ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 1

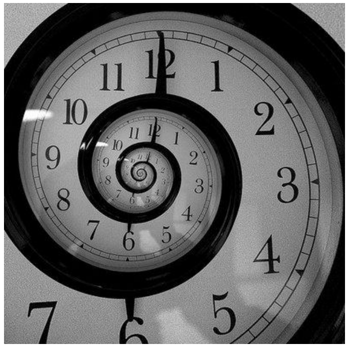
Prof. Marcelo S. Lauretto
marcelolauretto@usp.br
www.each.usp.br/lauretto

Introdução à Teoria da Computação

- Por que estudar teoria?

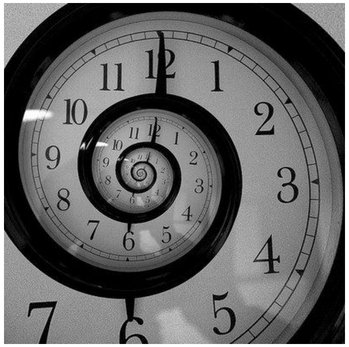
Introdução à Teoria da Computação

- Complexidade
- Computabilidade
- Teoria dos autômatos
- Linguagens formais



Complexidade

- Em quanto “tempo” um problema pode ser resolvido por um computador?
- Benefícios de conhecer complexidade:



Complexidade

- Em quanto “tempo” um problema pode ser resolvido por um computador?
- Benefícios de conhecer complexidade:
  - Estimar o tempo correto
  - Adaptar o problema
  - Solução de aproximação
  - Satisfazer-se com o que não for o pior caso
  - Tipos alternativos de computação (aleatorizada)
  - Criptografia

Computabilidade

- Classificação dos problemas em solúveis e não solúveis
- Benefícios:

Computabilidade

- Classificação dos problemas em solúveis e não solúveis
- Benefícios:
  - Adaptar o problema

Teoria dos autômatos

- Modelo matemático de computação
- Autômatos x Gramáticas
- Aplicações teóricas
- Aplicações práticas:

Teoria dos autômatos

- Modelo matemático de computação
- Autômatos x Gramáticas
- Aplicações teóricas
- Aplicações práticas:
  - Compilação, linguagens de programação
  - Processamento de texto
  - Projeto de hardware
  - Inteligência artificial
    - Bioinformática
    - Processamento de linguagens naturais
    - Visão computacional
    - ...

Disciplina - Bibliografia

- Livro base:
  - SIPSER, M. Introdução à Teoria da Computação. Ed. Thomson

- Livro complementar:
  - RAMOS, M. V. M.; NETO, J. J.; VEGA, I. S. Linguagens Formais. Ed. Bookman

Disciplina - Conteúdo

- Ordem dos temas:
  - Modelos de computação:
    - Cap 1: Autômatos Finitos x Linguagens Regulares
    - Cap 2: Autômatos a Pilha x Gramáticas Livres do Contexto
    - Cap. 3: Máquinas de Turing
  - Cap. 4: Computabilidade
  - Cap. 7: Complexidade

Disciplina – Avaliação

- 3 provas teóricas
    - P1: 20/09
    - P2: 31/10
    - P3: 29/11
    - Sub (fechada): 05/12
- 2 exercícios-programas (em duplas)
    - EP1: 07/10
    - EP2: 18/11
- Listas de exercícios (individuais)
  - Datas serão divulgadas ao longo do curso

Avaliação

- Média 1a. aval:
  - M1 = (3*P1 + 3*P2 + 3*P3 + EP1 + EP2 + ML)/12
    - ML: média das notas das listas

- Média 2a aval. = (M1+REC)/2

Outras Informações:

- Caio Santiago – caio.santiago@usp.br
- Horários de atendimento de dúvidas:
  - 3as feiras das 18:00 às 19:00 no Lab7
  - Dia 16/08 (5ª feira) das 20:00 às 21:30 no Lab7
- Não haverá aulas nos dias:
  - 15, 16/08 (Congresso)
  - 22, 23/08 (3ª Semana de Sistemas de Informação)

Cap 1 – Linguagens regulares
(agradecimentos à Profa. Ariane)

- Autômatos finitos
- Não determinismo
- Relação com modelos de Markov
- Expressões regulares
- Gramáticas regulares
- Linguagens não-regulares

Autômatos finitos

- Necessidade de um modelo para entender (estudar) um computador
- Vários modelos computacionais com diferentes características (e complexidades)
- O modelo mais simples:
  - Máquina de estados finitos ou
  - Autômato de estados finitos ou
  - Autômato finito
  - *Finite State Automaton (FSA)*

Autômatos finitos

- O exemplo de um controlador de portas

Autômatos finitos

- O que esse controlador precisa guardar em memória?

Autômatos finitos

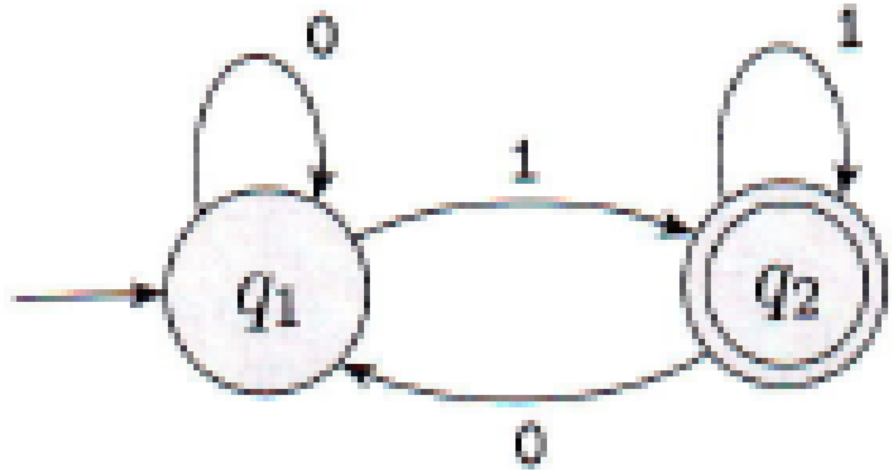
- O que esse controlador precisa guardar em memória?
  - Estado atual (aberto/fechado: 1 bit)
- Vários outros dispositivos (ex: eletrodomésticos) podem ser implementados de forma semelhante, com uma memória limitada

Autômatos finitos

- Autômatos finitos são mecanismos RECONHECEDORES
- Ex: como seria o autômato para reconhecer strings binárias que começam e terminam com zero, podem ter 0s ou 1s no meio, com tamanho pelo menos 1?
  - 0, 00, 010, 000000, 0101110, ...

