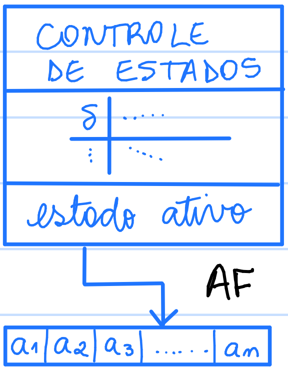
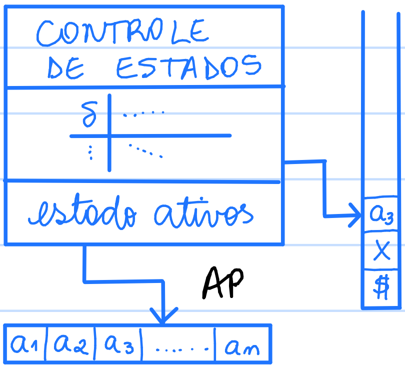
Máquinas de Turing

→ Como AFs e APs, são dispositivos reconhecedores.

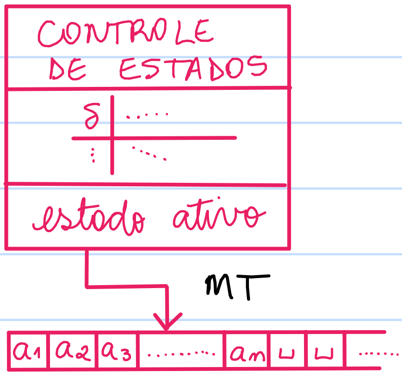
→ não são muito mais elaborados, mas o suficiente para que novas classes de linguagens apareçam.



Ling. Regulares
a^n, $a^n b^m$



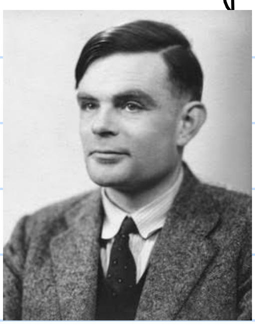
Ling. Livres de Contexto
$a^n b^n$, $a^n b^n c^m$



Ling. Decidíveis, Turing-Reconhecíveis
$a^n b^n c^n$

Máquinas de Turing

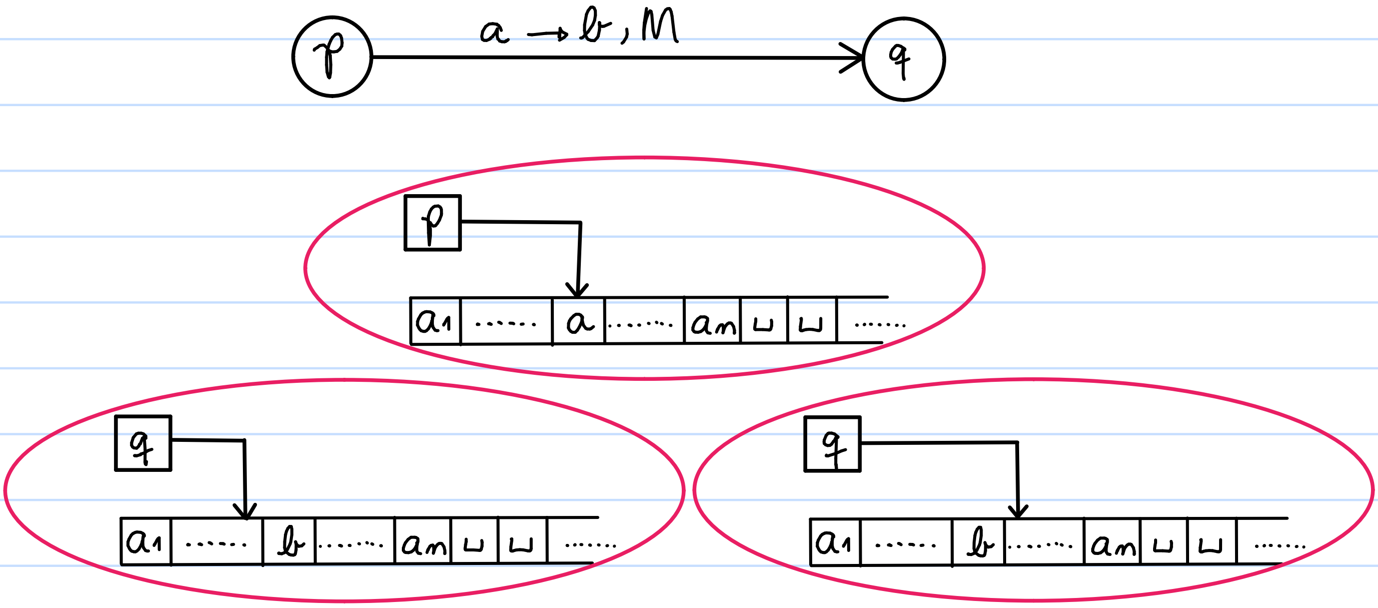
→ Foram propostas por Alan Turing em 1936.



