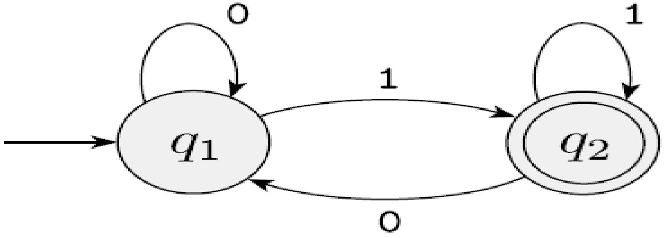
ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 6
Fechamentos de Linguagens Regulares

Profa. Ariane Machado Lima
ariane.machado@usp.br

Aulas anteriores

AFD

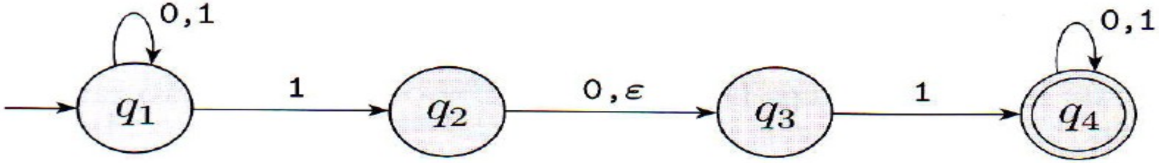


Para cada par (estado atual, próximo símbolo) está DETERMINADO qual é o próximo estado

Um ***autômato finito*** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

1. Q é um conjunto finito conhecido como os ***estados***,
2. Σ é um conjunto finito chamado o ***alfabeto***,
3. $\delta : Q \times \Sigma \longrightarrow Q$ é a ***função de transição***,[1]
4. $q_0 \in Q$ é o ***estado inicial***, e
5. $F \subseteq Q$ é o ***conjunto de estados de aceitação***.[2]

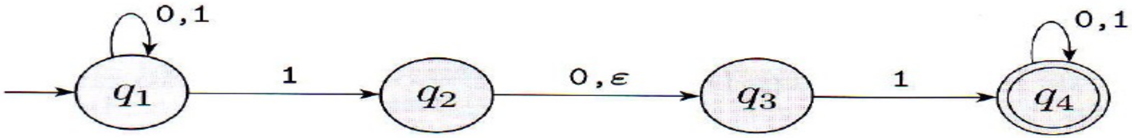
AFN

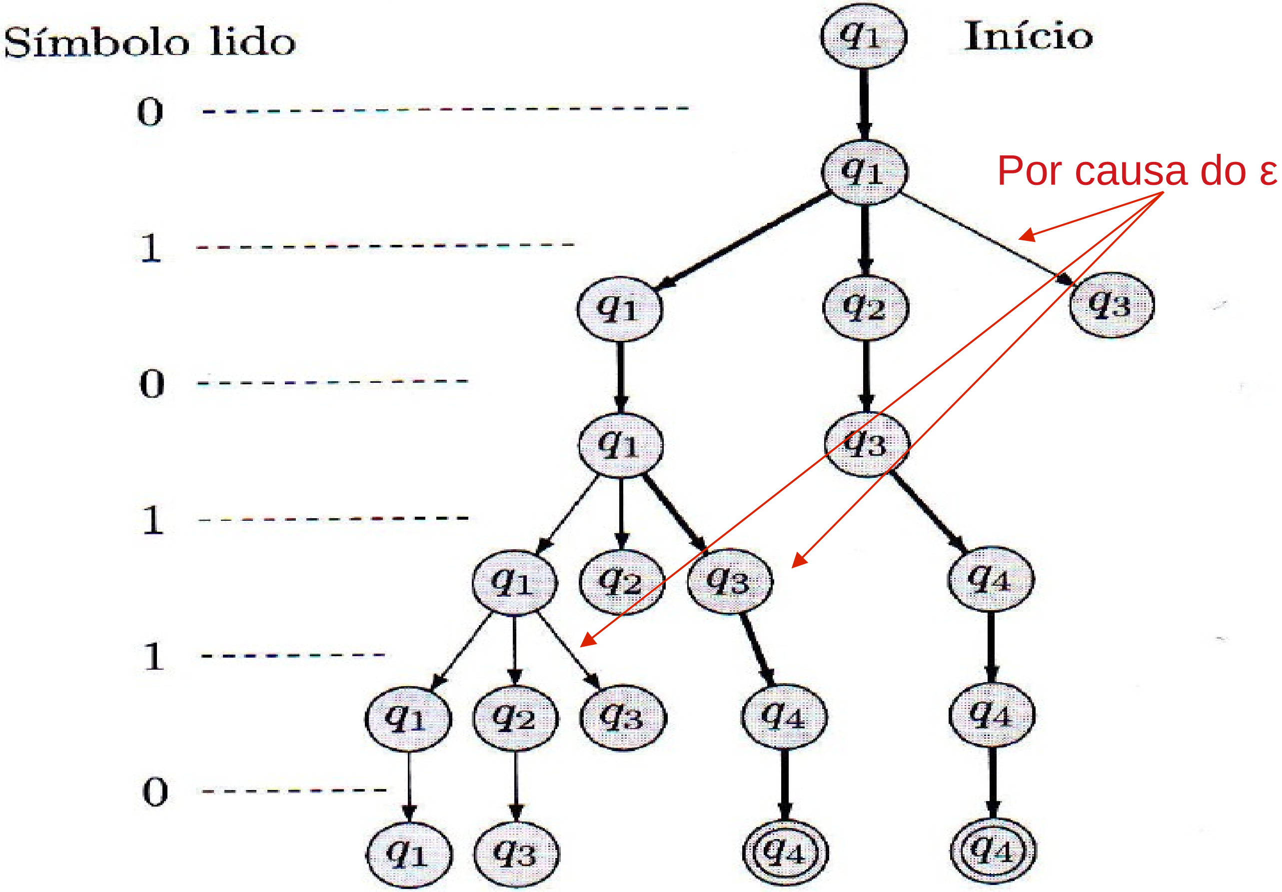


Um ***autômato finito não-determinístico*** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

1. Q é um conjunto finito de estados,
2. Σ é um alfabeto finito,
3. $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ 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.

Para cada par (estado atual, próximo símbolo – incluindo ε) há um conjunto de estados possíveis


0,1
1
0, ε
1
0,1
q1
q2
q3
q4


Símbolo lido
Início
Por causa do ε
0
1
0
1
1
0

Equivalência entre AFDs e AFNs

- Duas máquinas são equivalentes se elas reconhecem a mesma linguagem

TEOREMA 1.39

Todo autômato finito não-determinístico tem um autômato finito determinístico equivalente.

Linguagens Regulares

DEFINIÇÃO 1.16

Uma linguagem é chamada de uma ***linguagem regular*** se algum autômato finito a reconhece.

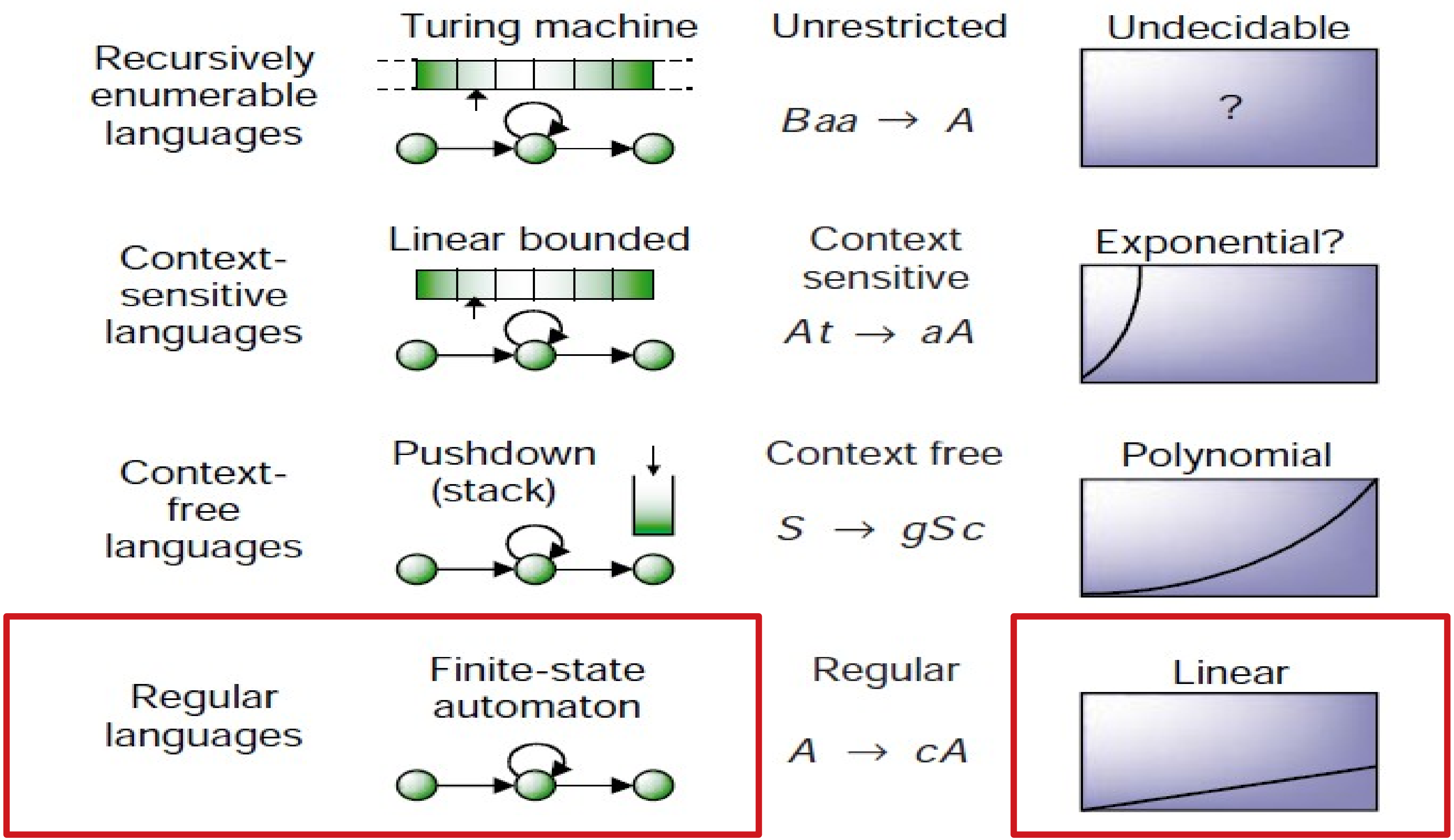
COROLÁRIO 1.40

Uma linguagem é regular se e somente se algum autômato finito não-determinístico a reconhece.

