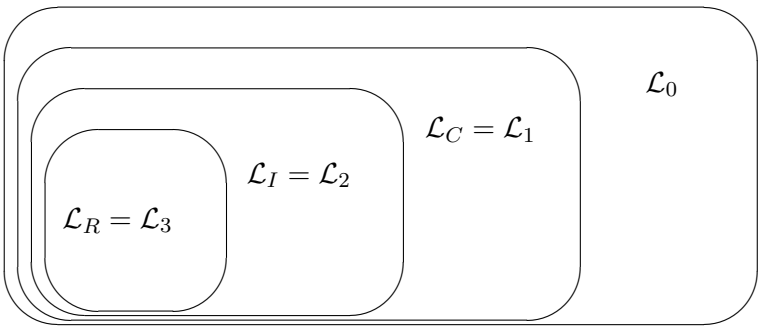
# Llenguatges formals i teoria de la computació

- Què hi ha més enllà dels llenguatges incontextuals?
- Quina diferència hi ha entre llenguatges de tipus 0 i 1?
- Com són els autòmats acceptors d'aquests llenguatges?



## Història de les Màquines de Turing (MT)

- Es va concebir com a formalisme matemàtic per tal de respresentar còmputs.
- A.M. Turing volia demostrar que certs problemes (algorísmics) no tenen solució (problema de la decisió de Hilbert, 1900).
- En 1936 ho va poder demostrar gràcies a la màquina que va inventar.

**Tesi de Church-Turing:** el poder computacional de la MT és equivalent al de qualsevol sistema computacional possible.

# Altres treballs relacionats

**1931** K. Gödel va demostrar que hi havia certs teoremes (relativament simples) que no es poden demostrar.

**1934** A. Church va proposar una forma de representar computacions ($\lambda$-càlcul).

**1936** S. Kleene va proposar un altre formalisme basat en funcions recursives.

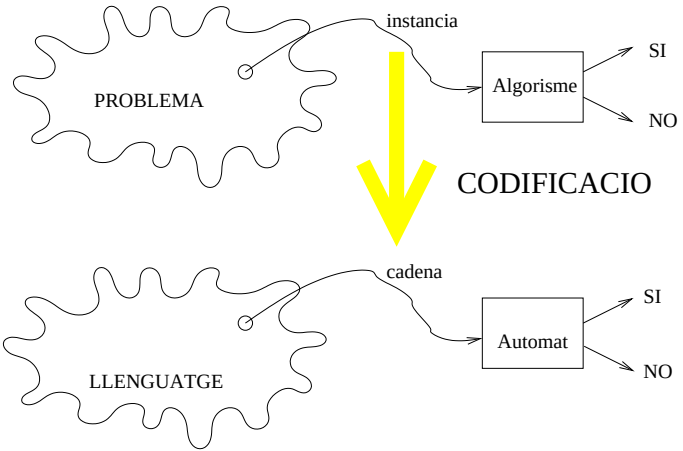
**1946** E. Post va estudiar un problema combinatori per al qual va demostrar que no existia solució.

**1956** N. Chomsky publica el seu treball sobre gramàtiques.

**1963** Es proposen diferents models formals més pareguts als computadors "reals".

# Relació entre llenguatges i problemes

La solució (algorísmica) d'un **problema** consisteix en la obtenció d'un **mètode** que compute la solució correcta per a qualsevol **instància** del problema.



## Relació entre llenguatges i problemes

Donada una codificació (efectiva) de les instàncies d'un problema (de decisió) les cadenes que corresponen a instàncies per a les quals la resposta ha de ser SI, formen un llenguatge sobre un determinat alfabet.

La (no) existència d'un algorisme per a un problema i la (no) existència d'un autòmat per a un llenguatge estan directament relacionats.

Els autòmats (en general) es poden veure coma acceptors de llenguatges o com a models de computació.

## Computabilitat

**Efectivament computable:** un problema que es pot resoldre d'alguna manera metòdica a partir d'una descripció finita.

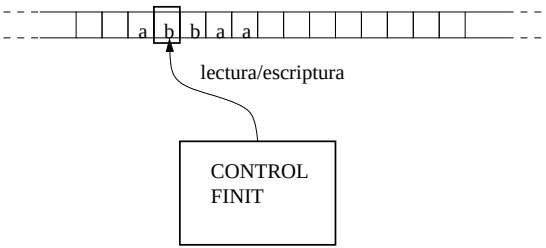
(qualsevol tipus de computació en què es puga pensar)

**$\mathcal{M}$-computable:** un problema que es pot resoldre fent servir un model (formal) de computació $\mathcal{M}$.

En particular, parlarem de problemes Turing-computables.

# Màquina de Turing. Definició

Una cinta d'entrada (doblement) infinita. Un capçal de lectura/escriptura. Un control finit que governa el moviment (dreta/esquerra) del capçal i el que escriu.




L'entrada és una cadena de símbols.

L'eixida és qualsevol cosa que quede escrita en la cinta (es pot produir una computacio infinita!).

# Exemple

MT que computa si el nombre de símbols $a$ és parell.

| | a | b |
|---|---|---|
| parell | esborrar, canviar d'estat i anar a la dreta | esborrar i anar a la dreta |
| senar | esborrar, canviar d'estat i anar a la dreta | esborrar i anar a la dreta |

$(p,\underline{a}ababa) \vdash (s,\underline{a}baba) \vdash (p,\underline{b}aba) \vdash (p,\underline{a}ba) \vdash (s,\underline{b}a) \vdash (s,\underline{a}) \vdash (p,\varepsilon)$

# MT. Definició formal

$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \triangle, F \rangle$

- $Q$ és un conjunt (finit) d'estats.
- $\Sigma$ és l'alfabet d'entrada.
- $\Gamma$ és l'alfabet de cinta ($\Sigma \subset \Gamma$).
- $q_0 \in Q$ és l'estat inicial.
- $\triangle \in \Gamma$ ($\triangle \notin \Sigma$) és un símbol especial anomenat **símbol blanc**.
- $F \subseteq Q$ és el conjunt d'estats finals.
- $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ és una funció parcial que rep el nom de **funció de transició** de la MT.

