Capítulo 4: Máquinas de Turing

Newton José Vieira   Isabel Gomes Barbosa

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

23 de setembro de 2010

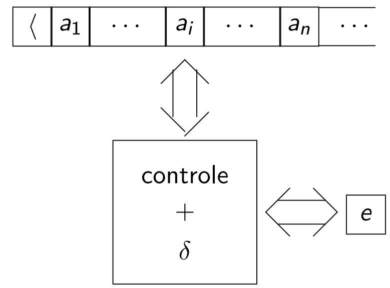
Sumário

Sumário

Sumário

Sumário

O que É Máquina de Turing

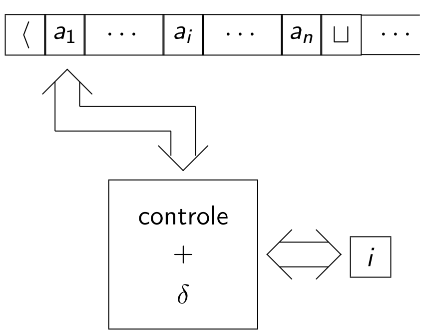


Componentes de uma MT:

1. fita de leitura e escrita;
2. cabeçote de leitura;
3. registrador;
4. função de transição;
5. unidade de controle.

Operação de uma MT

No início, sendo i o estado inicial da máquina de turing e $a_1 a_2 ... a_n$ a palavra de entrada:



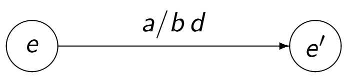
Operação de uma MT (cont.)

Enquanto $\delta(e, a)$ é definido, em que e refere-se ao estado no registrador da máquina, a é o símbolo sob o cabeçote e $\delta(e, a) = [e', b, d]$:

1. coloca-se no registrador o estado e';
2. substitui-se a por b na posição sob o cabeçote;
3. avança-se o cabeçote para a célula da esquerda, se $d = E$, ou para a da direita, se $d = D$.

Propriedades

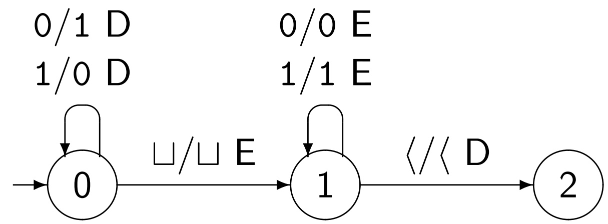
1. Uma MT é **determinística**: para cada estado e e símbolo a, há, no máximo, uma transição especificada pela função de transição.
2. Uma transição $\delta(e, a) = [e', b, d]$ será representada em um diagrama de estados da seguinte forma:



3. Uma MT pode ser usada como:
   - reconhecedora de linguagens;
   - transdutora de linguagens: $\Sigma^* \rightarrow \Gamma^*$.

Um exemplo

Uma MT que, recebendo como entrada uma palavra de $\{0, 1\}^*$, produz o complemento:



Definição de MT

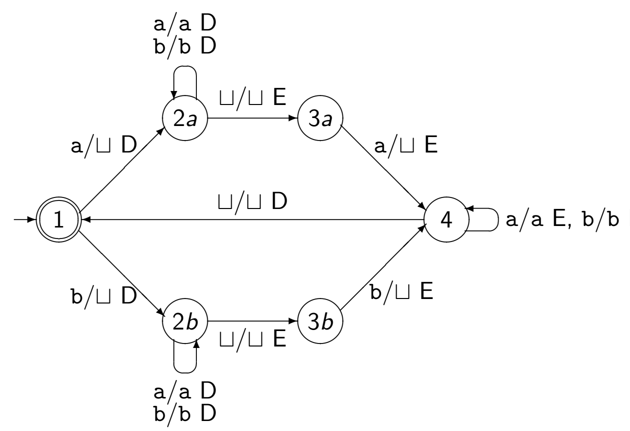
O que é MT

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* ($\sqcup \in \Gamma - \Sigma$, $\sqcup \neq \langle$);
- $\delta : E \times \Gamma \rightarrow E \times \Gamma \times \{E, D\}$ é a função de transição;
- i é o estado inicial;
- F é o conjunto de estados finais.