↳ Aos seus 24 anos!
↳ Confiram sua história!
↳ Considerado pai da teoria da computação.

→ Assim como AFs e APs, são modelos simples de computação
↳ Mas são equivalentes em poder aos computadores de propósito geral.

Transições



Exemplo

→ Como reconhecer $\{a^n b^n c^n \mid n \geq 1\}$?

Inicialmente a fita contém apenas a cadeia e todo o resto está em branco.

a	a	a	a	b	b	b	b	c	c	c	c	⊔	⊔	.....

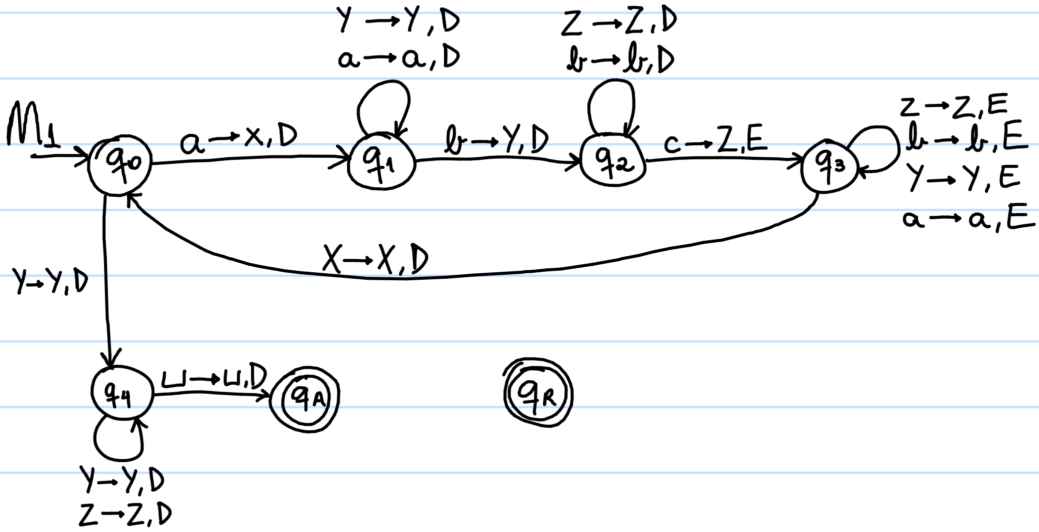
a	a	a	b	b	c	c	a	a	b	⊔	⊔	⊔	⊔	.....

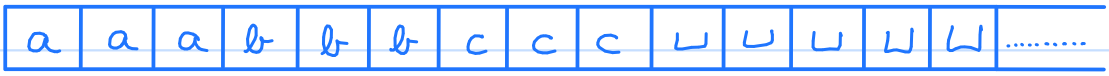
a	a	b	b	c	c	a	a	b	b	⊔	⊔	⊔	⊔	.....

① Percorra a fita marcando, para cada a, um b e um c.
Se não for possível, rejeite.

② Se todos os símbolos forem marcados, aceite.
Caso contrário, rejeite.

Representação gráfica





Máquinas de Turing

DEFINIÇÃO: Uma máquina de Turing (MT) é uma 7-upla $(Q, \Sigma, \Gamma, \delta, q_0, q_{ACEITA}, q_{REJEITA})$ em que:

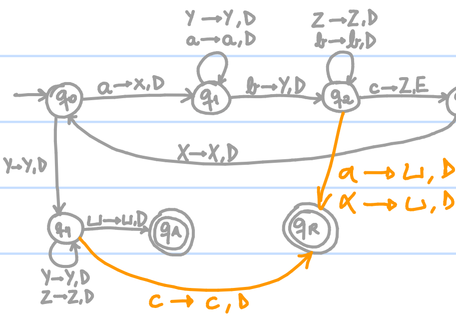
- Q é um conjunto finito de estados
- Σ é o alfabeto de entrada e $\sqcup \notin \Sigma$
- Γ é o alfabeto da fita, $\Sigma \subseteq \Gamma$ e $\sqcup \in \Gamma$
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{D, E\}$ é a função de transição.
- $q_0 \in Q$ é o estado inicial
- $q_{ACEITA} \in Q$ é o estado de aceitação
- $q_{REJEITA} \in Q$ é o estado de rejeição, $q_{REJEITA} \neq q_{ACEITA}$