AFDs e AFNs

- Por que o teorema de equivalência é importante?
- Pode-se optar por um ou outro dependendo do objetivo
- AFDs são mais eficientes
- AFNs podem:
  - ser mais fáceis de serem projetados
  - facilitar demonstração de teoremas
  - ser úteis em versões probabilísticas

Linguagens, modelos computacionais (dispositivos, gramáticas) e suas complexidades



Aula de hoje

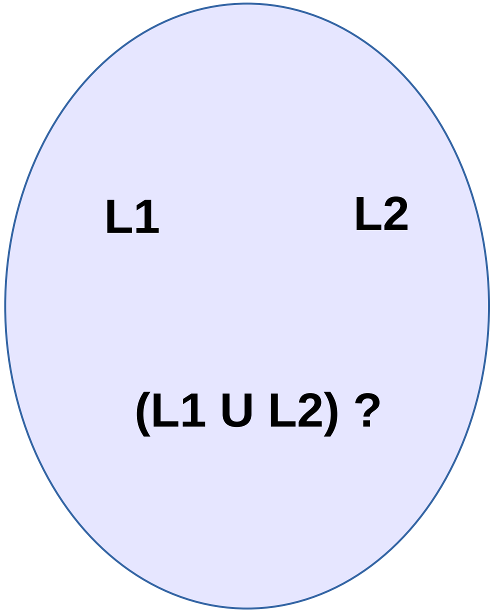
Fechamentos de Linguagens Regulares

Hoje - Motivação

- Estou projetando um analisador léxico para a linguagem de programação que estou criando
- Já criei dois autômatos finitos (AF):
  - um para reconhecer constantes numéricas
  - outro para reconhecer variáveis
- Será que é possível criar AFs para reconhecer:
  - O que é constantes numéricas OU variável ?
  - O que não é uma variável ?
- Ou seja, será que essas linguagens são regulares?

Hoje

- Fechamento de linguagens regulares sob as operações de união, intersecção, concatenação, estrela, complemento



Conjunto de todas as linguagens regulares

Linguagem Regular

- Uma linguagem é chamada linguagem regular se algum autômato finito a reconhece

- Vamos ver suas propriedades
  - Saber se uma linguagem é regular ou não é importante para sabermos se podemos ou não implementar um autômato finito que a reconheça

Operações regulares

Sejam A e B linguagens. Definimos as operações regulares ***união***, ***concatenação*** e ***estrela*** da seguinte forma.

- **União**: $A \cup B = \{x | x \in A \text{ ou } x \in B\}$.
- **Concatenação**: $A \circ B = \{xy | x \in A \text{ e } y \in B\}$.
- **Estrela**: $A^* = \{x_1 x_2 \ldots x_k | k \geq 0 \text{ e cada } x_i \in A\}$.

União: $A \cup B = \{x | x \in A \text{ ou } x \in B\}$.

Concatenação: $A \circ B = \{xy | x \in A \text{ e } y \in B\}$.

Estrela: $A^* = \{x_1 x_2 \dots x_k | k \geq 0 \text{ e cada } x_i \in A\}$.

Ex:

Suponha que o alfabeto Σ seja o alfabeto padrão de 26 letras $\{\mathtt{a}, \mathtt{b}, \dots, \mathtt{z}\}$. Se $A = \{\mathtt{legal}, \mathtt{ruim}\}$ e $B = \{\mathtt{garoto}, \mathtt{garota}\}$, então

$$A \cup B =$$

União: $A \cup B = \{x | \; x \in A \text{ ou } x \in B\}$.

Concatenação: $A \circ B = \{xy | \; x \in A \text{ e } y \in B\}$.

Estrela: $A^* = \{x_1 x_2 \ldots x_k | \; k \geq 0 \text{ e cada } x_i \in A\}$.

Ex:

Suponha que o alfabeto Σ seja o alfabeto padrão de 26 letras $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$. Se $A = \{\mathtt{legal}, \mathtt{ruim}\}$ e $B = \{\mathtt{garoto}, \mathtt{garota}\}$, então

$$A \cup B = \{\mathtt{legal}, \mathtt{ruim}, \mathtt{garoto}, \mathtt{garota}\}$$

União: $A \cup B = \{x |\ x \in A \text{ ou } x \in B\}$.

Concatenação: $A \circ B = \{xy |\ x \in A \text{ e } y \in B\}$.

Estrela: $A^* = \{x_1 x_2 \ldots x_k |\ k \geq 0 \text{ e cada } x_i \in A\}$.

Ex:

Suponha que o alfabeto Σ seja o alfabeto padrão de 26 letras $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$. Se $A = \{\mathtt{legal}, \mathtt{ruim}\}$ e $B = \{\mathtt{garoto}, \mathtt{garota}\}$, então

$$A \cup B = \{\mathtt{legal}, \mathtt{ruim}, \mathtt{garoto}, \mathtt{garota}\}$$

$$A \circ B =$$

Ex:

União: $A \cup B = \{x | \, x \in A \text{ ou } x \in B\}$.

Concatenação: $A \circ B = \{xy | \, x \in A \text{ e } y \in B\}$.

Estrela: $A^* = \{x_1 x_2 \ldots x_k | \, k \geq 0 \text{ e cada } x_i \in A\}$.

Suponha que o alfabeto Σ seja o alfabeto padrão de 26 letras $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$. Se $A = \{\mathtt{legal}, \mathtt{ruim}\}$ e $B = \{\mathtt{garoto}, \mathtt{garota}\}$, então

$$A \cup B = \{\mathtt{legal}, \mathtt{ruim}, \mathtt{garoto}, \mathtt{garota}\}$$

$$A \circ B = \{\mathtt{legalgaroto}, \mathtt{legalgarota}, \mathtt{ruimgaroto}, \mathtt{ruimgarota}\}$$

União: $A \cup B = \{x |\ x \in A \text{ ou } x \in B\}$.

Concatenação: $A \circ B = \{xy |\ x \in A \text{ e } y \in B\}$.

Estrela: $A^* = \{x_1 x_2 \dots x_k |\ k \geq 0 \text{ e cada } x_i \in A\}$.

Ex:

Suponha que o alfabeto Σ seja o alfabeto padrão de 26 letras $\{\mathtt{a}, \mathtt{b}, \dots, \mathtt{z}\}$. Se $A = \{\mathtt{legal}, \mathtt{ruim}\}$ e $B = \{\mathtt{garoto}, \mathtt{garota}\}$, então

$$A \cup B = \{\mathtt{legal}, \mathtt{ruim}, \mathtt{garoto}, \mathtt{garota}\}$$

$$A \circ B = \{\mathtt{legalgaroto}, \mathtt{legalgarota}, \mathtt{ruimgaroto}, \mathtt{ruimgarota}\}$$

$$A^* =$$

Ex:

União: $A \cup B = \{x | x \in A \text{ ou } x \in B\}$.

Concatenação: $A \circ B = \{xy | x \in A \text{ e } y \in B\}$.

Estrela: $A^* = \{x_1 x_2 \ldots x_k | k \geq 0 \text{ e cada } x_i \in A\}$.

Suponha que o alfabeto Σ seja o alfabeto padrão de 26 letras $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$. Se $A = \{\mathtt{legal}, \mathtt{ruim}\}$ e $B = \{\mathtt{garoto}, \mathtt{garota}\}$, então

$$A \cup B = \{\mathtt{legal}, \mathtt{ruim}, \mathtt{garoto}, \mathtt{garota}\}$$

$$A \circ B = \{\mathtt{legalgaroto}, \mathtt{legalgarota}, \mathtt{ruimgaroto}, \mathtt{ruimgarota}\}$$

$$A^* = \{\varepsilon, \mathtt{legal}, \mathtt{ruim}, \mathtt{legallegal}, \mathtt{legalruim}, \mathtt{ruimlegal}, \mathtt{ruimruim}, \mathtt{legallegallegal}, \mathtt{legallegalruim}, \mathtt{legalruimlegal}, \mathtt{legalruimruim}, \ldots\}.$$

Fechamento sob união

TEOREMA 1.25

A classe de linguagens regulares é fechada sob a operação de união.

Em outras palavras, se A_1 e A_2 são linguagens regulares, o mesmo acontece com $A_1 \cup A_2$.

Fechamento sob união

TEOREMA 1.25 ....

A classe de linguagens regulares é fechada sob a operação de união.

Em outras palavras, se A_1 e A_2 são linguagens regulares, o mesmo acontece com $A_1 \cup A_2$.

Ex: cadeias de 0's com nr de zeros sendo múltiplo de 2 OU 3

Fechamento sob união

TEOREMA 1.25 ....

A classe de linguagens regulares é fechada sob a operação de união.

Em outras palavras, se A_1 e A_2 são linguagens regulares, o mesmo acontece com $A_1 \cup A_2$.

Ex: cadeias de 0's com nr de zeros sendo múltiplo de 2 OU 3

PRIMEIRO prova usando AFDs (depois usando AFNs)

Fechamento sob união

TEOREMA 1.25 ....

A classe de linguagens regulares é fechada sob a operação de união.