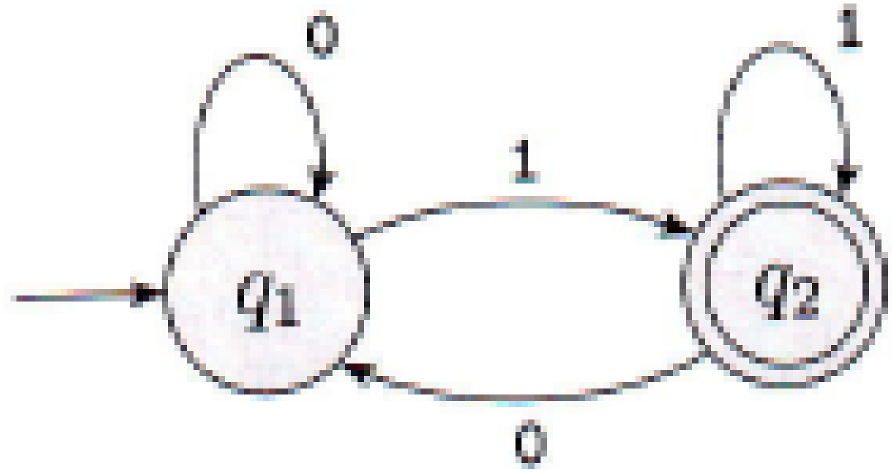
Autômatos finitos

- Diagrama de estados
- Ex: o que esse autômato A3 reconhece?



Autômatos finitos

- Diagrama de estados
- Ex: o que esse autômato A3 reconhece?



- Sequência binárias que terminam em 1

Autômatos finitos

- A **linguagem** reconhecida por um autômato é o conjunto das **cadeias** (de símbolos de entrada) aceitas pelo autômato
- L(A3) = {w | w é uma string binária e termina em 1}

Autômatos finitos

- Definição formal:

Autômatos finitos

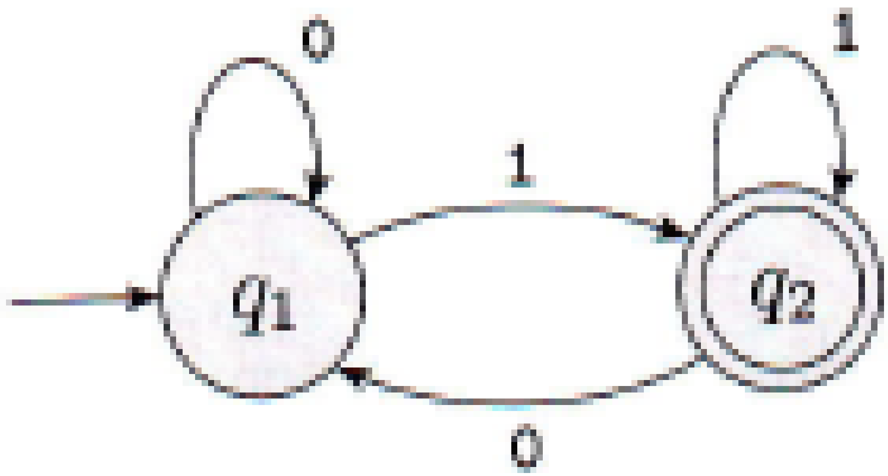
- Definição formal:

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]

Autômatos finitos

- Qual a definição formal do autômato A3?

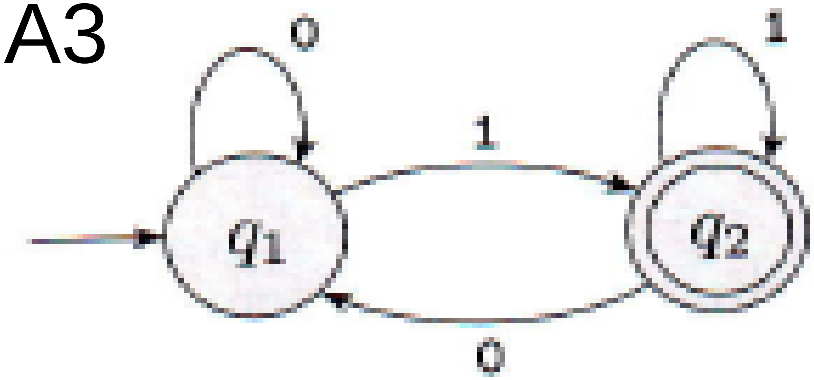


Um ***autômato finito*** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

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]

Autômatos finitos

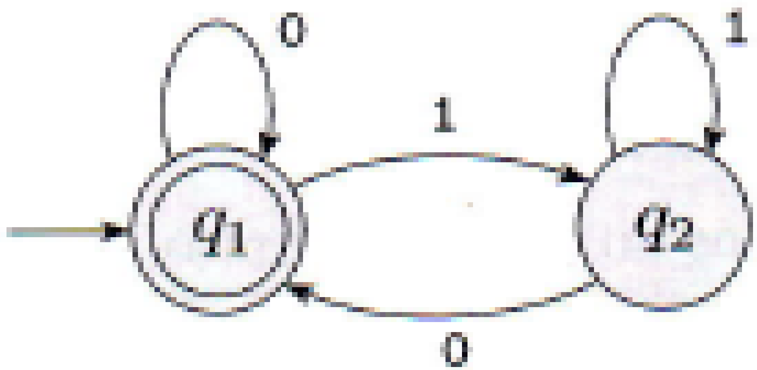
A3



L(A3) = {w | w é binária e termina com 1}

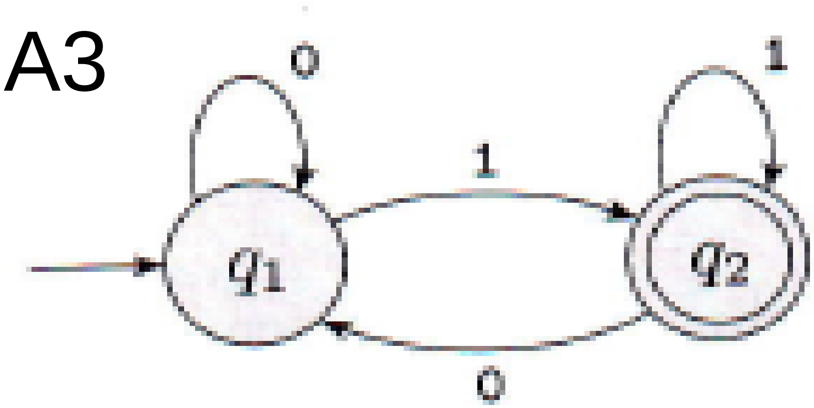
- Que linguagem esse autômato reconhece? (apenas mudou o estado final)