Exemplo

$M_1 = (\{q_0, q_1, q_2, q_3, q_4, q_A, q_R\}, \{a, b, c\}, \{a, b, c, X, Y, Z, \sqcup\}, \delta, q_0, q_A, q_R)$,

sendo δ dada por:

δ	a	b	c	X	Y	Z	$\sqcup$
q_0	(q_1, X, D)	$(q_R, \sqcup, D)$	$(q_R, \sqcup, D)$	$(q_R, \sqcup, D)$	(q_4, Y, D)	$(q_R, \sqcup, D)$	$(q_R, \sqcup, D)$
q_1	(q_1, a, D)	(q_2, Y, D)	(q_R, c, D)	$(q_R, \sqcup, D)$	(q_1, Y, D)		
q_2		(q_2, b, D)	(q_3, Z, E)	$(q_R, \sqcup, D)$		(q_2, Z, D)	
q_3	(q_3, a, E)	(q_3, b, E)		$(q_R, \sqcup, D)$	(q_3, Y, E)	(q_3, Z, E)	
q_4					(q_4, Y, D)	(q_4, Z, D)	$(q_A, \sqcup, D)$
q_A							
q_R							

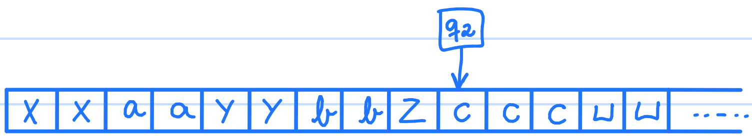


Configuração

DEFINIÇÃO: Uma configuração de uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R)$ é descrita por

$$\alpha q \beta$$

onde $\alpha, \beta \in \Gamma^*$, $\alpha\beta$ é o conteúdo da fita, $q \in Q$, e a cabeça está sobre o primeiro símbolo de β.



X X a a Y Y b b Z q_2 c c c

Sequências de configurações

DEFINIÇÃO: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R)$ uma MT e sejam $x, y, z \in \Gamma$, $\alpha, \beta \in \Gamma^*$ e $q_i, q_j \in Q$.
Se $\delta(q_i, y) = (q_j, z, E)$, então $\alpha x q_i y \beta \vdash \alpha q_j x z \beta$ é um movimento de M.
Se $\delta(q_i, y) = (q_j, z, D)$, então $\alpha x q_i y \beta \vdash \alpha x z q_j \beta$ é um movimento de M.

Caso especial: $q_i y \beta \vdash q_j z \beta$ para $\delta(q_i, y) = (q_j, z, E)$.

Usamos $\vdash^*$ para representar uma sequência de zero ou mais movimentos em M.

Configurações especiais

DEFINIÇÃO: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R)$ uma MT e $w \in \Sigma^*$.
Dizemos que $q_0 w$ é a configuração inicial
Dizemos que $\alpha q_A \beta$ é uma configuração de aceitação, com $\alpha, \beta \in \Gamma^*$.
Dizemos que $\alpha q_R \beta$ é uma configuração de rejeição, com $\alpha, \beta \in \Gamma^*$.

Configurações de aceitação e rejeição são chamadas de configurações de parada.

Computação

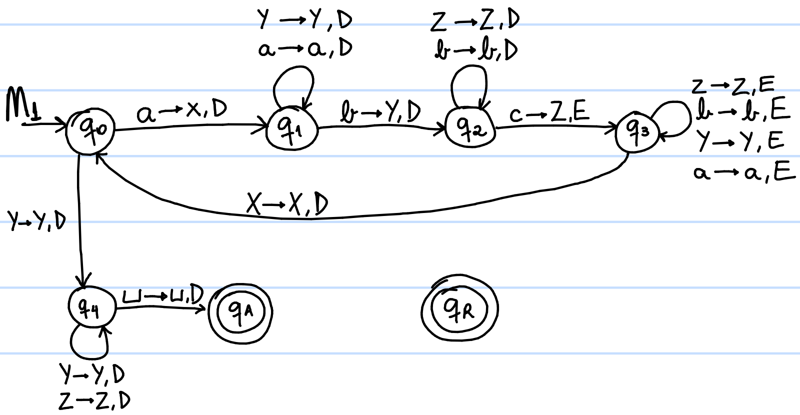
DEFINIÇÃO: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R)$ uma MT e $w \in \Sigma^*$.
Dizemos que M aceita w se

$$q_0 w \vdash^* \alpha q_A \beta$$

Dizemos que M decide w se

$$q_0 w \vdash^* \alpha q_A \beta \quad \text{ou} \quad q_0 w \vdash^* \alpha q_R \beta$$

