Máquinas

Prof^a Jerusa Marchi

Departamento de Informática e Estatística

Universidade Federal de Santa Catarina

e-mail: jerusa@inf.ufsc.br

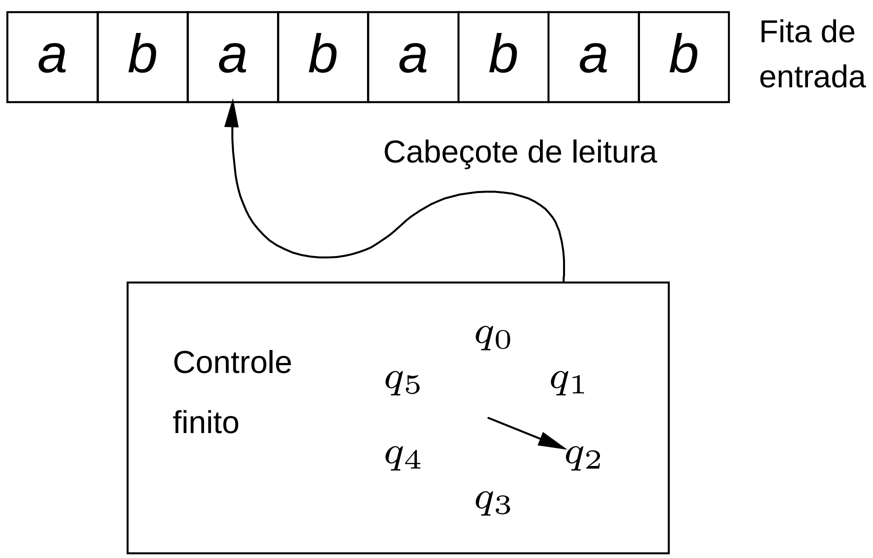
Tipos de Máquinas...

... e sua relação com geradores e reconhecedores

Máquina	Linguagem	Gramática	Hierarquia
MT	Recursivamente Enumerável	Irrestrita	Tipo 0
MT	Recursiva	Irrestrita	
MT	Sensível ao Contexto	Sensível ao Contexto	Tipo 1
Pilha	Livre de Contexto	Livre de Contexto	Tipo 2
Finito	Regular	Regular	Tipo 3

Autômatos Finitos

- Dispositivo simples
  - fita de entrada
  - cabeçote de leitura
  - unidade central de processamento (estados)
  - memória limitada - conceito de estado



Autômatos Finitos

- Aplicações:
  - Análise léxica (compiladores)
  - Busca em texto
  - Implementação de sistemas de controle simples baseados em estados (máquinas de refrigerante, jornais, salgadinhos, chocolates e elevadores).

Autômatos Finitos

- Há dois tipos de máquinas de estados finitos:
  - transdutores de linguagens - com entrada e saída
  - reconhecedores de linguagens - com duas saídas possíveis
    - aceitação
    - rejeição

Autômatos Finitos

- Podem ser:
  - Determinímisticos
  - Não Determinísticos

Autômatos Finitos Determinísticos

- Um autômato finito (AFD) é um quíntupla:

$$M = (K, \Sigma, \delta, s, F)$$

  Onde:
  - K = conjunto finito de estados
  - Σ = conjunto finito de símbolos de entrada
  - $\delta : K \times \Sigma \rightarrow K$ = função de transição
  - s = estado inicial ($s \in K$)
  - F = conjunto de estados finais ($F \subseteq K$)

Autômatos Finitos Determinísticos

- O autômato é dito determinístico pois pela definição da função de transição δ, cada par (*estado*, *símbolo*) mapeia para **exatamente** *um* estado.
  - $q, p \in K$ e $a \in \Sigma$

$$\delta(q, a) = p$$

Autômatos Finitos Não Determinísticos

- Um autômato finito não determinístico (AFN) é um quíntupla:

$$M = (K, \Sigma, \delta, s, F)$$

Onde:

- K = conjunto finito de estados
- Σ = conjunto finito de símbolos de entrada
- $\delta : K \times (\Sigma \cup \{\varepsilon\}) \times K$ = função de transição
- s = estado inicial ($s \in K$)
- F = conjunto de estados finais ($F \subseteq K$)

