ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 1

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

Introdução à Teoria da Computação

- Por que estudar teoria?

Introdução à Teoria da Computação

- Complexidade
- Computabilidade
- Teoria dos autômatos

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

- Ordem dos temas:
  - Teoria dos autômatos (teoria e prática)
  - Computabilidade
  - Complexidade
- Avaliação:
  - 2 provas teóricas
  - 1 exercício-programa (individual)
  - Listas de exercícios (individual)
  - Seminários (grupos de 6 a 7 alunos – LISTA DE NOMES NA AULA DESTA QUARTA-FEIRA!)

Avaliação

- MP = (P1 + 2*P2)/3
- MT = média de listas de exercícios e seminários (seminário tem peso 2)
- Média 1a. aval. M1 = (7*MP + 3*EP+1*MT)/11
- Uma prova substitutiva (fechada) – substitui a prova em que faltou
- Média 2a aval. = (M1+REC)/2

Material

- 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

Próximas aulas

- Hoje e 01/08 – cap 1 – linguagens regulares
- 01/08 – sorteio dos primeiros seminários
- 06/08 – não haverá aula, MAS vocês devem:
  - estudar o capítulo 0 do SIPSER
  - fazer a lista referente a esse capítulo (exercícios e problemas de 1 a 12) - entregar no dia 08/08
  - Preparar o seminário do dia 08/08
- 08/08:
  - entrega da lista do cap. 0
  - Seminários sobre aplicações de autômatos

Cap 1 – Linguagens regulares

- 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

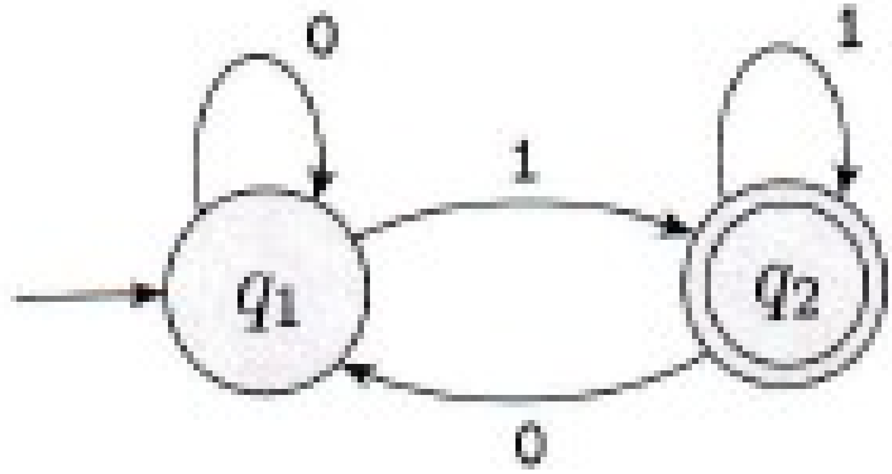
- O que esse controlador precisa guardar em memória?
  - Estado atual (aberto/fechado: 1 bit)
- Vários outros dispositivos (ex: eletrodomésticos) podem ser implementados de forma semelhante, com uma memória limitada

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, ...