Exemplo



$q_0 aabbac \vdash Xq_1abbac$
$\vdash Xaq_1bbac \vdash XaYq_2bac$
$\vdash XaYbq_2ac \vdash XaYb\sqcup q_R c$

$q_0 aabbcc \vdash Xq_1abbcc \vdash Xaq_1bbcc \vdash XaYq_2bcc$
$\vdash XaYbq_2cc \vdash XaYq_3bZc \vdash Xaq_3YbZc$
$\vdash Xq_3aYbZc \vdash q_3XaYbZc \vdash Xq_0aYbZc$
$\vdash XXq_1YbZc \vdash XXYq_1bZc \vdash XXYYq_2Zc$
$\vdash XXYYZq_2c \vdash XXYYq_3ZZ \vdash XXYq_3YZZ$
$\vdash XXq_3YYZZ \vdash Xq_3XYYZZ \vdash XXq_0YYZZ$
$\vdash XXYq_4YZZ \vdash XXYYq_4ZZ \vdash XXYYZq_4Z$
$\vdash XXYYZZq_4 \vdash XXYYZZq_A$

Com grandes poderes vêm grandes responsabilidades

→ Se a máquina não entrar em um estado de aceitação ou de rejeição, então ela entrou em loop e simplesmente não para.

DEFINIÇÃO: Seja $L \subseteq \Sigma^*$. Dizemos que uma MT M *decide* L se para toda $w \in \Sigma^*$, M decide w.

DEFINIÇÃO: Seja $L \subseteq \Sigma^*$. Dizemos que uma MT M *reconhece* L se $L = \{w \in \Sigma^* : M \text{ aceita } w\}$.

Novas classes de linguagem

DEFINIÇÃO: Uma linguagem é Turing-decidível (ou recursiva) se existe uma MT que a decide.

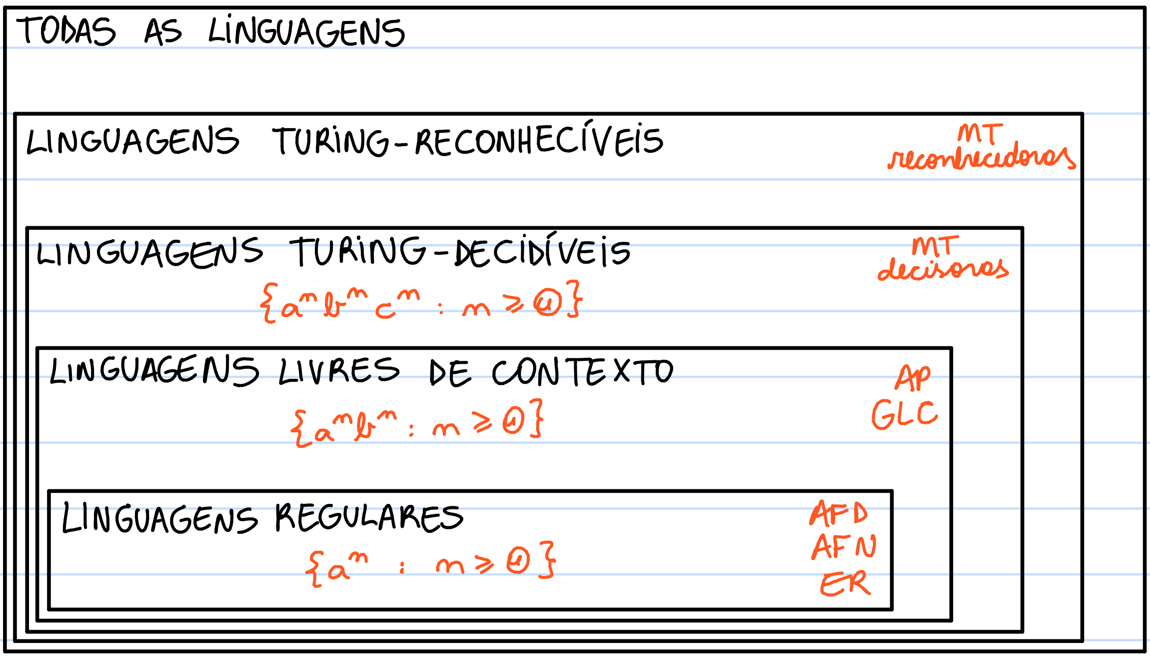
$\Rightarrow$ Dada $w \in \Sigma^*$, a MT decide se $w \in L$ ou $w \notin L$.