Autômatos Finitos Não Determinísticos

- O autômato é dito não determinístico se há pelo menos uma transição δ, para um par (*estado*, *símbolo*) que mapeia para um **subconjunto de estados.**
  - $q, p, r \in K$ e $a \in \Sigma$

$$\delta(q, a) = \{p, r\}$$

Autômatos Finitos

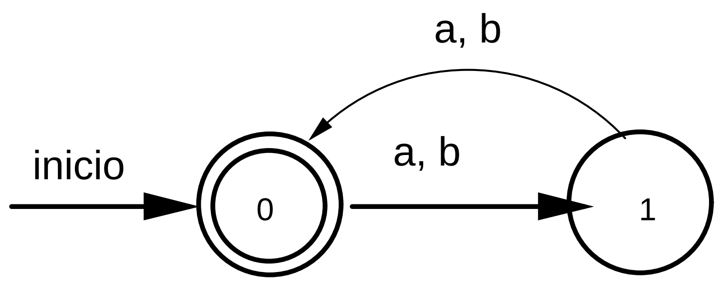
- Há duas formas de representar um AF:
  - Diagrama de transição
  - Tabela de transição

Autômatos Finitos

- Diagrama de Transição:
  - é um grafo direcionado e rotulado
  - os vértices representam os estados (círculos)
  - o estado inicial é diferenciado por uma seta
  - os estados finais são representados por círculos duplos
  - as arestas representam as transições $\delta(p, a) \rightarrow q$

Autômatos Finitos

- Exemplo: $L_1 = \{w \mid \in \Sigma^\star = \{a, b\}$ e $\mid w \mid$ é par $\}$
  Diagrama de transição



Autômatos Finitos

- Tabela de Transição
  - forma tabular de representar um AF onde a primeira coluna lista os estados e a primeira linha, os símbolos do alfabeto. O conteúdo da posição (q, a) será p se existir uma transição $\delta(q, a) \rightarrow p$.

Autômatos Finitos

- Exemplo: $L_1 = \{w \mid \in \Sigma^\star = \{a, b\}$ e $\mid w \mid$ é par $\}$

δ	a	b
$\rightarrow *q_0$	q_1	q_1
q_1	q_0	q_0

Autômatos Finitos

- Configuração:
  - uma configuração é determinada pelo estado corrente e pela parte ainda não processada da palavra

    $$[q_0, abab]$$

    que representa a configuração inicial para a palavra $w = abab$

Autômatos Finitos

- Computação:
  - é uma sequência de configurações
  - usa-se a relação $\vdash$ (resulta em) para indicar que a máquina passa de uma configuração à outra. Diz-se que:

$$[q_1, w] \vdash [q_2, y]$$

    se e somente se existe uma transição de q_1 para q_2 sob a, onde $a \in \Sigma$ e $w = ay$
    - Exemplo:

$$[q_0, abab] \vdash [q_1, bab] \vdash [q_0, ab] \vdash [q_1, b] \vdash [q_0, \varepsilon]$$

Autômatos Finitos

- Uma sentença w é aceita (reconhecida) por um autômato finito $M = (K, \Sigma, \delta, q_0, F)$ sse $\widehat{\delta}(q_0, w) \rightarrow q$ e $q \in F$, ou seja, há uma computação

