a segunda parte (sufixo) for aceita por N_2

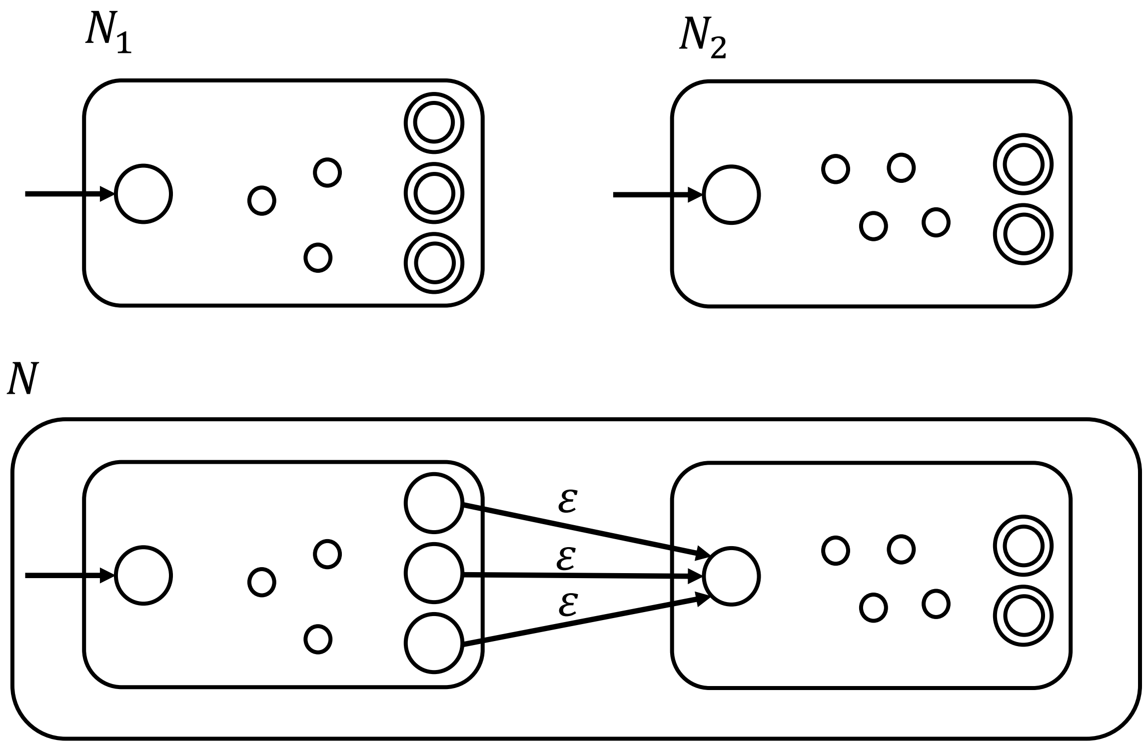
Fecho da Concatenação

$N = \{\omega | \omega = \omega_1\omega_2, \ \ \omega_1 \in A_1, \ \ \omega_2 \in A_2\}$

O AFN N aceita as cadeias que possam ser divididas em duas partes: a primeira aceita por N_1 e a segunda aceita por N_2. O AFN "**adivinha**" quando termina o prefixo e inicia o sufixo. Os estados de aceitação são os mesmos de N_2, isto é, $F = F_2$. O estado inicial do AFN N é o mesmo de N_1. **Todos os estados** de F_1 têm arestas ε para q_2.



Fecho da Concatenação



Fecho da Concatenação

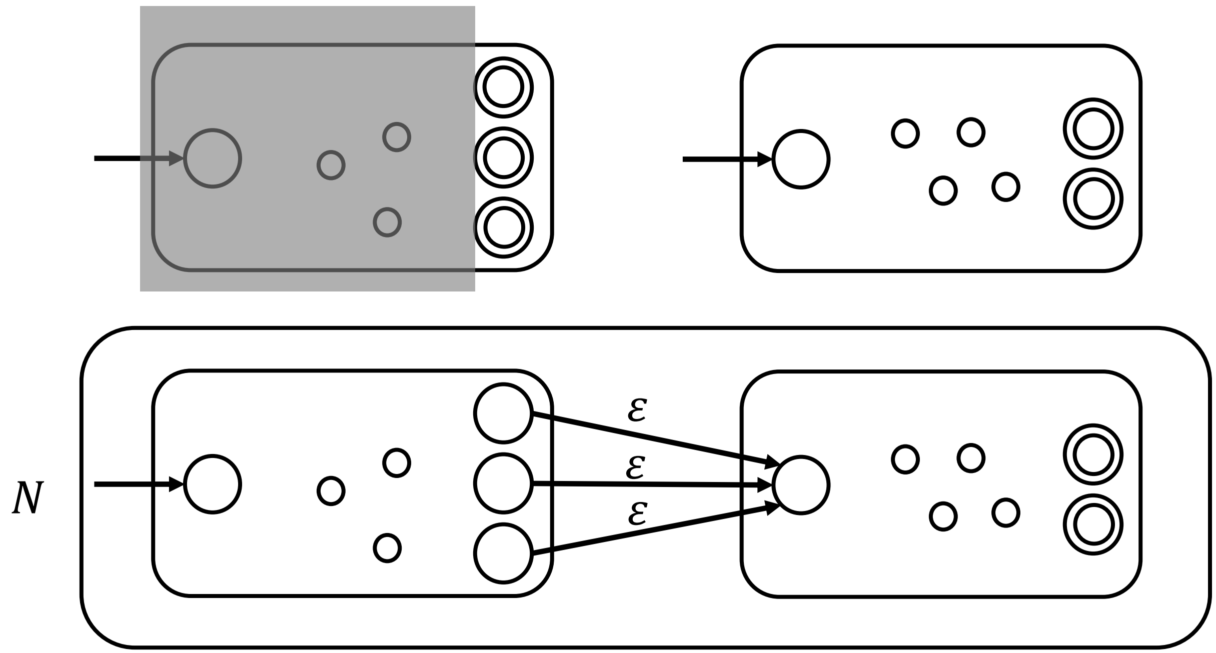
Formalizando

Suponha que $N_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ reconhece A_1 e $N_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$ reconhece A_2. Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \circ A_2$ (ou simplesmente $A_1 A_2$), de tal forma que: $Q = Q_1 \cup Q_2$, $q_0 = q_1$ é o estado inicial, $F = F_2$ e a função de transição é:

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \delta_2(q,a), & \text{se } q \in Q_2 \end{cases}$$