# MT. Definició formal

Inicialment el capçal apunta al primer símbol de la cadena d'entrada (sobre $\Sigma$).

La configuració d'una MT ve donada pel contingut de la cinta, la posició del capçal i l'estat.

La computació s'atura quan no hi ha transició definida a partir de la configuració actual.

Un estat s'anomena de parada si no té cap transició definida.

# MT. Definicions

**Configuració (descripció instantània):** en el cas de la MT ve donada únivocament per l'estat actual i el *contingut* de la cinta $(q, \alpha\underline{x}\beta) \equiv \alpha q x \beta$.

**Configuració de parada:** si no hi ha cap transició que es puga aplicar $\alpha q x \beta \dashv$.

**Configuració d'acceptació:** configuració de parada on l'estat actual és final $\alpha q x \beta \odot$.

**Moviment:** canvi de una configuració a la següent $\alpha q x \beta \vdash_M \alpha x p \beta$.

**Computació:** seqüència de configuracions des de la inicial fins a una de parada (final o no) $\alpha q x \beta \vdash_M^\star \alpha' q' \beta' \dashv$.

**Computació infinita:** seqüència de configuracions que mai arriba a una configuració de parada. $qw \vdash_M^\star \infty$

# La MT de l'exemple

- $Q = \{q_0, q_1, q_2\}$.
- $\Sigma = \{a, b\}$.
- $\Gamma = \Sigma \cup \{\triangle\}$.
- estat inicial $q_0$.
- $F = \{q_2\}$.
- $\delta(q_0, a) = (q_1, \triangle, \rightarrow)$ $\quad$ $\delta(q_1, a) = (q_0, \triangle, \rightarrow)$
- $\delta(q_0, b) = (q_0, \triangle, \rightarrow)$ $\quad$ $\delta(q_1, b) = (q_1, \triangle, \rightarrow)$
- $\delta(q_0, \triangle) = (q_2, \triangle, \leftarrow)$

# La MT com a acceptor de llenguatges

$M_1 = \{Q, \Sigma, \Gamma, \delta, q_0, \triangle, F\}$
on $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, $\Gamma = \Sigma \cup \{X, Y, \triangle\}$ i $F = \{q_4\}$.

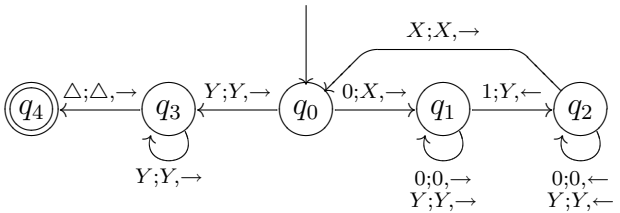
$\delta(q_0, 0) = (q_1, X, \rightarrow)$ $\delta(q_1, 0) = (q_1, 0, \rightarrow)$ $\delta(q_1, Y) = (q_1, Y, \rightarrow)$
$\delta(q_1, 1) = (q_2, Y, \leftarrow)$ $\delta(q_2, 0) = (q_2, 0, \leftarrow)$ $\delta(q_2, Y) = (q_2, Y, \leftarrow)$
$\delta(q_2, X) = (q_0, X, \rightarrow)$ $\delta(q_0, Y) = (q_3, Y, \rightarrow)$ $\delta(q_3, Y) = (q_3, Y, \rightarrow)$
$\delta(q_3, \triangle) = (q_4, \triangle, \rightarrow)$

Taula de transició:

| | 0 | 1 | $X$ | $Y$ | $\triangle$ |
|---|---|---|---|---|---|
| $q_0$ | $(q_1, X, \rightarrow)$ | | | $(q_3, Y, \rightarrow)$ | |
| $q_1$ | $(q_1, 0, \rightarrow)$ | $(q_2, Y, \leftarrow)$ | | $(q_1, Y, \rightarrow)$ | |
| $q_2$ | $(q_2, 0, \leftarrow)$ | | $(q_0, X, \rightarrow)$ | $(q_2, Y, \leftarrow)$ | |
| $q_3$ | | | | $(q_3, Y, \rightarrow)$ | $(q_4, \triangle, \rightarrow)$ |
| $\textcircled{\textcircled{q_4}}$ | | | | | |

# La MT com a acceptor de llenguatges

Diagrama de transició:




Computació:

$$
\begin{array}{c}
q_0 0011 \vdash Xq_1 011 \vdash X0q_1 11 \vdash Xq_2 0Y1 \vdash \\
\vdash q_2 X0Y1 \vdash Xq_0 0Y1 \vdash XXq_1 Y1 \vdash XXYq_1 1 \vdash \\
\vdash XXq_2 YY \vdash Xq_2 XYY \vdash XXq_0 YY \vdash XXYq_3 Y \vdash \\
\vdash XXYYq_3 \vdash XXYY \triangle q_4 \odot
\end{array}
$$

# Llenguatges associats a MT

Llenguatge acceptat per una MT:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash_M^\star \alpha_1 p \alpha_2 \odot, \quad p \in F, \alpha_1, \alpha_2 \in \Gamma^*\}$$

Diem que $M$ **accepta** $L$ si i només si

$$q_0 w \vdash_M^\star \alpha q \beta \odot \text{ per a tota cadena } w \text{ de } L$$

Diem que $M$ **decideix** $L$ si i només si

a) $q_0 w \vdash_M^\star \alpha q \beta \odot$ per a tota cadena $w$ de $L$

b) $q_0 w \vdash_M^\star \alpha' q' \beta' \dashv$ per a tota cadena $w$ de $\overline{L}$

Diem que $M$ **semidecideix** $L$ si l'accepta però no el decideix.