$$[q_0, w] \vdash_M^* [q, \varepsilon]$$

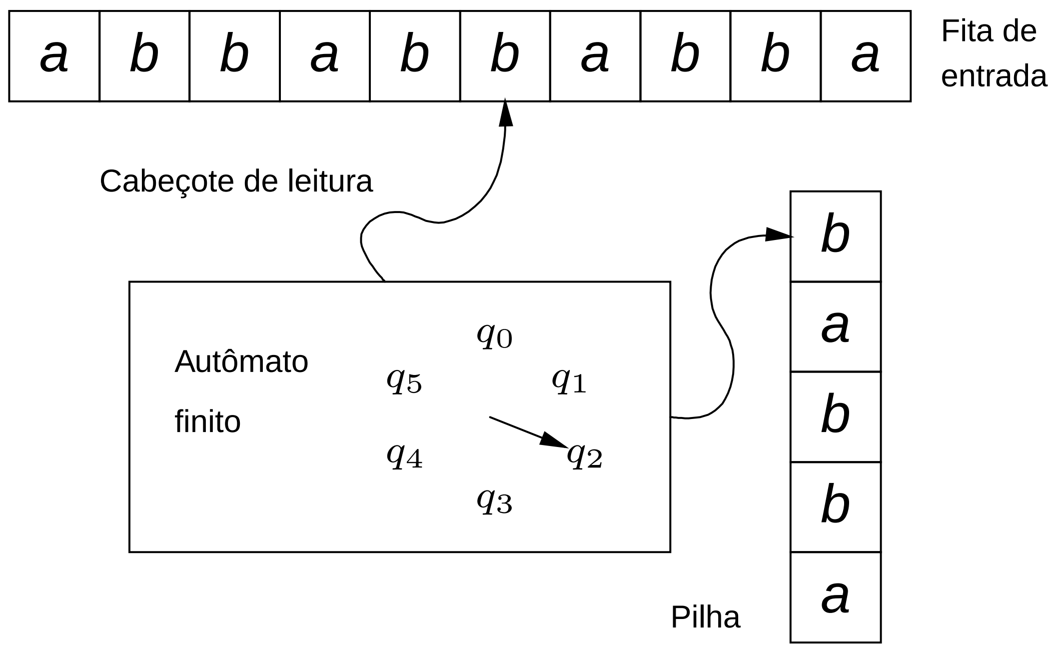
- A linguagem aceita por um autômato M é aquela cujo conjunto de sentenças é aceito por M

$$L(M) = \{w \mid \widehat{\delta}(q_0, w) \rightarrow q \text{ e } q \in F\}$$

- Dois autômatos finitos M_1 e M_2 são ditos equivalentes sse $L(M_1) = L(M_2)$
- Uma linguagem é regular sse ela for aceita por um autômato finito

Autômatos de Pilha

- Autômato finito acrescido de memória auxiliar (*pilha*)
- A manipulação é permitida somente no topo da pilha (LIFO)



Autômatos de Pilha

- Aplicações:
  - Análise Sintática (compiladores)
  - Verificação de parênteses em editores de texto e/ou ambientes de programação (emacs/xemacs)

Autômatos de Pilha

- Um autômato de pilha (AP) é um sêxtupla:

$$M = (K, \Sigma, \Gamma, \Delta, s, F)$$

Onde:

- K = conjunto finito de estados
- Σ = conjunto finito de símbolos de entrada
- Γ = conjunto finito de símbolos de pilha
- $\delta : (K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*) \times (K \times \Gamma^*)$ = relação de transição
- s = estado inicial ($s \in K$)
- F = conjunto de estados finais ($F \subseteq K$)

Autômatos de Pilha

- Se $((p, a, \beta), (q, \gamma)) \in \Delta$ então sempre que o autômato estiver no estado p com β no topo da pilha, poderá ler o símbolo a na fita de entrada (se $a = \varepsilon$ então a entrada não é consultada), substituindo β por γ no topo da pilha e passando para o estado q
- Empilhar - $((p, u, \varepsilon), (q, a))$
- Desempilhar - $((p, u, a), (q, \varepsilon))$

Autômatos de Pilha

- Configuração
  - uma configuração é definida como um membro de $K \times \Sigma^\star \times \Gamma^\star$

$$[q, w, abc]$$

    - $q \in K$ é o estado atual da máquina
    - w é a parte da sentença de entrada ainda não processada
    - abc é o conteúdo armazenado da pilha, lido a partir do topo

Autômatos de Pilha

- Computação:
  - Como nos autômatos finitos, é uma sequência de configurações, separadas pelo símbolo $\vdash$ (resulta em) que indica que a máquina passa de uma configuração à outra

$$[p, x, \alpha] \vdash [q, y, \iota]$$

se e somente se existe uma transição de $((p, a, \beta), (q, \gamma)) \in \delta$, tal que $x = ay$, $\alpha = \beta\eta$ e $\iota = \gamma\eta$ para algum $\eta \in \Gamma^\star$