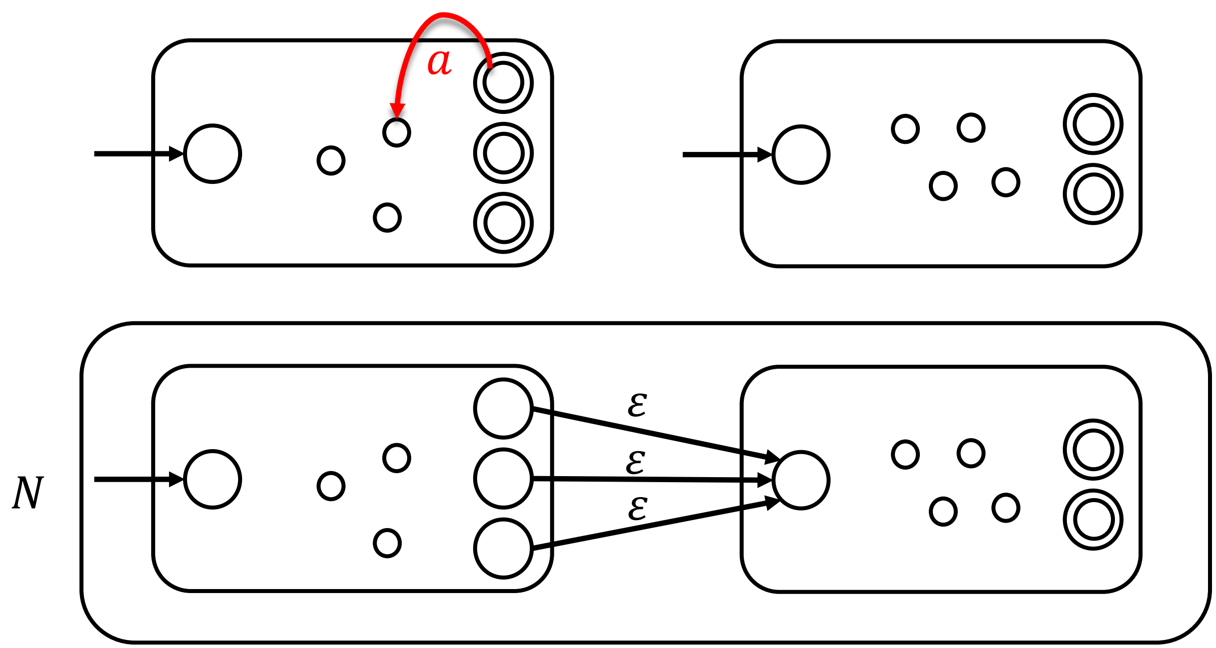
Fecho da Concatenação

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \delta_2(q,a), & \text{se } q \in Q_2 \end{cases}$$



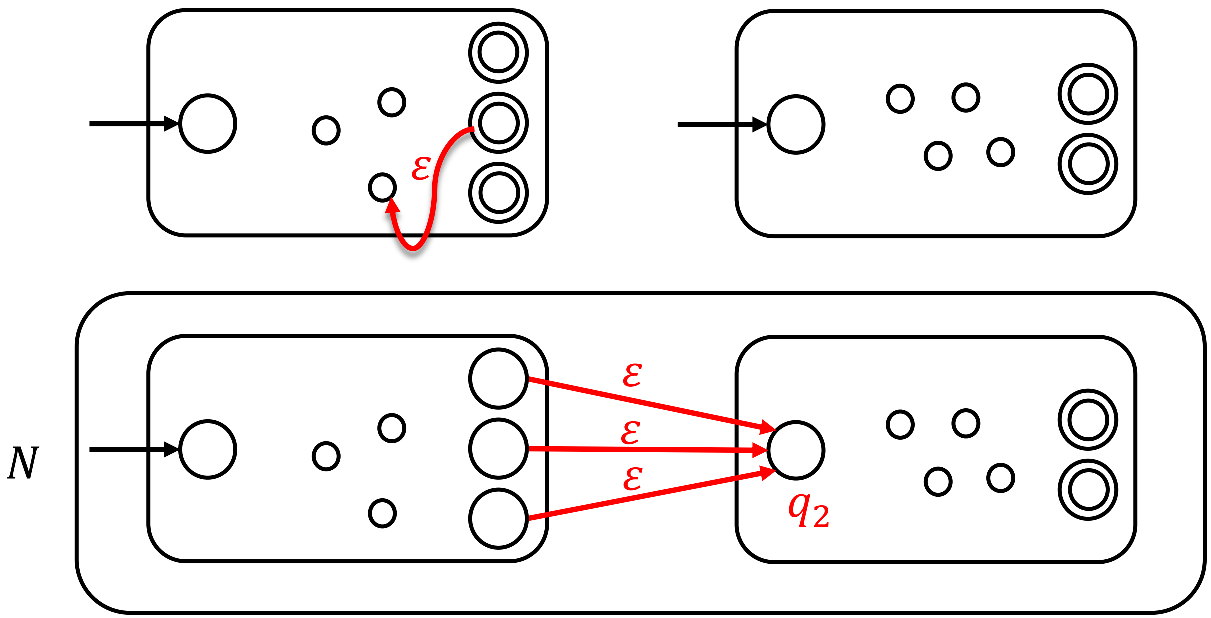
Fecho da Concatenação

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \delta_2(q,a), & \text{se } q \in Q_2 \end{cases}$$



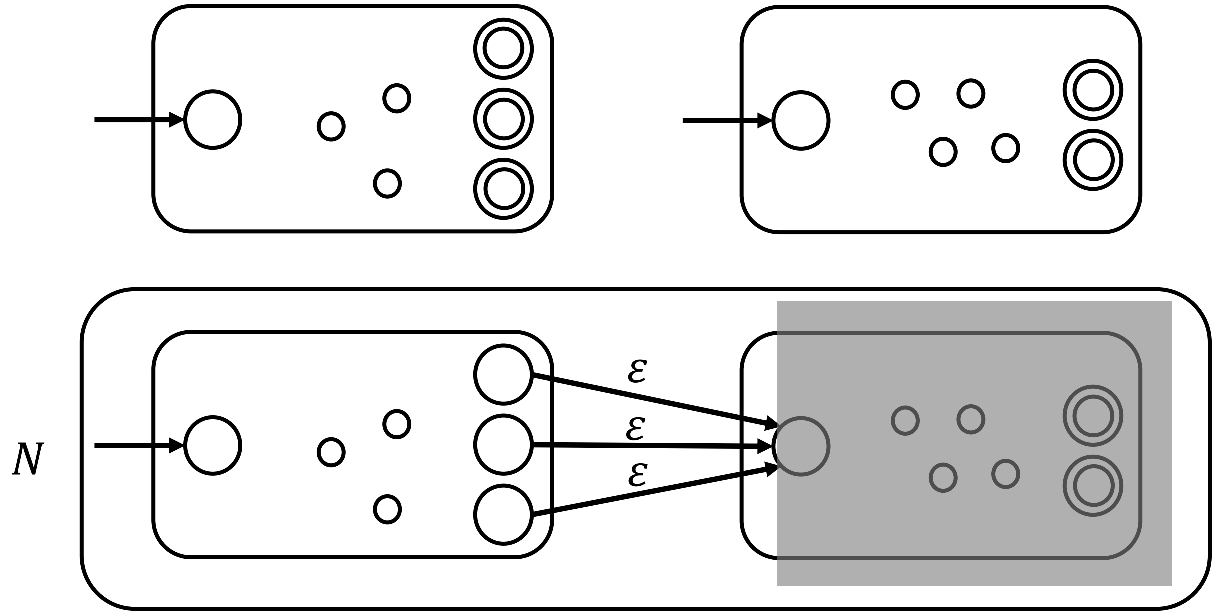
Fecho da Concatenação

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \delta_2(q,a), & \text{se } q \in Q_2 \end{cases}$$



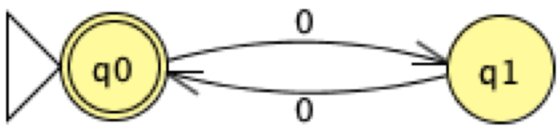
Fecho da Concatenação

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \delta_2(q,a), & \text{se } q \in Q_2 \end{cases}$$



