CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

Linguagens Formais e Autômatos

Máquinas de Turing (MT)

Andrei Rimsa Álvares
andrei@cefetmg.br

Sumário

- Introdução
- Máquinas de Turing
- Propriedades
  - Linguagens Recursivas
  - Linguagens Recursivamente Enumeráveis

INTRODUÇÃO

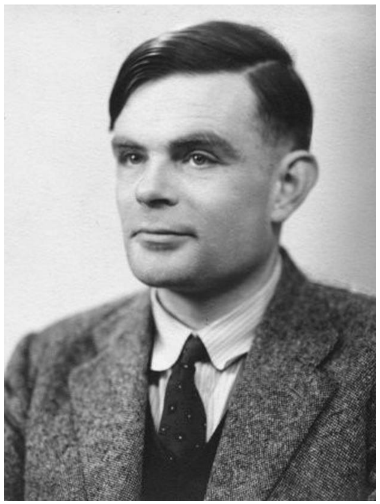
Introdução

- Os autômatos finitos e autômatos de pilha, embora importantes do ponto de vista prático e teórico, possuem limitações quanto ao poder de reconhecimento

- Mesmo linguagens relativamente simples não podem ser reconhecidas por essas máquinas, como
  - $\{xx \mid x \in \{a, b\}^*\}$
  - $\{a^n b^n c^n \mid n \geq 0\}$
  - $\{a^n b^k c^n d^k \mid n, k \geq 0\}$

E agora, quem poderá nos defender?

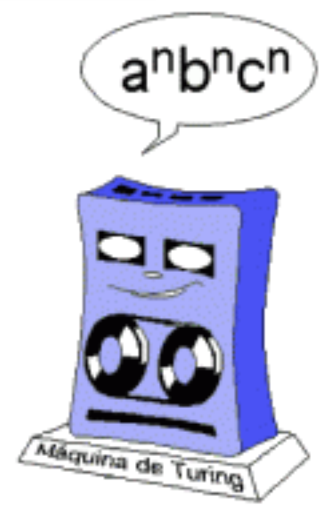
Máquinas de Turing



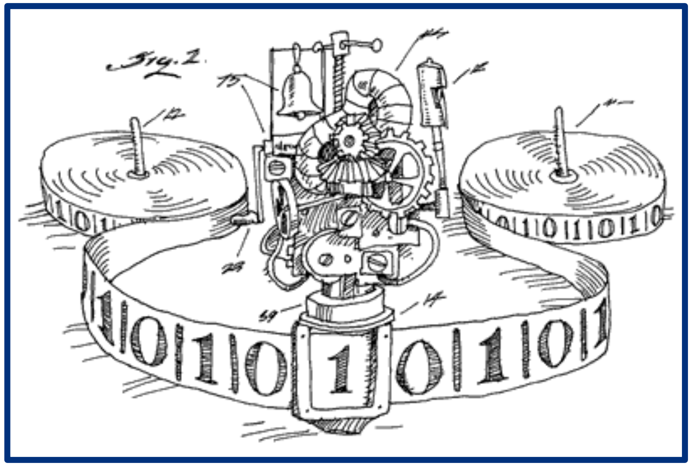
- As máquinas de Turing foram propostas por volta de 1930 pelo matemático inglês Alan Turing

Dica: o filme **O Jogo da Imitação** mostra as contribuições de Turing para desvendar mensagens cifradas pela máquina Enigma

- Esse tipo de máquina é tão poderosa que até hoje não se conseguiu nenhum outro tipo de máquina que tenha maior poder computacional
  - Nem mesmo os computadores atuais possuem poder computacional maior que o das máquinas de Turing



MÁQUINAS DE TURING (MT)

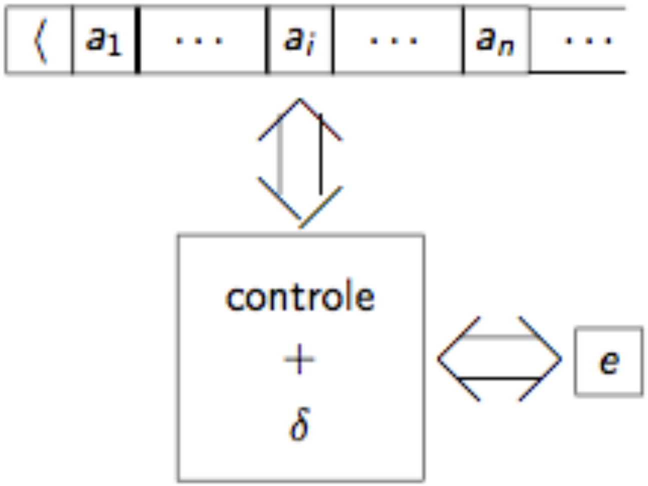


O que é uma Máquina de Turing

- Uma máquina de Turing (MT) opera sobre uma fita que, ao contrário dos AFs e APs, pode-se **ler** e **escrever** nela