Em outras palavras, se A_1 e A_2 são linguagens regulares, o mesmo acontece com $A_1 \cup A_2$.

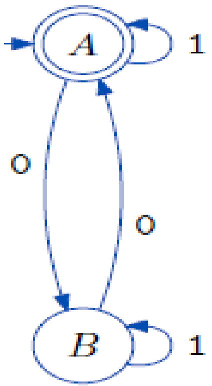
1-) Se A_1 e A_2 são regulares, então existem AFDs que reconhecem essas linguagens (M_1 e M_2, respectivamente)

2) Se pudermos construir um AFD que reconheça $A_1 \cup A_2$ (e só temos esses M_1 e M_2 genéricos para isso), então $A_1 \cup A_2$ é regular

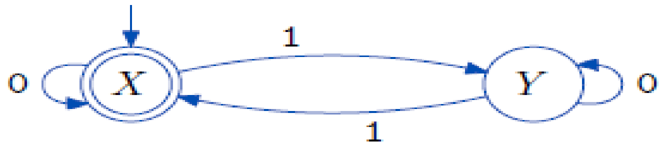
Fechamento sob união

- Prova:
  - Usando AFDs
  - Se L1 e L2 são regulares, então existem M1 e M2 tal que L1 = L(M1) e L2 = L(M2)
  - Construímos um autômato M que simule ao mesmo tempo M1 e M2
  - Tal prova nos dá o "algoritmo" para construir AFDs para linguagens do tipo "cadeias do tipo *x* ou do tipo *y* "

- Exemplo:
  - L1: sequências binárias com número par de 0's
  - L2: sequências binárias com número par de 1's



M1



M2

Fechamento sob união – Prova (AFDs)

Suponha que M_1 reconheça A_1, onde $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, e que
M_2 reconheça A_2, onde $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Construa M para reconhecer $A_1 \cup A_2$, onde $M = (Q, \Sigma, \delta, q_0, F)$.

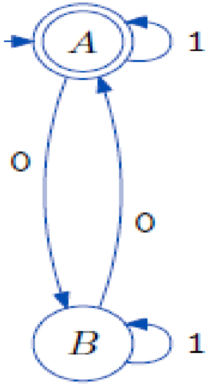
Fechamento sob união – Prova (AFDs)

Suponha que M_1 reconheça A_1, onde $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, e que
M_2 reconheça A_2, onde $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

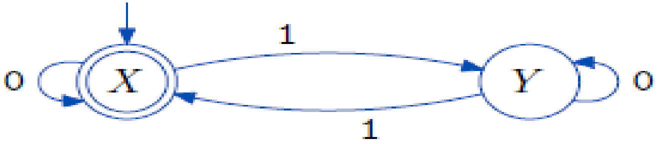
Construa M para reconhecer $A_1 \cup A_2$, onde $M = (Q, \Sigma, \delta, q_0, F)$.

1. $Q = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$.
Esse conjunto é o ***produto cartesiano*** dos conjuntos Q_1 e Q_2 e é escrito $Q_1 \times Q_2$. Trata-se do conjunto de todos os pares de estados, sendo o primeiro de Q_1 e o segundo de Q_2.

Q =



M1



M2

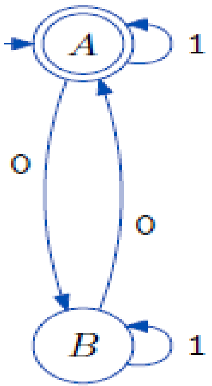
Fechamento sob união – Prova (AFDs)

Suponha que M_1 reconheça A_1, onde $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, e que M_2 reconheça A_2, onde $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

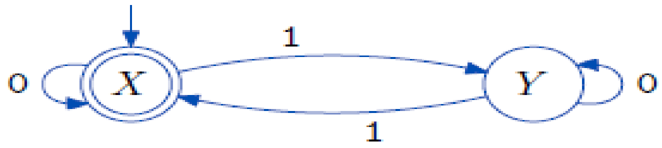
Construa M para reconhecer $A_1 \cup A_2$, onde $M = (Q, \Sigma, \delta, q_0, F)$.

1. $Q = \{(r_1, r_2) | \; r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$.
Esse conjunto é o ***produto cartesiano*** dos conjuntos Q_1 e Q_2 e é escrito $Q_1 \times Q_2$. Trata-se do conjunto de todos os pares de estados, sendo o primeiro de Q_1 e o segundo de Q_2.

Q = {(A,X), (A,Y), (B,X), (B,Y)}



M1



M2

Fechamento sob união – Prova (AFDs)

Suponha que M_1 reconheça A_1, onde $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, e que
M_2 reconheça A_2, onde $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Construa M para reconhecer $A_1 \cup A_2$, onde $M = (Q, \Sigma, \delta, q_0, F)$.

1. $Q = \{(r_1, r_2) | \; r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$.
Esse conjunto é o ***produto cartesiano*** dos conjuntos Q_1 e Q_2 e é escrito $Q_1 \times Q_2$. Trata-se do conjunto de todos os pares de estados, sendo o primeiro de Q_1 e o segundo de Q_2.

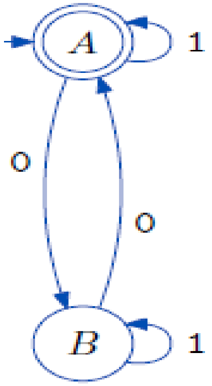
2. Σ, o alfabeto, é o mesmo em M_1 e M_2. Neste teorema e em todos os teoremas similares subseqüentes, assumimos por simplicidade que ambas M_1 e M_2 têm o mesmo alfabeto de entrada Σ. O teorema permanece verdadeiro se elas tiverem alfabetos diferentes, Σ_1 e Σ_2. Aí então modificaríamos a prova para tornar $\Sigma = \Sigma_1 \cup \Sigma_2$.

Fechamento sob união – Prova (AFDs)

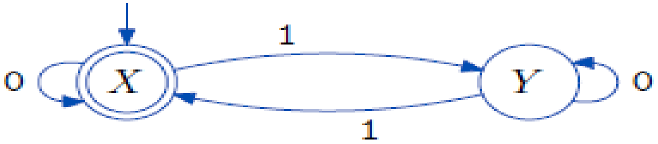
3. δ, a função de transição, é definida da seguinte maneira. Para cada $(r_1, r_2) \in Q$ e cada $a \in \Sigma$, faça

$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)).$$

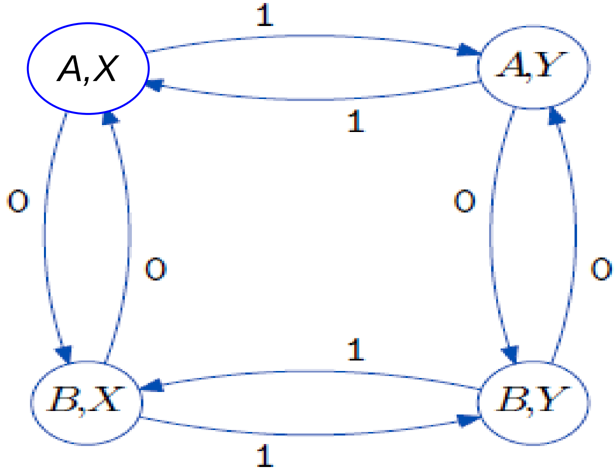
Logo, δ obtém um estado de M (que na realidade é um par de estados de M_1 e M_2), juntamente com um símbolo de entrada, e retorna o próximo estado de M.



M1



M2

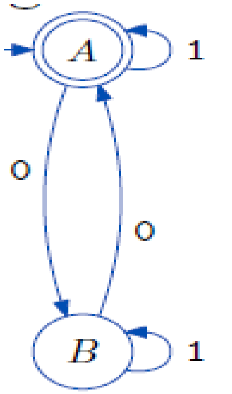


Fechamento sob união – Prova (AFDs)

3. δ, a função de transição, é definida da seguinte maneira. Para cada $(r_1, r_2) \in Q$ e cada $a \in \Sigma$, faça

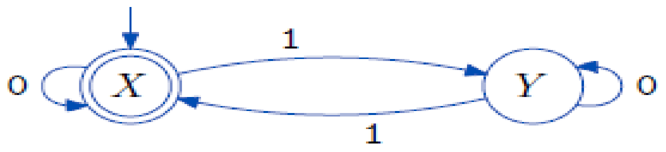
$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)).$$

Logo, δ obtém um estado de M (que na realidade é um par de estados de M_1 e M_2), juntamente com um símbolo de entrada, e retorna o próximo estado de M.

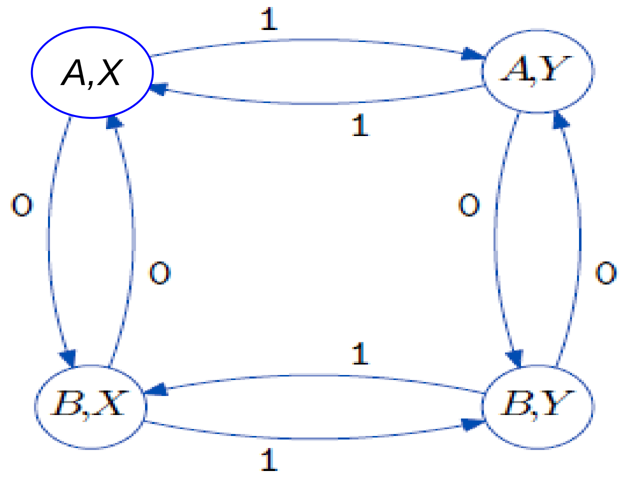


M1

4. q_0 é



M2



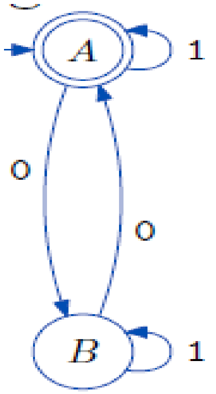
Fechamento sob união – Prova (AFDs)

3. δ, a função de transição, é definida da seguinte maneira. Para cada $(r_1, r_2) \in Q$ e cada $a \in \Sigma$, faça

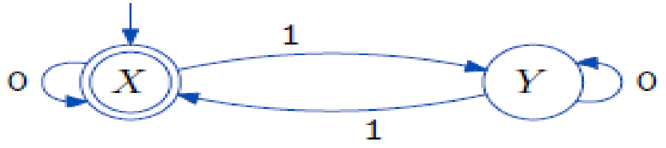
$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)).$$

Logo, δ obtém um estado de M (que na realidade é um par de estados de M_1 e M_2), juntamente com um símbolo de entrada, e retorna o próximo estado de M.