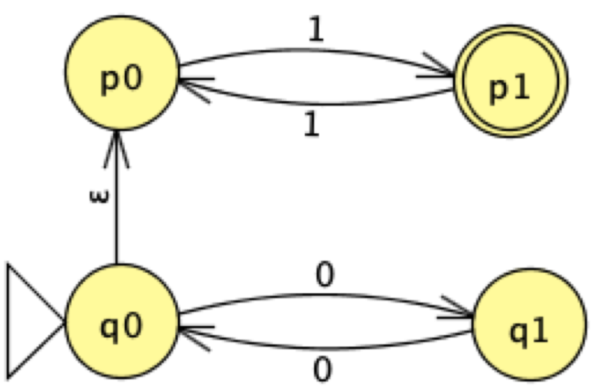
Exemplo de Concatenação

Mostre que $\{0^i1^j : \;\; i$ é par e j é ímpar$\}$ é regular



$\{0^i : \;\; i$ é par$\}$

$\{1^j : \; j$ é ímpar$\}$



$\{0^i1^j : \;\; i$ é par e j é ímpar$\}$

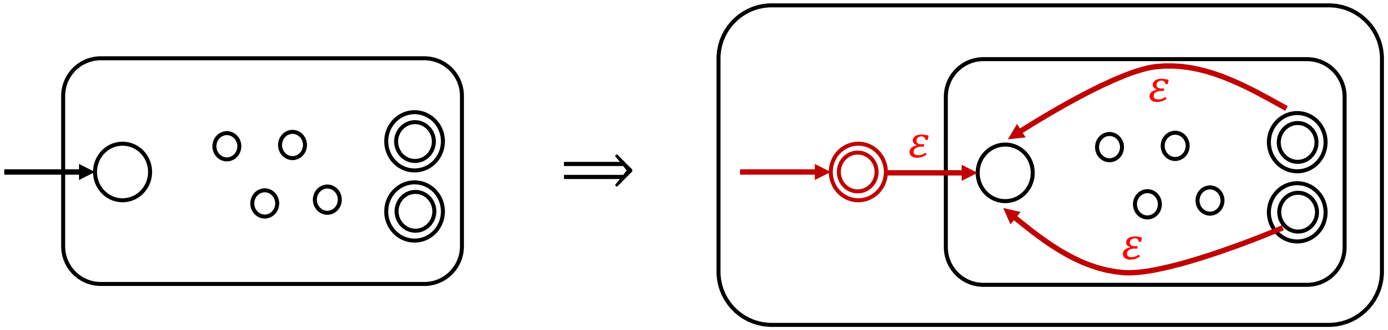
Estrela

Estrela

- A estrela de uma linguagem regular é regular
- Se $\omega \in A^*$, então $\omega = \omega_1\omega_2 \cdots \omega_k$, com $\omega_i \in A$ e $k \geq 0$. Isso quer dizer que a concatenação de quaisquer cadeias pertencentes à linguagem A também pertencem a A^*, ou seja, se $\alpha, \beta \in A$, necessariamente $\alpha\beta \in A^*$
- A diferença da concatenação anterior é que aqui temos k pedaços
- Sabendo quem é o AFN A vamos construir um AFN para A^*

Estrela

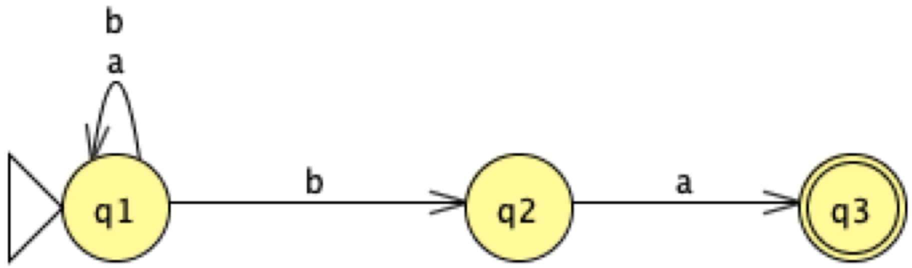
A geraçã do AFN para A^* será da seguinte forma



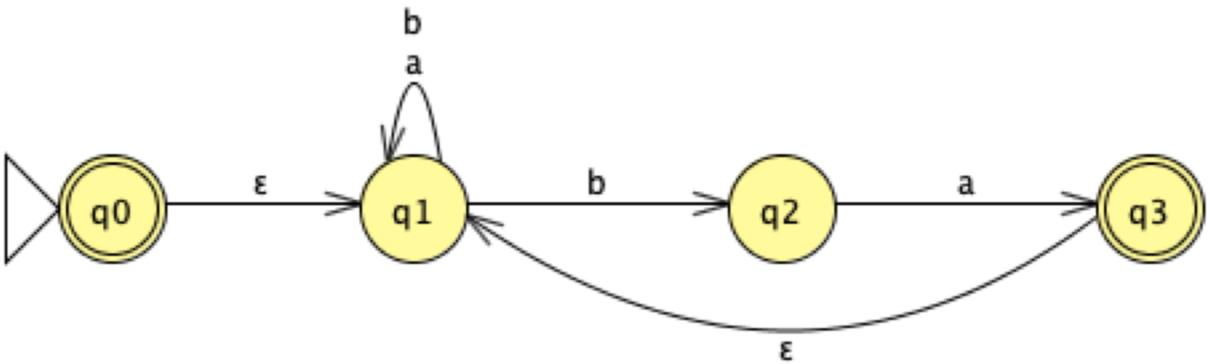
Note que uma cadeia ω só é aceita se o AFN ler todos os símbolos de ω e terminar em um estado de aceitação

Exemplo de geração de estrela

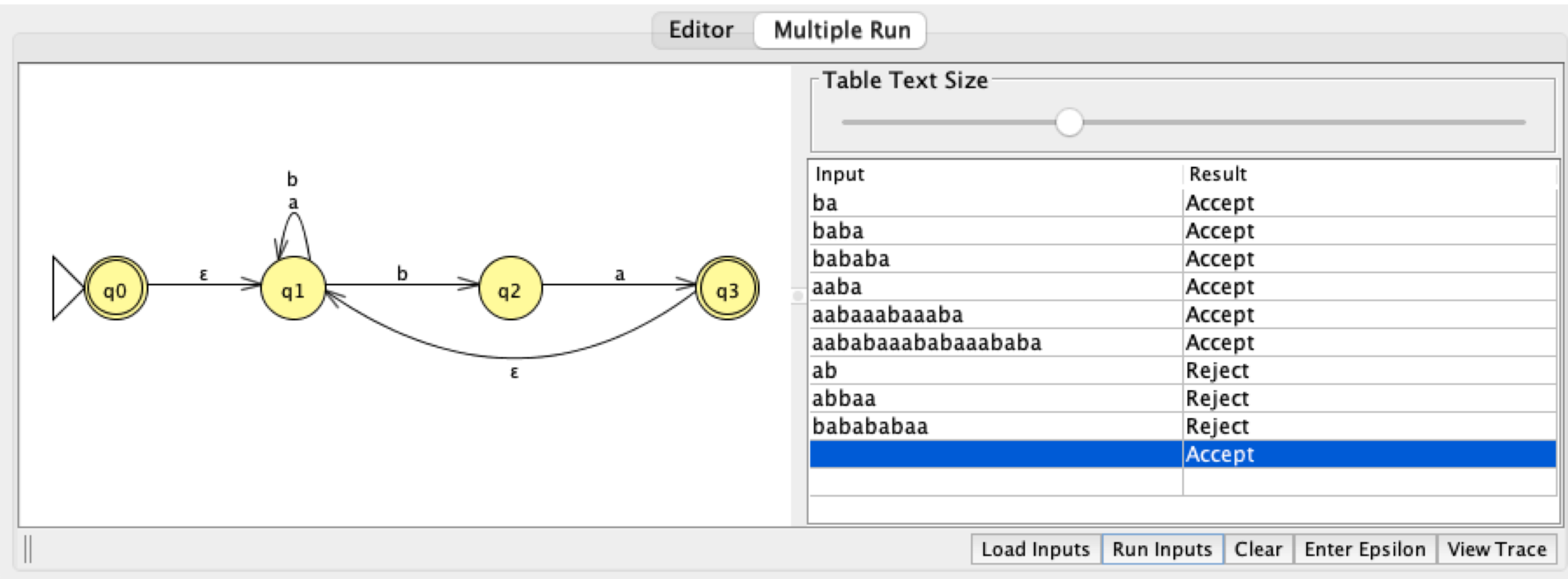
Considere o AFN que reconheçe a seguinte linguagem
$A = \{\omega \in \{a,b\}^*: \omega \text{ termina em } ba\}$