- A fita é
  - Dividida em células que comportam apenas um símbolo cada uma
  - Limitada à esquerda, possui um símbolo especial para marcar seu início que não pode repetir em nenhuma outra célula
  - Ilimitada à direita

- A máquina possui
  - Um cabeçote de leitura que pode se mover para a direita e para a esquerda
  - Um registrador que contém o estado atual
  - Um conjunto de instruções (δ)
  - Uma unidade de controle



Função de Transição

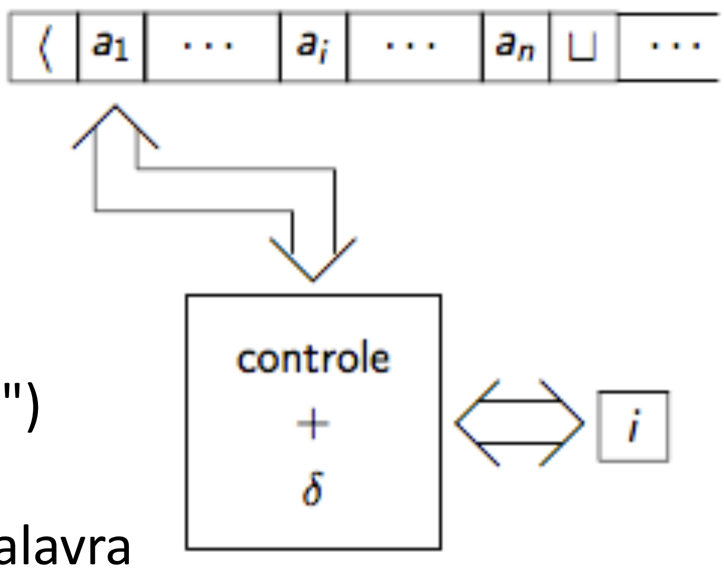
- A função de transição é uma função parcial, que dá, para cada par (e, a), onde e é um estado e a é um símbolo do alfabeto, uma tripla $[e', b, d]$, onde
  - e' é o próximo estado
  - b é o símbolo a substituir a
  - d é a direção, esquerda (E) ou direita (D), em que o cabeçote deve se mover

Notação:
$\delta(e, a) = [e', b, d]$

- **Cuidado**
  - Quando $a = \langle$, obrigatoriamente $b = \langle$ e $d = \mathrm{D}$ ou $\delta(e, \langle)$ é indefinido (não se pode mover o cabeçote para a esquerda da primeira célula)
  - O símbolo $\langle$, além de não poder ser escrito em qualquer outra célula da fita, não pode ser apagado da primeira célula da fita

Configuração Inicial

- O registrador da máquina contém o estado inicial (i)
- A fita contém
  - A primeira célula que contém ⟨ (marcador de início)
  - A palavra de entrada a partir da sua segunda célula ($a_1 \ldots a_n$)
  - O restante da fita contém somente o símbolo ⊔ (branco ou "célula vazia")



- O cabeçote é posicionado no início da palavra de entrada (segunda célula)

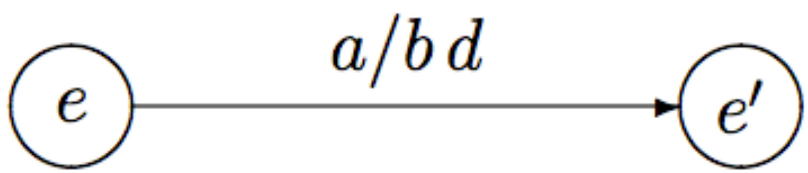
Algoritmo

- **Enquanto** $\delta(e, a)$ é definido, onde e é o estado no registrador da máquina, a é o símbolo sob o cabeçote e $\delta(e, a) = [e', b, d]$
  - Colocar no registrador o estado e'
  - Substitui-se a por b na posição sob o cabeçote
  - Avança-se o cabeçote para a célula da esquerda, se $d = \mathrm{E}$, ou para a direita, se $d = \mathrm{D}$