A4



Autômatos finitos

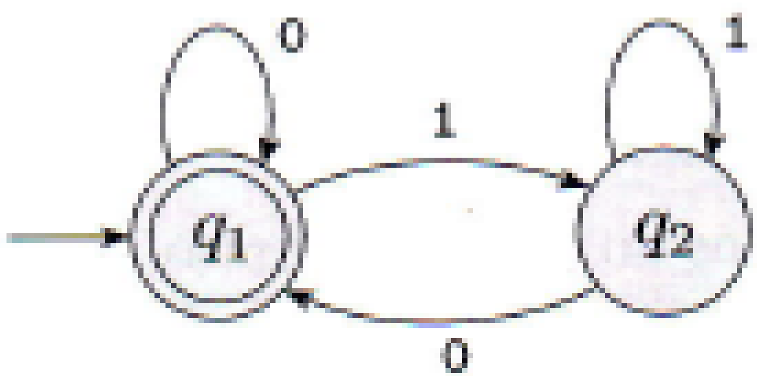
A3



L(A3) = {w | w é binária e termina com 1}

- Que linguagem esse autômato reconhece? (apenas mudou o estado final)

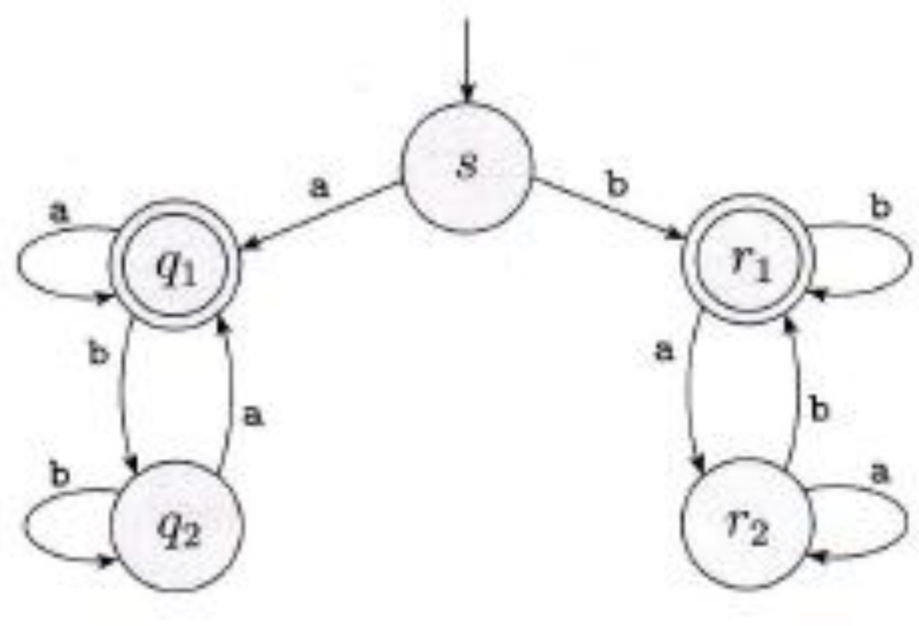
A4



L(A4) = {w | w é binária e começa com 0

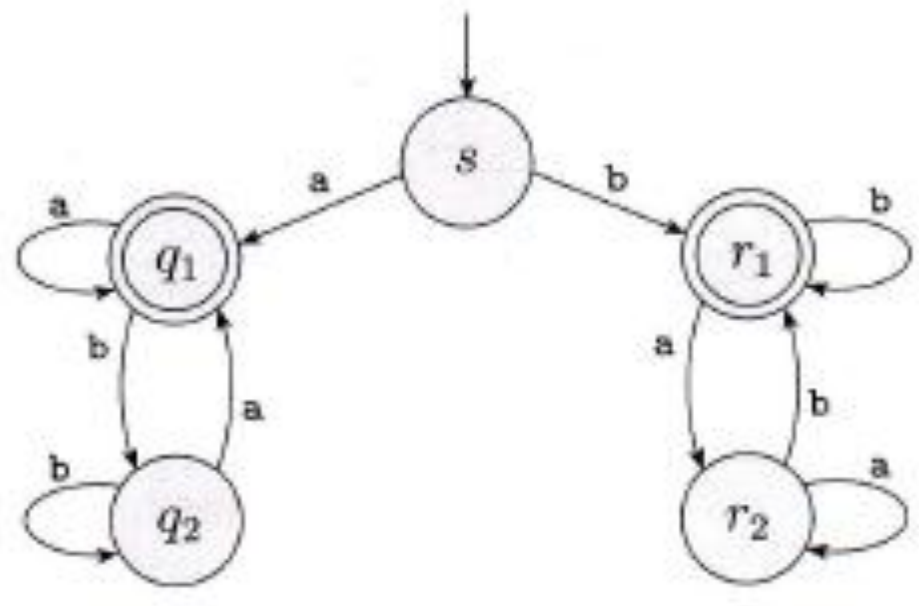
Autômatos finitos

- Que linguagem esse autômato reconhece?



Autômatos finitos

- Que linguagem esse autômato reconhece?