Exemplo

Diagrama de estados para uma MT que reconhece $\{ww^R \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$:



Configuração de uma MT

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; e
- $y \in \Gamma^*$ 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 (1)

Função $\pi : \Gamma^* \to \Gamma^*$: $\pi(w)$ elimina de w os brancos à direita do último símbolo diferente de branco.

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

A relação $\vdash \subseteq (E \times \Gamma^+)^2$, para uma MT M, é tal que:

1. 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 $[e, x\underline{a}] \vdash [e', xb\underline{\sqcup}]$;
2. se $\delta(e, a) = [e', b, \mathrm{E}]$, então $[e, xc\underline{a}y] \vdash [e', x\underline{c}\pi(by)]$ para $c \in \Gamma$;
3. se $\delta(e, a)$ é indefinido, então não existe configuração f tal que $[e, x\underline{a}y] \vdash f$.

Mudança de Configuração (2)

- 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: $f \overset{n}{\vdash} f'$.

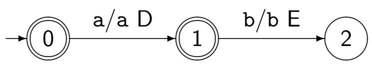
Linguagem Reconhecida por uma MT

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

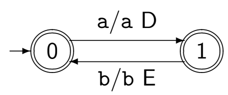
$$L(M) = \{w \in \Sigma^* | [i, \langle\underline{w}] \overset{*}{\vdash} [e, x\underline{a}y], \delta(e, a) \text{ é indefinido e } e \in F\}$$

Duas MTs

MTs para $\{a, b, c\}^* - (\{ab\}\{a, b, c\}^*)$:



Máquina que pára sempre.



Máquina que pára se aceita.

Linguagens recursivamente enumeráveis e recursivas

Linguagem recursivamente enumerável

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 pare para todas as palavras do alfabeto de entrada.

Modelos alternativos de reconhecimento

Por estado final

Seja uma MT $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$. A linguagem reconhecida por M por estado final é:

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

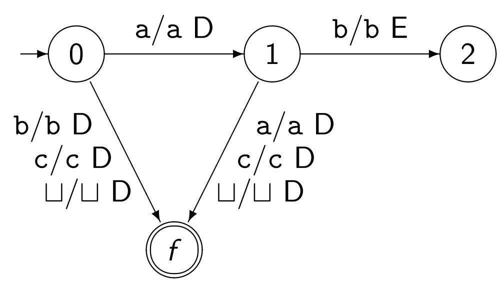
Por parada

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

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

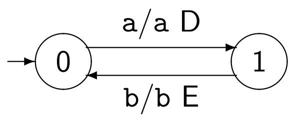
Exemplo/reconhecimento por estado final

Máquina que reconhece $\{a, b, c\}^* - (\{ab\}\{a, b, c\}^*)$ por estado final:



Exemplo/reconhecimento por parada

Máquina que reconhece $\{a, b, c\}^* - (\{ab\}\{a, b, c\}^*)$ por parada:



Equivalência dos modelos alternativos 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.

Máquina com cabeçote imóvel

A máquina é uma óctupla $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i)$, sendo que:

- δ é uma função de $E \times \Gamma$ para $E \times \Gamma \times \{D, E, I\}$.

Uma transição $\delta(e, a) = [e', b, \mathrm{I}]$ pode ser simulada por transições das formas a seguir, sendo d um *novo* estado:

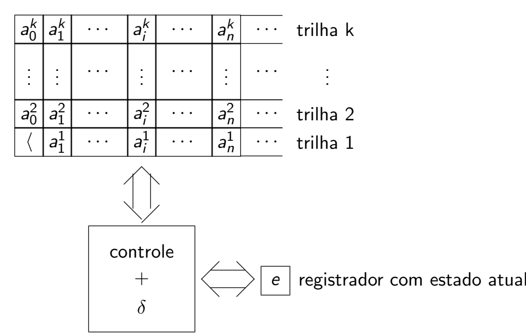
- $\delta(e, a) = [d, b, \mathrm{D}]$;
- $\delta(d, c) = [e', c, \mathrm{E}]$ para cada $c \in \Gamma - \{\langle\}$.