Após a transformação teremos:



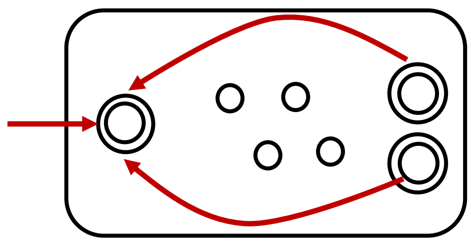
Exemplo de geração de estrela



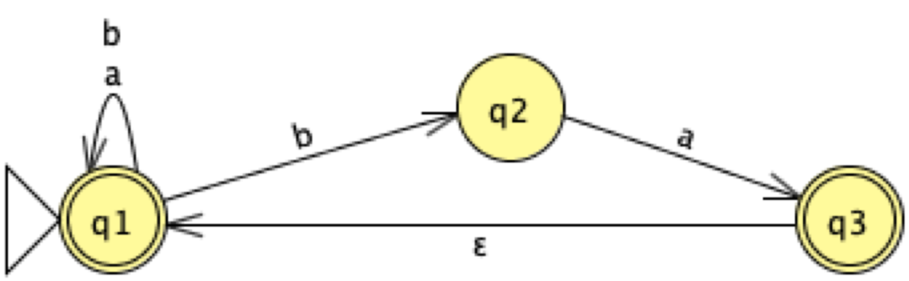
Essa linguagem é finita ou infinita? A cadeia vazia (ε) foi aceita ou não? Qual o valor de k para cada caso de teste acima?

Estrela

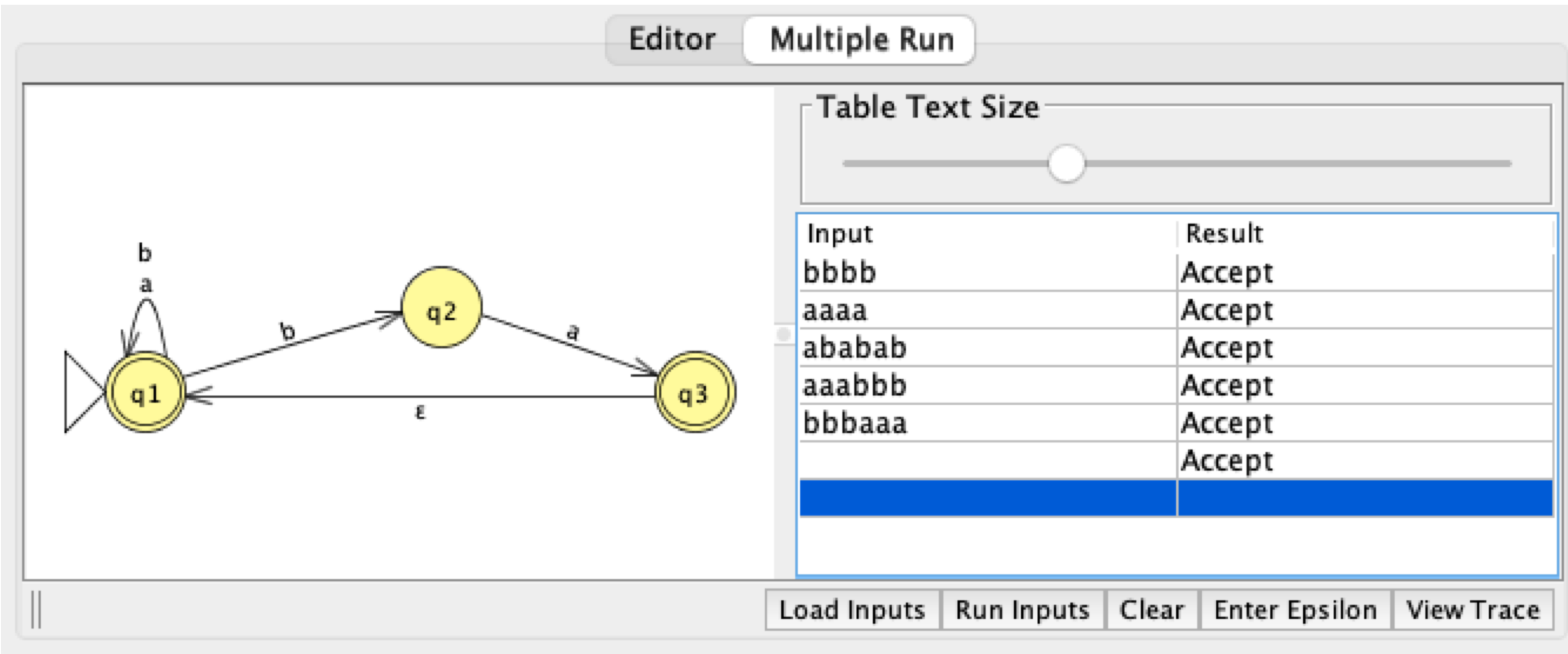
Observe que simplesmente transformar o estado inicial em estado final pode falhar



Veja como ficaria o autômato na mesma linguagem anterior



Estrela



O problema é que este autômato aceita qualquer cadeia de $\{a, b\}^*$

Estrela

Formalizando

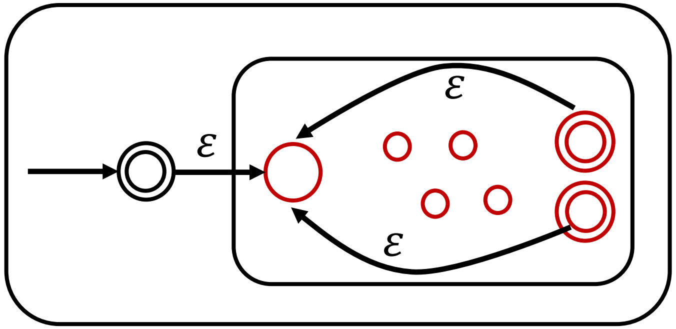
Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconhece a linguagem A. Vamos criar $N = (Q, \Sigma, \delta, q_0, F)$ em que:

$Q = Q_1 \cup \{q_0\}$, $F = F_1 \cup \{q_0\}$ e a função de transição é:

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_0\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \{q_1\}, & \text{se } q = q_0 \text{ e } a = \varepsilon \\ \emptyset & \text{se } q = q_0 \text{ e } a \neq \varepsilon \end{cases}$$

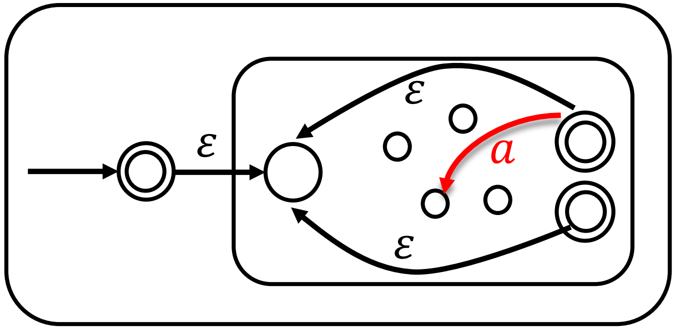
Estrela

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_0\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \{q_1\}, & \text{se } q = q_0 \text{ e } a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \text{ e } a \neq \varepsilon \end{cases}$$