Autômatos de Pilha

- Uma sentença w é aceita (reconhecida) por um autômato de pilha $M = (K, \Sigma, \Gamma, \delta, q_0, F)$ sse

$$[q_0, w, \varepsilon] \vdash_M^* [p, \varepsilon, \varepsilon]$$

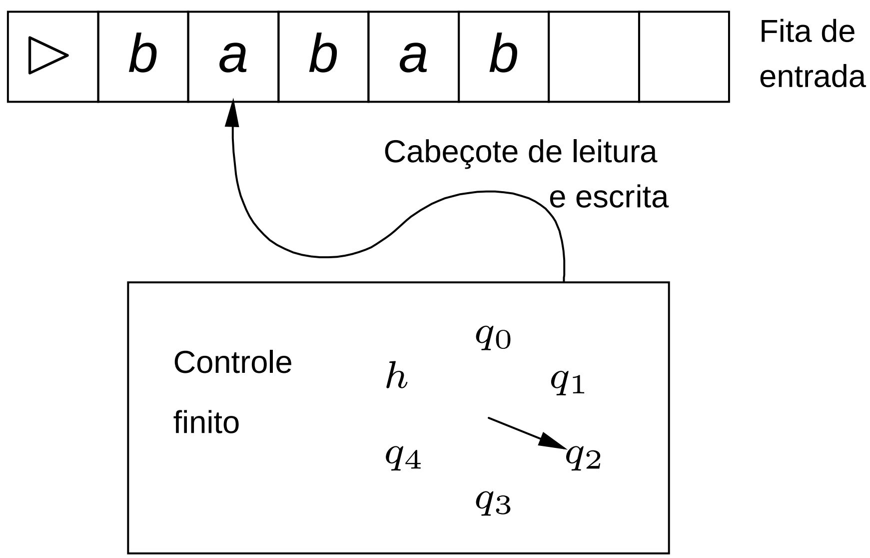
  para algum $p \in F$
- A linguagem aceita por um autômato M é aquela cujo conjunto de sentenças é aceito por M
- A classe de linguagens aceitas por autômatos de pilha é exatamente a classe de linguagens livres de contexto

Autômatos de Pilha

- Exemplo
  - $L = \{w | w \in \Sigma = \{a, b, c\}^\star$ e $w = xcx^R$ tal que $x \in \{a, b\}^\star\}$
  - $M = (K, \Sigma, \Gamma, \delta, s, F)$ onde:
    - $K = \{s, f\}$
    - $\Sigma = \{a, b, c\}$
    - $\Gamma = \{a, b\}$
    - $F = \{f\}$
    - $\delta = \{$
      1. $((s, a, \varepsilon), (s, a))$
      2. $((s, b, \varepsilon), (s, b))$
      3. $((s, c, \varepsilon), (f, \varepsilon))$
      4. $((f, a, a), (f, \varepsilon))$
      5. $((s, b, b), (f, \varepsilon))\}$

Máquinas de Turing

- unidade de Controle Finito
- fita
- cabeçote de leitura/escrita



Máquinas de Turing

- Ao contrário dos demais tipos de autômatos, MT não são um modelo a ser suplantado
- Ainda que este modelo seja fortalecido (multifitas, acesso aleatório), todos os melhoramentos mostram-se equivalentes, em poder computacional, às MT

Máquinas de Turing

- Operações da Unidade de Controle

1. Levar a unidade de controle para um novo estado
2. (a) Gravar um símbolo na célula apontada pelo cabeçote, substituindo algum símbolo lá encontrado ou não

   (b) Mover o cabeçote de leitura/escrita para apontar para uma célula à direita ou à esquerda na fita em relação a posição corrente

Máquinas de Turing

- Quanto a fita:
  - infinita à direita
  - delimitada à esquerda pelo símbolo $\triangleright$
  - a cadeia de entrada é gravada logo a direita do símbolo $\triangleright$ e o restante da fita é preenchido com espaços em branco (denotados por $\sqcup$)
- Quanto ao cabeçote de leitura/escrita
  - por convenção, sempre que o cabeçote for posicionado sobre o símbolo delimintador de fita, ele é movido automaticamente para a direita
  - são usados os símbolos $\rightarrow$ e $\leftarrow$ para denotar o movimento do cabeçote para a direita e para a esquerda, respectivamente
  - os símbolos $\rightarrow$ e $\leftarrow$ não fazem parte de nenhum alfabeto

Máquinas de Turing

- Uma Máquina de Turing (MT) é uma quíntupla:

$$M = (K, \Sigma, \delta, s, H)$$

Onde:

- K = conjunto finito de estados
- Σ = alfabeto de entrada $\cup \{\sqcup, \triangleright\}$
- s = estado inicial ($s \in K$)
- H = conjunto de estados de parada ($H \subseteq K$)
- $\delta : (K - H) \times \Sigma$ para $(K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$, tal que
  1. para todos os $q \in K - H$, se $\delta(q, \triangleright) = (p, b)$ então $b = \rightarrow$
  2. para todos os $q \in K - H$ e $a \in \Sigma$, se $\delta(q, a) = (p, b)$ então $b \neq \triangleright$