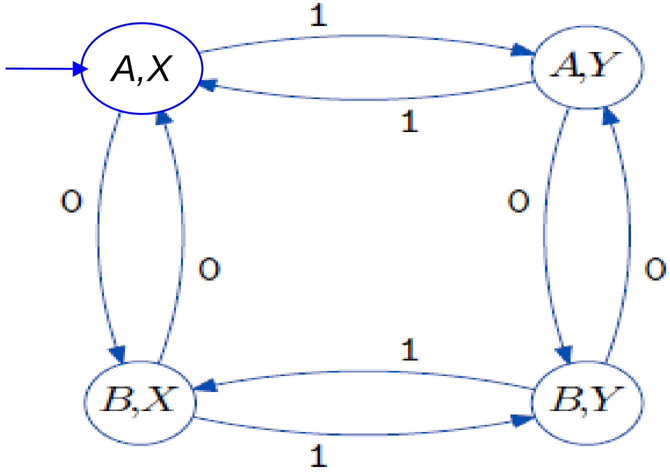
4. q_0 é o par (q_1, q_2).



M1

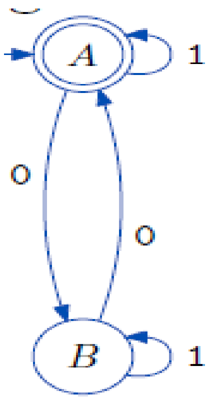


M2

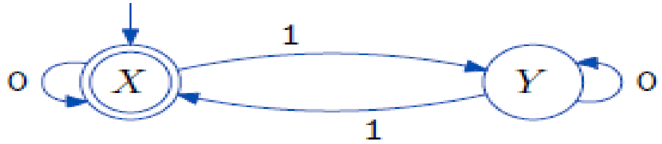


Fechamento sob união – Prova (AFDs)

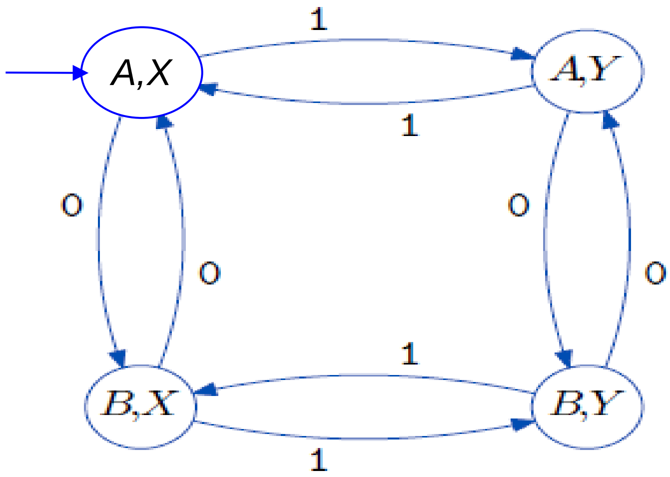
5. F



M1



M2

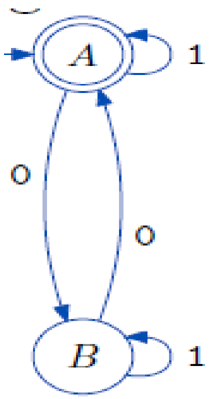


Fechamento sob união – Prova (AFDs)

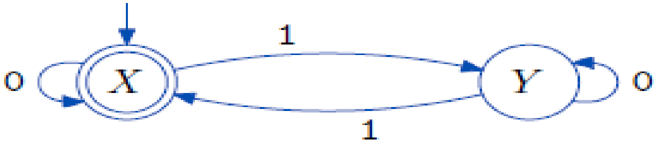
5. F é o conjunto de pares nos quais um dos membros é um estado de aceitação de M_1 ou M_2. Podemos escrevê-lo como

$$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ ou } r_2 \in F_2\}.$$

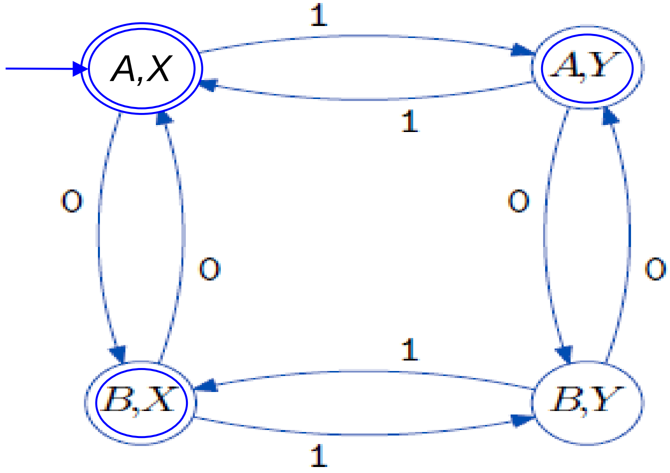
Essa expressão é a mesma que $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.



M1



M2



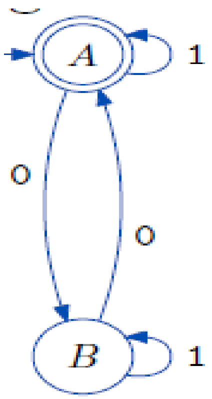
Fechamento sob união – Prova (AFDs)

5. F é o conjunto de pares nos quais um dos membros é um estado de aceitação de M_1 ou M_2. Podemos escrevê-lo como

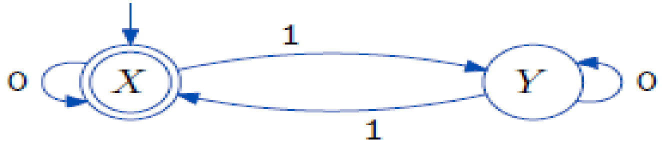
$$F = \{(r_1, r_2) | \, r_1 \in F_1 \text{ ou } r_2 \in F_2\}.$$

E se fosse “e”?

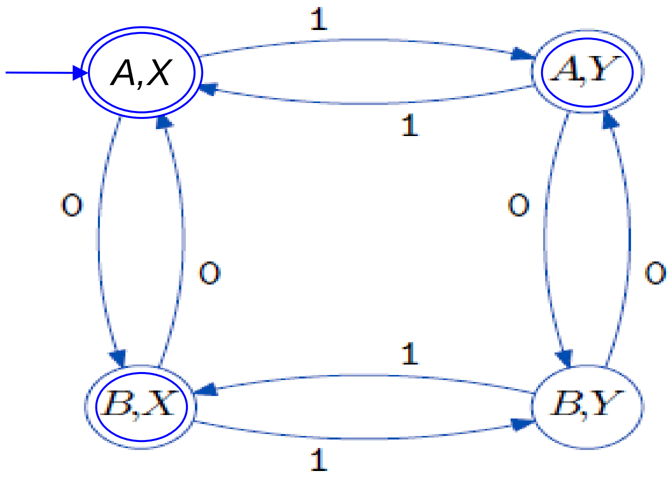
Essa expressão é a mesma que $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.



M1



M2



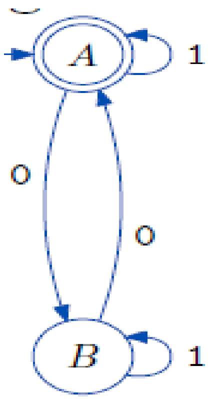
Fechamento sob intersecção – Prova (AFDs)

5. F é o conjunto de pares nos quais um dos membros é um estado de aceitação de M_1 ou M_2. Podemos escrevê-lo como

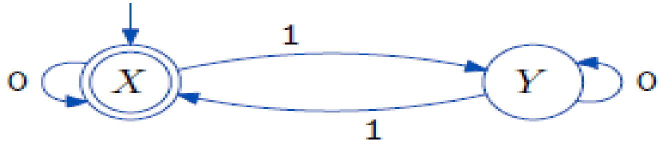
$$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ ou } r_2 \in F_2\}.$$

Essa expressão é a mesma que $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

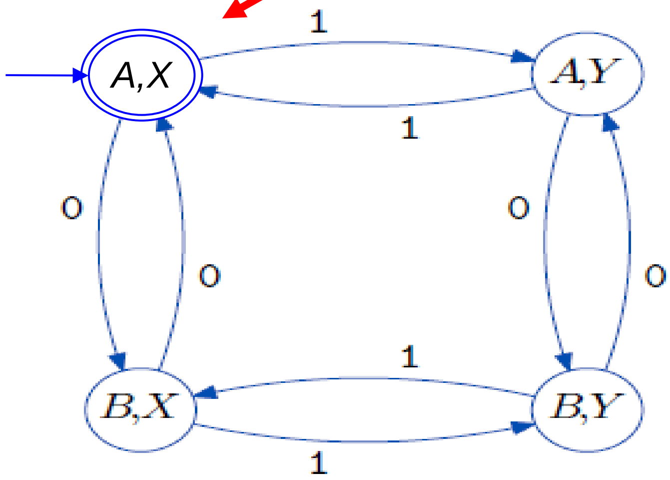
E se fosse "e"?
Intersecção!



