ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 1

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

Introdução à Teoria da Computação

Introdução à Teoria da Computação

- Complexidade
- Computabilidade
- Teoria dos autômatos

Por que estudar teoria?

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:
  - Saber o que dá para computar, e o que não der saber como 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:
  - 3 provas teóricas

Avaliação

- M1 = média aritmética simples das 3 provas
- Uma prova substitutiva (FECHADA) – substitui a prova em que faltou
- Média 2a aval. = (M1+REC)/2
- Provas Sub e Rec:
  - Cai a matéria toda
  - Mais difíceis que as 3 provas normais

Material

- Slides de aula no Moodle (https://edisciplinas.usp.br)

- 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, listas de exercícios

- Hoje – Iniciaremos no cap 1 MAS vocês devem:
  - estudar o capítulo 0 do SIPSER

- FAZER todas as listas de exercícios (do livro do Sipser, capítulos estudados): Não haverá entrega.

- Não haverá aula na semana de SI, mas a lista de presença circulará no evento

Cap 1 – Linguagens regulares

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

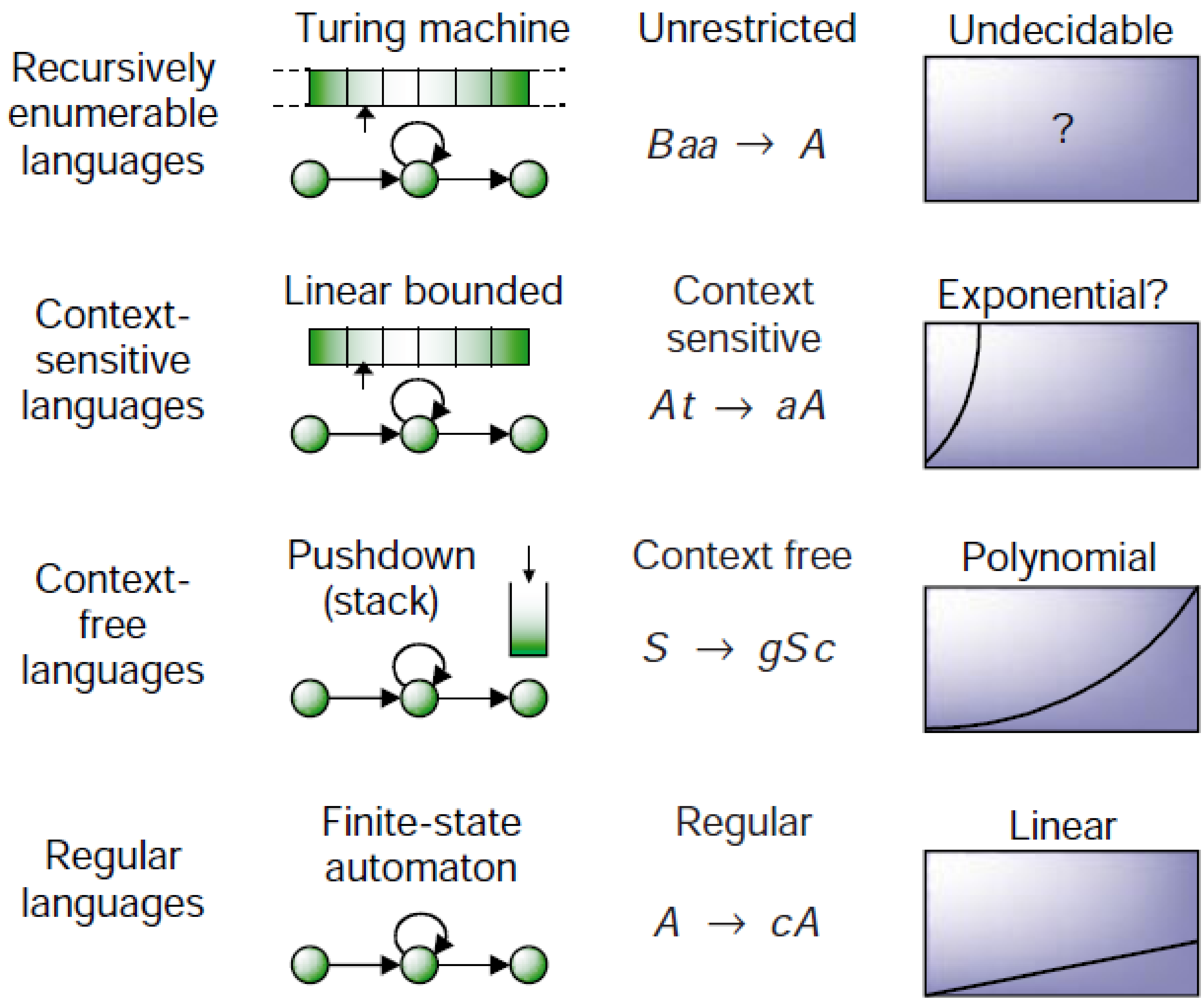
Cap 1 – Linguagens regulares

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

Autômatos finitos

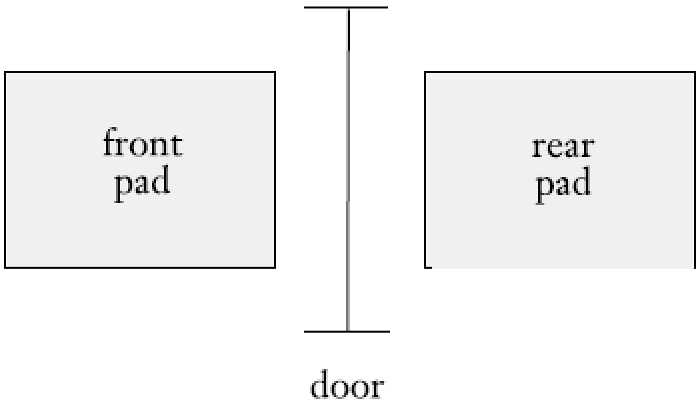
- Necessidade de um modelo para entender (estudar) um computador
- Vários modelos computacionais com diferentes características (e complexidades)
- O modelo mais simples:
  - Máquina de estados finitos ou
  - Autômato de estados finitos ou
  - Autômato finito
  - *Finite State Automaton (FSA)*

Linguagens, dispositivos, gramáticas e complexidades



Autômatos finitos

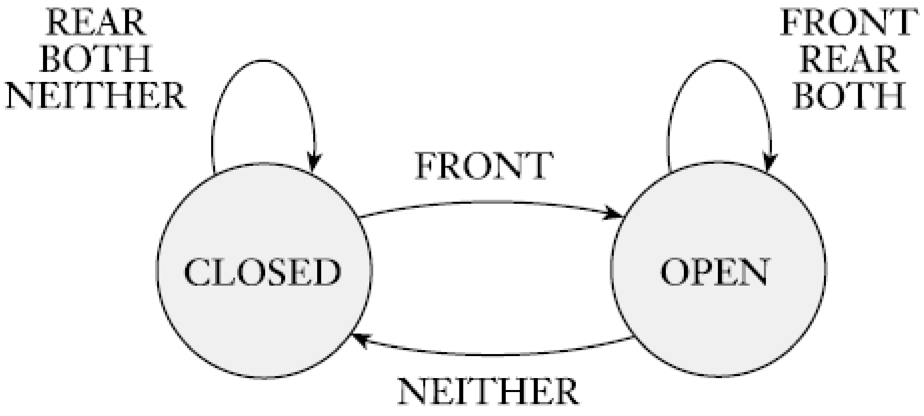
- O exemplo de um controlador de portas



- Entradas possíveis (presença de pessoas):
  - FRONT (na frente)
  - REAR (atrás)
  - BOTH (pessoas na frente e atrás)
  - NEITHER (ninguém na frente nem atrás)

Autômatos finitos

- O exemplo de um controlador de portas



Autômatos finitos

- O que esse controlador precisa guardar em memória?