- Cadeias que comecem e terminem com o mesmo símbolo

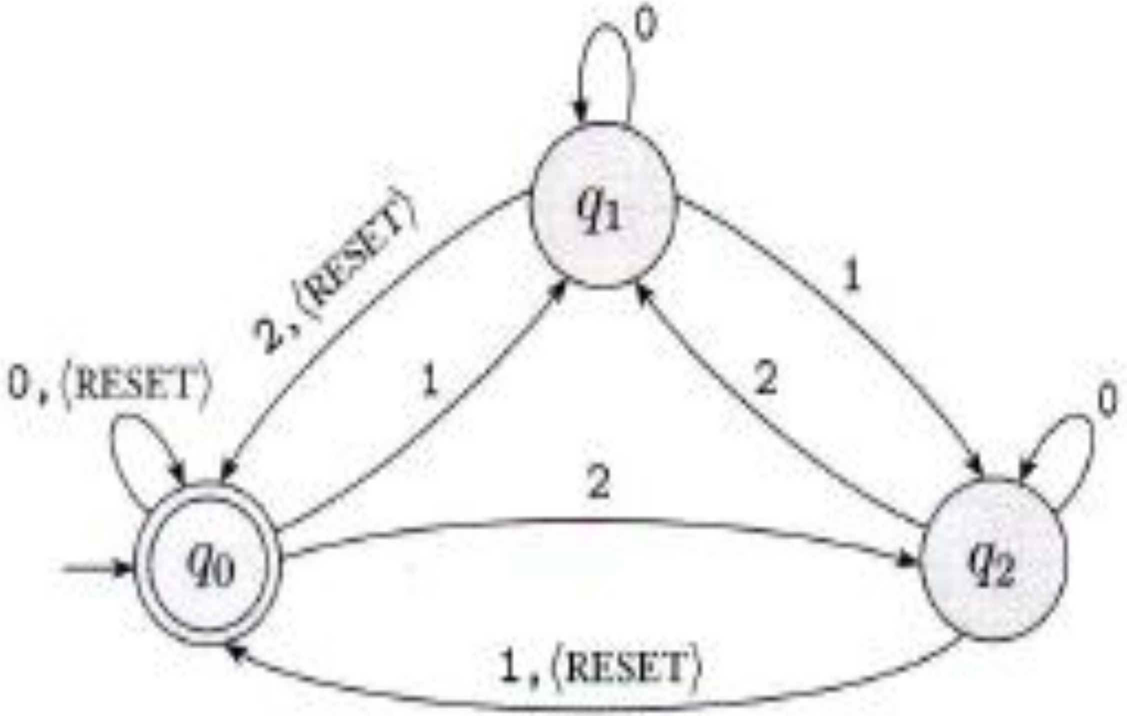
Projetando autômatos

- Pense que você é um autômato
- A cadeia de entrada pode ser arbitrariamente grande
- Sua memória é finita (o número de estados é finito)
- A transição se dá dados o estado atual e o próximo símbolo de entrada
- Você recebe um símbolo por vez, e não sabe quando a cadeia vai acabar (você precisa ter sempre uma “resposta corrente”)

Exercício

- Projete um autômato (diagrama de estados) que, dado $\Sigma = \{0,1,2,$<RESET>$\}$, aceita a cadeia de entrada se a soma dos números for igual a 0 módulo 3 (ou seja, se a soma for um múltiplo de 3). <RESET> zera o contador

Exercício - solução



Autômatos finitos

- Pode ser mais conveniente projetar o autômato usando a definição formal ao invés do diagrama de estados
- Ex: generalização do autômato anterior para aceitar somas múltiplas de i

Autômatos finitos

- Pode ser mais conveniente projetar o autômato usando a definição formal ao invés do diagrama de estados
- Ex: generalização do autômato anterior para aceitar somas múltiplas de i

autômato finito B_i, reconhecendo A_i. Descrevemos a máquina B_i formalmente da seguinte forma: $B_i = (Q_i, \Sigma, \delta_i, q_0, \{q_0\})$, onde Q_i é o conjunto de i estados $\{q_0, q_1, q_2, \ldots, q_{i-1}\}$, e desenhamos a função de transição δ_i de modo que para cada j, se B_i está em q_j, a soma corrente é j, módulo i. Para cada q_j faça

$$\delta_i(q_j, 0) = q_j,$$
$$\delta_i(q_j, 1) = q_k, \text{ onde } k = j + 1 \text{ módulo } i,$$
$$\delta_i(q_j, 2) = q_k, \text{ onde } k = j + 2 \text{ módulo } i, \text{ e}$$
$$\delta_i(q_j, \langle\text{RESET}\rangle) = q_0.$$

Definição formal de computação

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um autômato finito e suponha que $w = w_1 w_2 \cdots w_n$ seja uma cadeia onde cada w_i é um membro do alfabeto Σ. Então M ***aceita*** w se existe uma seqüência de estados $r_0, r_1, \ldots, r_n$ em Q com três condições:

1. $r_0 = q_0$,

2. $\delta(r_i, w_{i+1}) = r_{i+1}$, para $i = 0, \ldots, n-1$, e

3. $r_n \in F$.

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

Operações regulares

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

Operações regulares

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

$$A \cup B =$$

Operações regulares

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

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

Operações regulares

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

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

$$A \circ B =$$

Operações regulares

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

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

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

Operações regulares

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

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

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

$$A^* =$$

Operações regulares

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

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

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

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

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

- Prova:
  - sugestões?

Fechamento sob união

- Prova:
  - sugestões?
  - construímos um autômato M que simule ao mesmo tempo M1 e M2

Fechamento sob união - Prova

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

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.

Fechamento sob união - Prova

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. $Q = \{(r_1, r_2) | \; r_1 \in Q_1$ 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

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

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

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.

Fechamento sob união - Prova

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

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

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).

Fechamento sob união - Prova

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

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

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).

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

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

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

Fechamento sob união - Prova

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

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

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.

E se fosse "e"?

4. q_0 é o par (q_1, q_2).

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

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

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

Fechamento sob união - Prova

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.

E se fosse "e"?
Intersecção!

4. q_0 é o par (q_1, q_2).

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

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

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

Fechamento sob concatenação

- O que acham?

Fechamento sob concatenação

- O que acham?

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 $A_1 \circ A_2$.

Fechamento sob concatenação

- O que acham?

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?

Fechamento sob concatenação

- O que acham?

TEOREMA 1.26

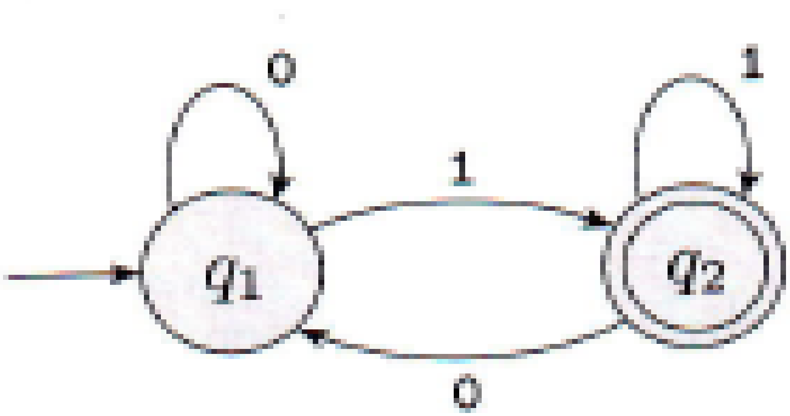
A classe de linguagens regulares é fechada sob a operação de concatenação.