# La MT com a model de computació

Funció calculada per una MT:

$$f_M(w) = \begin{cases} x & \text{si } \exists x \in \Gamma^* : q_0 x \vdash_M^\star qx \dashv \\ \\ \text{indefinit} & \text{si no} \end{cases}$$

# Exemples de MT

- Es poden marcar símbols de la cadena d'entrada amb un nombre finit de marques diferents.
- Es poden desplaçar cadenes a la dreta o l'esquerra.
- Es pot comprovar la igualtat de cadenes en la cinta
- Es pot "comptar" el nombre de símbols en cadenes.
- Es pot sumar, restar, multiplicar, dividir, ...

## Variants i extensions de la MT

- es pot definir una MT amb una cinta semi-infinita.
- la funció de transició es pot redefinir per contemplar la posibilitat que el capçal no es moga després de llegir un símbol.

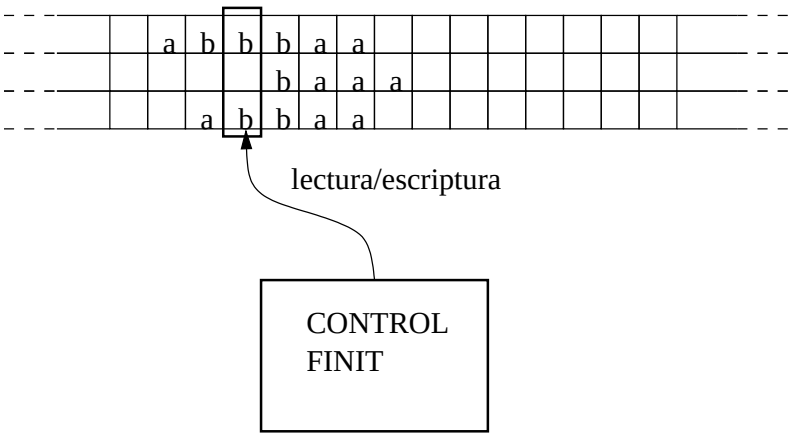
$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \bullet, \rightarrow\}$$

Les MT obtingudes són equivalents a la original

# La MT multipista

El capçal llegeix i escriu $k$ símbols a l'hora.

$$\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{\leftarrow, \bullet, \rightarrow\}$$



# MT multipista ($k = 2$) que accepta un llenguatge no incontextual

$L = \{wcw \in \{a, b, c\}^\star \mid w \in \{a, b\}^\star\}$ inicialment la cadena d'entrada es troba en la primera pista. Estratègia:

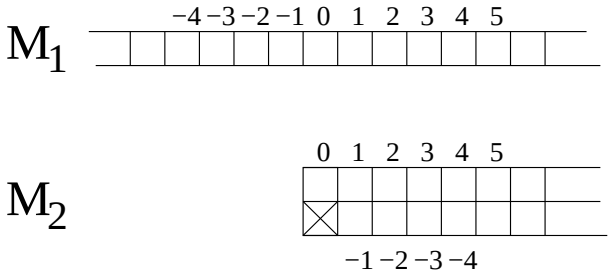
- un camí en la MT per cada símbol. $a$ i $b$.
- farem una marca en la segona cinta i entrarem en el camí corresponent al símbol actual.
- cercarem el mateix símbol (no marcat) despres de la $c$ (i farem una marca en la segona cinta)
- tornem al principi i cerquem el primer símbol no marcat

  ...

# MT amb cinta semi-infinita: equivalència

Donada una MT $M_1 = \langle Q, \Sigma, \Gamma, \delta, q_0, \triangle, F \rangle$, existeix una MT amb cinta semi-infinita i dues pistes que simula exactament $M_1$.

s'assigna una numeració a les caselles de $M_1$ i a les de la nova MT $M_2$:



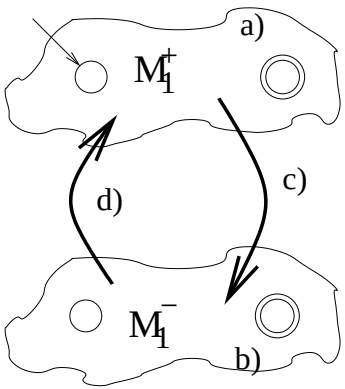

per cada $q \in Q$ s'introueixen dos estats, $q^+, q^-$.

$$M_2 = \langle Q^+ \cup Q^-, \Sigma', (\Gamma \cup \sharp)^2, \delta', q_0^+, \begin{pmatrix} \triangle \\ \triangle \end{pmatrix}, F^+ \cup F^- \rangle$$

# MT amb cinta semi-infinita: equivalència

Es repeteix la MT $M_1$ com a $M_1^+$ i $M_1^-$ que simulen la primera mentre el capçal es troba en caselles positives o negatives, respectivament.

Caldrà a més a més afegir transicions entre les dues per tal de simular el pas de la part positiva a la negativa i al revés.



# MT amb cinta semi-infinita: equivalència

Per cada transició de la forma
$\delta(p, \sigma) = (q, \tau, D)$, $D \in \{\leftarrow, \bullet, \rightarrow\}$, afegim

a) $\delta'(p^+, \begin{pmatrix}\sigma\\ \alpha\end{pmatrix}) = (q^+, \begin{pmatrix}\tau\\ \alpha\end{pmatrix}, D)$, $\forall \alpha \in \Gamma$

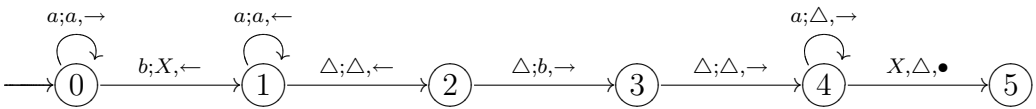
$\alpha = \sharp \wedge D \in \{\bullet, \rightarrow\}$

b) $\delta'(p^-, \begin{pmatrix}\alpha\\ \sigma\end{pmatrix}) = (q^-, \begin{pmatrix}\alpha\\ \tau\end{pmatrix}, D^{-1})$, $\forall \alpha \in \Gamma$

c) $\delta'(p^+, \begin{pmatrix}\sigma\\ \sharp\end{pmatrix}) = (q^-, \begin{pmatrix}\tau\\ \sharp\end{pmatrix}, \rightarrow)$, si $D = \leftarrow$.

d) $\delta'(p^-, \begin{pmatrix}\sigma\\ \sharp\end{pmatrix}) = (q^+, \begin{pmatrix}\tau\\ \sharp\end{pmatrix}, \rightarrow)$, si $D = \rightarrow$.