Autômatos finitos

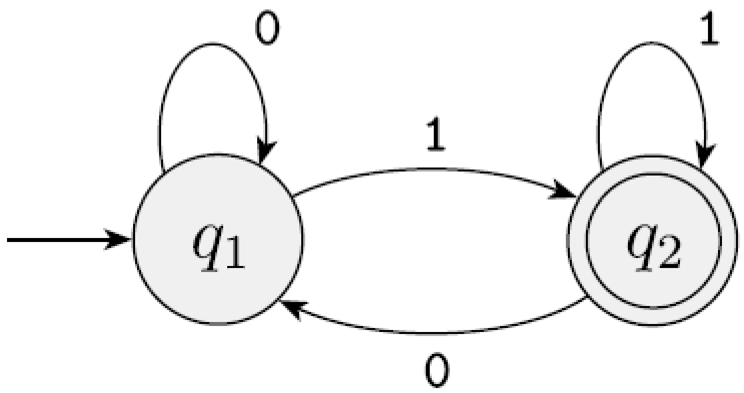
- O que esse controlador precisa guardar em memória?
  - Estado atual (aberto/fechado: 1 bit)
- Vários outros dispositivos (ex: eletrodomésticos) podem ser implementados de forma semelhante, com uma memória limitada

Autômatos finitos

- Autômatos finitos são mecanismos RECONHECEDORES
- Ex: reconhecer strings binárias (compostas por 0's e 1's) que comecem e terminem com zero, com tamanho pelo menos 1
  - 0, 00, 010, 000000, 0101110, ...

Autômatos finitos

- Diagrama de estados



Círculos: estados

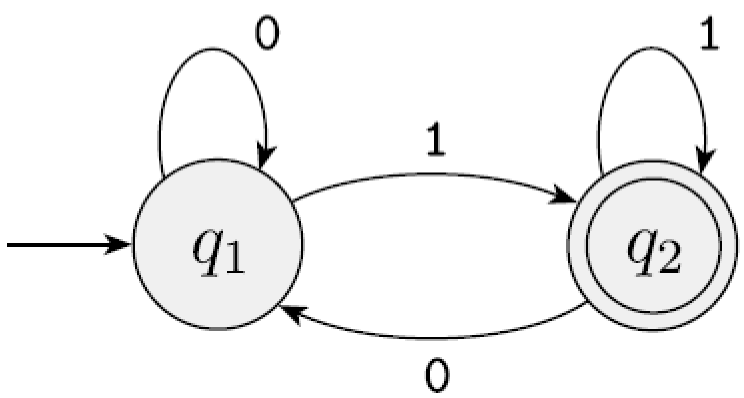
Círculos duplos: estados finais ou de aceitação

Seta sem início (somente uma): indica quem é o estado inicial

Setas entre estados: define a transição entre dois estados após a leitura do símbolo que está sobre sobre a seta

Autômatos finitos

- Diagrama de estados



Processo de reconhecimento:

Dados um autômato M e uma cadeia w:

- comece pelo estado inicial
- faça a transição de estados a cada símbolo lido (da cadeia de entrada w)
- ao finalizar a leitura, se M parou em um estado de aceitação aceite, e rejeite caso contrário

Círculos: estados

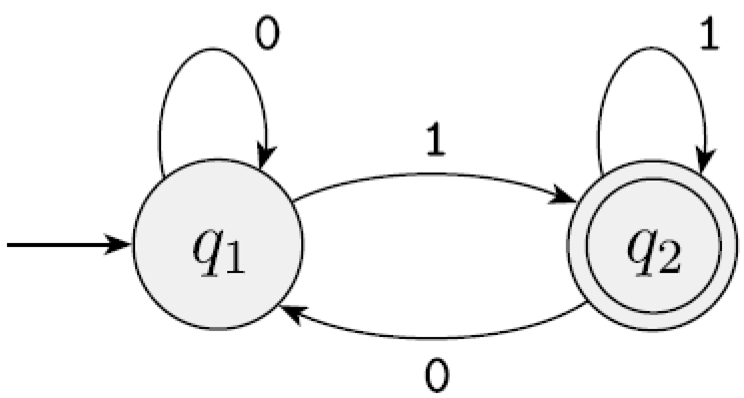
Círculos duplos: estados finais ou de aceitação

Seta sem início (somente uma): indica quem é o estado inicial

Setas entre estados: define a transição entre dois estados após a leitura do símbolo que está sobre sobre a seta

Autômatos finitos

- Diagrama de estados



Processo de reconhecimento:

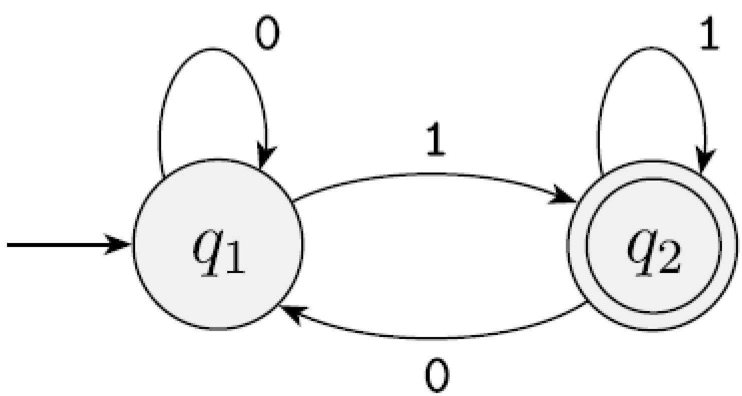
Dados um autômato M e uma cadeia w:

- comece pelo estado inicial
- faça a transição de estados a cada símbolo lido (da cadeia de entrada w)
- ao finalizar a leitura, se M parou em um estado de aceitação aceite, e rejeite caso contrário

Que tipos de cadeias esse autômato M1 aceita?

Autômatos finitos

- Diagrama de estados



Processo de reconhecimento:

Dados um autômato M e uma cadeia w:

- comece pelo estado inicial
- faça a transição de estados a cada símbolo lido (da cadeia de entrada w)
- ao finalizar a leitura, se M parou em um estado de aceitação aceite, e rejeite caso contrário

Que tipos de cadeias esse autômato M1 aceita?

Sequência binárias que terminam em 1

Autômatos finitos

- Um **alfabeto** Σ é um conjunto de símbolos
- Uma **cadeia** w é uma sequência de símbolos de um alfabeto Σ (w ϵ Σ*)
- Uma **linguagem** L é um conjunto de cadeias sobre um alfabeto Σ (L é subconjunto de Σ*)
- A **linguagem** reconhecida por um autômato é o conjunto das **cadeias** (de símbolos de entrada) aceitas pelo autômato
- L(M1) = {w | w é uma string binária e termina em 1}

Autômatos finitos

- Definição formal:

Autômatos finitos

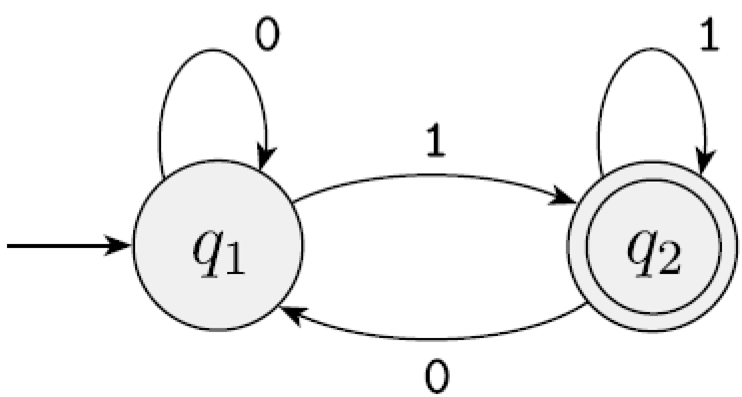
- Definição formal:

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 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]

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. $r_n \in F$.

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:

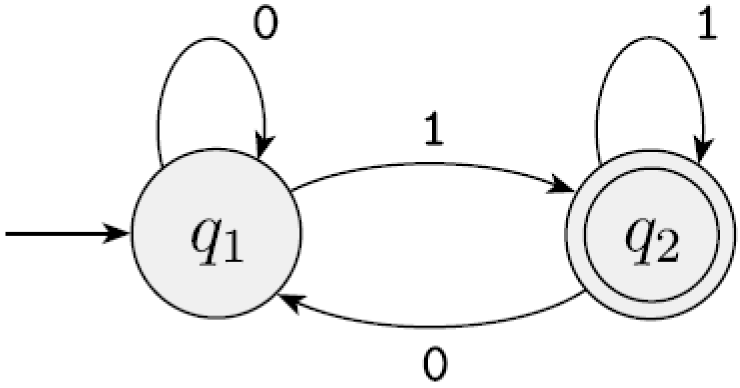
1. $r_0 = q_0$,
2. $\delta(r_i, w_{i+1}) = r_{i+1}$, para $i = 0, \ldots, n-1$, e
3. $r_n \in F$.