Determinismo

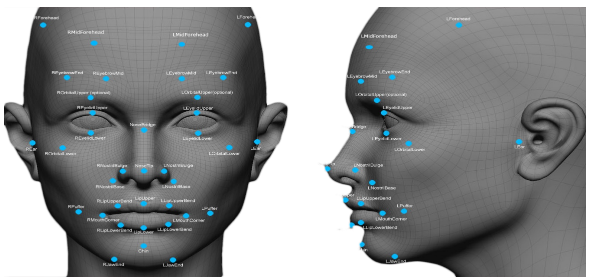
- A Máquina de Turing é **determinística**
  - Para cada estado e e símbolo a há, no máximo, uma transição especificada pela função de transição

- Uma transição $\delta(e, a) = [e', b, d]$, onde $d \in \{\mathrm{E}, \mathrm{D}\}$, será representada da seguinte maneira em um diagrama de estados



Aplicações

- Uma Máquina de Turing pode ser usada como

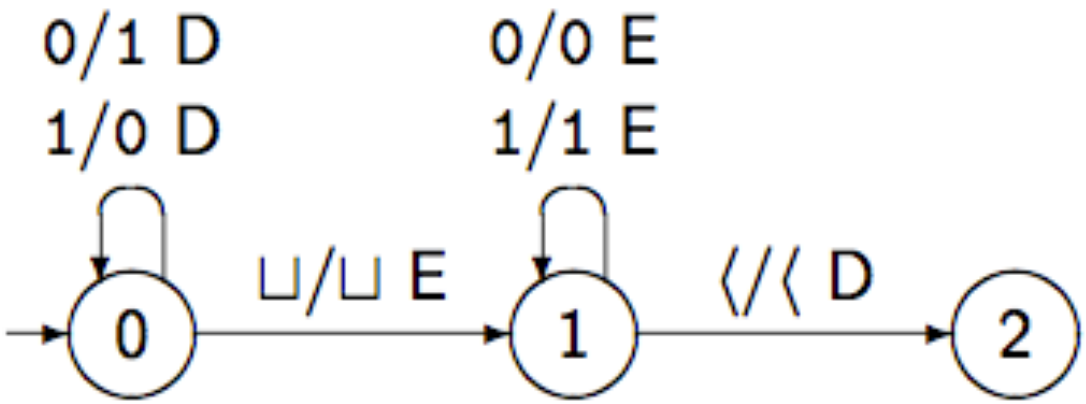


  – Reconhecedora de linguagens

  – Transdutora de linguagens: $\Sigma^* \rightarrow \Gamma^*$ (recebe uma palavra *w* na entrada e produz na própria fita uma saída)

Um Primeiro Exemplo

- Uma Máquina de Turing que recebe como entrada uma palavra de $\{0, 1\}^*$ e produz seu complemento como saída (inverso)