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

- Prova? Precisamos do conceito de não-determinismo

Autômatos Finitos Determinísticos (AFD)

- Dado um estado atual e um símbolo de entrada sabemos exatamente para onde ir (está determinado)



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

1. Q é um conjunto finito conhecido como os ***estados***,

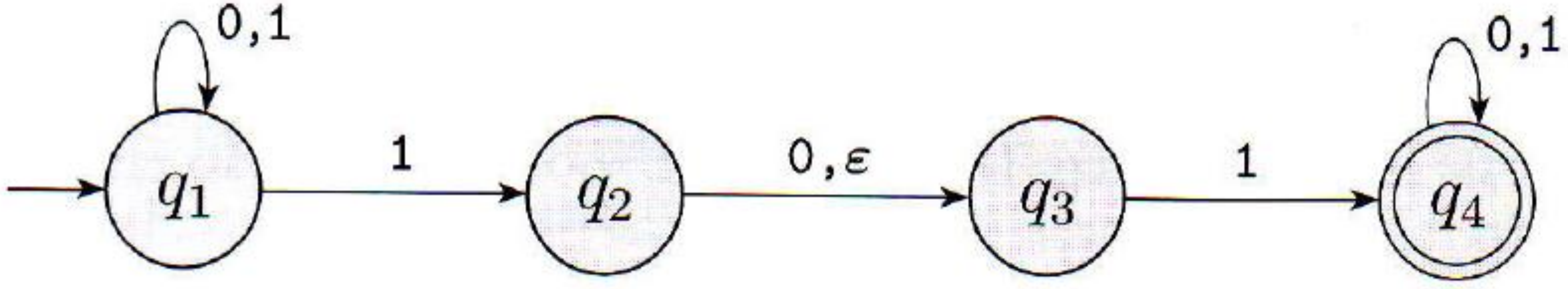
2. Σ é um conjunto finito chamado o ***alfabeto***,

3. $\delta\colon Q \times \Sigma \longrightarrow Q$ é a ***função de transição***,[1]

4. $q_0 \in Q$ é o ***estado inicial***, e

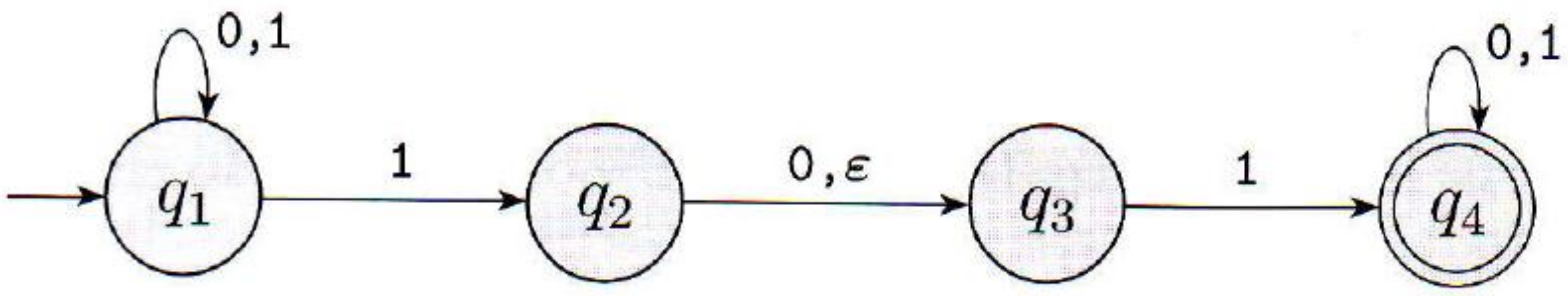
5. $F \subseteq Q$ é o ***conjunto de estados de aceitação***.[2]

Autômatos Finitos Não Determinísticos (AFN)



- Um estado pode ter 0 ou mais transições (setas saindo) para cada símbolo de Σ
- Um estado pode ter setas rotuladas por ε

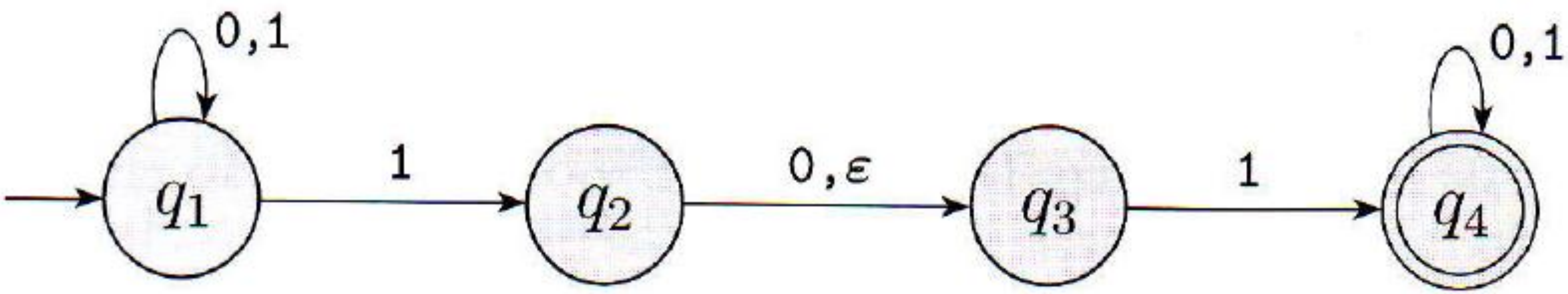
Autômatos Finitos
Não Determinísticos (AFN)



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. $\delta \colon Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ 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.