# MT amb cinta semi-infinita: equivalència




$0abba \vdash a0bba \vdash$

$\vdash 1aXba \vdash 1 \triangle aXba \vdash$

$\vdash 2 \triangle \triangle aXba \vdash$

$\vdash b3 \triangle aXba \vdash$

$\vdash b \triangle 4aXba \vdash b \triangle \triangle 4Xba \vdash$

$\vdash b \triangle \triangle 5 \triangle ba$

$0^+ \begin{pmatrix} a\ b\ b\ a \\ \sharp \triangle \triangle \triangle \end{pmatrix} \vdash \begin{pmatrix} a \\ \sharp \end{pmatrix} 0^+ \begin{pmatrix} b\ b\ a \\ \triangle \triangle \triangle \end{pmatrix} \vdash$

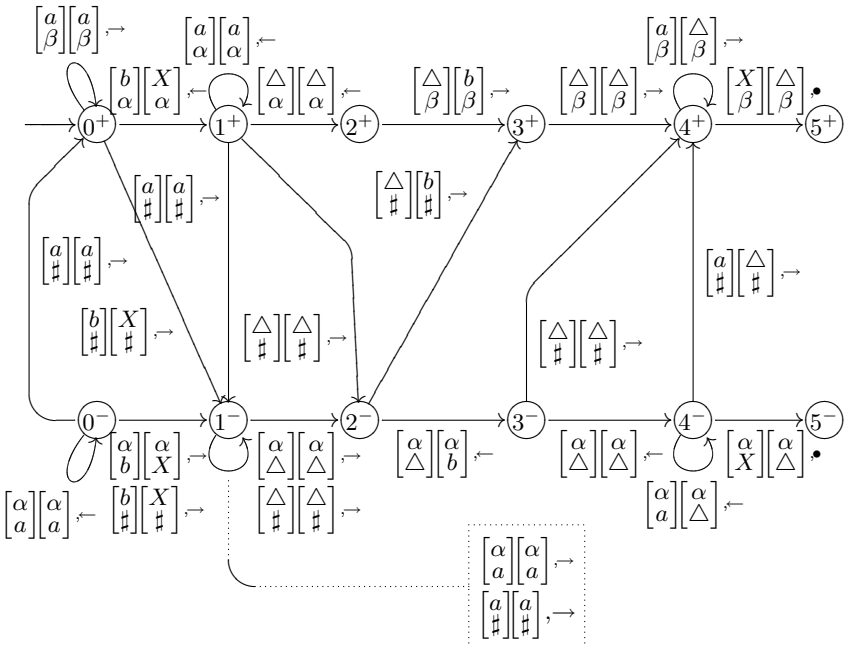
$\vdash 1^+ \begin{pmatrix} aX\ b\ a \\ \sharp \triangle \triangle \triangle \end{pmatrix} \vdash \begin{pmatrix} a \\ \sharp \end{pmatrix} 1^- \begin{pmatrix} X\ b\ a \\ \triangle \triangle \triangle \end{pmatrix} \vdash$

$\vdash \begin{pmatrix} aX \\ \sharp \triangle \end{pmatrix} 2^- \begin{pmatrix} b\ a \\ \triangle \triangle \end{pmatrix} \vdash \begin{pmatrix} a \\ \sharp \end{pmatrix} 3^- \begin{pmatrix} Xb\,a \\ \triangle b \triangle \end{pmatrix} \vdash$

$\vdash 4^- \begin{pmatrix} aXb\,a \\ \sharp \triangle b \triangle \end{pmatrix} \vdash \begin{pmatrix} \triangle \\ \sharp \end{pmatrix} 4^+ \begin{pmatrix} Xb\,a \\ \triangle b \triangle \end{pmatrix} \vdash$

$\vdash \begin{pmatrix} \triangle \\ \sharp \end{pmatrix} 5^+ \begin{pmatrix} \triangle b\,a \\ \triangle b \triangle \end{pmatrix}$

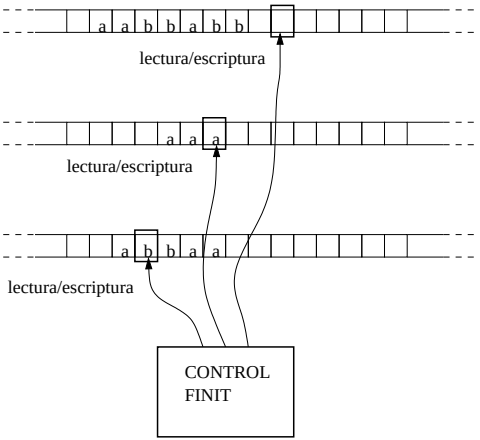
# MT amb cinta semi-infinita: equivalència



# La MT multicinta

Hi ha $k$ capçals que operen sobre $k$ cintes.

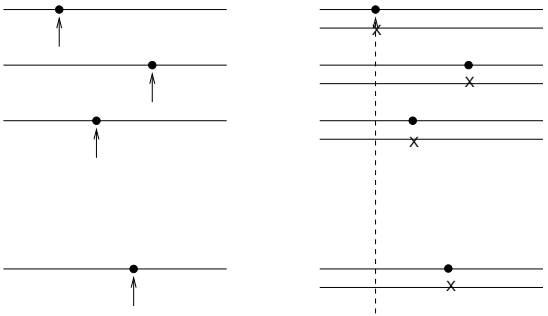
$$\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{\leftarrow, \bullet, \rightarrow\}^k$$




també es pot definir la MT multicapçal (cinta única).