Exemplo

Máquina com múltiplas trilhas

A fita é composta por múltiplas trilhas: cada célula contém uma k-upla de símbolos.

No início, a palavra de entrada está na trilha 1.



Máquina com múltiplas trilhas

Uma MT com k trilhas é uma óctupla $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ em que:

- δ é uma função de $E \times \Gamma^k$ para $E \times \Gamma^k \times \{D, E\}$.

Uma configuração instantânea tem a forma:

- $[e, x_1 a_1 y_1, x_2 a_2 y_2, ..., x_k a_k y_k]$, onde $|x_i| = |x_j|$ para $i \neq j$.

A linguagem aceita é o conjunto de toda palavra $w \in \Sigma^*$ tal que

$$[i, \langle \underline{w}, \sqcup\underline{\sqcup}, \ldots, \sqcup\underline{\sqcup}] \overset{*}{\vdash} [e, x_1\underline{a_1}y_1, x_2\underline{a_2}y_2, \ldots, x_k\underline{a_k}y_k]$$

onde $e \in F$ e $\delta(e, a_1, a_2, \ldots, a_k)$ é indefinido.

MT-padrão e MT com múltiplas trilhas

Uma MT-padrão $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ pode ser simulada por uma com k trilhas $(E, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$ em que:

- se $\delta(e, a) = [e', b, d]$, então $\delta'(e, a, \sqcup, \ldots, \sqcup) = [e', b, \sqcup, \ldots, \sqcup, d]$.

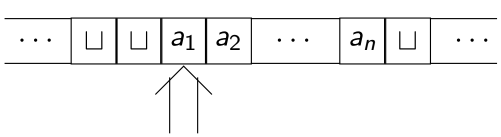
Dada uma MT de k trilhas $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$, obtém-se uma MT-padrão equivalente $M' = (E, \Sigma \times \{\sqcup\}^{k-1}, \Gamma \times (\Gamma - \{\langle\})^{k-1}, \{\langle\} \times \{\sqcup\}^{k-1}, \{\sqcup\}^k, \delta', i, F)$ de forma que:

- se $\delta(e, a_1, a_2, \ldots, a_k) = [e', b_1, b_2, \ldots, b_k, d]$, então $\delta'(e, [a_1, a_2, \ldots, a_k]) = [e', [b_1, b_2, \ldots, b_k], d]$.

Máquina com fita ilimitada em ambas as direções

A fita é ilimitada também à esquerda;

No início, o cabeçote está posicionado no primeiro símbolo da palavra de entrada, se esta não for λ:



De MT-padrão a MT com fita ilimitada em ambas as direções

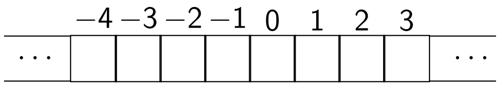
Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT-padrão e $i', j \notin E$.

Uma MT com fita ilimitada em ambas as direções, equivalente a M, seria $M' = (E \cup \{i', j\}, \Sigma, \Gamma, \sqcup, \delta', i', F)$ em que δ' consta das mesmas transições que M acrescidas de:

- $\delta'(i', a) = [j, a, \mathrm{E}]$ para cada $a \in \Gamma$;
- $\delta'(j, \sqcup) = [i, \langle, \mathrm{D}]$.

De MT com fita ilimitada em ambas as direções a MT-padrão

Seja $M = (E, \Sigma, \Gamma, \sqcup, \delta, i, F)$ uma máquina com fita ilimitada em ambas as direções. Considerando a fita com as células indexadas assim:



Pode-se obter uma máquina de duas trilhas que simula M, como segue.