Autômatos Finitos Não Determinísticos (AFN)



1. $Q = \{q_1, q_2, q_3, q_4\}$,
2. $\Sigma = \{0,1\}$,
3. δ é dado como

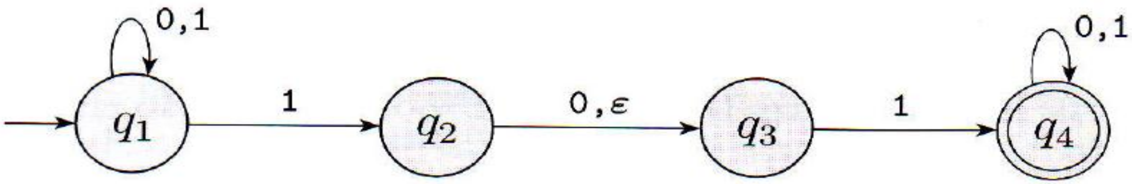
	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

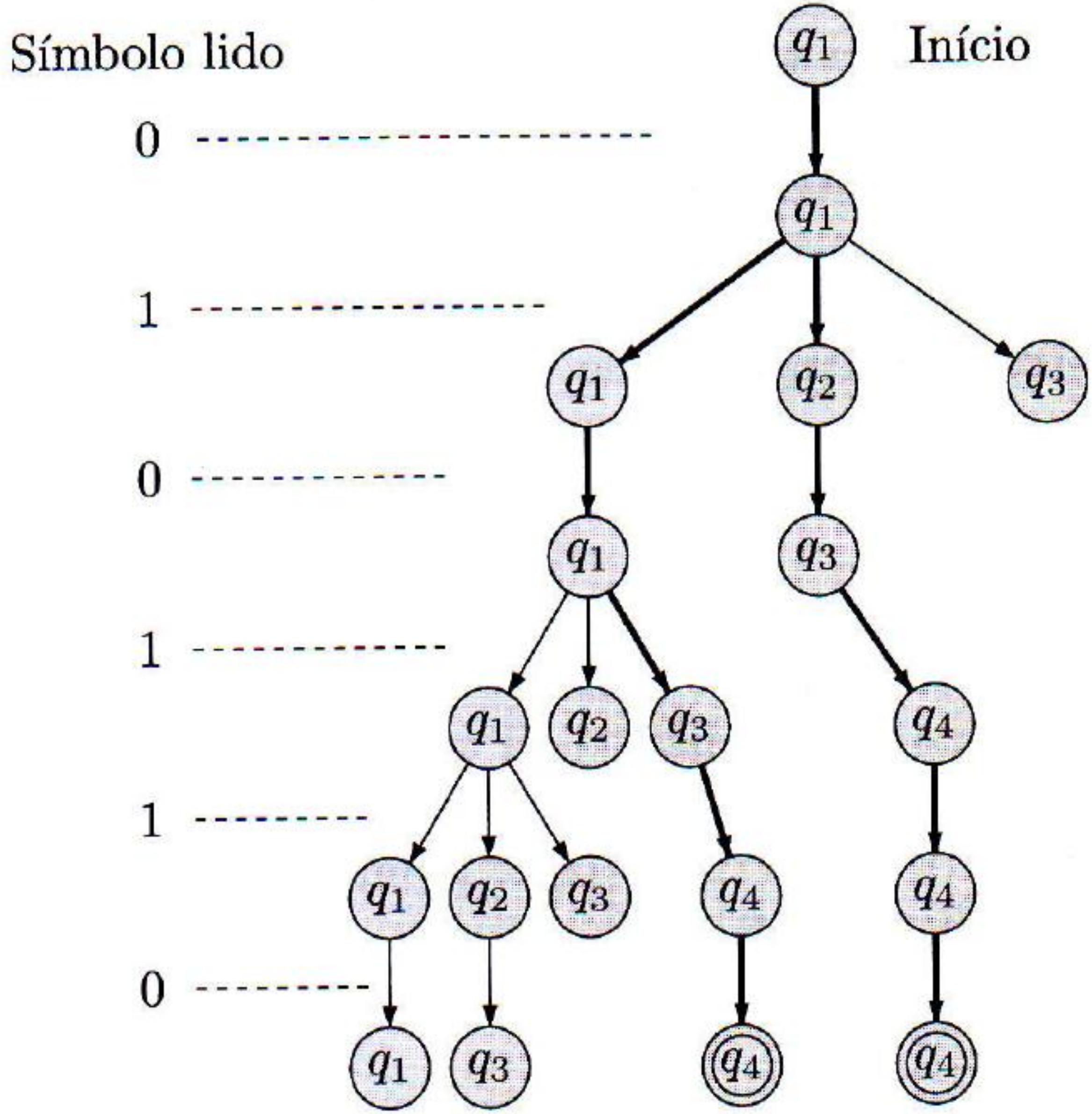
,

4. q_1 é o estado inicial, e
5. $F = \{q_4\}$.

Funcionamento de um AFN

- Sempre que o autômato se depara com um não-determinismo (símbolo repetido ou ε) faz uma cópia de si e cada cópia segue com uma alternativa, em paralelo.
- Se uma cópia aceitar a cadeia, então o AFN aceita a cadeia


0,1
1
0,ε
1
0,1
q_1
q_2
q_3
q_4


Símbolo lido
Início
0
1
0
1
1
0
q_1
q_2
q_3
q_4

AFDs e AFNs

- Quem reconhece mais linguagens?

AFDs e AFNs

- Quem reconhece mais linguagens?
- Os dois reconhecem a mesma classe de linguagens

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.

Equivalência entre AFDs e AFNs

- Prova: um estado para cada subconjunto

- Primeiro vamos desconsiderar setas ε

PROVA Seja $N = (Q, \Sigma, \delta, q_0, F)$ o AFN que reconhece alguma linguagem A. Construímos um AFD $M = (Q', \Sigma, \delta', q_0', F')$ que reconhece A. Antes de realizar a construção completa, vamos primeiro considerar o caso mais fácil no qual N não tem setas ε. Mais adiante levamos as setas ε em consideração.

1. $Q' = \mathcal{P}(Q)$.
Todo estado de M é um conjunto de estados de N. Lembre-se de que $\mathcal{P}(Q)$ é o conjunto de subconjuntos de Q.

2. Para $R \in Q'$ e $a \in \Sigma$ seja $\delta'(R, a) = \{q \in Q | q \in \delta(r, a) \text{ para algum } r \in R\}$.
Se R é um estado de M, é também um conjunto de estados de N. Quando M lê um símbolo a no estado R, ele mostra para onde a leva cada estado em R. Dado que cada estado pode ir para um conjunto de estados, tomamos a união de todos esses conjuntos. Outra maneira de escrever essa expressão é

$$\delta'(R, a) = \bigcup_{r \in R} \delta(r, a).$$ [4]

3. $q_0' = \{q_0\}$.
M começa no estado correspondente à coleção contendo somente o estado inicial de N.