# La MT multicinta

Qualsevol MT amb $k$ cintes pot ser simulada per una MT amd $2k$ pistes



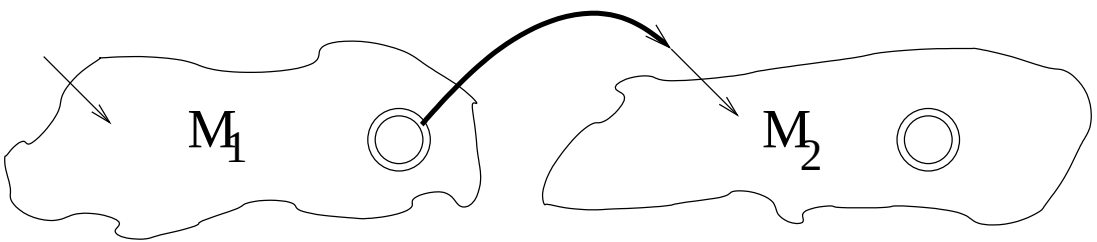
Cada transició de la MT multicinta seria simulada per una seqüència de transicions.
La MT multipista faria servir les pistes parelles per tal de marcar les posicions dels capçals.

Des d'alguna posició tal que totes les marques estiguen a la dreta, la MT buscaria les marques i reproduiria l'acció de cada capçal en les cintes senars i tornaria a una posició a l'esquerra de les marques.

# MT Modulars

És possible dissenyar MT mols simples que realitzen tasques bàsiques de forma que es puguen després combinar per crear-ne de més complexes.

La forma bàsica de combinar dues MT és mijantçant **composició**.



# MT Modulars

$$M_1 = \langle Q_1, \Sigma, \Gamma_1, \delta_1, q_1, \triangle, F_1 \rangle$$
$$M_2 = \langle Q_2, \Sigma, \Gamma_2, \delta_2, q_2, \triangle, F_2 \rangle$$

$$M_1 M_2 = \langle Q_1 \cup Q_2, \Sigma, \Gamma_1 \cup \Gamma_2, \delta, q_1, \triangle, F_2 \rangle$$

$$\delta(q,\sigma) = \begin{cases} \delta_1(q,\sigma) \text{ si} & q \in Q_1 \wedge \delta_1(q, \sigma \neq (p,\tau,D) \\ & \forall p \in F_1 \\ \delta_2(q,\sigma) \text{ si} & q \in Q_2 \\ \\ (q_2,\tau,D) \text{ si } q \in & Q_1 \wedge \exists p \in F_1 \ : \\ & \delta_1(q,\sigma = (p,\tau,D) \end{cases}$$

(cal suposar algunes coses sobre les MT originals)

# MT Modulars

La idea és que s'apliquen MT de forma seqüencial una darrere de l'altra. Cada MT comença en el seu estat inicial però amb el contingut de cinta i posició de capçal que ha deixat l'anterior.
Es pot pensar també en composicions condicionals i en bucles (una mateixa MT s'aplica mentre es complesca certa condició)

composició: $M_1 M_2$ o $M_1 \longrightarrow M_2$

composició condicional: $M_1 \xrightarrow{a} M_2$

bucle: $M_1 \overset{a}{\circlearrowleft}$

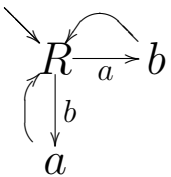
# MT simples (MT modulars)

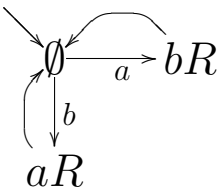
| | |
|---|---|
| $\emptyset$ | MT que no fa res. |
| $R$, $L$ | MT que desplacen el capçal a dreta o esquerra, resp. |
| $a$ | MT que escriu $a \in \Gamma$ en la posició actual sense canviar-la. |
| $R_X$, $L_X$ | MT que cerquen algun símbol de $X \subseteq \Gamma$ a dreta o esquerra (sense comptar el símbol actual) |
| $\circledcirc$ | MT que accepta |
| $S_R, S_L$ | MT que desplaça una posició a dreta o esquerra la cadena que va des de la posició actual fins al primer blanc a la dreta ($S_L$ sobreescriu el símbol actual). |

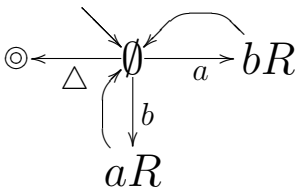
# MT simples (MT modulars)

$R_{\triangle}RaL^2_{\triangle}$ se situa al final de la cadena d'entrada, es desplaça una posició, escriu una $a$ i torna al principi.

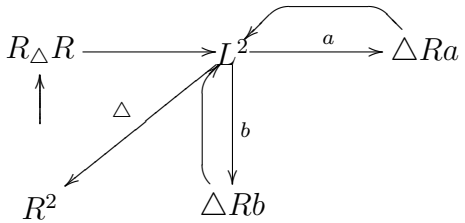
$R_{\{a,b\}}1$ cerca una $a$ o una $b$ i la canvia per 1 (pot no parar).