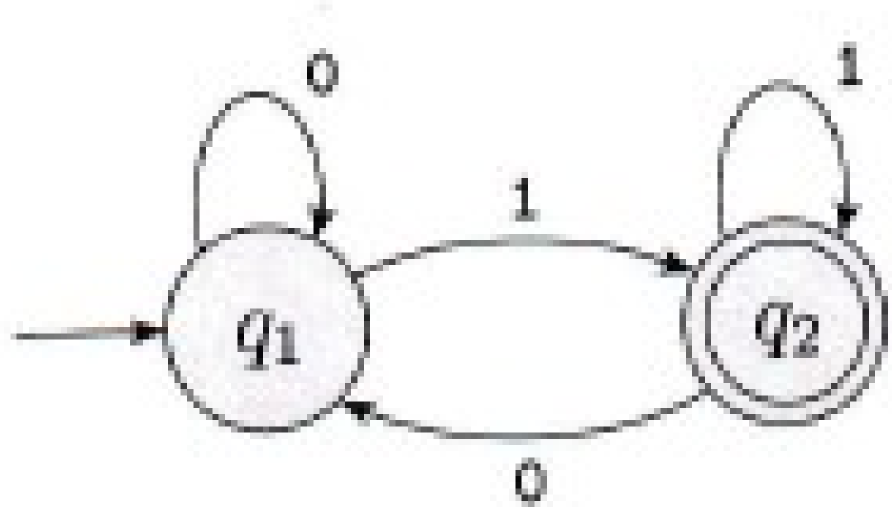
Autômatos finitos

- 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

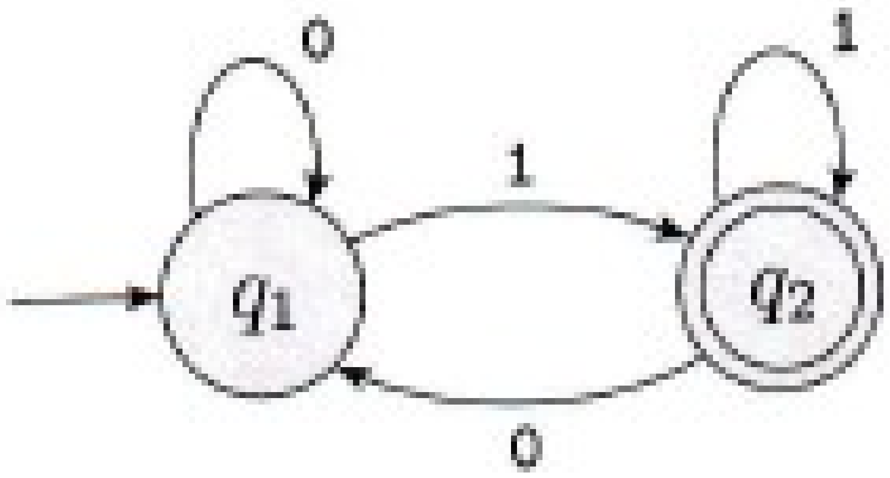
- Definição formal:

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?