4. $F' = \{R \in Q' \mid R$ contém um estado de aceitação de $N\}$.
A máquina M aceita se um dos possíveis estados nos quais N poderia estar nesse ponto é um estado de aceitação.

- Agora considerando setas ε:

$$E(R) = \{q|\ q \text{ pode ser atingido a partir de } R \text{ viajando-se ao longo de 0 ou mais setas } \varepsilon\}.$$

$$\delta'(R, a) = \{q \in Q|\ q \in E(\delta(r, a)) \text{ para algum } r \in R\}.$$

$$q_0{}' = E(\{q_0\})$$

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 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

AFNs mais fáceis de serem projetados

AFNs facilitando provas de teoremas

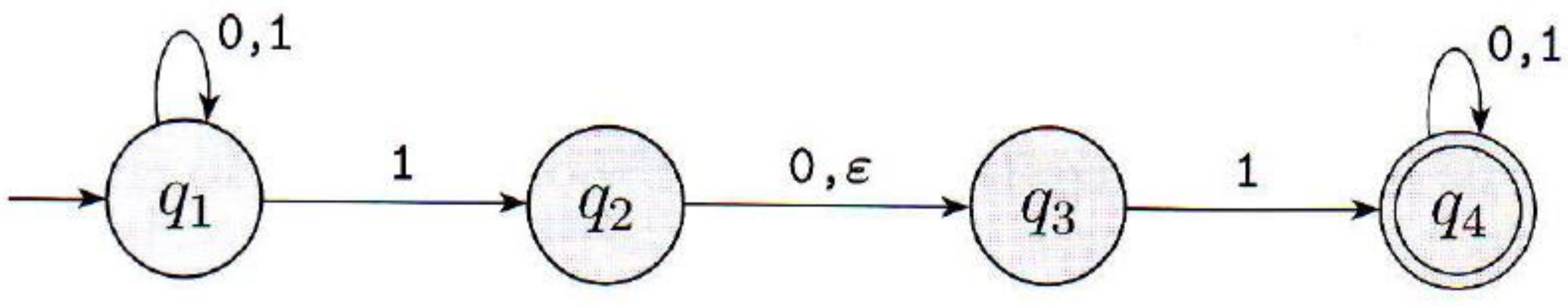
AFNs probabilísticos

- Um autômato probabilístico possui uma distribuição de probabilidades sobre as transições de cada estado
- $\delta: Q \times \Sigma \rightarrow P(Q)$
- $P: Q \times \Sigma \rightarrow [0,1]$

onde $\sum_j (q_i, a_j) = 1$ para q_i em Q

AFNs probabilísticos
Exemplo

- Transformar esse autômato em probabilístico

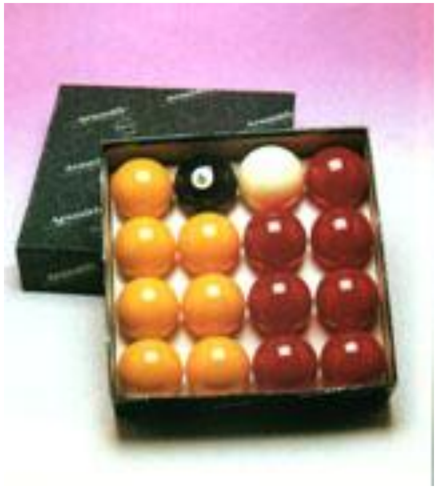


Modelo Oculto de Markov
Hidden Markov Model (HMM)

- Imagine que eu tenho uma urna de bolas coloridas (ou seja, tenho uma distribuição de probabilidades sobre essas cores)

- Alguém tem que adivinhar qual a próxima cor. Como isso poderia ser feito?

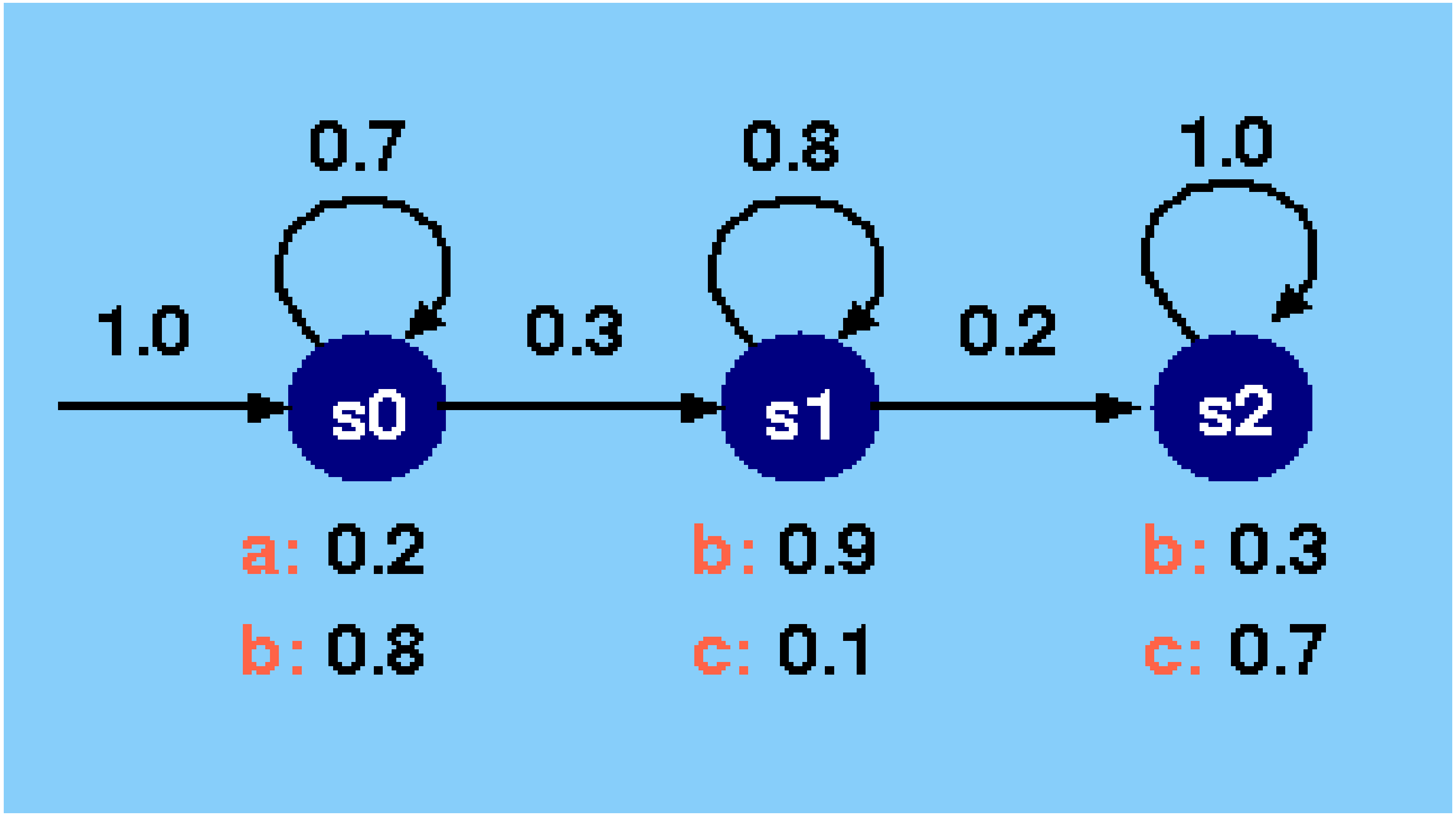
Modelo Oculto de Markov
Hidden Markov Model (HMM)