DEFINIÇÃO: Uma linguagem é Turing-reconhecível (ou recursivamente enumerável) se existe uma MT que a reconhece.

$\Rightarrow$ Dada $w \in \Sigma^*$, a MT aceita se $w \in L$ e rejeita ou entra em loop se $w \notin L$.

- $\{a^n b^n c^n : n \geq 0\}$ é decidível
- Toda decisora é reconhecedora

Relação entre as classes



Daria para usar uma MT para simular um AP?

- → Certamente a pilha não é um problema
- → Nem o não determinismo: podemos converter o AP em uma MT não determinística e esta para uma MT determ.
  - ↳ Porém, alguns ramos de computação do AP podem não parar, e a MT não seria decisora.

Mais exemplos

$A = \{ w\#w \; : \; w \in \{0,1\}^* \}$

0	1	0	0	1	#	0	1	0	0	1	⊔	⊔	⊔	.....

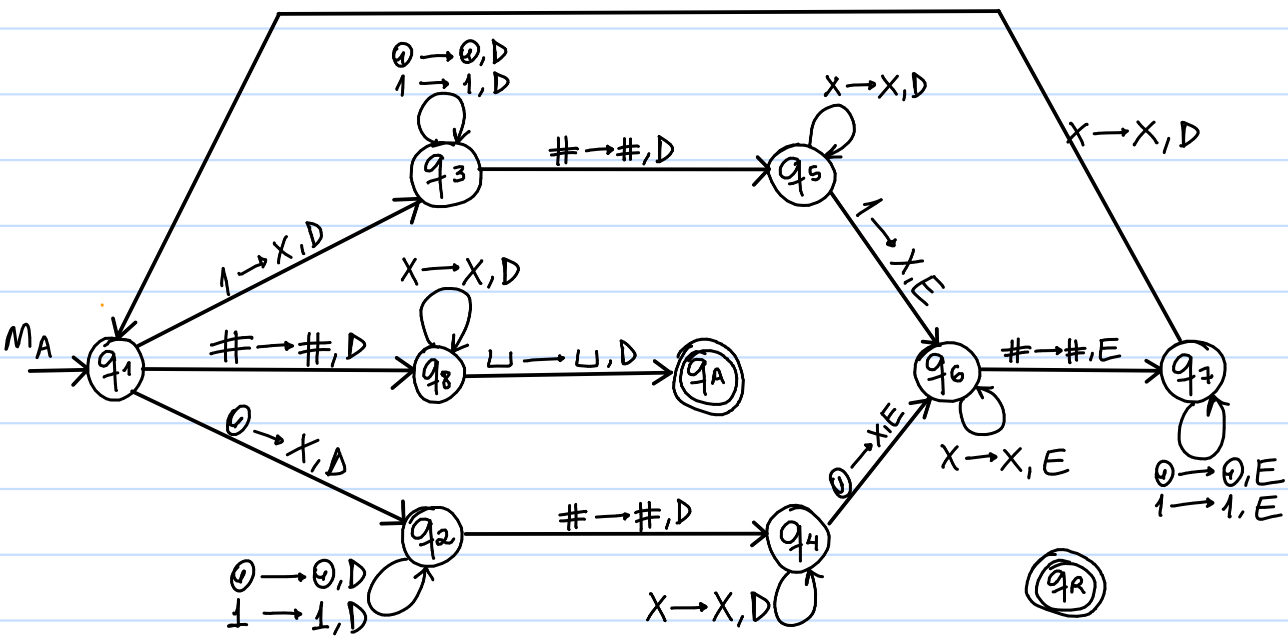
M_A = " Sobre a cadeia w de entrada:

(1) Faça um zigue-zague ao longo da fita, checando posições correspondentes de ambos os lados do símbolo # para verificar se elas contêm o mesmo símbolo. Se alguma não contém ou se # não for encontrado, rejeite.

Marque os símbolos à medida que eles são verificados.

(2) Quando todos os símbolos à esquerda do # forem marcados, verifique se há símbolos não marcados à direita do #.

Se houver, rejeite. Caso contrário, aceite. "



1	0	0	1	1	#	1	0	0	1	1	⊔	⊔	⊔	.....