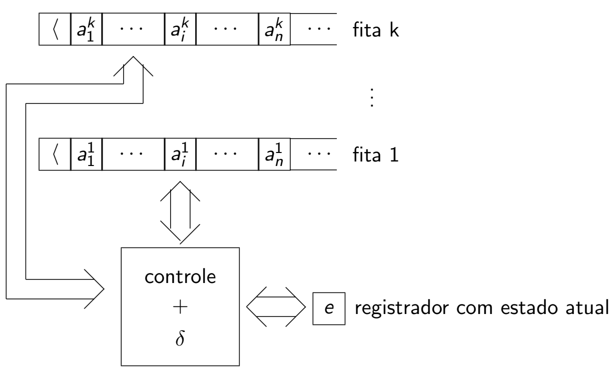
A MT ilimitada em ambas as direções

$M' = (E', \Sigma, \Gamma', \langle, \sqcup, \delta', i', F')$

- $E' = E \times \{1, 2\}$; $i' = [i, 1]$; $F' = F \times \{1, 2\}$; $\Gamma' = \Gamma \cup \{\langle\}$, $\langle \notin \Gamma$;
- δ' é obtida de δ assim:
  - para cada transição $\delta(e, a) = [e', b, \mathrm{D}]$, deve-se ter:
    - $\delta'([e, 1], a, c) = [[e', 1], b, c, \mathrm{D}]$ para cada $c \in \Gamma$;
    - $\delta'([e, 2], c, a) = [[e', 2], c, b, \mathrm{E}]$ para cada $c \in \Gamma$;
    - $\delta'([e, 1], \langle, a) = \delta'([e, 2], \langle, a) = [[e', 1], \langle, b, \mathrm{D}]$;
  - para cada transição $\delta(e, a) = [e', b, \mathrm{E}]$, deve-se ter:
    - $\delta'([e, 1], a, c) = [[e', 1], b, c, \mathrm{E}]$ para cada $c \in \Gamma$;
    - $\delta'([e, 2], c, a) = [[e', 2], c, b, \mathrm{D}]$ para cada $c \in \Gamma$;
    - $\delta'([e, 1], \langle, a) = \delta'([e, 2], \langle, a) = [[e', 2], \langle, b, \mathrm{D}]$.

Máquina com múltiplas fitas

Nessa variação, cada fita tem seu cabeçote de leitura/escrita.



No início, a palavra de entrada está na fita 1.

Máquina com múltiplas fitas

Uma MT de k fitas é uma óctupla $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ onde:

- δ é uma função de $E \times \Gamma^k$ para $E \times (\Gamma^k \times \{D, E, I\})^k$.

Uma configuração instantânea tem a forma:

- $[e, x_1 a_1 y_1, x_2 a_2 y_2, ..., x_k a_k y_k]$.

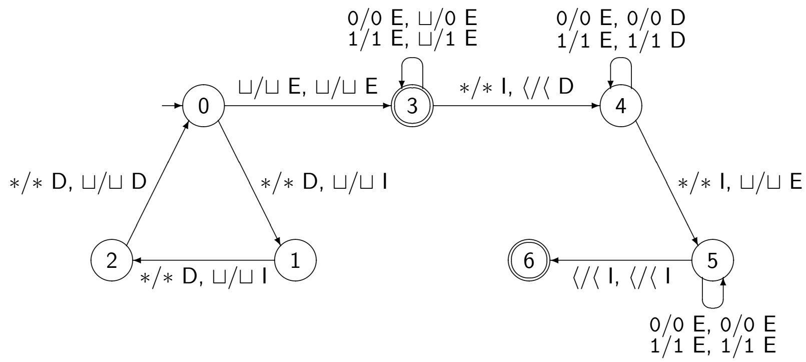
A linguagem aceita é o conjunto de toda palavra $w \in \Sigma^*$ tal que

$$[i, \langle \underline{w}, \langle \underline{\sqcup}, \ldots, \langle \underline{\sqcup}] \overset{*}{\vdash} [e, x_1 \underline{a_1} y_1, x_2 \underline{a_2} y_2, \ldots, x_k \underline{a_k} y_k]$$

onde $e \in F$ e $\delta(e, a_1, a_2, \ldots, a_k)$ é indefinido.