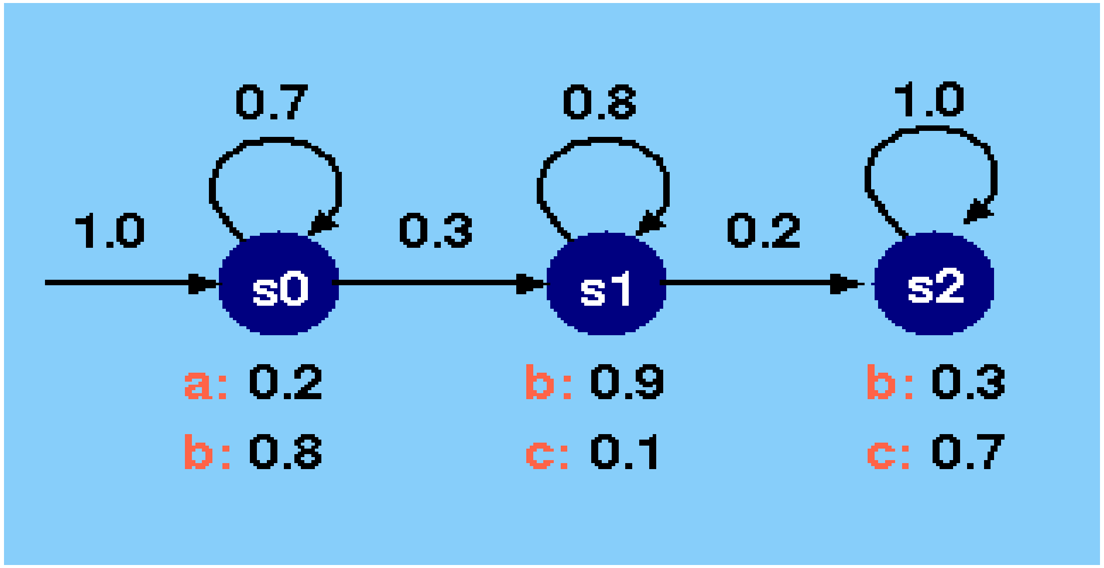
Modelo Oculto de Markov
Hidden Markov Model (HMM)

- Símbolos de emissão
- Estados ocultos
- Uma distribuição de probabilidades de emissão de símbolos associada a cada estado
- Probabilidade de transição entre estados
- Distribuição de probabilidades do estado inicial

Modelo Oculto de Markov
Hidden Markov Model (HMM)

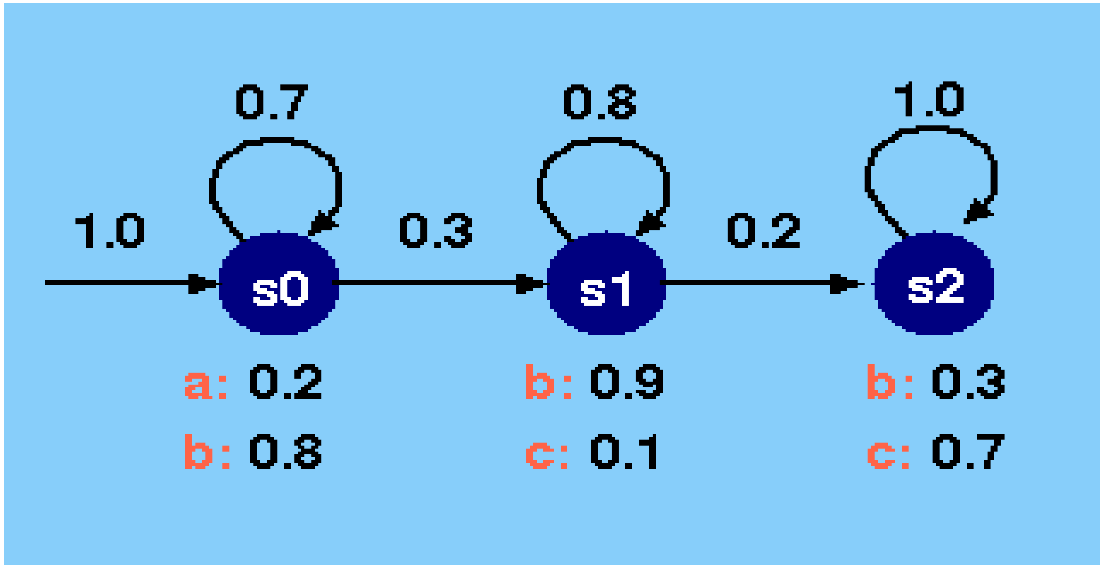


Modelo Oculto de Markov
Hidden Markov Model (HMM)



- Semelhança com algo?

Modelo Oculto de Markov
Hidden Markov Model (HMM)



- Semelhança com algo?
- Como transformo essa HMM em um AF probabilístico?

Modelo Oculto de Markov
Hidden Markov Model (HMM)

- Autômatos finitos (não determinísticos) probabilísticos são equivalentes a modelos ocultos de Markov