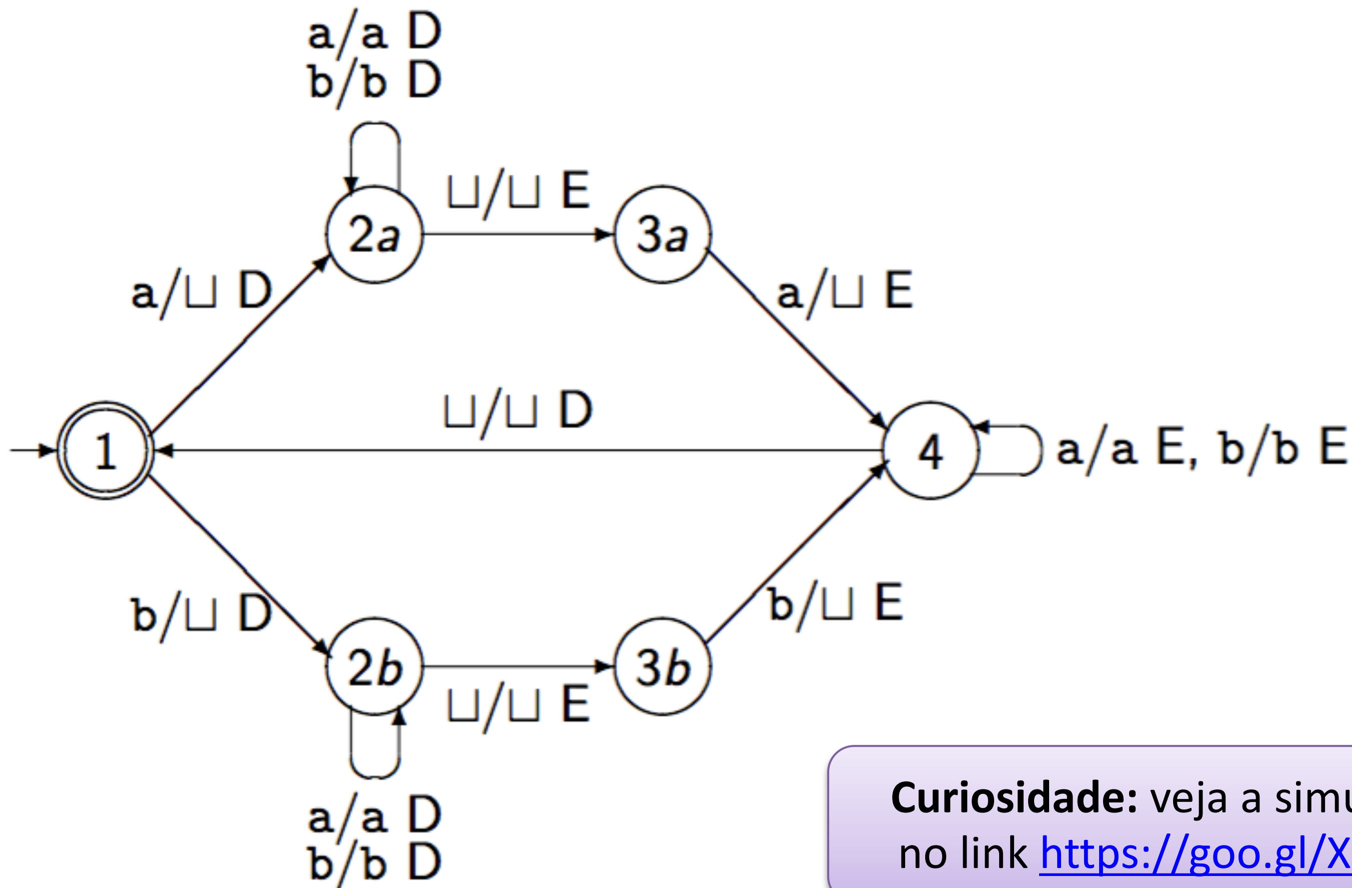
Curiosidade: veja a simulação no link https://goo.gl/3bj613

Máquina de Turing

- **Definição:** uma Máquina de Turing é uma óctupla $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$, em que
  - E é um conjunto finito de estados
  - $\Sigma \subseteq \Gamma$ é o alfabeto de entrada
  - Γ é o alfabeto da fita
  - $\langle$ é o primeiro símbolo da fita ($\langle \in \Gamma - \Sigma$)
  - $\sqcup$ é o branco ($\sqcup \in \Gamma - \Sigma$, $\sqcup \neq \langle$)
  - δ: $E \times \Gamma \rightarrow E \times \Gamma \times \{\mathrm{E}, \mathrm{D}\}$ é a função de transição
  - i é o estado inicial
  - F é o conjunto de estados finais

Exemplo

- Máquina de Turing que reconhece $L = \{ww^R \mid w \in \{a, b\}^*\}$



Curiosidade: veja a simulação no link https://goo.gl/XzjAAV

Configuração Instantânea

- Uma configuração instantânea de uma Máquina de Turing é um par $[e, x\underline{a}y]$, em que
  - $e \in E$ é o estado atual
  - $x \in \Gamma^*$ é a palavra situada à esquerda do cabeçote de leitura
  - $a \in \Gamma$ é o símbolo sob o cabeçote
  - $y \in \Gamma^*$ é a palavra à direita do cabeçote até o último símbolo diferente de $\sqcup$

- A configuração inicial é
  - $[i, \langle\underline{a_1}a_2...a_n]$, caso a palavra de entrada seja $a_1a_2...a_n$
  - $[i, \langle\underline{\sqcup}]$, caso a palavra de entrada seja λ

Mudança de Configuração

- Como uma MT pode entrar em loop, não será definida uma função que retorne o estado alcançado a partir de uma certa configuração instantânea
  - A relação $\vdash$ será definida mais à frente

- Para facilitar a definição, será utilizada a função $\pi : \Gamma^* \rightarrow \Gamma^*$

$$\pi(w) = \begin{cases} \lambda & \text{, se } w \in \{\sqcup\}^* \\ xa & \text{, se } w = xay,\ a \in \Gamma - \{\sqcup\} \text{ e } y \in \{\sqcup\}^* \end{cases}$$

Informalmente, $\pi(w)$ elimina de w os brancos à direita do último símbolo diferente de branco