Mais exemplos

$B = \{ 0^{2^n} : n \geq 0 \}$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	⊔	⊔	...

M_B = " Sobre a cadeia w de entrada:

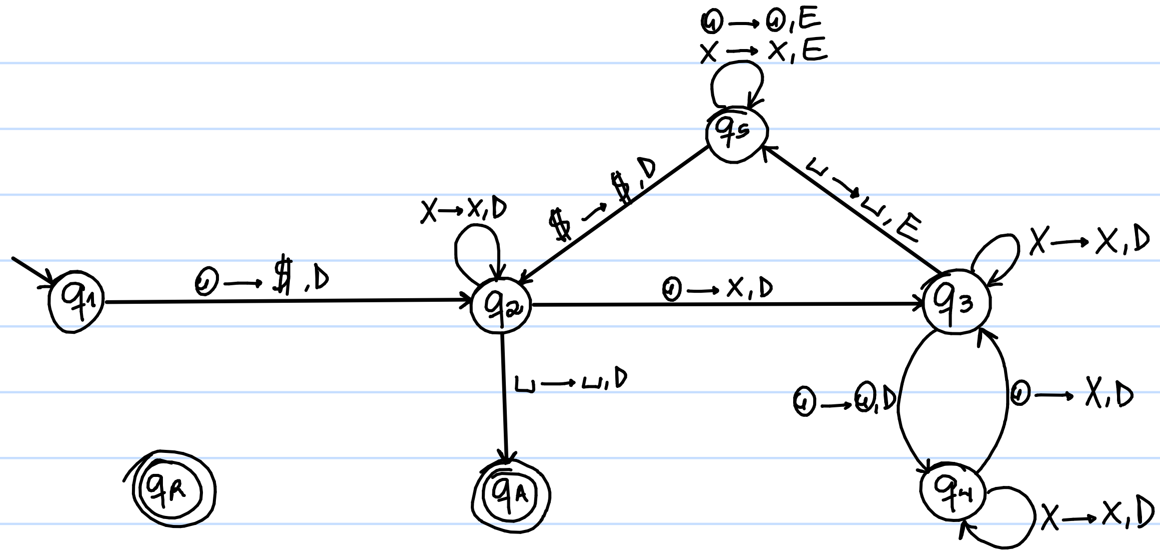
(1) Faça uma varredura da esquerda para a direita na fita, marcando um 0 não e outro sim.

(2) Se em (1) a fita tinha um único 0, aceite.

(3) Se em (1) a fita tinha mais de um 0 e o número de zeros é ímpar, rejeite.

(4) Retorne para a extremidade esquerda.

(5) Vá para (1). "



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	⊔	⊔	...

Mais exemplos

$C = \{ ww : w \in \{0,1\}^* \}$

0	1	1	0	1	0	1	1	0	1	⊔	⊔	⊔	.....

M_C = " Sobre a cadeia w de entrada:

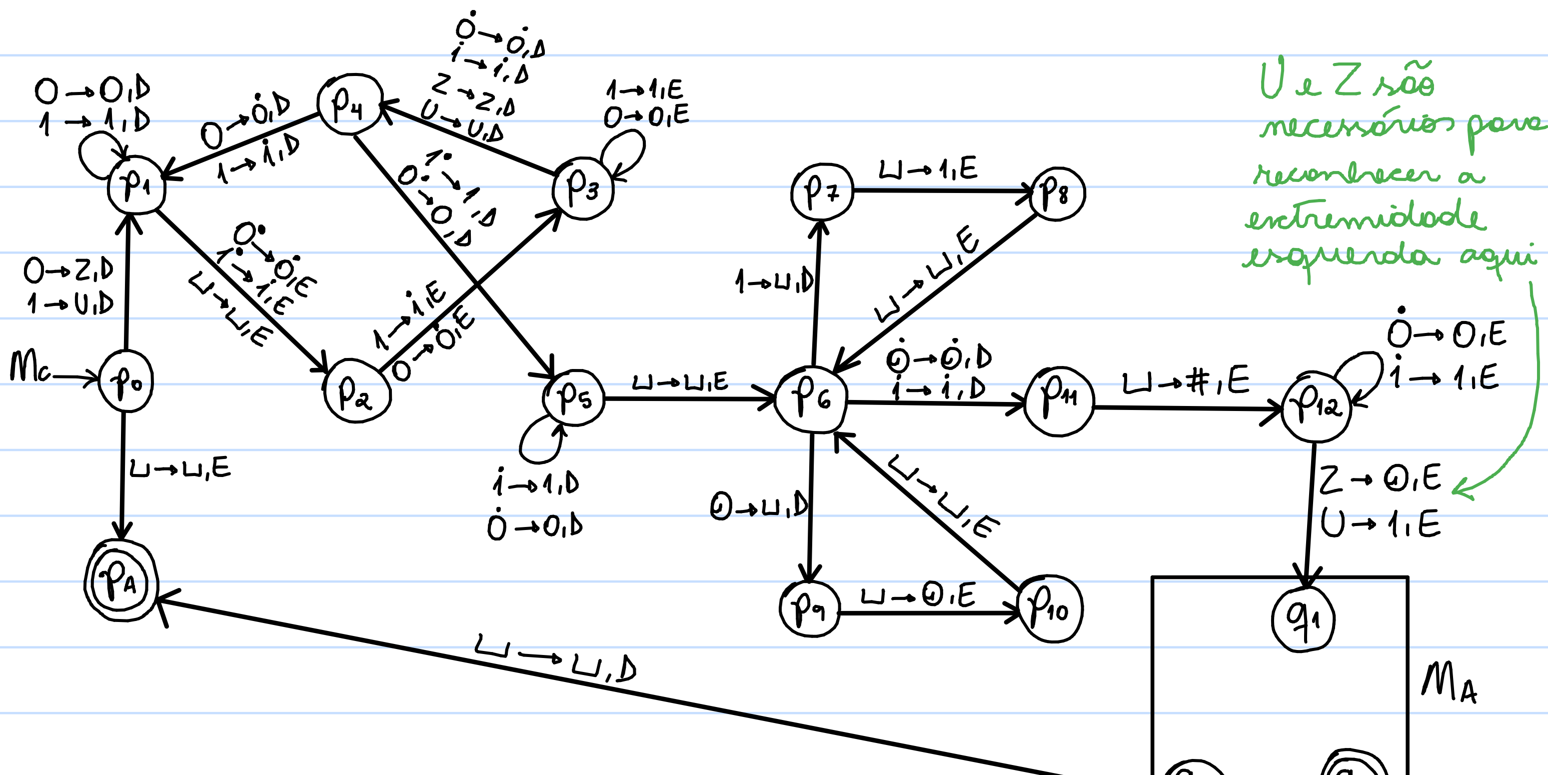
(1) Enquanto houver símbolos não marcados, vá e volte na fita marcando o primeiro e o último símbolos ainda não marcados. Se não for possível, rejeite. Se não houver símbolos, aceite.

(2) A partir da primeira posição da 2ª metade, desmarque os símbolos até chegar no primeiro ⊔.

(3) A partir do último símbolo e para cada símbolo não marcado, copie os símbolos uma posição para a direita.

(4) Escreva o símbolo # na primeira posição da 2ª metade e desmarque os símbolos até a extremidade esquerda.

(5) Execute M_A sobre a entrada. Aceite se M_A aceitar e rejeite se M_A rejeitar. "



0	1	0	1	1	0	1	0	1	1	⊔	⊔	.....

