

# Calculabilité - Décidabilité (ICC)

## Cours n°8

Stef Graillat

Sorbonne Université



- **Machine de Turing** : une machine de Turing est un modèle de calculateur qui a la même puissance de calcul que les ordinateurs actuels et que les autres définitions mathématiques de calculateurs. Une MT consiste en un **automate fini**, un **ruban** infini composé de cellules et d'une **tête de lecture** qui est au dessus d'une cellule. Un mouvement de la MT dépend de l'état de l'automate et du symbole contenu dans la cellule sous la tête de lecture. Lors d'un mouvement, la MT change d'état, écrit dans la cellule sous la tête de lecture et déplace la tête de lecture vers la gauche ou vers la droite

# Exemple de machines de Turing reconnaissant $\{0^n1^n : n \geq 1\}$

## Idée :

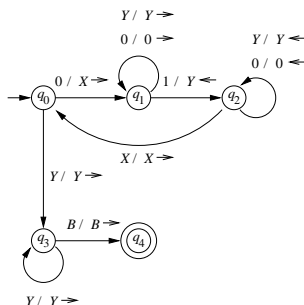
- On écrit le mot sur le ruban et alternativement on change les 0 en X et les 1 en Y
- On commence la lecture à gauche de l'entrée et on remplace un 0 par X et on bouge à droite tant qu'on lit des 0 et des Y jusqu'à ce qu'on lise un 1
- On change un 1 en Y et on bouge à gauche sur des Y et des 0 jusqu'à ce qu'on lise un X
- À ce niveau, on regarde si on a un 0 immédiatement à droite. Si oui on le change en X et on repète le procédé en changeant 1 en Y

À la fin, si la MT n'arrive pas à faire un mouvement alors on s'arrête et on n'accepte pas le mot

Si à la fin, on a changé alternativement tous les 0 en X et tous les 1 en Y alors l'entrée était de la forme  $0^n1^n$

# Exemple de machines de Turing reconnaissant $\{0^n 1^n : n \geq 1\}$

Soit la MT  $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$  où  $\delta$  est donné par le diagramme de transition



L'expression sur le ruban est de la forme  $\mathbf{X^*0^*Y^*1^*}$

# Exemple de machines de Turing reconnaissant $\{0^n 1^n : n \geq 1\}$

- $q_0$  est l'état initial et  $M$  revient en  $q_0$  à chaque fois qu'il revient sur le 0 restant le plus à gauche
  - si  $M$  est en  $q_0$  et qu'il lit un 0 alors il passe à l'état  $q_1$  et change 0 en  $X$  et se déplace à droite
  - une fois à l'état  $q_1$ ,  $M$  se déplace à droite tant qu'il lit des 0 et des  $Y$ .
  - s'il lit un  $X$  ou un  $B$ , il s'arrête
  - s'il lit un 1 en étant en  $q_1$ , il change 1 en  $Y$  et passe à l'état  $q_2$  et se déplace vers la gauche
  - en  $q_2$ , il se déplace vers la gauche tant qu'il lit des 0 et des  $Y$
  - quand  $M$  atteint le  $X$  le plus à droite, il retourne en  $q_0$  et se déplace vers la droite
- 2 cas sont alors possibles :

## Exemple de machines de Turing reconnaissant $\{0^n 1^n : n \geq 1\}$

- Si  $M$  lit un 0 alors on recommence le cycle précédent
- Si  $M$  lit un  $Y$  alors c'est qu'il a changé tous les 0 en  $X$ . Si tous les 1 ont été changés en  $Y$  alors l'entrée était de la forme  $0^n 1^n$  et  $M$  doit accepter. Par conséquent,  $M$  entre dans l'état  $q_3$  et se déplace à droite tant qu'il lit des  $Y$ . Si le premier symbole rencontré différent de  $Y$  est  $B$  alors  $M$  va à l'état  $q_4$  et accepte. Par contre, si  $M$  rencontre un 1 alors c'est qu'il y a trop de 1 et  $M$  s'arrête sans accepter. Si  $M$  rencontre un 0,  $M$  s'arrête aussi sans accepter.