Mudança de Configuração

- Definição: A relação $\vdash \subseteq (E \times \Gamma^{+})^2$, para uma MT M, é tal que para todo $e \in E$ e todo $a \in \Gamma$

a) se $\delta(e, a) = [e', b, \mathrm{D}]$, então $[e, x\underline{a}cy] \vdash [e', xb\underline{c}y]$ para $c \in \Gamma$;
$[e, x\underline{\mathrm{a}}] \vdash [e', xb\underline{\sqcup}]$

b) se $\delta(e, a) = [e', b, \mathrm{E}]$, então $[e, xc\underline{a}y] \vdash [e', x\underline{c}\pi(by)]$ para $c \in \Gamma$

c) se $\delta(e, a)$ é indefinido, então não existe configuração f tal que $[e, x\underline{a}y] \vdash f$

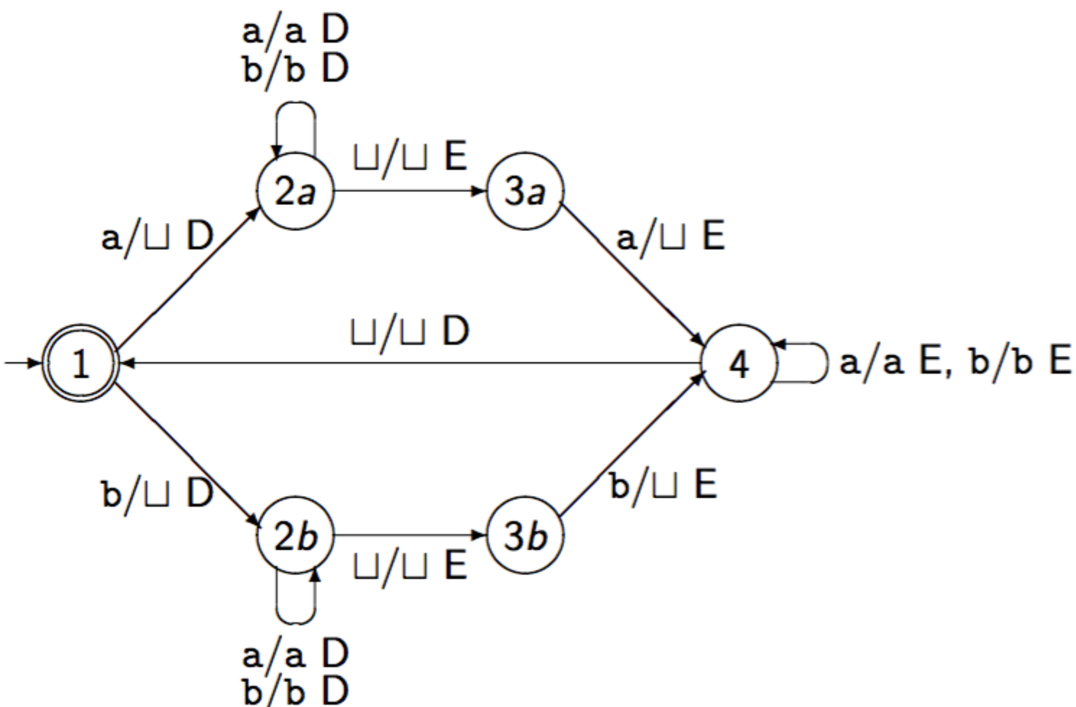
Mudança de Configuração

- Como usual, serão definidos
  - O fecho reflexivo e transitivo de $\vdash$ será denotado por $\overset{*}{\vdash}$
  - Configuração instantânea f' é obtida a partir de f percorrendo-se $n \geq 0$ transições: $f \overset{n}{\vdash} f'$

Quando as palavras são aceitas?

Mudança de Configuração

- Por exemplo, para a MT do exemplo anterior
  - $[1, \langle\underline{\sqcup}] \overset{0}{\vdash} [1, \langle\underline{\sqcup}]$ e $\delta(1, \sqcup)$ é indefinido
  - $[1, \langle\underline{a}ab] \overset{4}{\vdash} [3a, \langle\sqcup a\underline{b}]$ e $\delta(3a, b)$ é indefinido
  - $[1, \langle\underline{a}bba] \overset{14}{\vdash} [1, \langle\sqcup\sqcup\underline{\sqcup}]$ e $\delta(1, \sqcup)$ é indefinido



Quando as palavras são aceitas?