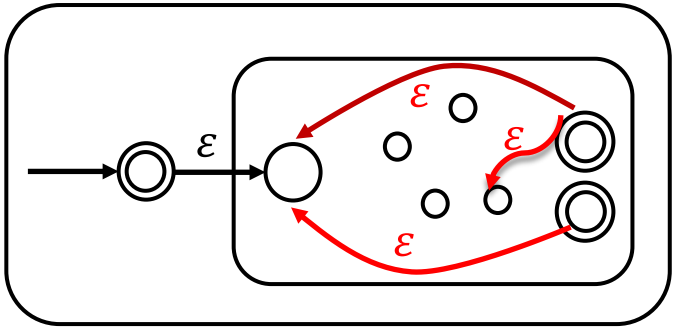
Estrela

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_0\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \{q_1\}, & \text{se } q = q_0 \text{ e } a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \text{ e } a \neq \varepsilon \end{cases}$$



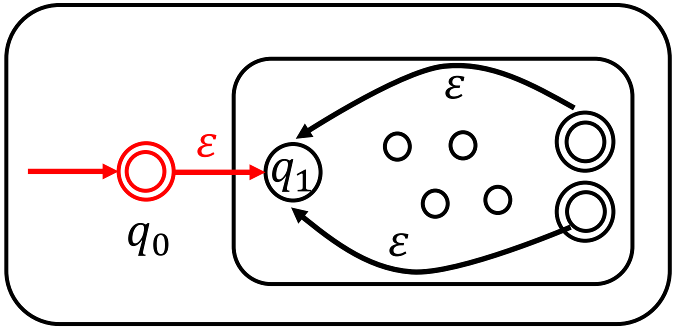
Estrela

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_0\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \{q_1\}, & \text{se } q = q_0 \text{ e } a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \text{ e } a \neq \varepsilon \end{cases}$$



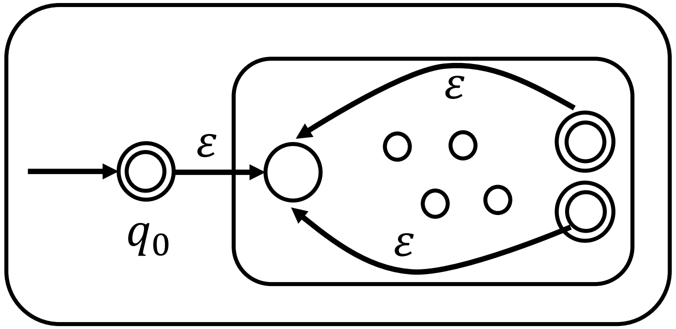
Estrela

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_0\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \{q_1\}, & \text{se } q = q_0 \text{ e } a = \varepsilon \\ \emptyset, & \text{se } q = q_0 \text{ e } a \neq \varepsilon \end{cases}$$



Estrela

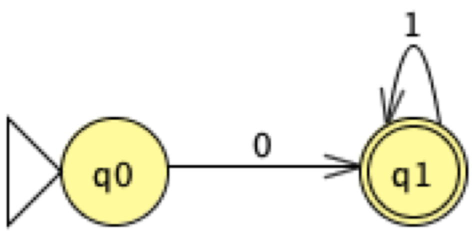
$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,a), & \text{se } q \in F_1 \text{ e } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_0\}, & \text{se } q \in F_1 \text{ e } a = \varepsilon \\ \{q_1\}, & \text{se } q = q_0 \text{ e } a = \varepsilon \\ \emptyset & \text{se } q = q_0 \text{ e } a \neq \varepsilon \end{cases}$$



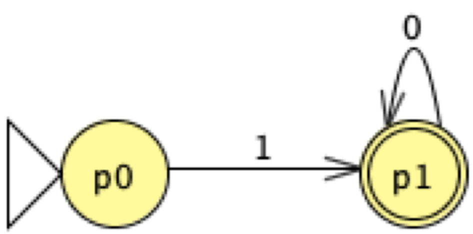
A partir de q_0 não pode processar nenhum símbolo do alfabeto

Exemplo de estrela

Seja $A = \{\omega \in \{0,1\}^* : \ \omega = 01^i$ com $i \geq 0$ ou $\omega = 10^j$ com $j \geq 0\}$
A estratégia é gerar AFNs das linguagens parciais, fazer a união e, depois, fazer a estrela



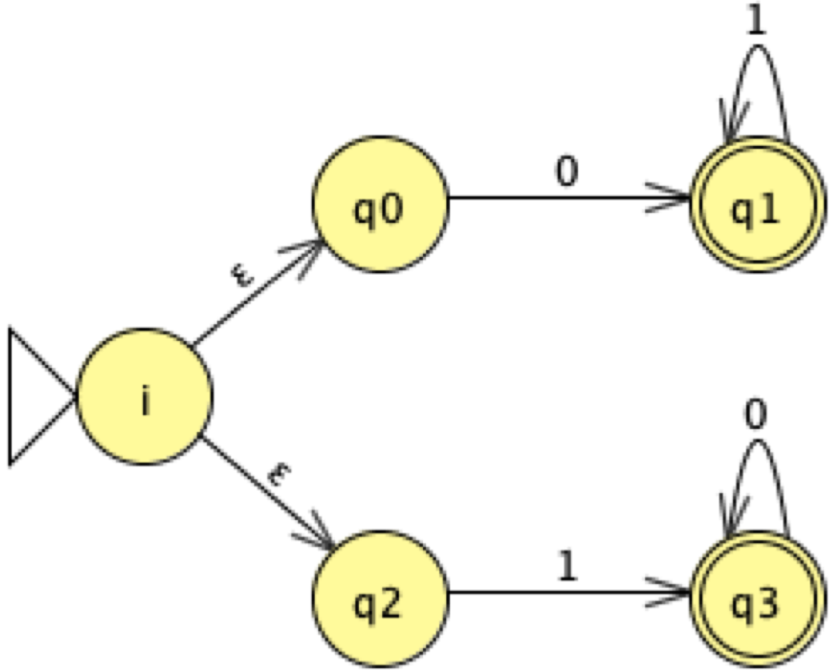
$A_1 = \{\omega \in \{0,1\}^* : \ \omega = 01^i$ com $i \geq$