# Exemple de machines de Turing reconnaissant $\{0^n 1^n : n \geq 1\}$

## Séquence avec comme entrée 0011

$$\begin{aligned} q_0 0011 \vdash X q_1 011 \vdash X 0 q_1 11 \vdash X q_2 0 Y 1 \vdash q_2 X 0 Y 1 \vdash \\ X q_0 0 Y 1 \vdash X X q_1 Y 1 \vdash X X Y q_1 1 \vdash X X q_2 Y Y \vdash X q_2 X Y Y \vdash \\ X X q_0 Y Y \vdash X X Y q_3 Y \vdash X X Y Y q_3 B \vdash X X Y Y B q_4 B \end{aligned}$$

## Séquence avec comme entrée 0010

$$\begin{aligned} q_0 0010 \vdash X q_1 010 \vdash X 0 q_1 10 \vdash X q_2 0 Y 0 \vdash q_2 X 0 Y 0 \vdash \\ X q_0 0 Y 0 \vdash X X q_1 Y 0 \vdash X X Y q_1 0 \vdash X X Y 0 q_1 B \end{aligned}$$

# Exemple de machines de Turing avec « sortie »

Soit la MT calcule  $m \ominus n = \max(m - n, 0)$

- on met  $0^m 10^n$  sur le ruban et à la fin on veut  $0^{m \ominus n}$  sur le ruban
- $M$  cherche le 0 le plus à gauche et le remplace par un  $B$
- $M$  cherche alors à droite pour trouver un 1
- après avoir trouvé un 1, il cherche à droite pour trouver un 0 et le remplace par un 1
- $M$  retourne à gauche en cherchant un 0 (il le fait en trouvant un  $B$  et en allant un cran à droite).
- on répète le procédé jusqu'à ce que



# Exemple de machines de Turing avec « sortie »

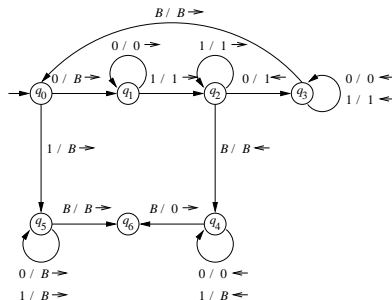
- on répète le procédé jusqu'à ce que
  - en cherchant un 0,  $M$  trouve un  $B$ . Alors les  $n$  0 dans  $0^m 10^n$  ont été changés en 1 et  $n + 1$  des  $m$  0 ont été changés en  $B$ .  $M$  remplace alors les  $n + 1$  1 par un 0 et  $n$   $B$  laissant  $m - n$  0 sur le ruban.
  - en commençant le cycle,  $M$  ne peut trouver un 0 à changer en  $B$  du fait que les  $m$  0 ont déjà été changés en  $B$ . Alors  $n \geq m$  et  $m \ominus n = 0$ .  $M$  remplace tous les 1 et 0 restant en  $B$

# Exemple de machines de Turing avec « sortie »

Soit la MT calcule  $m \ominus n = \max(m - n, 0)$  La table de transition est donnée par

	0	1	B
$\rightarrow q_0$	$(q_1, B, R)$	$(q_5, B, R)$	
$q_1$	$(q_1, 0, R)$	$(q_2, 1, R)$	
$q_2$	$(q_1, 1, L)$	$(q_2, 1, R)$	$(q_4, B, L)$
$q_3$	$(q_3, 0, L)$	$(q_3, 1, L)$	$(q_0, B, R)$
$q_4$	$(q_4, 0, L)$	$(q_4, B, L)$	$(q_6, 0, R)$
$q_5$	$(q_5, B, R)$	$(q_5, B, R)$	$(q_6, B, R)$
$\star q_6$			