Linguagem Reconhecida

- **Definição:** A linguagem reconhecida por uma MT $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ é

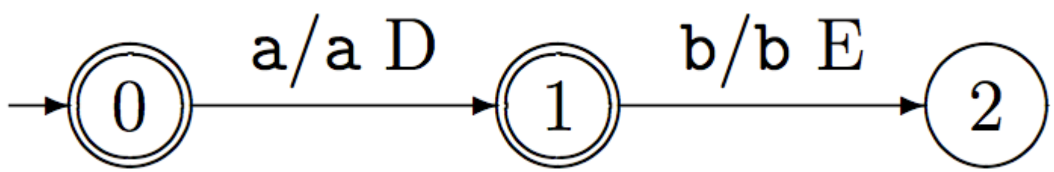
$$L(M) = \{w \in \Sigma^* \mid [i, \langle \underline{w}] \overset{*}{\vdash} [\mathrm{e}, \mathrm{x}\underline{\mathrm{a}}\mathrm{y}], \delta(e, a) \text{ é indefinido e } e \in F\}$$

A expressão $[i, \langle \underline{w}]$ é usada para denotar a configuração instantânea inicial

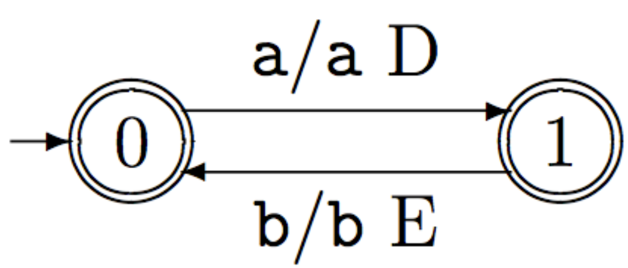
Exemplos de Reconhecimento

- Considere a linguagem sobre $\Sigma = \{a, b, c\}$ que não tem ab como prefixo, ou seja: $L = \{a, b, c\}^* - (\{ab\}\{a, b, c\}^*)$

  – Máquina que **sempre para**



  – Máquina que **para se aceita**



Diferente dos AFs e APs, não é necessário consumir toda a palavra de entrada para que ela seja aceita

Classe de Linguagens

- As definições a seguir dão nomes para as classes das linguagens que podem ser reconhecidas por MTs
  - **Linguagem Recursivamente Enumerável (LRE):** uma linguagem é dita ser uma linguagem recursivamente enumerável (LRE) se existe uma MT que a **reconhece**
  - **Linguagem Recursiva:** uma linguagem é dita ser uma linguagem recursiva se existe uma MT que a **reconhece** e que **para para todas as palavras** do alfabeto de entrada

Se existe uma MT que reconhece uma linguagem L, necessariamente existe uma MT que sempre para e que reconhece L?

Modelos Alternativos de Reconhecimento

- Existem outros dois modelos alternativos de reconhecimento que são úteis em certos contextos
  - **Por estado final:** seja uma MT $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$. A linguagem reconhecida por M por estado final é

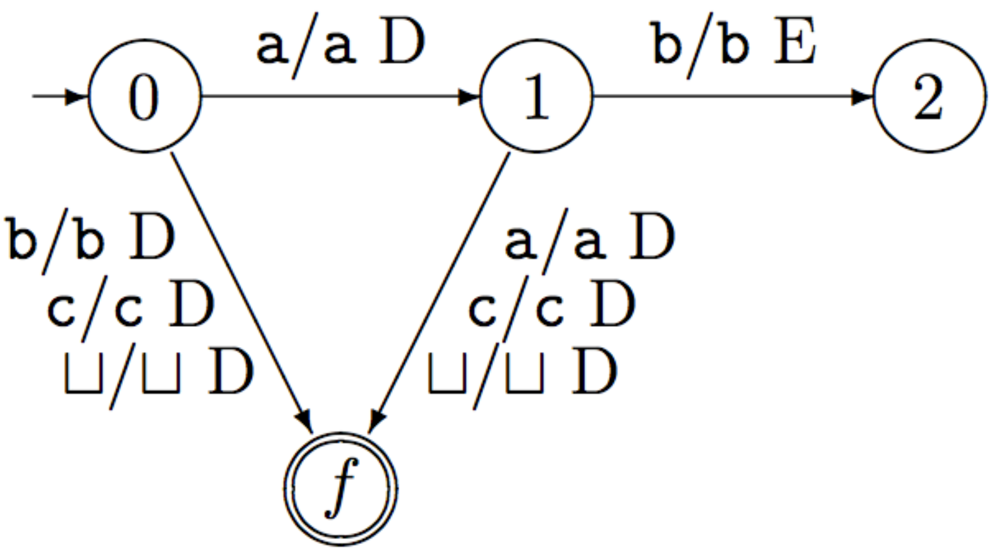
$$L_F(M) = \{\ w \in \Sigma^* \mid [i, \langle \underline{w}] \overset{*}{\vdash} [e, x\underline{a}y], a \in \Gamma \text{ e } e \in F\}$$