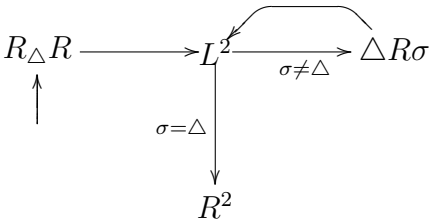


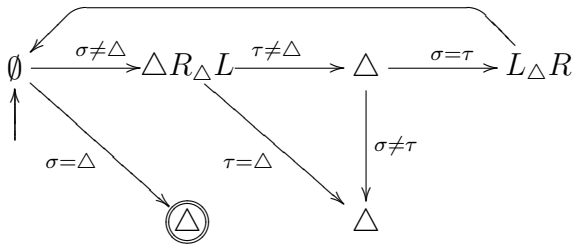



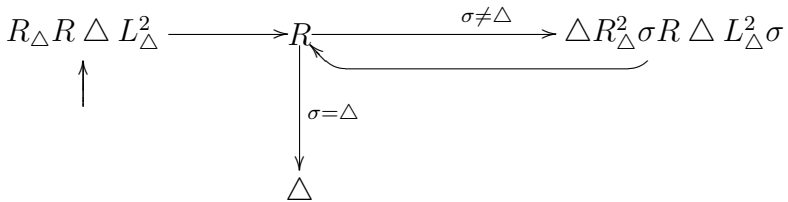
# MT modular: exemples

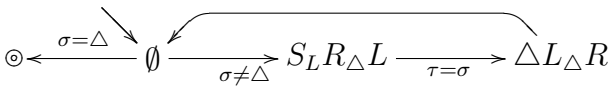


# MT modular: exemples



## La MT no determinista

Es pot definir la MT de forma no determinista de la mateixa manera que es fa en altre tipus d'autòmats.

$$\delta : Q \times \Gamma \longrightarrow \mathcal{P}(Q \times \Gamma \times \{\leftarrow, \bullet, \rightarrow\})$$

El comportament de la MTND pot variar entre **infinites** possibilitats:

$$\longrightarrow \underset{\circlearrowleft}{R} \longrightarrow \emptyset$$

Ara bé, en un instant concret, la MTND només pot optar entre un nombre **finit** d'opcions ($|\Gamma| \cdot |Q| \cdot 3$ com a màxim).

És possible simular qualsevol MTND mijantçant una MT(D)?

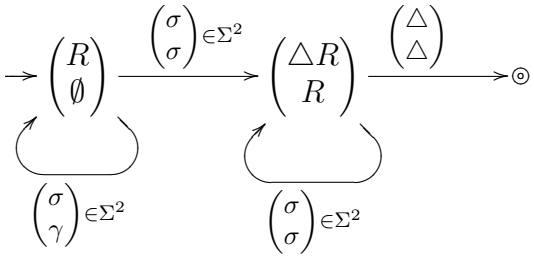
## La MT no determinista

Simulació d'una MTND:

- en cada pas de qualsevol computació la MTND ha elegit entre un nombre (finit) $r$ d'opcions.
- qualsevol de les (infinites) computacions de la MTND a partir d'una configuració inicial queda unívocament determinada per una seqüència (infinita) de decisions (una cadena de $\{s_0, \ldots s_{r-1}\}^\star$ ).
- cada un dels (infinits) comportaments de la MTND es pot simular amb una MT si se li dona la seqüència de decisions corresponent.

Caldrien almenys 3 cintes: una per la cadena d'entrada (només lectura), la segona per generar seqüències de decisions, i la tercera per fer la simulació.

# MTND que accepta $L = \{ww | w \in \Sigma^\star\}$




## MT multicapçal (2 capçals):

- un capçal es mou a la dreta i l'altre roman quiet.
- quan els dos capçals llegeixen el mateix símbol, es pot canviar d'estat o no (indeterminisme).
- si canvia d'estat, els dos capçals es mouen cap a la dreta mentre llegesquen el mateix símbol
- el primer capçal (més a la dreta) va esborrant.
- si els dos llegeixen $\triangle$ al mateix temps la cadena és acceptada.

## Classes de llenguatges relacionades amb MT

Un llenguatge $L$ és **recursivament enumerable** (r.e.) si és acceptat per alguna màquina de Turing. És a dir, si

$\exists\ M = (Q, \Sigma, \Gamma, \delta, q_0, \triangle, F)$ tal que $L = L(M)$

$\mathcal{L}_{re} = \{L \subseteq \Sigma^\star | \exists M : L = L(M)\}$

Un llenguatge $L$ és **recursiu** (rec) si és acceptat per alguna màquina de Turing que arriba a una configuració de parada per a tota cadena d'entrada.

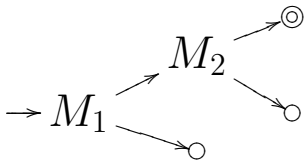
$\mathcal{L}_{rec} = \{L \subseteq \Sigma^\star \ \mid \ \exists M : L = L(M) \quad \wedge$

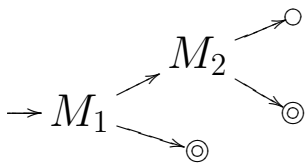
$q_0 w \vdash_M^\star \alpha_1 q \alpha_2 \dashv \ \forall w \in \Sigma^\star$, on $\alpha_1, \alpha_2 \in \Gamma^\star$ i $q \in Q$

| |
|---|
| recursiu $\equiv$ decidible, no recursiu $\equiv$ indecidible |
| indecidible $\wedge$ r.e. $\equiv$ semidecidible |

# Propietats

- $\mathcal{L}_{rec} \subseteq \mathcal{L}_{re}$ (trivial. $\mathcal{L}_{rec} = \mathcal{L}_{re}$?)
- $\mathcal{L}_I \subseteq \mathcal{L}_{rec}$ (les MT poden simular AP)
- $L \in \mathcal{L}_{rec} \Rightarrow \overline{L} \in \mathcal{L}_{rec}$ (trivial)
- $L_1 \in \mathcal{L}_{rec}, L_2 \in \mathcal{L}_{rec} \Rightarrow L_1 \cap L_2 \in \mathcal{L}_{rec},\ L_1 \cup L_2 \in \mathcal{L}_{rec}$



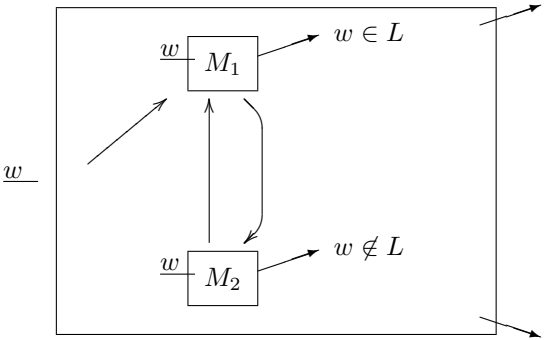




- $\mathcal{L}_I \subset \mathcal{L}_{rec}$

# Propietats

- $\mathcal{L}_{re}$ és tancada respecte de la intersecció i la unió:


- $L \in \mathcal{L}_{re}, \overline{L} \in \mathcal{L}_{re} \Rightarrow L \in \mathcal{L}_{rec}$



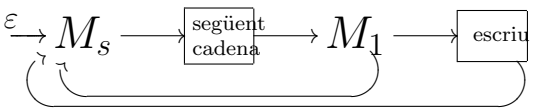
## MT que enumeren llenguatges

Un llenguatge $L$ és **enumerat per una MT** si existeix alguna MT que, en alguna cinta dessignada com d'eixida, escriu **totes** les cadenes de $L$ separades per algun símbol especial. Inicialment totes les cintes estan en blanc. S'escriu $L = G(M)$.

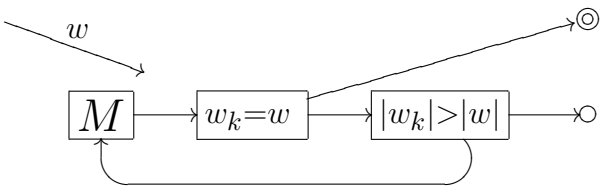
Si la MT és capaç de generar les cadenes de $L$ en ordre **canònic** (lèxicogràfic) es diu que $L$ és **canònicament enumerat per una MT**. $L = G_c(M)$.

# MT que enumeren llenguatges

$\boxed{L \in \mathcal{L}_{rec} \implies \exists M : L = G_c(M)}$ ($M_1$ decideix $L$ i $M_s$ calcula la cadena següent.)

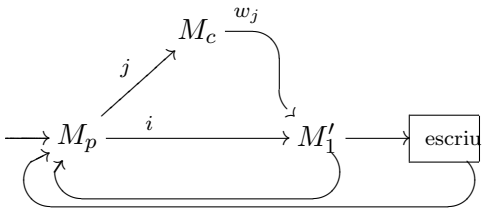



$\boxed{L \in \mathcal{L}_{rec} \impliedby \exists M : L = G_c(M)}$

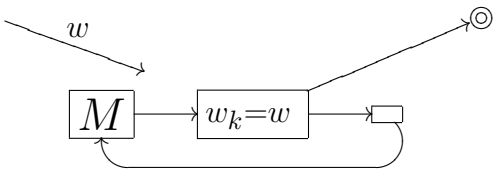


# MT que enumeren llenguatges

$\boxed{L \in \mathcal{L}_{re} \implies \exists M : L = G(M)}$ ($M_1$ accepta $L$, $M_1'$ és com $M_1$ però para després de $i$ moviments, $M_c$ genera la cadena $j$-èssima)




$\boxed{L \in \mathcal{L}_{re} \impliedby \exists M : L = G(M)}$



# Relació amb la jerarquia de Chomsky

Si $L \in \mathcal{L}_0 \Leftrightarrow L \in \mathcal{L}_{re}$

($\Rightarrow$) $L \in \mathcal{L}_0 \Rightarrow \exists G : L = L(G)$.

Donada $G$ es pot simular el procés de generació de cadenes de forma ordenada.

- Primer es generen les cadenes que pot generar $G$ mijantçant 1 derivació (nombre finit). A continuació les que genera amb 2 derivacions (n. finit)

  ⋮

- A continuació les que genera amb $i$ derivacions (són un nombre finit)

  .

La MT així construïda enumeraria $L$ per tant $L \in \mathcal{L}_{re}$.

## Demostració alternativa:

Es construeix $M' : L = L(M')$: s'apliquen produccions de forma no determinista sobre formes sentencials en una cinta i es compara amb la cadena d'entrada.

# Relació amb la jerarquia de Chomsky

## $(\Leftarrow)$ $L \in \mathcal{L}_{re} \Rightarrow \exists M : L = L(M)$.

Es pot construir una gramàtica de forma que les seues formes sentencials corresponguen a configuracions de la MT $M$ en el format $xqy$.

Donada $M$ (que puguem suposar semiinfinita, sense moviments nuls i que no deixa blancs intercalats):

$$\alpha apb\beta \vdash_M \begin{cases} \alpha ab'q\beta & \underline{\texttt{si}} \quad \delta(p,b) = (q,b',\rightarrow) \quad \equiv \quad b'q \rightarrow pb \\ \alpha qab'\beta & \underline{\texttt{si}} \quad \delta(p,b) = (q,b',\leftarrow) \quad \equiv \quad qab' \rightarrow apb \end{cases}$$

$$\begin{array}{l} S \rightarrow S \triangle \,|\sharp A \\ A \rightarrow aA|Aa|q \qquad \forall q \in F, \forall a \in \Gamma - \{\triangle\} \\ \sharp q_0 \rightarrow \varepsilon \\ \triangle \rightarrow \varepsilon \end{array}$$

Aleshores

$$S \quad \stackrel{\star}{\Rightarrow} \quad \sharp xqy\triangle^\star \quad \stackrel{\star}{\Rightarrow} \quad \sharp q_0 w\triangle^\star \quad \stackrel{\star}{\Rightarrow} \quad w$$

## MT i els contextuals

$\boxed{L \in \mathcal{L}_C \Rightarrow L \in \mathcal{L}_{rec}}$

Si $L \in \mathcal{L}_C \Rightarrow \exists G : L = L(G)$ i les produccions de $G$ són de la forma

$$\alpha \rightarrow \beta : |\beta| \geq |\alpha|$$

En una derivació de $G$, les formes sentencials mai decreixen.

Aquest fet es pot utilitzar per a que una eventual simulació amb MT puga parar també per a cadenes de $\overline{L}$ quan la forma sentencial siga més gran que la cadena.