Le diagramme de transition est donnée par



# Langage reconnu par une machine de Turing

Écriture du mot donné en entrée sur le ruban et  $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

## Définition 1

*Le langage accepté par la MT  $M$  noté  $L(M)$  est défini par*

$$L(M) = \{w \in \Sigma^* : q_0 w \vdash^* \alpha p \beta \text{ avec } p \in F\}$$

## Définition 2

*L'ensemble des langages acceptés par une machine de Turing est appelé ensemble des **langages récursivement énumérables**.*

**Autre mode d'acceptation** : arrêt de la machine si  $\delta(q, X)$  n'est pas défini.

On peut toujours supposer qu'une machine de Turing s'arrête si elle rentre dans un état acceptant.

Mais une machine de Turing peut ne pas s'arrêter même si elle n'accepte pas son entrée.

## Définition 3

*Un langage est **rékursif** s'il est reconnu par une machine de Turing qui s'arrête toujours que le mot soit accepté ou non.*

Cela correspond à la notion d'**algorithme**

- Mémorisation de données dans le contrôleur  $\leadsto$  le contrôleur est doté d'une capacité de mémorisation finie.
- Ruban à plusieurs pistes. L'alphabet  $\Gamma$  est alors constitué de  $n$ -uplets.
- Sous-routines : une tâche est associée à un sous-ensemble d'états

Ceci ne change en rien la puissance de calcul des MT

# Extensions de la machine de Turing simple – MT Multi-ruban

- L'entrée est copiée sur le première ruban.
- Toutes les autres cellules contiennent le symbole blanc.
- La machine est à l'état initial.
- Le contrôleur est situé à gauche du premier symbole de l'entrée

Un changement d'état :

- Les déplacements du contrôleur sur chacun des rubans sont indépendants les uns des autres.
- Sur chaque ruban, les changements de symboles sont indépendants les uns des autres.

## Théorème 1

*Tout langage accepté par une MT multi-ruban est récursivement énumérable.*

Nombre de pas, de transitions, effectuées par une MT  $M$  lorsque l'entrée est  $w$ .

Si  $M$  ne s'arrête pas, ce nombre est infini.

## Définition 4

*Complexité en temps :  $T(n)$  = le maximum des pas effectués par  $M$  pour tous les mots de longueur  $n$ .*

Le temps pris par une MT à un ruban pour simuler  $n$  changements dans une MT à  $k$  rubans est  $\mathcal{O}(n^2)$  (pour  $k$  fixé).

# Extensions de la machine de Turing simple – MT non-déterministe

Dans le cas non-déterministe,  $\delta(q, X)$  est un ensemble fini de triplets :

$$\{(q_1, Y_1, D_1), \dots, (q_k, Y_k, D_k)\}$$

## Théorème 2

*Soit  $M_N$  une MT non-déterministe. Alors il existe une MT déterministe  $M_D$  telle que  $L(M_N) = L(M_D)$ .*



Tout langage accepté par une Machine de Turing  $M_2$  est aussi accepté par une machine de Turing  $M_1$  soumise aux restrictions suivantes :

- Le contrôleur de  $M_1$  est toujours situé à gauche de sa position initiale.
- $M_1$  n'écrit jamais le symbole blanc

Pour la preuve considérer plusieurs pistes du ruban, la seconde piste décrivant ce qui se passe à gauche de la position initiale du contrôleur dans  $M_2$ .

## Exemple de langage non hors-contexte reconnu par une MT

Que se passe-t-il si on dote un AP de plusieurs piles ?

Une machine à  $k$  piles est un AP avec  $k$  piles.

Un changement de configuration est induit par :

- L'état de l'AP
- Le symbole d'entrée lu
- Le haut de chaque pile.

$$\delta(q, a, X_1, X_2, \dots, X_k) = (p, \gamma_1, \gamma_2, \dots, \gamma_k)$$

Acceptation par état final.

Marqueur de fin \$ pour l'entrée.