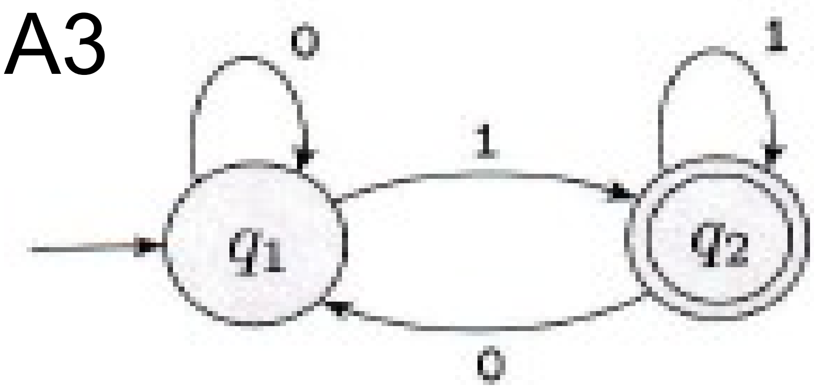


Um ***autômato finito*** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

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

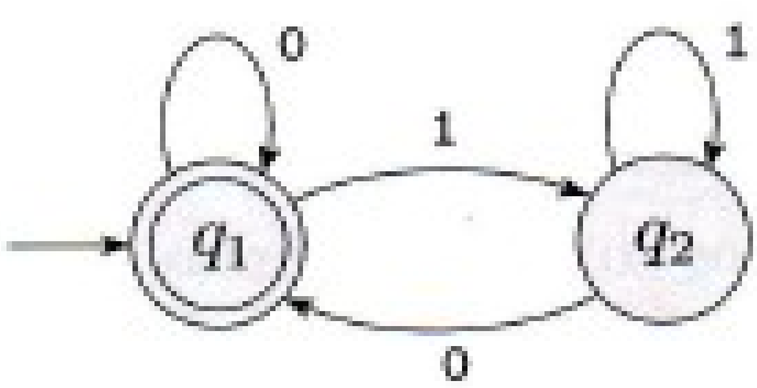
A3



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

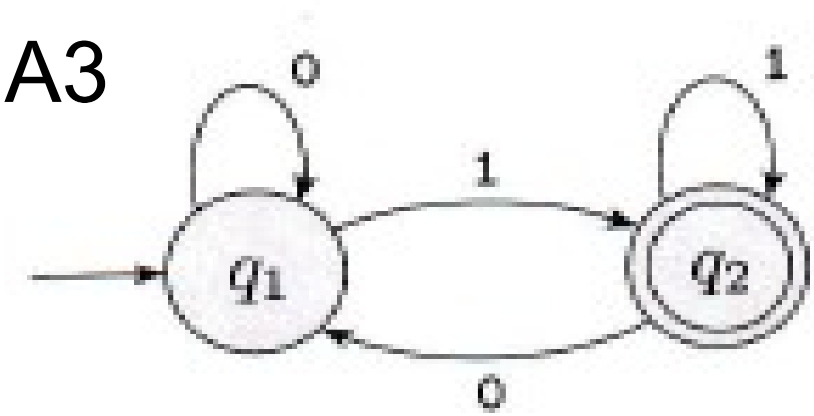
- Que linguagem esse autômato reconhece? (apenas mudou o estado final)