$\boxed{L \in \mathcal{L}_{rec} \not\Rightarrow L \in \mathcal{L}_C}$

Només ;-) cal trobar un llenguatge que ens servesca com a contraexemple

## Un llenguatge recursiu i no contextual

Siga $S \rightarrow \beta_0, \alpha_1 \rightarrow \beta_1, \ldots, \alpha_k \rightarrow \beta_k$, una gramàtica contextual sobre $\{a, b\}$. Podem codificar aquesta gramàtica de la següent manera:

$$\begin{aligned} a =& 00 \\ b =& 001 \\ A_i =& 01^i \\ \rightarrow =& 0011 \\ , =& 00111 \end{aligned}$$

Tota gramàtica contextual queda unívocament descrita per una cadena sobre $\{0, 1\}$. Aquesta codificació ens dóna a més a més una forma d'ordenar totes les gramàtiques contextuals $\{G_i\}_{i=1}^{\infty}$

# Un llenguatge recursiu i no contextual

Podem ara definir el llenguatge

$$L_c = \{w_i | w_i \notin L(G_i)\}$$

$L_c$ és recursiu. Exercici!

Si $L_c$ fóra contextual, aleshores $L_c = L(G_k)$.
Siga doncs $w_k$ la $k$-èssima cadena de $\{a, b\}^\star$. Hi ha dues possibilitats:

$$w_k \in L_c \implies w_k \notin L(G_k) = L_c \qquad \text{contradicció!}$$
$$w_k \notin L_c = L(G_k) \implies w_k \in L_c \qquad \text{contradicció!}$$

$L_c$ és recursiu i no contextual!

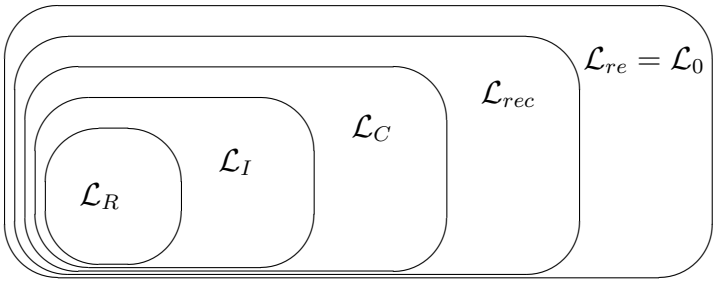
# Jerarquia de Chomsky estesa

Només falta aclarir si és cert o no que $\mathcal{L}_{rec} = \mathcal{L}_{re}$.

Definirem després un llenguatge r.e. que no és recursiu.

La jerarquia de Chomsky afegint-hi les noves classes de llenguatges queda com

$$\mathcal{L}_R \subset \mathcal{L}_I \subset \mathcal{L}_C \subset \mathcal{L}_{rec} \subset \mathcal{L}_{re} = \mathcal{L}_0$$



# La MT Universal

La MT universal, $M_U$, és una MT que pren com a entrada la codificació d'una altra MT, $M$ i una cadena $w$ i és capaç de **simular** exactament el comportament de $M$ quan se li presenta $w$ com a cadena d'entrada.
Com que una MT té una descripció **finita** (estats, transicions, alfabet de cinta, moviments) es pot trobar una codificació efectiva d'aquesta, $\langle M \rangle$.

$$\underline{\langle M, w \rangle} \boxed{M_u} \underline{\langle c \rangle}$$

La cadena d'entrada (per a $M$) es pot donar codificada en el mateix alfabet que $\langle M \rangle$. $\langle c \rangle$ és una codificació de la configuració de parada a què arribaria $M$ amb $w$.

La simulació no para quan $M$ no para amb $w$!

# La MT Universal

Es pot mostrar que la MT universal es pot construir com a una MT de tres cintes que pren com a entrada una codificació d'una MT qualsevol sobre l'alfabet $\{0, 1\}$.

Com que qualsevol cadena (sobre un alfabet finit) també es pot codificar com a una cadena de $\{0, 1\}^\star$, es pot reduir tota la simulació a una MT amb alfabet d'entrada $\{0, 1\}$.

()

# Un llenguatge no r.e.

Siga $M_j$ una MT amb alfabet d'entrada $\Sigma$ tal que la seua codificació correspon a l'enter $j$ escrit en binari, $\langle M_j \rangle = \langle j \rangle$.
(A tota MT li correspon un enter)

Siga $w_i$ la $i$-èssima cadena de $\Sigma^\star$ en ordre canònic.

$$L_d = \{w_i \in \Sigma^\star | w_i \notin L(M_i)\}$$

$\boxed{L_d \text{ no és r.e.}}$ Per reducció a l'absurd:
Suposem que $\exists M : L_d = L(M)$ i suposem que a eixa MT li correspon l'enter $k$, $M = M_k$. Aleshores, $L_d = L(M_k)$ (*).
Considerem la cadena $w_k$ tenim que

- si $w_k \in L_d \overset{def.L_d}{\Longrightarrow} w_k \notin L(M_k) \overset{(*)}{=} L_d$. contradicció.
- si $w_k \notin L_d \overset{(*)}{=} L(M_k) \overset{def.L_d}{\Longrightarrow} w_k \in L_d$. contradicció.

Per tant, $L_d$ **no** és r.e.

# ... i un llenguatge r.e. que no és recursiu

Siga $L_r = \{w_i \in \Sigma^\star | w_i \in L(M_i)\}$

$L_r$ és r.e.

Donada $w$,

1. calcular $\langle i \rangle : w = w_i$.
2. si $\langle i \rangle$ no es correspon a cap MT, NO ACCEPTAR.
3. si no, Simular $M_i$ amb entrada $w_i$ (MTU).
4. si la simulació para i accepta, ACCEPTAR.

$L_r$ no és recursiu

Si ho fóra, com que $L_r = \overline{L_d}$, implicaria que $L_d$ també és recursiu. contradicció.