M1



M2



Logo…
Fechamento sob a intersecção

- A classe de linguagens regulares é fechada sob a operação de intersecção

Linguagens do tipo “cadeias do tipo *x* e do tipo *y* ”

Dica de construção de AFDs

- Essa prova de fechamento sob as operações de união e intersecção nos dão um método de construção de AFDs para linguagens que sejam a união e/ou intersecção de outras linguagens

Exercícios do Sipser (2ª. edição)

1.4 Each of the following languages is the intersection of two simpler languages. In each part, construct AFDs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 50) to give the state diagram of a AFD for the language given. In all parts $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

a. $\{w \mid w$ has at least three a's and at least two b's$\}$

R**b.** $\{w \mid w$ has at exactly two a's and at least two b's$\}$

c. $\{w \mid w$ has an even number of a's and one or twc

R**d.** $\{w \mid w$ has an even number of a's and each a is fo

e. $\{w \mid w$ starts with an a and has at most one b$\}$

f. $\{w \mid w$ has an odd number of a's and ends with a

g. $\{w \mid w$ has even length and an odd number of a's

1.6 Give state diagrams of AFDs recognizing the following languages. In all parts the alphabet is $\{0,1\}$

a. $\{w \mid w$ begins with a 1 and ends with a 0$\}$

b. $\{w \mid w$ contains at least three 1s$\}$

c. $\{w \mid w$ contains the substring 0101, i.e., $w = x0101y$ for some x and $y\}$

d. $\{w \mid w$ has length at least 3 and its third symbol is a 0$\}$

e. $\{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$

f. $\{w \mid w$ doesn't contain the substring 110$\}$

g. $\{w \mid$ the length of w is at most 5$\}$

h. $\{w \mid w$ is any string except 11 and 111$\}$

i. $\{w \mid$ every odd position of w is a 1$\}$

j. $\{w \mid w$ contains at least two 0s and at most one 1$\}$

k. $\{\varepsilon, 0\}$

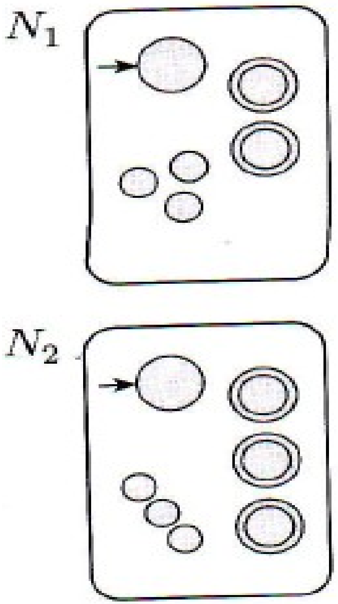
l. $\{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$

m. The empty set

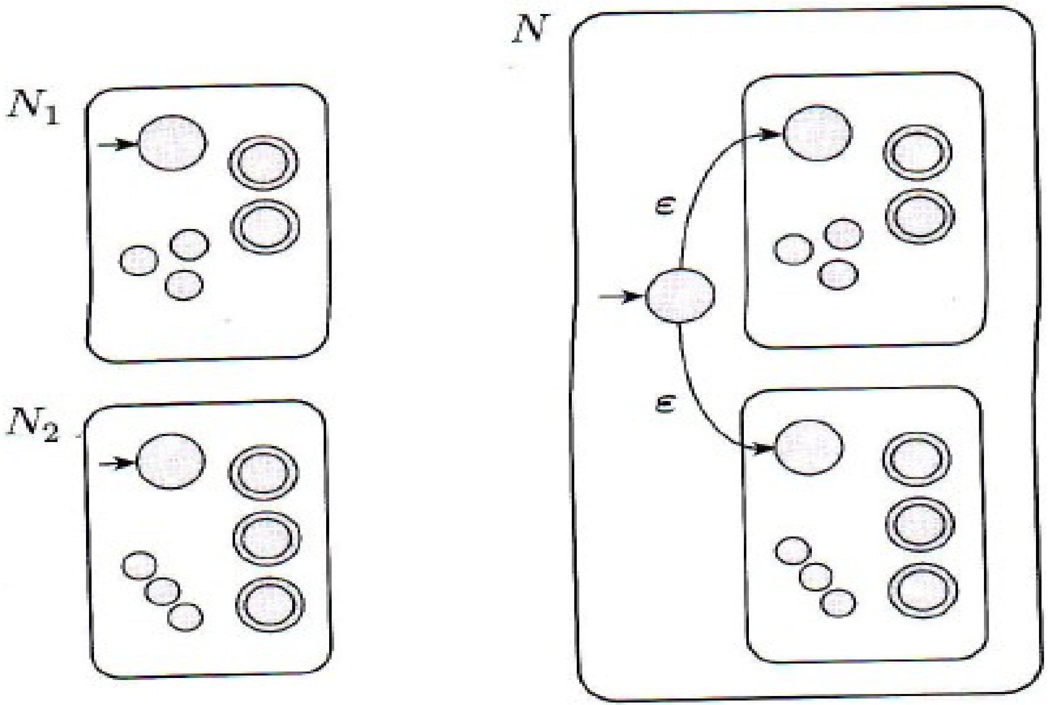
n. All strings except the empty string

Prova do fechamento em relação à união usando AFNs

- Sugestões?



Prova do fechamento em relação à união usando AFNs

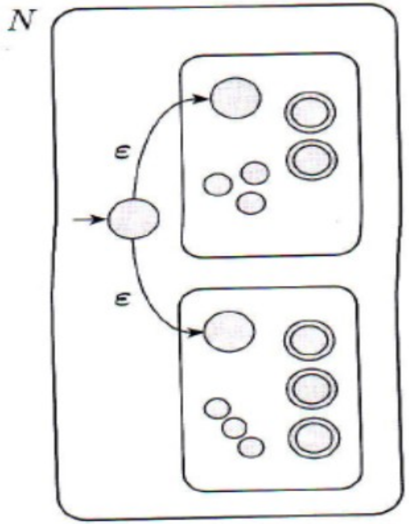


Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q =$

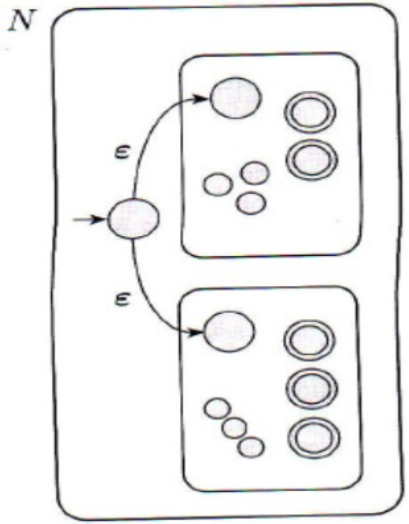


Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.



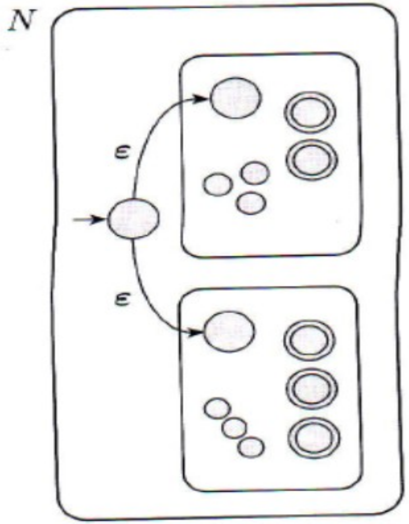
Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.

2. é o estado inicial de N.



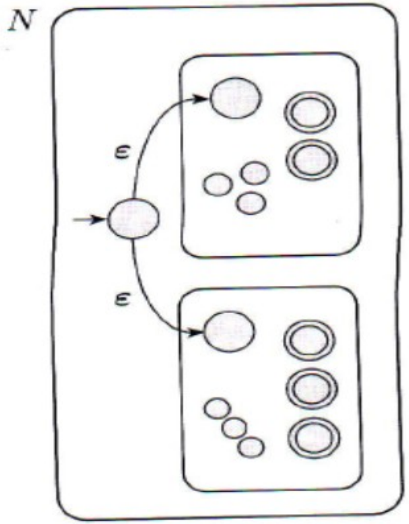
Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.

2. O estado q_0 é o estado inicial de N.

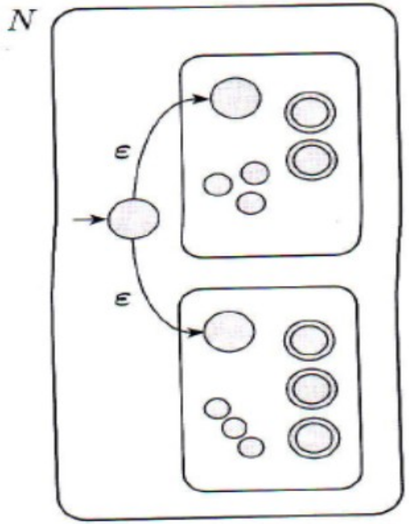


Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.
2. O estado q_0 é o estado inicial de N.
3. Os estados de aceitação $F =$



Prova (união) usando AFNs

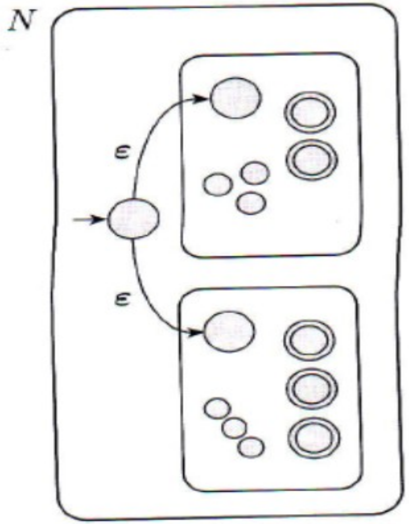
Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.