A4



Autômatos finitos

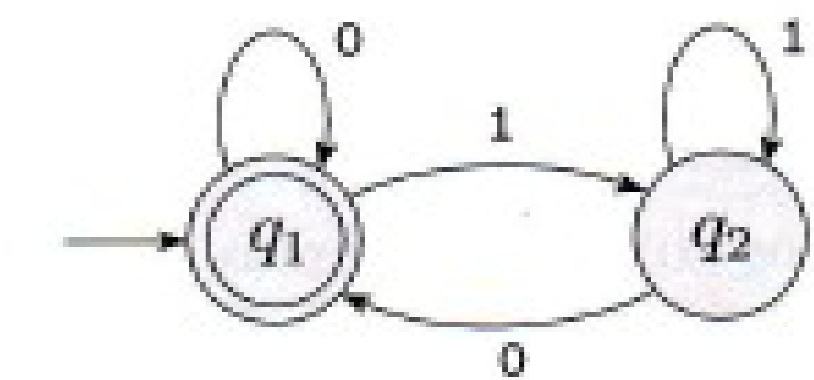
A3



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

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

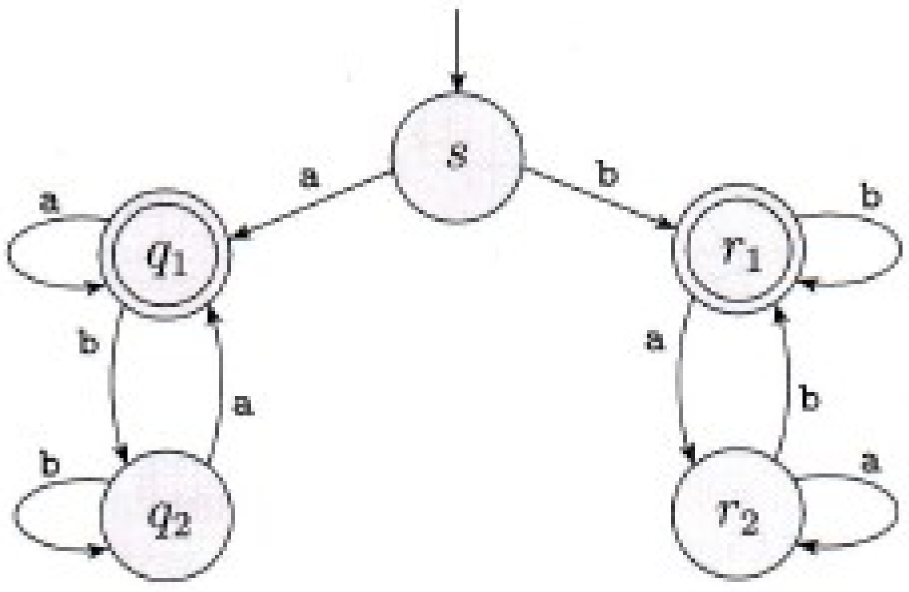
A4



L(A4) = {w | w é a cadeia vazia ou é binária e termina com 0}

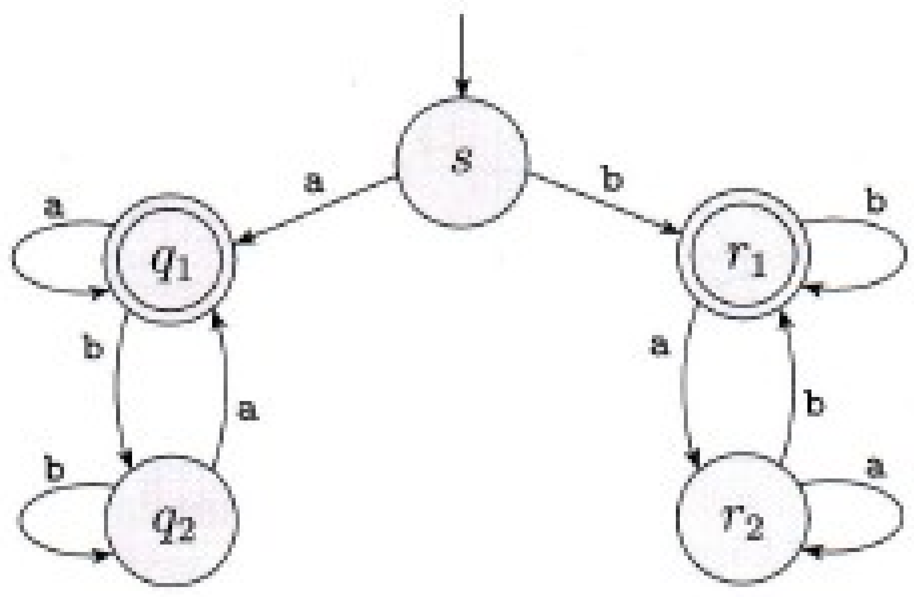
Autômatos finitos

- Que linguagem esse autômato reconhece?



Autômatos finitos

- Que linguagem esse autômato reconhece?



- Cadeias que comecem e terminem com o mesmo símbolo

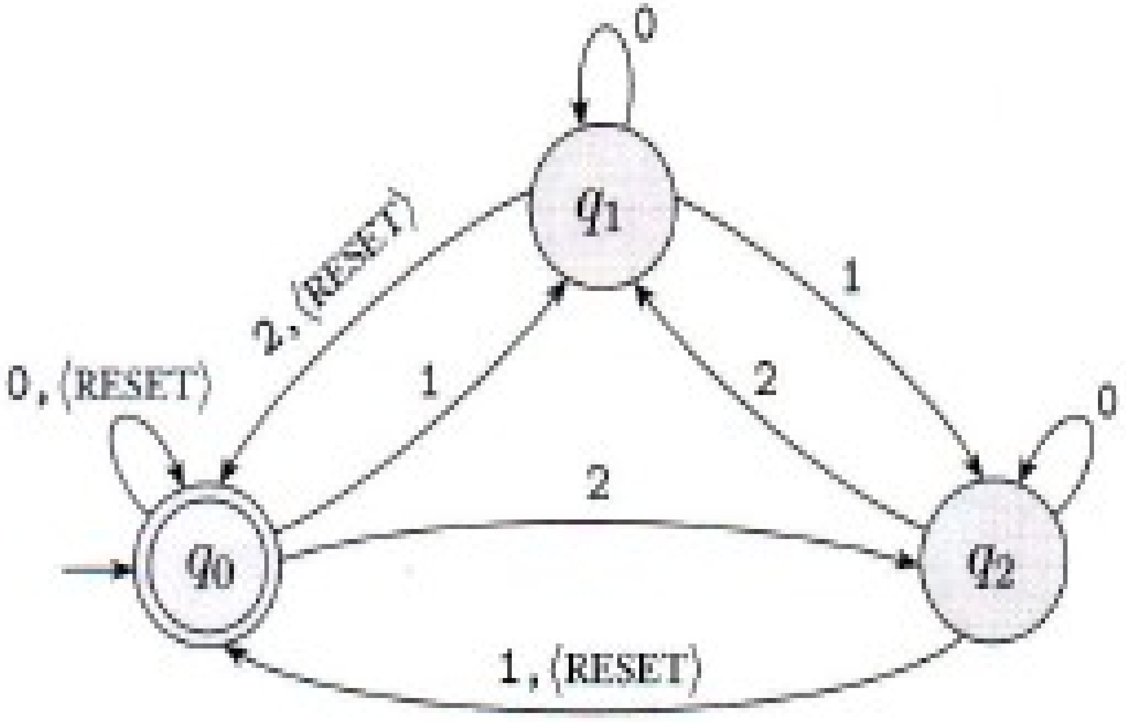
Projetando autômatos

- 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 $\Sigma = \{0,1,2,\text{<RESET>}\}$, 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