Exemplo de máquina com múltiplas fitas

Diagrama de estados de uma máquina de duas fitas que reconhece $\{ww^Rw \mid w \in \{0,1\}^*\}$.



Máquina não determinística

- Admite mais de uma transição partindo de certo estado sob determinado símbolo.
- Podem existir várias computações no processamento de uma palavra.
- Uma palavra é aceita quando **existe** uma computação para a qual a máquina pára em um estado final.

Máquina não determinística

Uma MT não determinística é uma óctupla $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$, onde:

- δ é uma função total de $E \times \Gamma$ para $\mathcal{P}(E \times \Gamma \times \{\text{D}, \text{E}\})$.

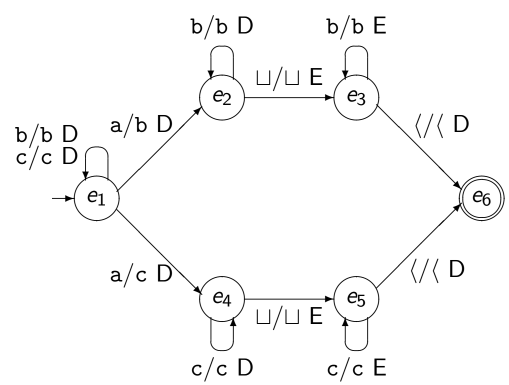
No caso em que $\delta(e, a) = \emptyset$ para certo estado e e símbolo a, não há transição do estado e sob a.

A linguagem aceita pela MT M é:

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

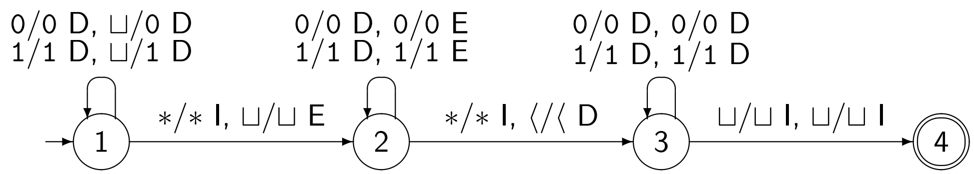
Exemplo de máquina não determinística

Diagrama de estados de uma MT não determinística que aceita a linguagem $b^*ab^* + c^*ac^*$.



Exemplo/MT não determinística de duas fitas

MT de duas fitas não determinística para $\{ww^Rw \mid w \in \{0,1\}^*\}$:.



De gramática a máquina de Turing

A linguagem gerada por uma gramática irrestrita é uma LRE.

Seja uma gramática irrestrita $G = (V, \Sigma, R, P)$.

Constrói-se uma MT não determinística de duas fitas, M, tal que $L(M) = L(G)$.

- fita 1: palavra de entrada;
- fita 2: forma sentencial de G.

O algoritmo da máquina M é apresentado a seguir.

De gramática a máquina de Turing

1. Escreva P (a variável de partida) na fita 2.
2. **ciclo**
    - 2.1 selecione uma posição p na forma sentencial que está na fita 2;
    - 2.2 selecione uma regra $u \rightarrow v \in R$;
    - 2.3 **se** u ocorre a partir da posição p da fita 2 **então**
        - 2.3.1 substitua u por v na fita 2;
        - 2.3.2 **se** a forma sentencial na fita 2 é idêntica à palavra de entrada na fita 1 **então**
            - aceite
        - **fimse**
    - **senão**
        - rejeite
    - **fimse**

   fimciclo.

De máquina de Turing a gramática

Uma LRE pode ser gerada por uma gramática irrestrita.

Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT que aceita L.

Constrói-se, a partir de M, uma gramática irrestrita $G = (V, \Sigma, R, P)$ que gera L.