Variações de MTs

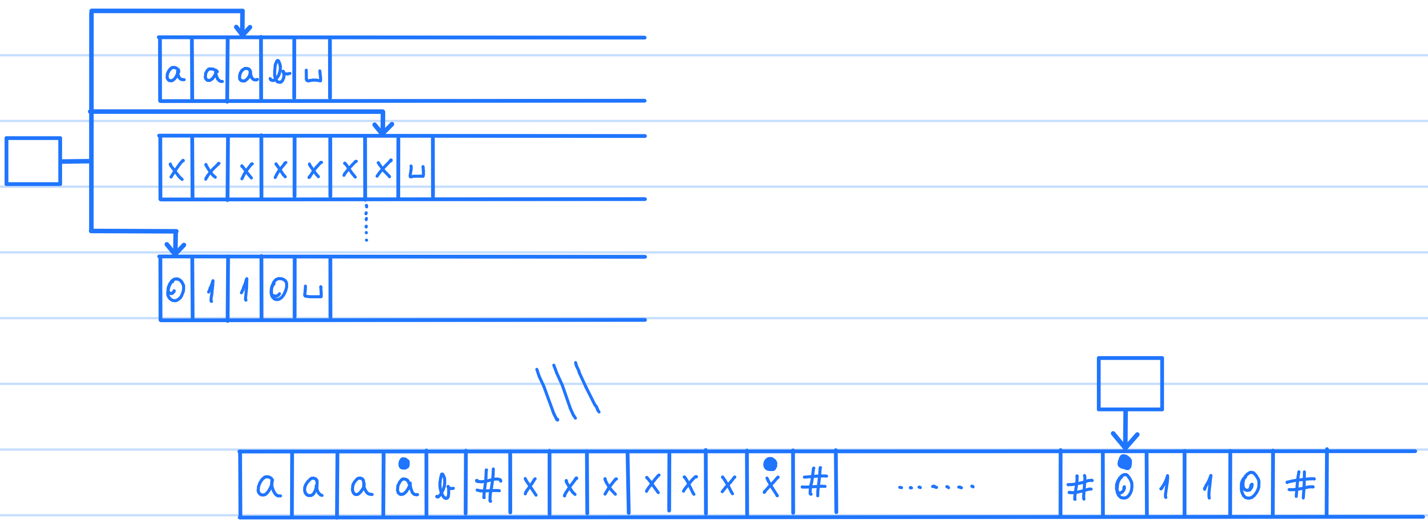
① MT que pode ficar parada

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D, P\}$$

Variações de MTs

② MT com $k \geq 2$ fitas

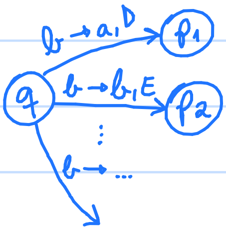
$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{E, D, P\}^k$$



Variações de MTs

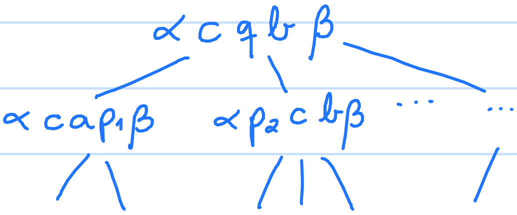
③ MT não determinística

$$\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{E, D\})$$



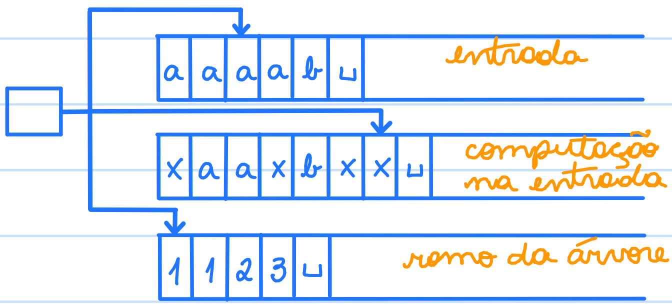
Se algum ramo chega em config. de aceitação, pare e aceite. Se todos os ramos chegam em config. de rejeição, pare e rejeite. Caso contrário, a computação é infinita.

Computação não linear



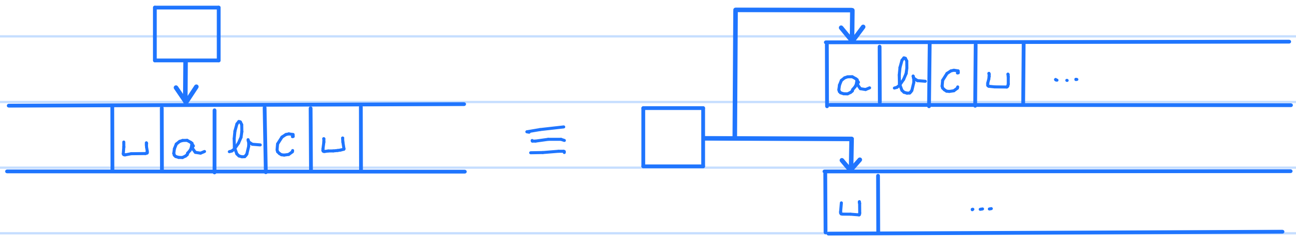
Vamos fazer a MT det. percorrer a árvore de configurações, mas não em uma busca em profundidade.

Seja 4 a maior ramificação possível da MT não det.



Variações de MTs

④ MT com fita ilimitada dos dois lados



Variações de MTs

⑤ Enumerador
"MT com impressora"

Começa com fita em branco e computa, imprimindo cadeias.
As cadeias impressas fazem parte da linguagem reconhecida.

Por que tanta equivalência?

→ Esses modelos têm as características essenciais das MTs:
- Acesso irrestrito a memória ilimitada

→ E satisfazem requisitos razoáveis
- Como fazer quantidade finita de trabalho em um único passo

→ Portanto, mesmo que possamos imaginar muitos modelos computacionais diferentes, a classe de algoritmos que eles descrevem será a mesma.