  - **Por parada:** seja uma MT $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i)$. A linguagem reconhecida por M por parada é

$$L_P(M) = \{\ w \in \Sigma^* \mid [i, \langle \underline{w}] \overset{*}{\vdash} [e, x\underline{a}y], a \in \Gamma \text{ e } \delta(e, a) \text{ é indefinido}\}$$

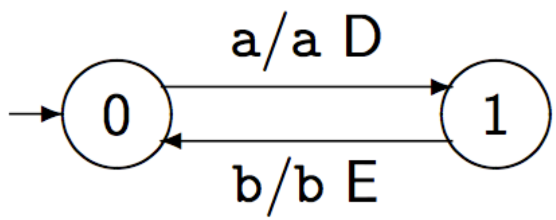
Exemplo

- Exemplo de reconhecimento por **estado final**:
  $L = \{a, b, c\}^* - (\{ab\}\{a, b, c\}^*)$



Exemplo

- Exemplo de reconhecimento por **parada**:
  $L = \{a, b, c\}^* - (\{ab\}\{a, b, c\}^*)$



Equivalência entre os Modelos de Reconhecimento

- Seja L uma linguagem. As seguintes afirmativas são **equivalentes**

a) L é uma LRE
b) L pode ser reconhecida por uma MT por estado final
c) L pode ser reconhecida por uma MT por parada

Será demonstrado que as seguintes transformações são possíveis:
(a) → (b), (b) → (c), (c) → (a)

(a) → (b)

- Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT padrão; uma máquina equivalente à M, que reconhece por estado final, seria $M' = (E \cup \{f\}, \Sigma, \Gamma, \langle, \sqcup, \delta', i, \{f\}), f \notin F$, onde

a) para todo $(e, a) \in E \times \Gamma$, se $\delta(e, a)$ é definido, então $\delta'(e, a) = \delta(e, a)$

b) para todo $(e, a) \in F \times \Gamma$, se $\delta(e, a)$ é indefinido, $\delta'(e, a) = [f, a, \mathrm{D}]$

c) para todo $(e, a) \in (E - F) \times \Gamma$, se $\delta(e, a)$ é indefinido, $\delta'(e, a)$ é indefinido

d) para todo $a \in \Gamma$, $\delta'(f, a)$ é indefinido

Observe que o cabeçote se move para a direita **(b)**, prevendo o caso em que $a = \langle$

(b) → (c)

- Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT que reconhece por estado final; uma máquina equivalente à M, que reconhece por parada, seria $M' = (E \cup \{l\}, \Sigma, \Gamma, \langle, \sqcup, \delta', i)$, $l \notin E$, onde

a) para todo $(e, a) \in (E - F) \times \Gamma$: se $\delta(e, a)$ é definido, então $\delta'(e, a) = \delta(e, a)$, senão $\delta'(e, a) = [l, a, \mathrm{D}]$

b) para todo $a \in \Gamma$, $\delta'(l, a) = [l, a, \mathrm{D}]$

c) para todo $(e, a) \in F \times \Gamma$, $\delta'(e, a)$ é indefinido

Observe que o cabeçote é movido indefinidamente para a direita ao ser atingido o estado l **(b)**

(c) → (a)

- Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i)$ uma MT que reconhece por parada; uma MT normal equivalente à M é obtida simplesmente tornando todos os estados de M estados finais: $M' = (E, \Sigma, \Gamma, \langle, \sqcup, \delta', i, E)$

Dica: quando não se disser o contrário, assume-se que o reconhecimento se dá **por parada em estado final**

Propriedades: Linguagens Recursivas e Linguagens Recursivamente Enumeráveis

Propriedades das LREs e Linguagens Recursivas

- A classe das **linguagens recursivas** é fechada sob
  - **União**
  - **Interseção**
  - **Complemento**
  - **Concatenação**
  - **Fecho de Kleene**
- A classe das linguagens **recursivamente enumeráveis** (LREs) é fechada sob
  - **União**
  - **Interseção**
  - **Concatenação**
  - **Fecho de Kleene**