$A_2 = \{\omega \in \{0,1\}^* : \ \omega = 10^j$ com $j \geq 0$

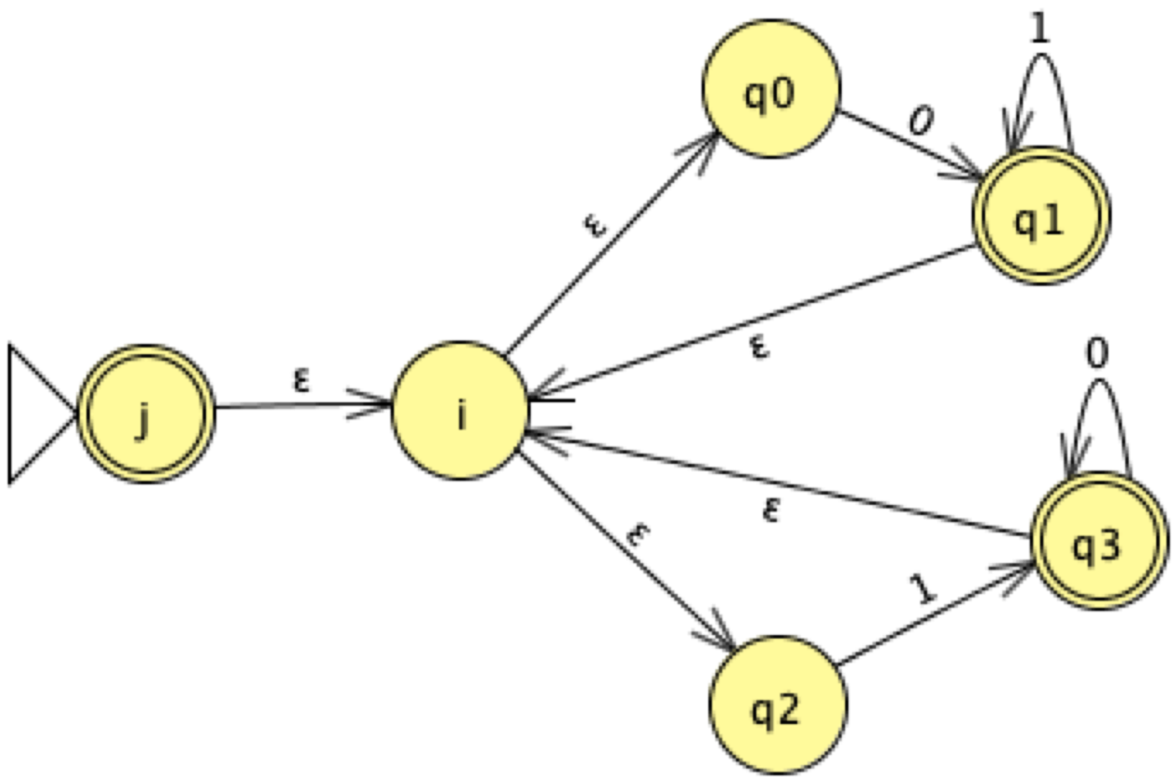
Exemplo de estrela

A união de A_1 com A_2 será $\mathrm{A} = A_1A_2$



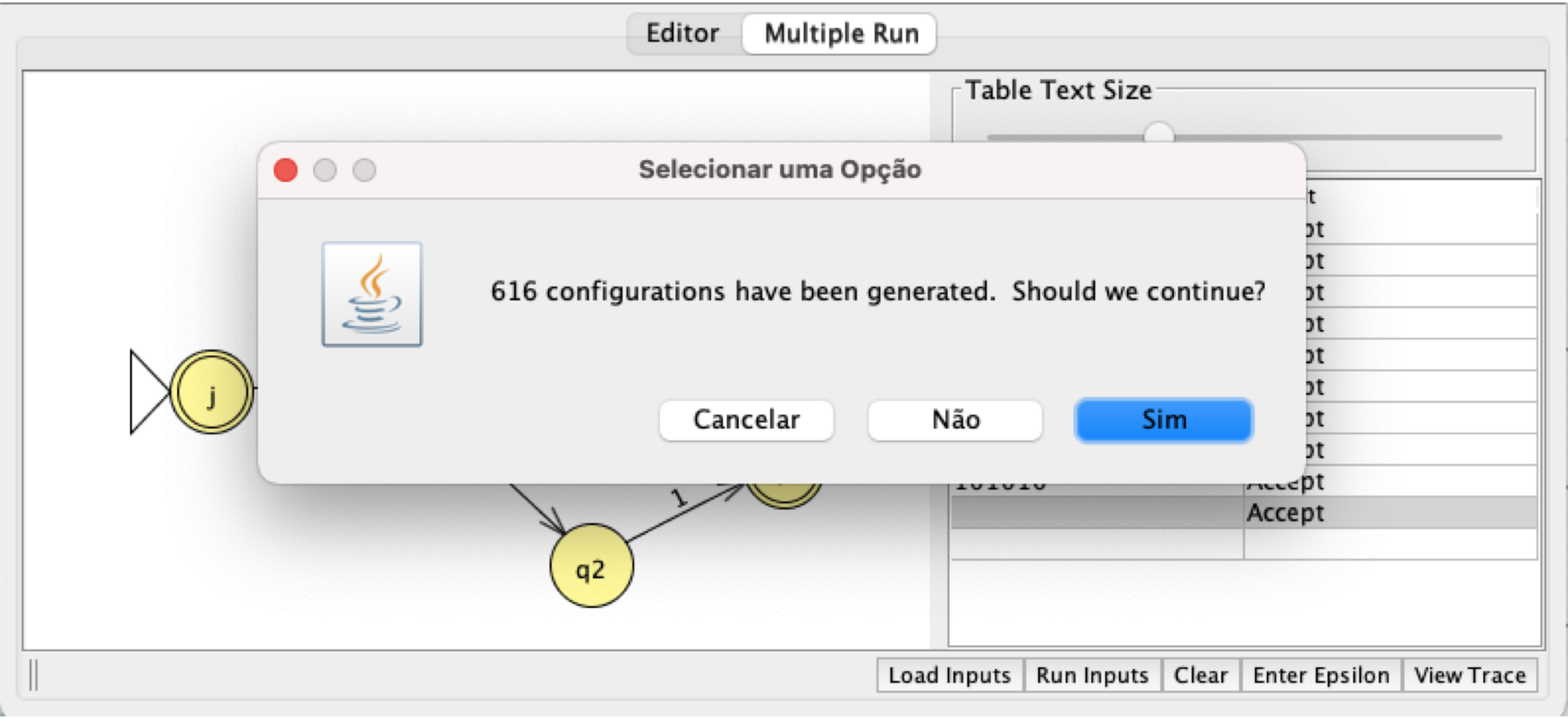
Exemplo de estrela

A estrela de A^* será



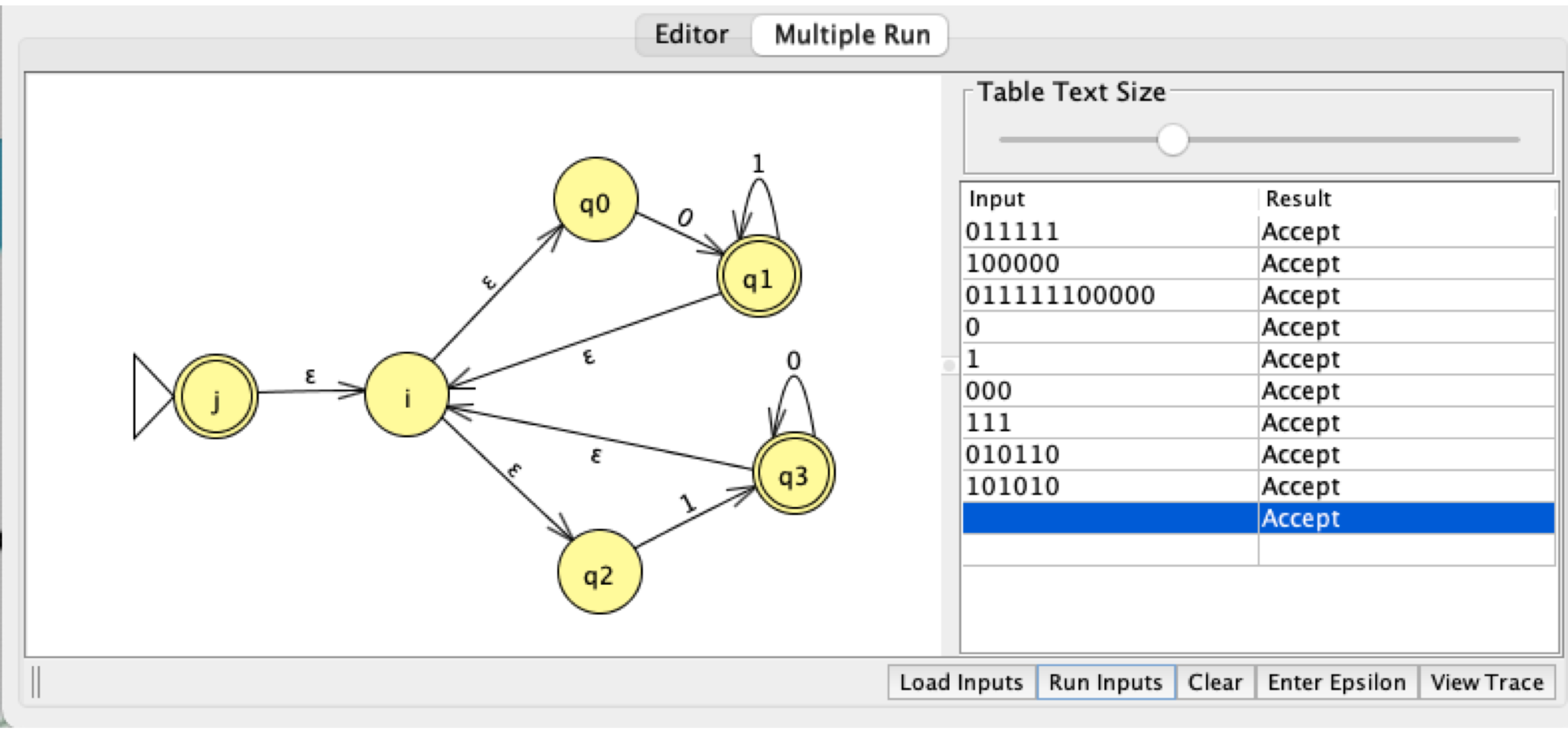
Exemplo de estrela

A estrela de A^* será



Exemplo de estrela

A estrela de A^* será



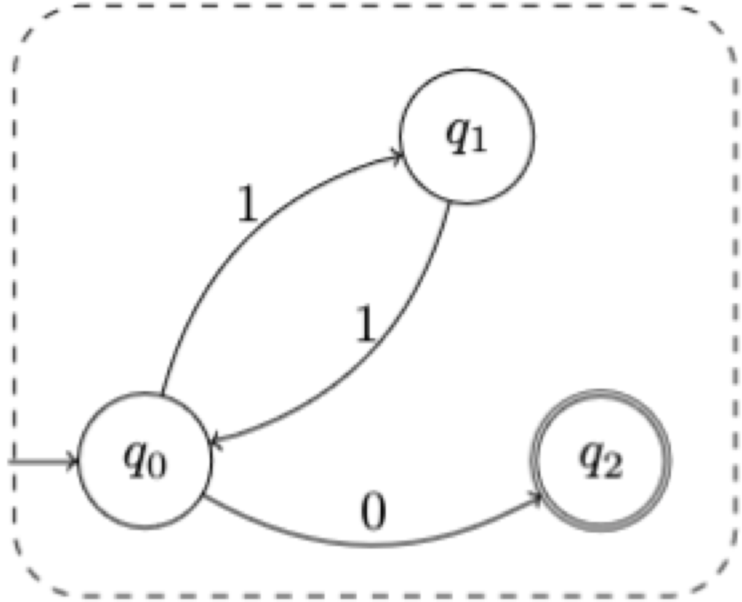