Um autômato:

- **aceita** ou não aceita uma **cadeia**
- **reconhece** ou não reconhece uma **linguagem**

Autômatos finitos

- Qual a definição formal do autômato M1?

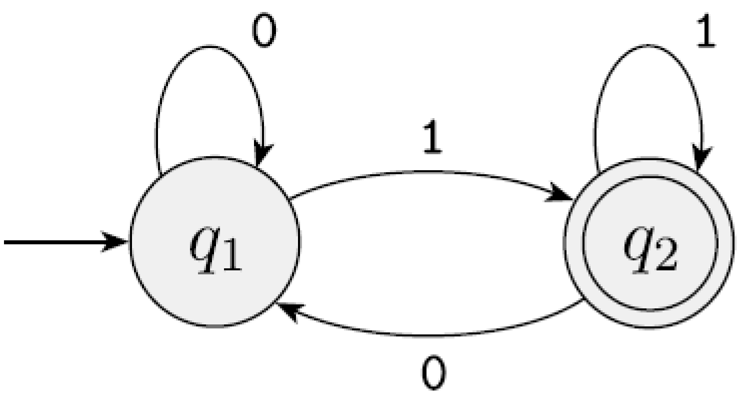


Um ***autômato finito*** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

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

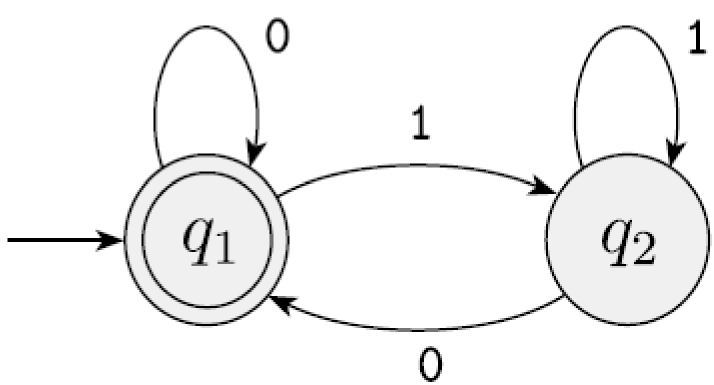
M1



L(M1) = {w | w é binária e termina com 1}

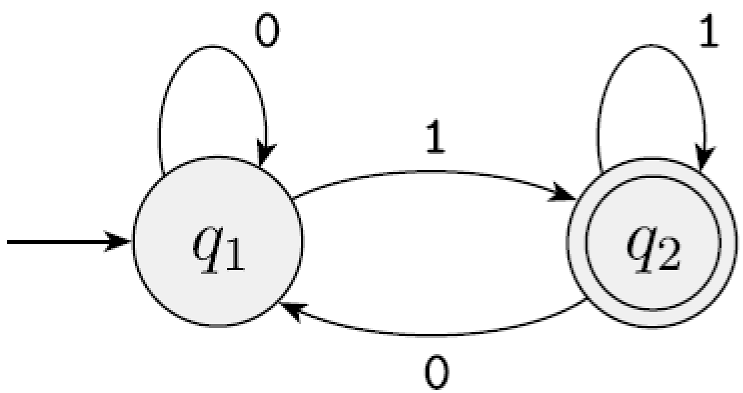
- Que linguagem o autômato M2 reconhece? (apenas mudou o estado final)

M2



Autômatos finitos

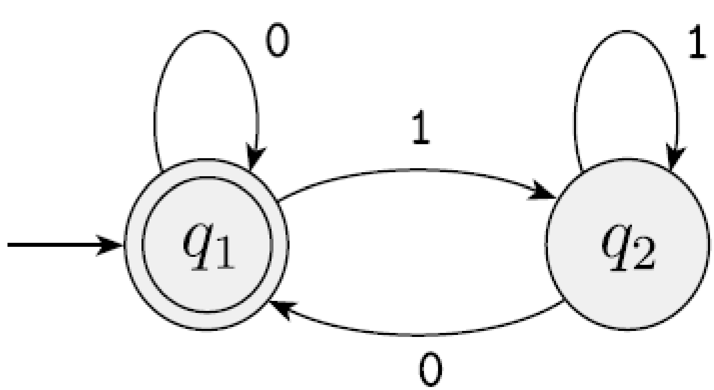
M1



L(M1) = {w | w é binária e termina com 1}

- Que linguagem o autômato M2 reconhece? (apenas mudou o estado final)