2. O estado q_0 é o estado inicial de N.

3. Os estados de aceitação $F = F_1 \cup F_2$.



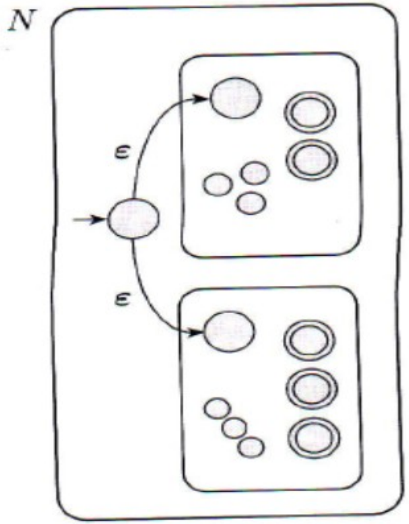
Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.
2. O estado q_0 é o estado inicial de N.
3. Os estados de aceitação $F = \quad F_1 \cup F_2$.
4.

$\delta(q, a) =$



Prova (união) usando AFNs

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

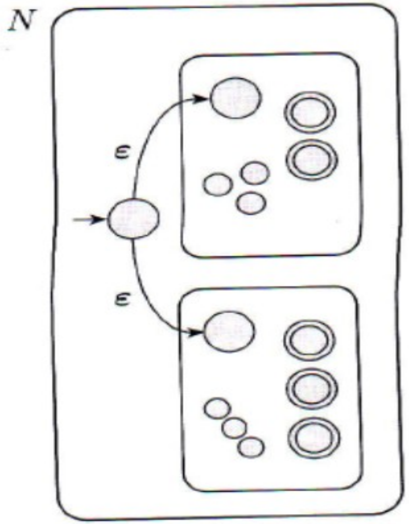
1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.

2. O estado q_0 é o estado inicial de N.

3. Os estados de aceitação $F = F_1 \cup F_2$.

4.

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



Fechamento sob concatenação

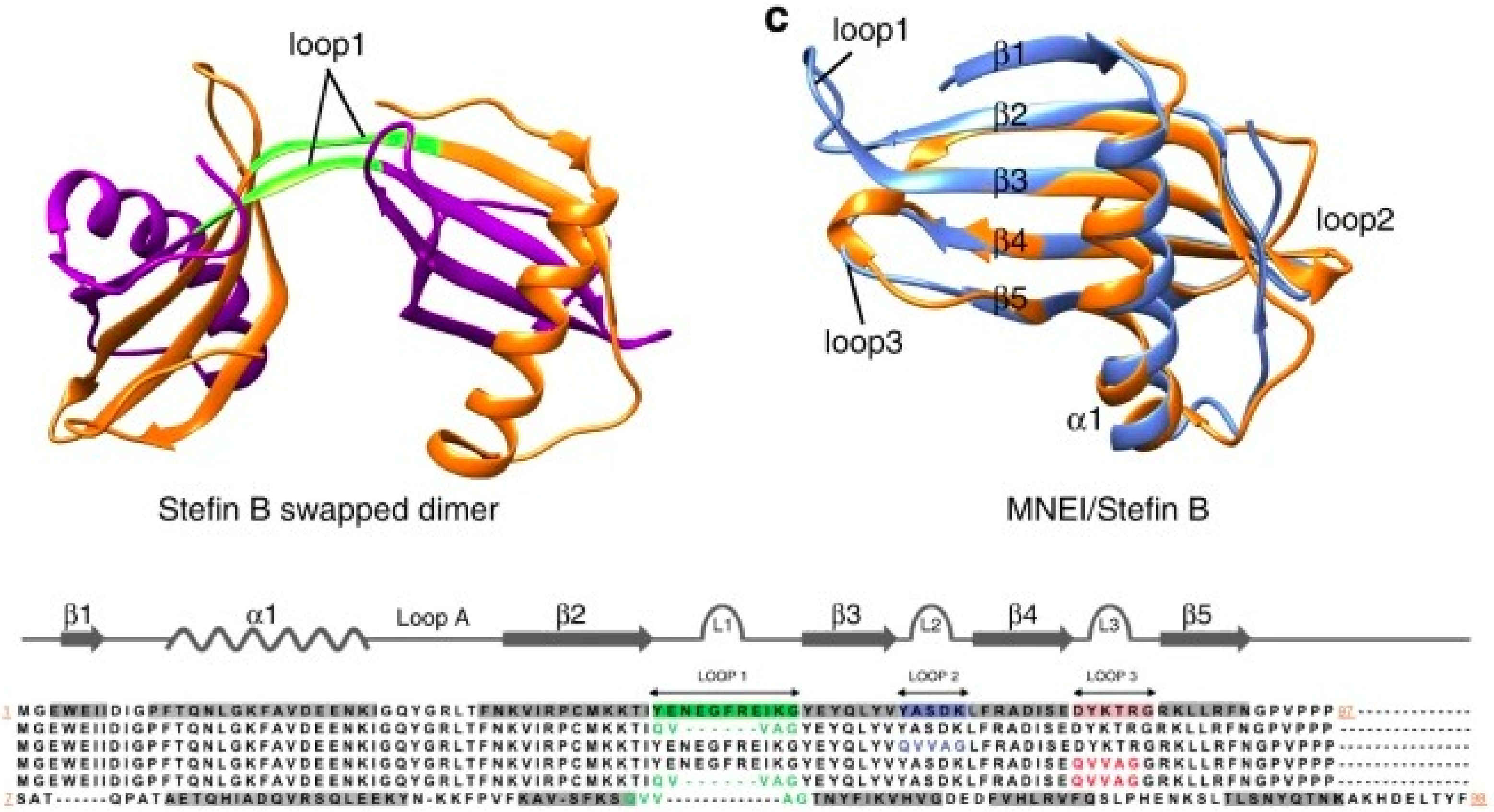
TEOREMA 1.26 ....

A classe de linguagens regulares é fechada sob a operação de concatenação.

Em outras palavras, se A_1 e A_2 são linguagens regulares, então o mesmo acontece com $A_1 \circ A_2$.

Concatenação: $A \circ B = \{xy |\ x \in A \text{ e } y \in B\}$

Exemplo em Bioinformática: domínios proteicos



https://www.nature.com/articles/s41467-019-08295-x

Fechamento sob concatenação

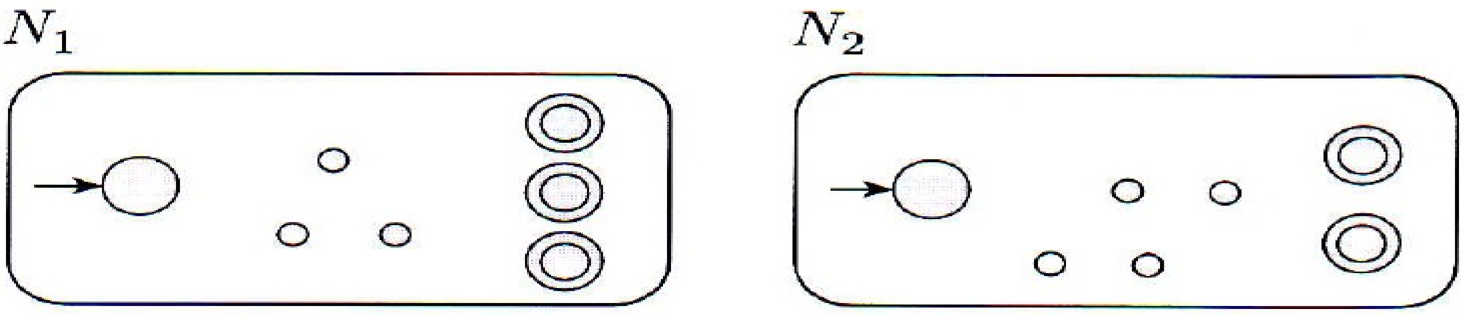
TEOREMA 1.26

A classe de linguagens regulares é fechada sob a operação de concatenação.

Em outras palavras, se A_1 e A_2 são linguagens regulares, então o mesmo acontece com $A_1 \circ A_2$.

- Prova? (usando AFNs)