Intuição sobre o complemento

Complemento

Se $A \subseteq \Sigma^*$ é uma linguagem regular, o mesmo vale para $\bar{A} = \Sigma^* - A = \{\omega \mid \omega \in \Sigma^* \text{e } \omega \notin A\}$

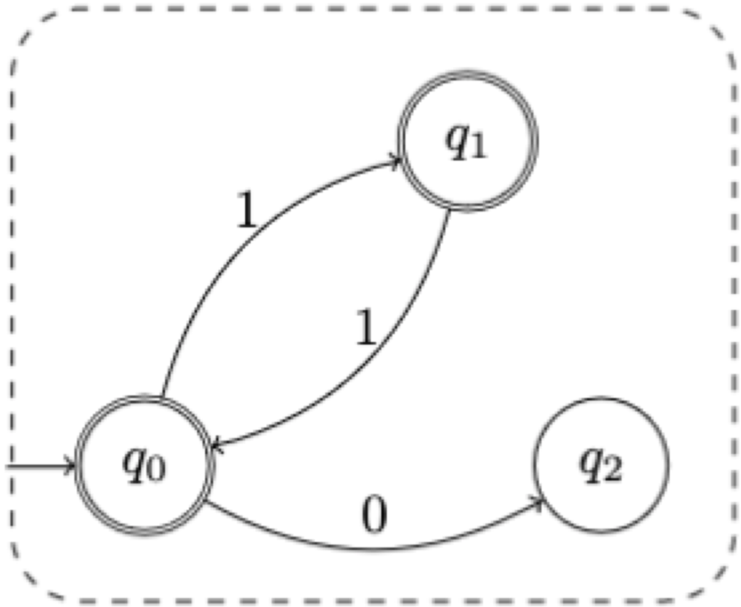
Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFD que reconhece a linguagem A

Vamos construir um novo autômato M, tal que, as condições "aceita/rejeita" de M sejam invertidas para reconhecer $\bar{A}$