## Théorème 3

*Si un langage  $L$  est accepté par une MT alors il est accepté par un AP à deux piles.*

# Restrictions sur les MT – Machines à compteurs

Même structure qu'une machine multi-pile mais chaque pile est remplacée par un compteur. Les changements dépendent de l'état courant et de la nullité ou non des compteurs. Ils induisent un changement d'état et l'ajout ou la soustraction de 1 à chacun des compteurs (mais le résultat des soustractions ne peut pas être négatif).

Évidemment, tout langage accepté par une machine à compteur est récursivement énumérable.

Si on n'a qu'un seul compteur, le langage accepté est récursivement énumérable.

## Théorème 4

*Tout langage récursivement énumérable est accepté par une machine à trois compteurs.*

## Théorème 5

*Tout langage récursivement énumérable est accepté par une machine à deux compteurs.*

# Partie IV :

## Décidabilité – Indécidabilité

**But :** Prouver que le langage récursivement énumérable  $L_u$  constitué de  $(M, w)$  où

- $M$  est une MT encodée en binaire
- $w$  est un mot de  $\{0, 1\}^*$
- $M$  accepte  $w$ .

est **indécidable**

Si ce problème est indécidable sur des entrées binaires alors il le sera sûrement avec un alphabet quelconque

**Première étape :** coder une machine de Turing comme un mot constitué de 0 et 1 et exhiber un langage qui n'est pas récursivement énumérable

# Codage d'une machine de Turing

Énumération des mots binaires :  $w \rightsquigarrow 1w$  et  $1w$  est vu comme le codage binaire d'un entier

$$\varepsilon \rightsquigarrow 1 \rightsquigarrow 1$$

$$0 \rightsquigarrow 10 \rightsquigarrow 2$$

$$01 \rightsquigarrow 101 \rightsquigarrow 5$$

On peut donc parler du  $i$ -ième mot de  $\{0,1\}^*$  !

Codage d'une machine de Turing :  $M = (Q, \{0,1\}, \Gamma, \delta, q_1, B, F)$

- on suppose que les états sont  $q_1, q_2, \dots, q_r$  avec  $q_1$  l'état initial et  $q_2$  l'unique état final
- on suppose que les symboles du ruban sont  $X_1, X_2, \dots, X_s$  avec  $X_1 = 0$ ,  $X_2 = 1$  et  $X_3 = B$
- $L = D_1$  et  $R = D_2$

# Codage d'une machine de Turing (suite)

**Code des transitions** :  $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$  est codé par  $0^i 10^j 10^k 10^\ell 10^m$

**Code pour une MT** : on concatène les codes  $C_i$  pour les transitions en les séparant avec 11  $C_1 11 C_2 11 \dots 11 C_{n-1} 11 C_n$

**Exemple** :  $M = (\{q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, B\}, \delta, q_1, B, \{q_2\})$  avec  
 $\delta(q_1, 1) = (q_3, 0, R)$ ,  $\delta(q_3, 0) = (q_1, 1, R)$ ,  $\delta(q_3, 1) = (q_2, 0, R)$ ,  
 $\delta(q_3, B) = (q_1, 1, L)$

Code pour les transitions : 0100100010100

0001010100100

00010010010100

0001000100010010

Code pour  $M$  :

01001000101001100010101001001100010010010100110001000100010010

## Codage d'une machine de Turing (suite)

Étant donnée une machine de Turing  $M$  de code  $w_i$ , on peut lui associer l'entier  $i \rightsquigarrow M_i = i$ -ième MT

Beaucoup d'entiers ne correspondent à une MT (par exemple 11001)

Si le code  $w_i$  de  $M_i$  n'est pas valide alors on prend pour  $M_i$  la MT avec un seul état et pas de transition qui s'arrête immédiatement sur n'importe quelle entrée. Par conséquent  $L(M_i) = \emptyset$ .

Une paire  $(M, w)$  constituée d'une MT et d'un mot est encodé par : code de  $M111w$