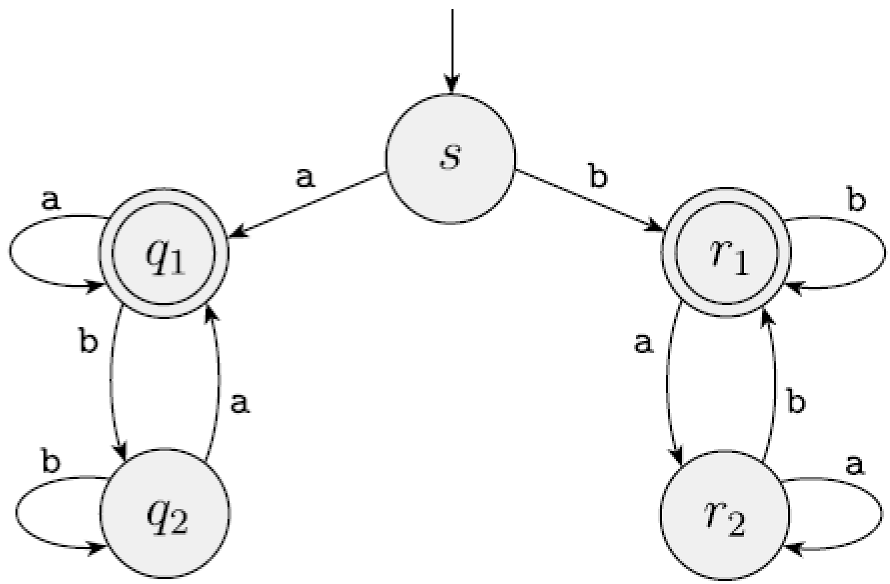
M2



L(M2) = {w | w é a cadeia vazia ou é binária e termina com 0}

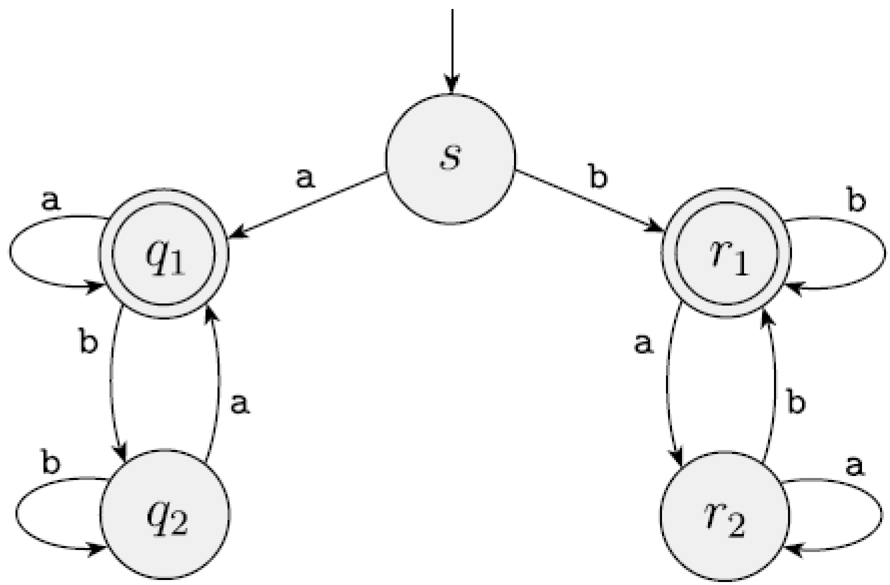
Autômatos finitos

- Que linguagem esse autômato reconhece?



Autômatos finitos

- Que linguagem esse autômato reconhece?



- Cadeias sobre o alfabeto {a,b} que comecem e terminem com o mesmo símbolo

Linguagem Regular

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

Projetando autômatos

- Pense que você é um autômato
- A cadeia de entrada pode ser arbitrariamente grande
- O número de estados é finito
- A transição se dá dados apenas 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 que reconheça cadeias binárias (compostas por 0's e 1's) que comecem e terminem com zero, com tamanho pelo menos 1

0, 00, 010, 000000, 0101110, ...

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. Cadeia vazia também é aceita.