Existirão regras em G para três propósitos, os quais são descritos a seguir.

De máquina de Turing a gramática/1

Gerar todas as formas sentenciais do tipo $w\langle iw\rangle$, onde $w \in \Sigma^*$:

$P \rightarrow B\rangle$

$B \rightarrow a_k B A_k$ para $1 \leq k \leq n$ (portanto, n regras)

$B \rightarrow \langle i$

$A_k\rangle \rightarrow a_k\rangle$ para $1 \leq k \leq n$ (portanto, n regras)

$A_j a_k \rightarrow a_k A_j$ para $1 \leq k, j \leq n$ (portanto, n^2 regras).

De máquina de Turing a gramática/2

Simular M sobre a configuração instantânea $\langle iw\rangle$, deixando o prefixo w inalterado:

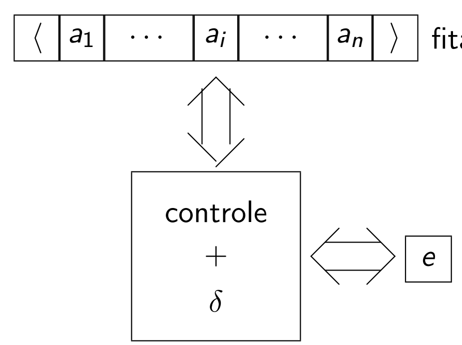
- para cada transição em M da forma $\delta(e, a) = [e', b, \mathrm{D}]$:
  $ea \rightarrow be'$
  $e\rangle \rightarrow be'\rangle$ se $a = \sqcup$
- para cada transição em M da forma $\delta(e, a) = [e', b, \mathrm{E}]$:
  $cea \rightarrow e'cb$ para cada $c \in \Gamma$
  $ce\rangle \rightarrow e'cb\rangle$ para cada $c \in \Gamma$, se $a = \sqcup$.

De máquina de Turing a gramática/3

Apagar a configuração instantânea quando ela for do tipo $\langle xeay\rangle$, onde $e \in F$ e $\delta(e, a)$ é indefinido:

- para cada par (e, a) tal que $e \in F$ e $\delta(e, a)$ é indefinido:
  $ea \rightarrow a\#$
  $\#c \rightarrow \#$ para cada $c \in \Gamma - \{\langle\}$
  $c\# \rightarrow \#$ para cada $c \in \Gamma - \{\langle\}$
  $\langle\#\rangle \rightarrow \lambda$

Arquitetura de um autômato linearmente limitado



Autômato linearmente limitado

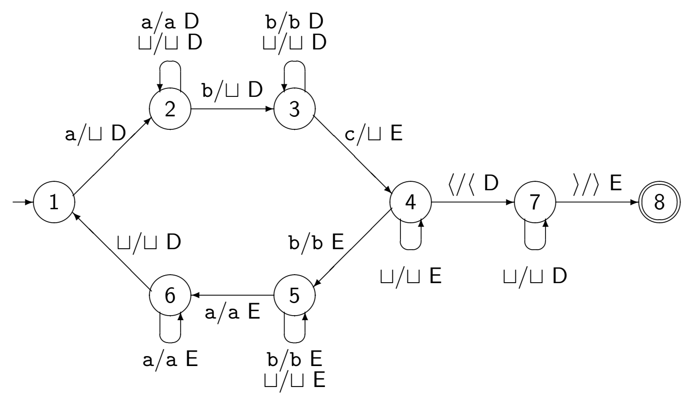
Autômato linearmente limitado

Um autômato linearmente limitado (ALL) é uma MT não determinística $M = (E, \Sigma, \Gamma, \langle, \rangle, \sqcup, \delta, i, F)$ em que:

- $\rangle$ é um símbolo especial de Γ que não pode ser escrito na fita;
- a configuração inicial é $[i, \langle \underline{w} \rangle]$; e
- se $\delta(e, \rangle)$ é definida, $\delta(e, \rangle) = [e', \rangle, \mathrm{E}]$ para algum $e' \in E$.