Cuidado: A classe das LREs **não é** fechada sob o complemento

Linguagens Não LRE

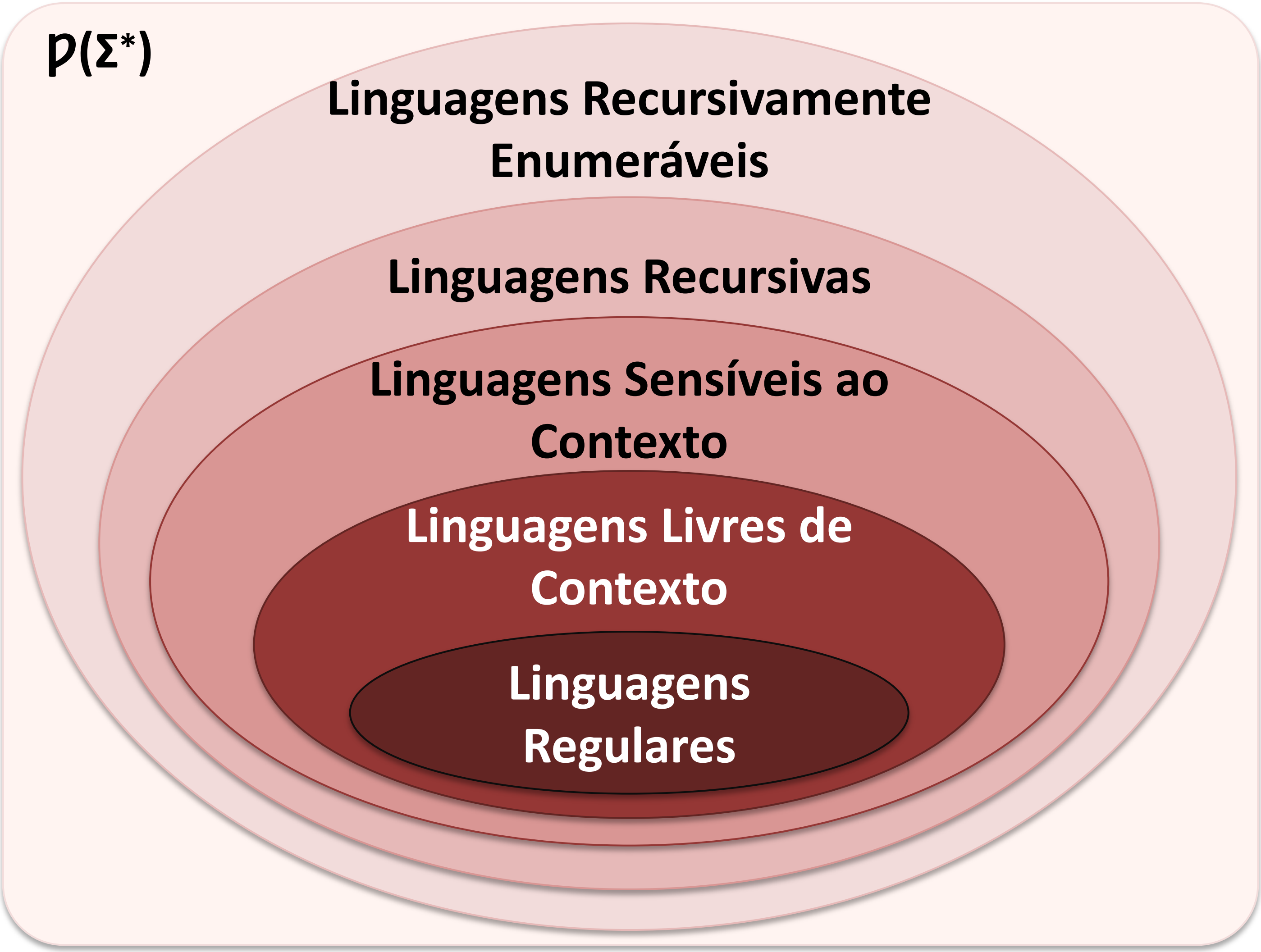
- Existem linguagens que não são LRE, considere o seguinte raciocínio
  - Seja R uma linguagem sobre Σ cujas palavras representam todas as MTs
  - Como Σ^* é um conjunto enumerável e $R \subseteq \Sigma^*$, R é enumerável
    - Ou seja, o conjunto das MTs é enumerável, independentemente da linguagem usada para representá-las
  - O conjunto de todas as linguagens de alfabeto Σ, $P(\Sigma^*)$, não é enumerável
  - Como o conjunto das MTs é enumerável e o conjunto das linguagens não, segue-se que não há como associar cada linguagem a uma MT
    - Não existe uma função injetiva de $P(\Sigma^*)$ para R
  - Logo, existem mais linguagens do que MTs

Um Teorema Importante

- **Teorema:** se L e $\overline{L}$ são LRE, então L é recursiva

  - Em particular, segue-se a contrapositiva
    - Se L é LRE e L não é recursiva, então $\overline{L}$ não é LRE

Espaço das Linguagens em $P(\Sigma^*)$



ISSO É TUDO, PESSOAL!