Máquinas de Turing

- Se $q \in K - H$, $a \in \Sigma$ e $\delta(q, a) = (p, b)$, então M, quando no estado q e tendo lido o símbolo a, transitará para o estado p e,
  1. se b for um símbolo contido em Σ, M irá substituir na fita o símbolo corrente a pelo símbolo b ou,
  2. se b for um dos símbolos $\rightarrow$ ou $\leftarrow$, M moverá sua cabeça na direção convencionada para o símbolo b
- M pára quando atingir algum dos estados de parada.

Máquinas de Turing

- Exemplo
  - $M = (K, \Sigma, \delta, s, \{h\})$ onde:
    - $K = \{q_0, q_1, h\}$
    - $\Sigma = \{a, \sqcup, \triangleright\}$
    - $s = \{q_0\}$
    - δ :

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

Máquinas de Turing

- Configuração
  - uma configuração é definida como um membro de $K \times \rhd\Sigma^{\star} \times (\Sigma^{\star}(\Sigma - \{\sqcup\}) \cup \{\varepsilon\})$

$$[q, \varepsilon, \rhd aaaa]$$

    - $q \in K$ é o estado atual da máquina
    - ε é a parte da sentença de entrada já processada
    - $\rhd aaaa$ é a parte da sentença ainda não processada
  - Para simplificar adota-se a notação $[q, \rhd \underline{a}aaa]$, indicando que o cabeçote encontra-se sob o símbolo sublinhado

Máquinas de Turing

- Computação:
  - Sejam duas configurações de uma Máquina de Turing MT, $[q_1, w_1\underline{a_1}u_1]$ e $[q_2, w2\underline{a_2}u_2]$ onde $a_1, a_2 \in \Sigma$. Então

$$[q_1, w_1\underline{a_1}u_1] \vdash [q_2, w2\underline{a_2}u_2]$$

sse para algum $b \in \Sigma \cup \{\rightarrow, \leftarrow\}$, $\delta(q_1, a_1) = (q_2, b)$ e também

1. $b \in \Sigma$, $w_1 = w_2$, $u_1 = u_2$ e $a_2 = b$, ou
2. $b =\leftarrow$, $w_1 = w_2a_2$ e
   (a) $u_2 = a_1u_1$, se $a_1 \neq \sqcup$ ou $u_1 \neq \varepsilon$, ou
   (b) $u_2 = \varepsilon$, se $a_1 = \sqcup$ e $u_1 = \varepsilon$, ou ainda
3. $b =\rightarrow$, $w_2 = w_1a_1$ e
   (a) $u_1 = a_2u_2$, ou
   (b) $u_1 = u_2 = \varepsilon$, e $a_2 = \sqcup$

Máquinas de Turing

- Para transformar uma máquina de Turing em um mecanismo reconhecedor de Linguagens, é necessário alterar minimamente a sua definição, para que, ao terminar de computar a entrada a máquina finalize sua execução em um de dois estados:
  - aceitação
  - rejeição
- Outra consideração útil é permitir que a máquina escreva símbolos na fita que não pertençam ao alfabeto Σ

Máquinas de Turing

- Uma Máquina de Turing (MT) é uma séptupla:

$$M = (K, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$$

Onde:

- K = conjunto finito de estados
- Σ = alfabeto de entrada $\cup \{\triangleright\}$
- Γ = alfabeto da fita, onde $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$
- q_0 = estado inicial ($q_0 \in K$)
- q_{accept} = conjunto de estados de parada que aceitam a entrada ($q_{accept} \in K$)
- q_{reject} = conjunto de estados de parada que rejeitam a entrada ($q_{reject} \in K$)
- $\delta : K \times \Gamma$ para $(K \times \Gamma \times \{\leftarrow, \rightarrow\})$

Máquinas de Turing

- Uma máquina de Turing MT **decide** uma linguagem $L \in \Sigma^\star$, se para qualquer cadeia $w \in \Sigma^\star$
  - MT aceita w, se $w \in L$
  - MT rejeita w, se $w \notin L$
- Se uma linguagem L é **decidível** então L é dita **Recursiva** ou **Turing Decidível**