Exemplo de autômato linearmente limitado

Diagrama de estados de um ALL para $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n > 0\}$:



Gramáticas sensíveis ao contexto

Gramática sensível ao contexto

Uma gramática sensível ao contexto (GSC) é uma gramática (V, Σ, R, P), em que cada regra tem a forma:

- $x \to y$, $x, y \in (V \cup \Sigma)^+$ e $|x| \leq |y|$.

Exemplo:

GSC que gera $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 1\}$:

$$P \to \mathtt{a}PB\mathtt{c} \mid \mathtt{abc}$$
$$\mathtt{c}B \to B\mathtt{c}$$
$$\mathtt{b}B \to \mathtt{bb}$$

Linguagens sensíveis ao contexto

Linguagem sensível ao contexto

Uma linguagem é dita ser uma linguagem sensível ao contexto (LLC) se existe uma GSC que a gere.

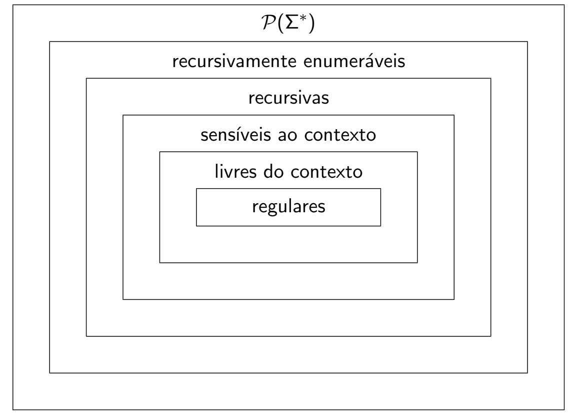
1. Toda LSC é reconhecida por algum ALL.
2. Se M é um ALL, então $L(M) - \{\lambda\}$ é LSC.

LLCs e LSCs

Desprezando o caso em que a linguagem contém λ, a classe das LLCs está propriamente contida na classe das LSCs.

1. Toda LLC que não contenha λ pode ser definida por meio de uma GLC sem regras λ. E toda GLC sem regras λ é uma GSC. Assim, toda LLC sem a palavra λ é uma LSC.
2. Existe *LSC* que não é *LLC*. Por exemplo, existe uma GSC para a linguagem $\{a^n b^n c^n \mid n \geq 1\}$, mas não existe GLC para essa mesma linguagem.

Espaço das linguagens em $\mathcal{P}(\Sigma^*)$



Propriedades das LREs e das Linguagens Recursivas

A classe das linguagens recursivas é fechada sob:

- união;
- interseção;
- complementação.

A classe das LREs é fechada sob:

- união;
- interseção.

A classe das LREs não é fechada sob complementação!

Existem linguagens que não são LREs

1. Seja R uma linguagem sobre Σ cujas palavras representam todas as MTs.
2. 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.
3. O conjunto de todas as linguagens de alfabeto Σ, $\mathcal{P}(\Sigma^*)$, não é enumerável.
4. 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 $\mathcal{P}(\Sigma^*)$ para R). Logo, existem mais linguagens do que MTs.

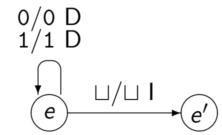
Um teorema importante

Se L e $\overline{L}$ são LREs, então L é recursiva.

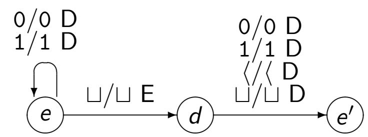
Em particular, segue-se a contrapositiva:

- Se L é LRE e L não é recursiva, então $\overline{L}$ não é LRE.

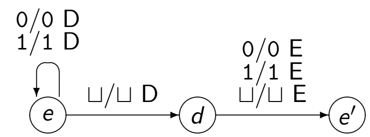
Máquina com cabeçote imóvel



(a) Trecho de MT com imobilidade



(b) Trecho de MT-padrão I



(c) Trecho de MT-padrão II

Retornar