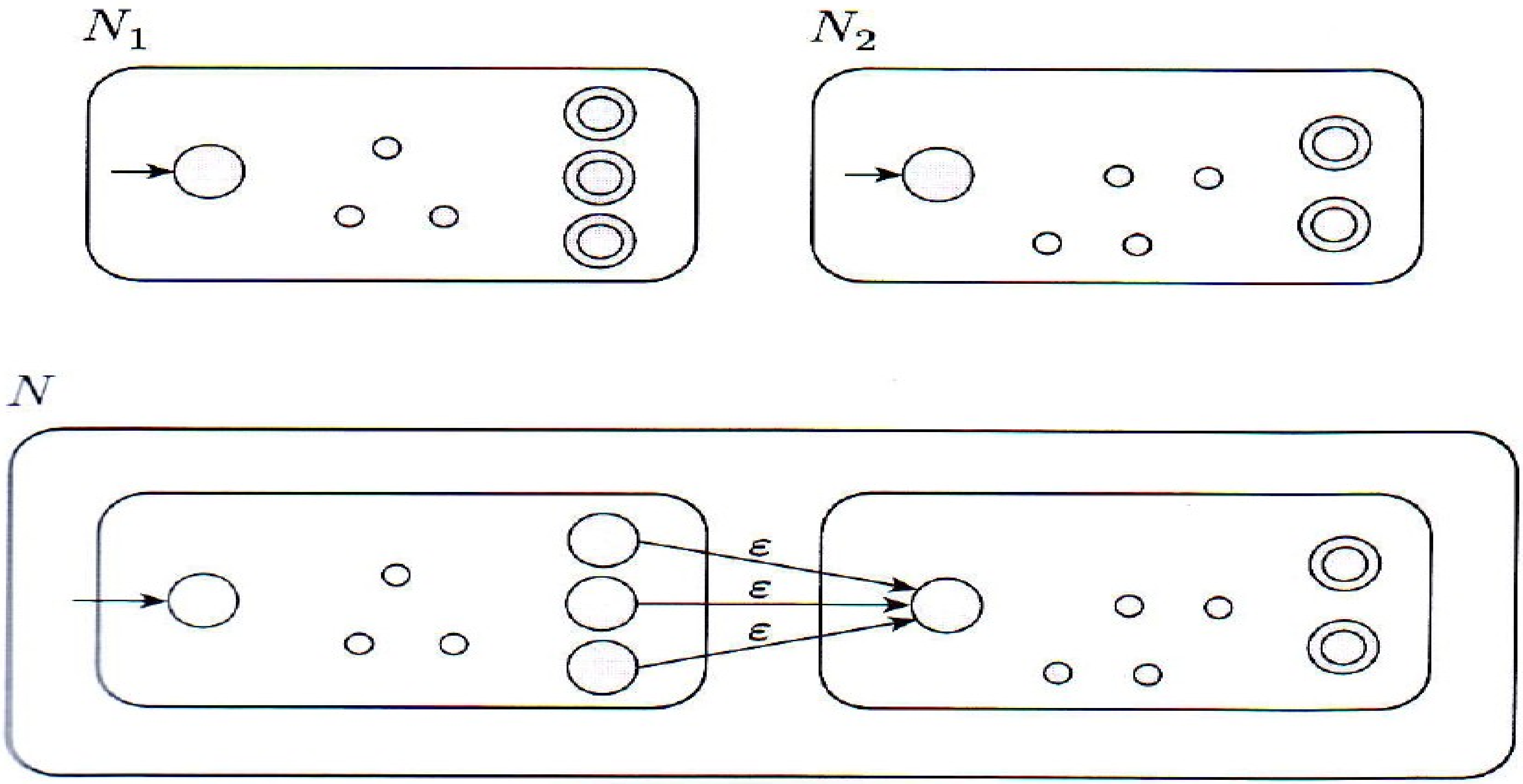
Fechamento sob concatenação: prova



1-) Se A_1 e A_2 são regulares, então existem AFNs que reconhecem essas linguagens (N_1 e N_2, respectivamente)

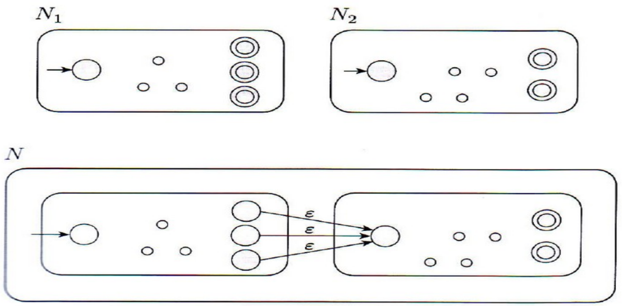
2) Se pudermos construir um AFN que reconheça $A_1 \circ A_2$ (e só temos esses N_1 e N_2 genéricos para isso), então $A_1 \circ A_2$ é regular

Fechamento sob concatenação: prova



Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

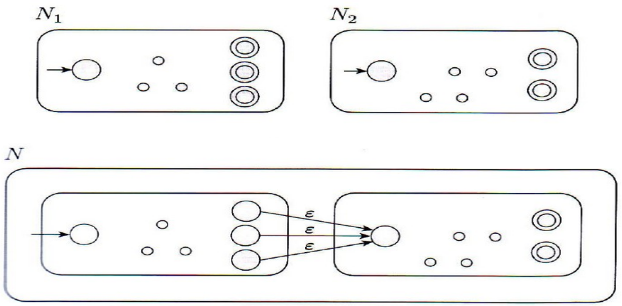


Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q =$

Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.



Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   Os estados de N são todos os estados de N_1 e N_2.

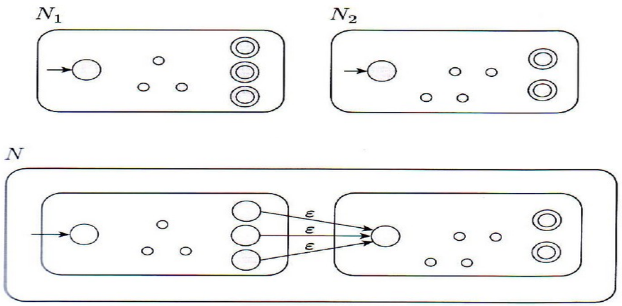
Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

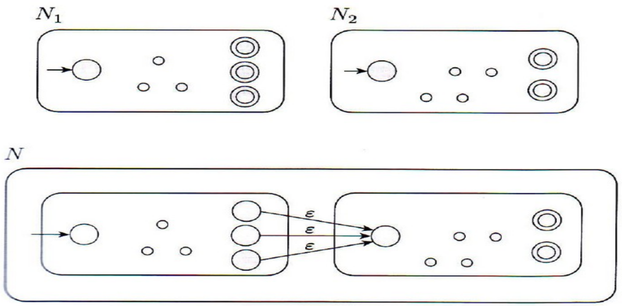
1. $Q = Q_1 \cup Q_2$.
Os estados de N são todos os estados de N_1 e N_2.

2. o estado inicial



Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.



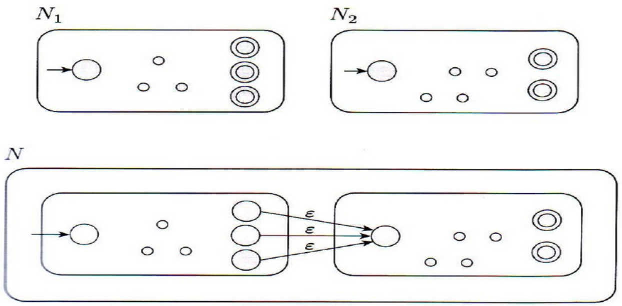
Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
Os estados de N são todos os estados de N_1 e N_2.

2. O estado q_1 é o mesmo que o estado inicial de N_1.

Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

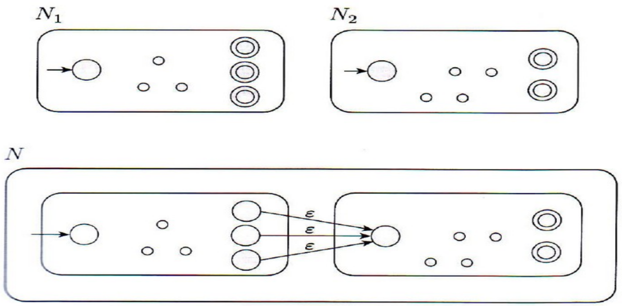


Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   Os estados de N são todos os estados de N_1 e N_2.
2. O estado q_1 é o mesmo que o estado inicial de N_1.
3. Os estados de aceitação

Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

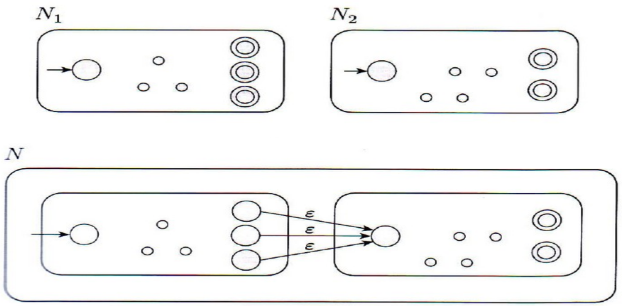


Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   Os estados de N são todos os estados de N_1 e N_2.
2. O estado q_1 é o mesmo que o estado inicial de N_1.
3. Os estados de aceitação F_2 são os mesmos que os estados de aceitação de N_2.

Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.

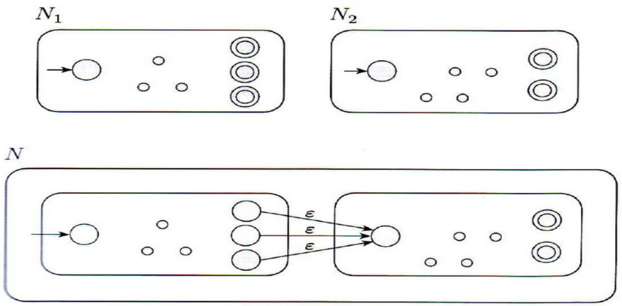


Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   Os estados de N são todos os estados de N_1 e N_2.
2. O estado q_1 é o mesmo que o estado inicial de N_1.
3. Os estados de aceitação F_2 são os mesmos que os estados de aceitação de N_2.
4. Defina δ de modo que para qualquer $q \in Q$ e qualquer $a \in \Sigma_\varepsilon$,

Fechamento sob concatenação: prova

Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1, e que
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ reconheça A_2.



Construa $N = (Q, \Sigma, \delta, q_1, F_2)$ para reconhecer $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   Os estados de N são todos os estados de N_1 e N_2.
2. O estado q_1 é o mesmo que o estado inicial de N_1.
3. Os estados de aceitação F_2 são os mesmos que os estados de aceitação de N_2.
4. Defina δ de modo que para qualquer $q \in Q$ e qualquer $a \in \Sigma_\varepsilon$,

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

Fechamento sob op. estrela

TEOREMA 1.49

A classe de linguagens regulares é fechada sob a operação estrela.