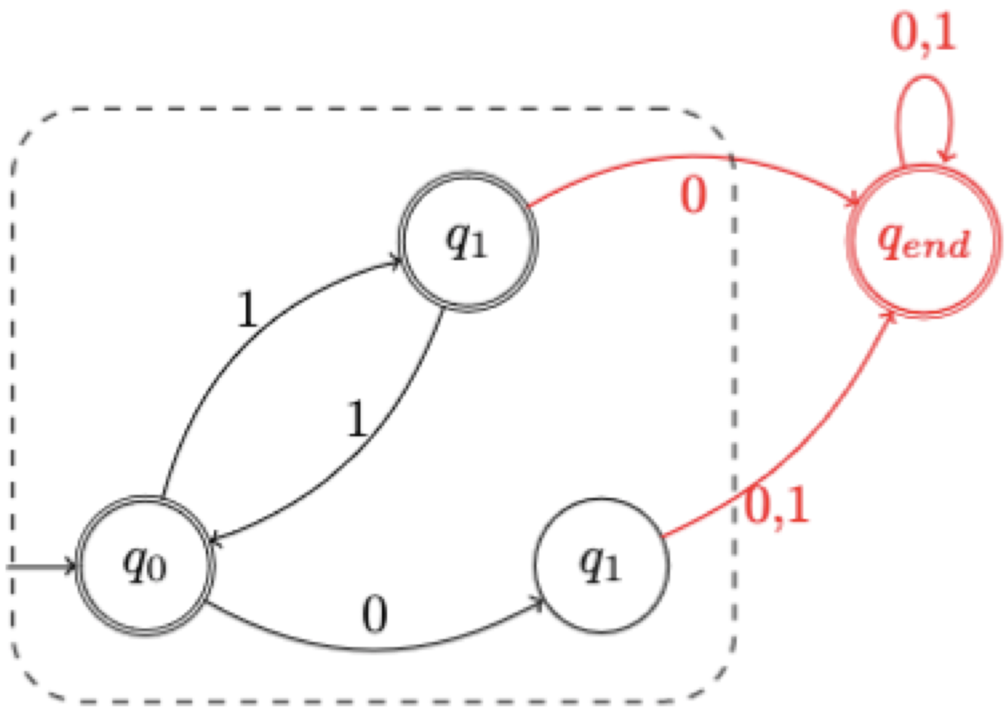
Complemento

Vamos inverter os estados finais em não-finais e vice-versa



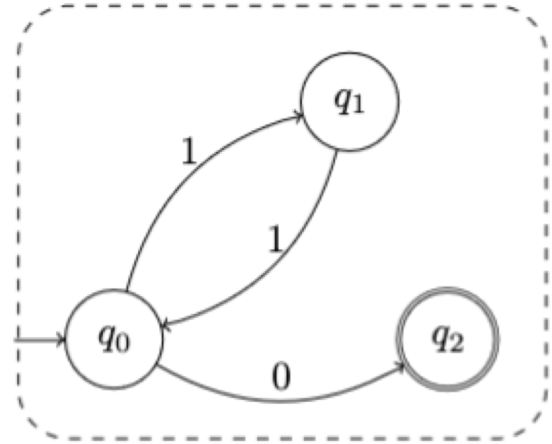
Complemento

Adicionamos um novo estado q_{end}, tal que $\delta(q,a) = q_{end}$ sempre que $\delta(q,a) = \emptyset$

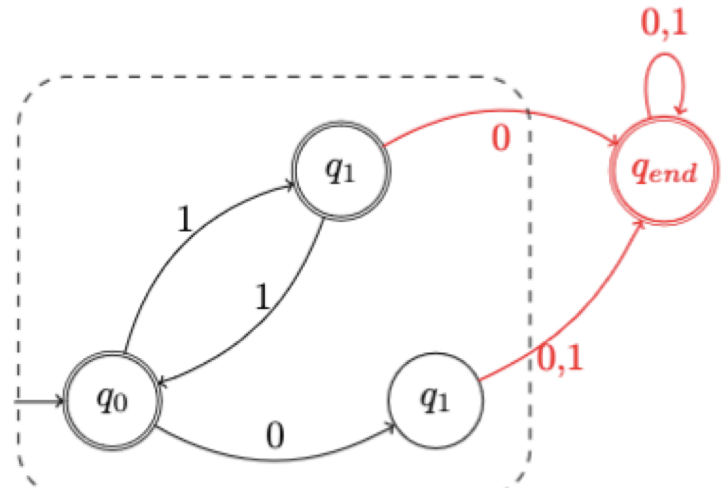


Complemento

$\omega_1 = 110$
$\omega_2 = 0$
$\omega_3 = 11110$
$\omega_4 = 11$
$\omega_5 = 1100$



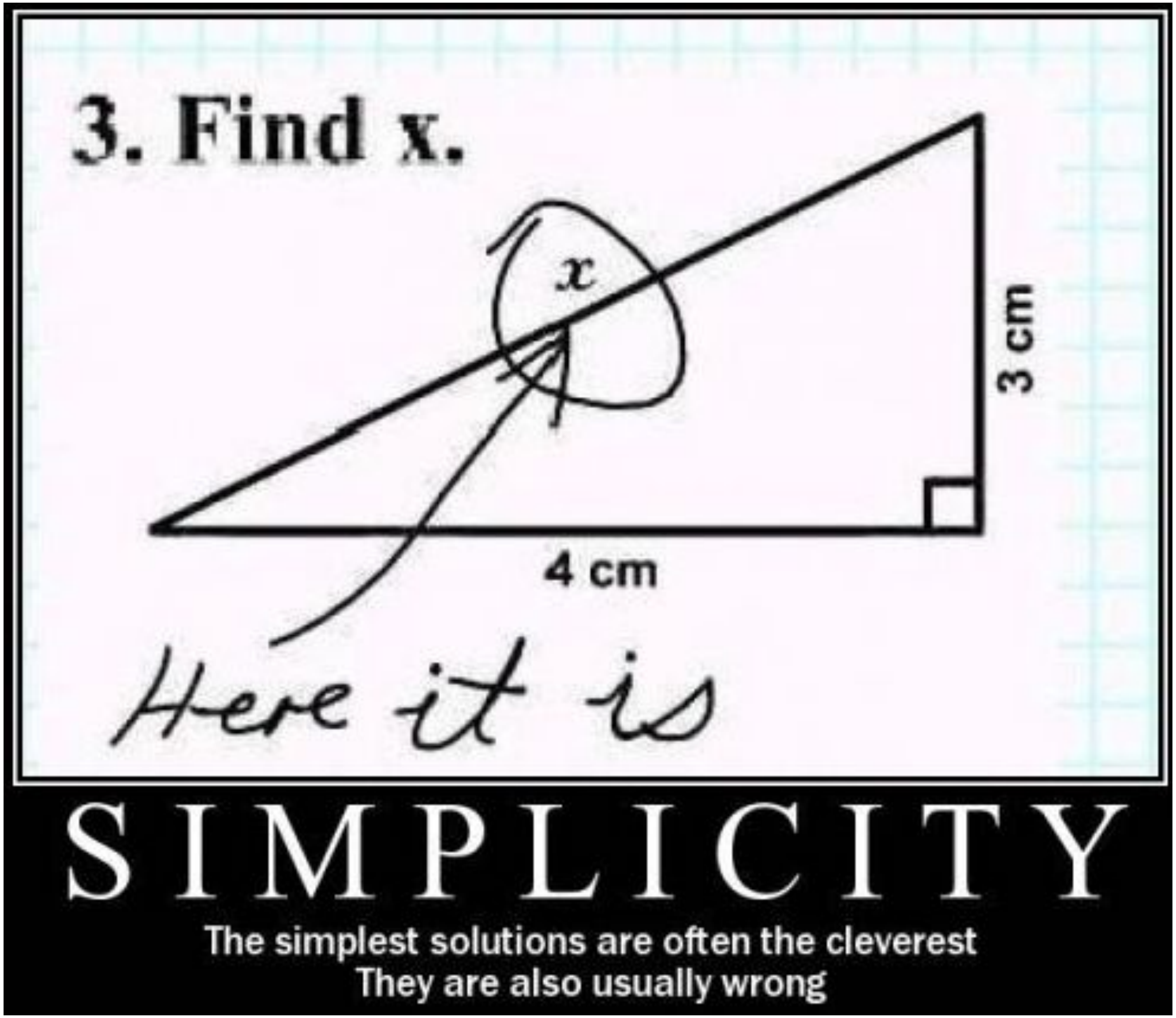
$\omega_1 = 110$
$\omega_2 = 0$
$\omega_3 = 11110$
$\omega_4 = 11$
$\omega_5 = 1100$



Intuição sobre o complemento

Fecho da União

- Se A_1 e A_2 são linguagens regulares, o mesmo vale para $A_1 \cap A_2 = \{x|x \in A_1 \text{ e } x \in A_2\}$
- Sejam N_1 e N_2 os AFN's que reconhecem as linguagens A_1 e A_2
- Construir um AFN que aceite uma cadeia ω em ambos os autômatos não é uma tarefa fácil
- Entretanto, podemos utilizar a lei de De Morgan: $A_1 \cap A_2 = \overline{\overline{A_1} \cup \overline{A_2}}$
- O que precisa ser feito é **só** aplicar as operações de complemento e união


3. Find x.
x
3 cm
4 cm
Here it is
SIMPLICITY
The simplest solutions are often the cleverest
They are also usually wrong