Estrela: $A^* = \{x_1 x_2 \ldots x_k | \; k \geq 0 \text{ e cada } x_i \in A\}$

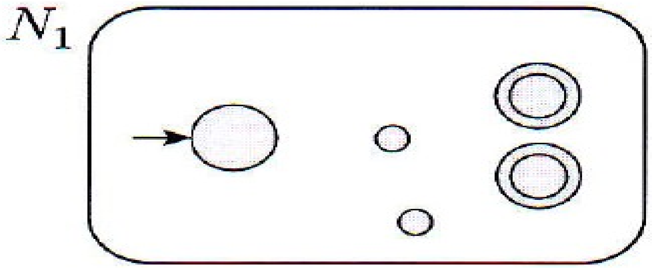
Fechamento sob op. estrela

TEOREMA 1.49

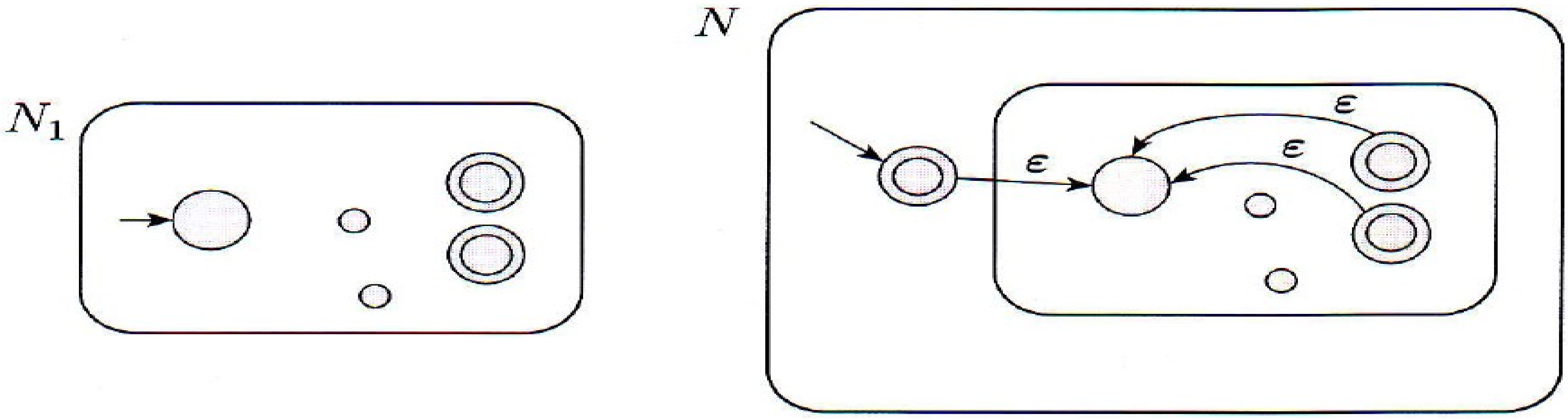
A classe de linguagens regulares é fechada sob a operação estrela.

- Prova?

Fechamento sob op. estrela - prova



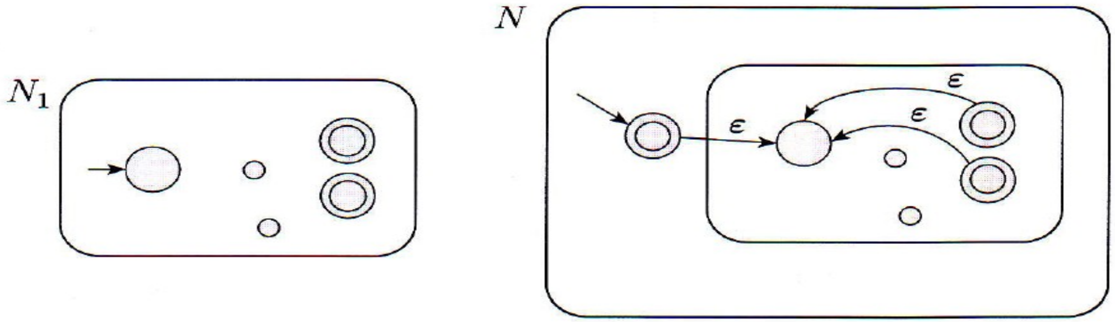
Fechamento sob op. estrela - prova



Fechamento sob op. estrela - prova

PROVA Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1.
Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer A_1^*.

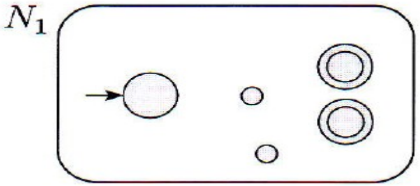
1. $Q = \{q_0\} \cup Q_1$.
   Os estados de N são os estados de N_1 mais um novo estado inicial.
2. O estado q_0 é o novo estado inicial.
3. $F = \{q_0\} \cup F_1$.
   Os estados de aceitação são os antigos estados de aceitação mais o novo estado inicial.
4. Defina δ de modo que para qualquer $q \in Q$ e qualquer $a \in \Sigma_\varepsilon$,

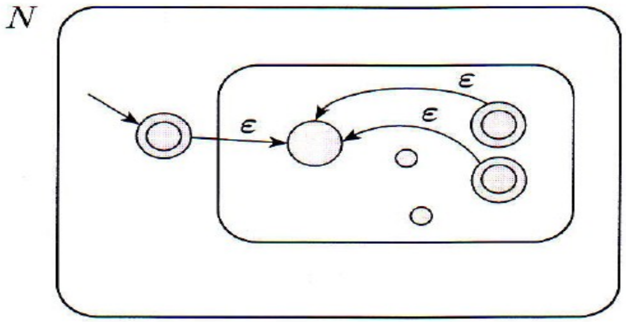


Fechamento sob op. estrela - prova

PROVA Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1. Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer A_1^*.

1. $Q = \{q_0\} \cup Q_1$.
   Os estados de N são os estados de N_1 mais um novo estado inicial.
2. O estado q_0 é o novo estado inicial.
3. $F = \{q_0\} \cup F_1$.
   Os estados de aceitação são os antigos estados de aceitação mais o novo estado inicial.
4. Defina δ de modo que para qualquer $q \in Q$ e qualquer $a \in \Sigma_\varepsilon$,





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

Ex: Fechamento sob o complemento?

- Será que as linguagens regulares são fechadas sob a operação de complemento?
  - L é uma linguagem regular
  - $\overline{L}$ é o complemento de L, isto é,

    $\overline{L}$ = { w | w não pertence a L}

- Dado um AFD que reconhece L, conseguiria criar um AFD que reconhece $\overline{L}$?

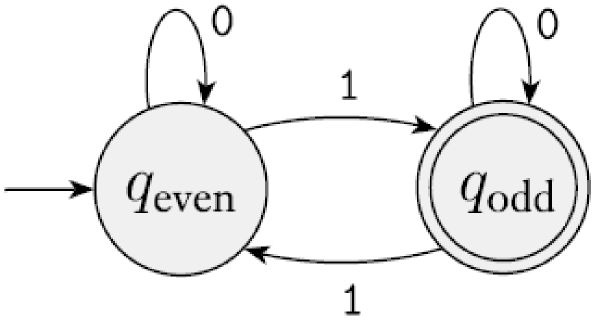
Fechamento sob o complemento

- SIM! As linguagens regulares são fechadas sob a operação de complemento!!!
  - M é um AFD que reconhece L, uma linguagem regular
  - $\overline{M}$ que reconhece $\overline{L}$, é um AFD obtido de M invertendo a propriedade **de cada estado ser final ou não** (estados finais de M deixam de ser finais em $\overline{M}$ e estados não finais em M são estados finais em $\overline{M}$

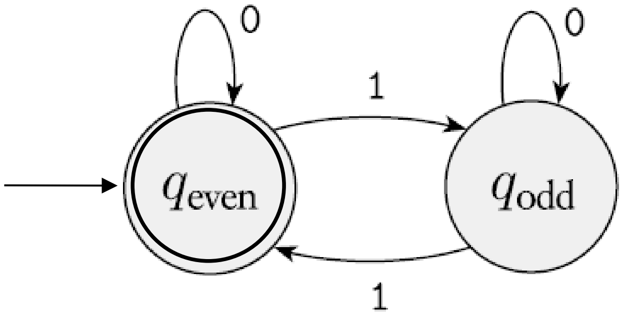
$$\overline{M} = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

Fechamento sob o complemento - Exemplo

L_1 = { x | soma dos dígitos de x é ímpar}



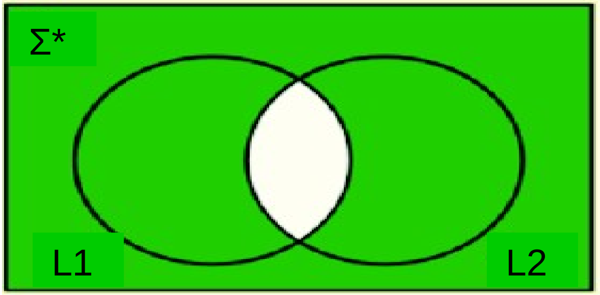
$L_2 = \overline{L_1}$ = { x | soma dos dígitos de x não é ímpar (par)}



Ex: Fechamento sob a intersecção

- Em slides anteriores mostramos que a classe de linguagens regulares é fechada sob a intersecção utilizando AFDs
- Também poderíamos provar utilizando o resultado do fechamento sob o complemento e união:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}.$$



Ex: Fechamento sob a intersecção

- Em slides anteriores mostramos que a classe de linguagens regulares é fechada sob a intersecção utilizando AFDs
- Também poderíamos provar utilizando o resultado do fechamento sob o complemento e união

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}.$$

- $\overline{L_1}$ and $\overline{L_2}$ are regular
- $\overline{L_1} \cup \overline{L_2}$ is regular
- Hence, $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ is regular.

Em resumo

- Vimos hoje como provar que a classe de linguagens regulares é fechada sob uma determinada operação
- Em particular, é fechada sob as operações:

ar um

Em resumo

- Vimos hoje como provar que a classe de linguagens regulares é fechada sob uma determinada operação
- Em particular, é fechada sob as operações:
  - União
  - Intersecção
  - Concatenação
  - Estrela
  - Complemento
- É também fechada sob outras… (ex: sufixo)
- Novamente, saber isso é importante para sabermos se podemos ou não criar um AFD/AFN para uma linguagem de interesse...

Dica de construção de AFDs/AFNs

- Essas provas de fechamento sob as operações regulares nos dão um método de construção de AFDs/AFNs para linguagens que sejam a união, intersecção, concatenação, estrela e complementação de outras linguagens

Exercícios Sipser

- 1.1 até 1.11 (já conseguem fazer todos!!!)