Théorie des Langages - EISTI - ING 1

Yannick Le Nir

Ecole Internationale des Sciences du Traitement de l'Information

yannick.lenir@eisti.fr

Historique

Problèmes de Hilbert

- David Hilbert : 1862-1943
- Liste de 23 problèmes (Exposition universelle de 1900 à Paris)
- Problème numéro 10 : *"Trouver un algorithme déterminant si une équation diophantienne à des solutions."*

Historique

Machine de Turing

- Alan Matheson Turing : 1912-1954
- Rôle actif pour décrypter la machine Enigma pendant la seconde guerre mondiale
- Inventeur de la machine de Turing (1936)

Machine de Turing

Généralités

- Tout algorithme peut être traduit en un programme pour la machine de Turing
- Modèle théorique de l'ordinateur (réalisations physiques dès 1940).

Description

- Ruban infini : suite de cases portant chacune un élément d'un alphabet
- Semblable aux automates finis, sauf qu'elle peut lire, écrire et se déplacer sur le ruban
- Déplacement d'une seule case (droite ou gauche) à la fois
- Diagramme de transition pour modéliser son comportement

Machine de Turing

Historique

- En 1935, Turing essaye de résoudre la question posée par Hilbert, reformulée par :
  "Existe-t-il, au moins théoriquement, une méthode ou un processus moyen duquel toutes les questions mathématiques peuvent être décidées"
- *Thèse de Church-Turing :* "Aucune procédure de calcul ne peut être considérée comme un algorithme à moins qu'on puisse la représenter comme une machine de Turing"
- Thèse de Turing : *"Tout ce qui est calculable peut être calculé par une machine de Turing"*
- *Vrai si calculer signifie manipuler un nombre fini de symboles et produire une réponse après un nombre fini d'étapes*

Machine de Turing

Représentation

1	1	0	1	1	0	1						...

$\uparrow$

$\square$ Mécanisme de contrôle : nombre d'états fini

- Règles de transitions : (état initial,caractère lu,état final,caractère écrit,déplacement)
- $M(n)$: résultat en écriture

Machine de Turing

Fonctionnement

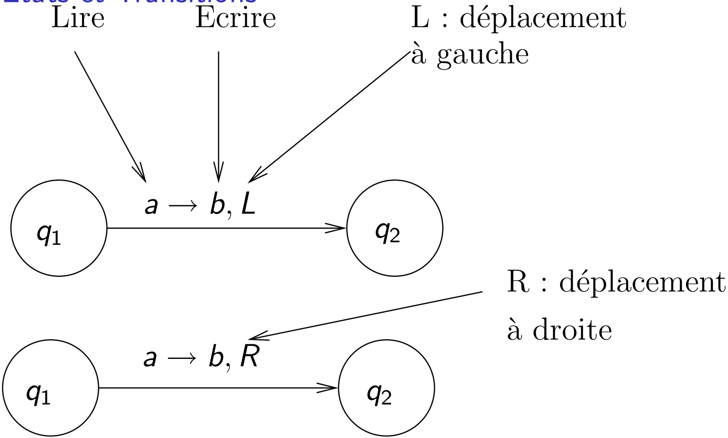
Initialisation : Un mot est inscrit sur le ruban et la tête est positionnée sur le caractère le plus à gauche

A chaque étape, la machine de Turing :

- lit un symbole
- fait une transition d'état
- fait l'une des trois actions suivantes :
  - écriture d'un symbole
  - déplacement de la tête vers la droite
  - déplacement de la tête vers la gauche

Machine de Turing

Etats et Transitions



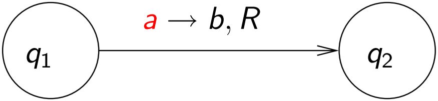
Machine de turing

Exemple de transition

a	b	a	c	◇	◇	◇						⋯

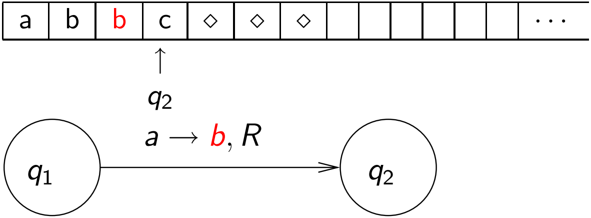
$\uparrow$

q_1



Machine de turing

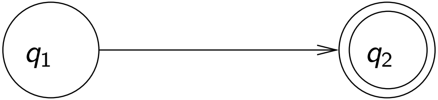
Exemple de transition



Machine de Turing

Acceptation et Langage

- La machine s'arrête s'il n'y a plus de transition possible
- Mot accepté si la machine s'arrête dans un état final :

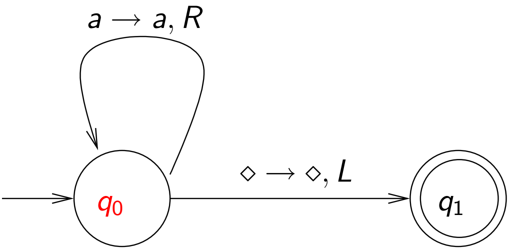


- Mot rejeté si la machine s'arrête dans un état non final ou entre dans une boucle
- L'ensemble des mots acceptés constituent le langage de la machine de Turing

Machine de Turing

Exemple de Machine de Turing

Soit la machine suivante qui accepte le langage $L = a^*$:

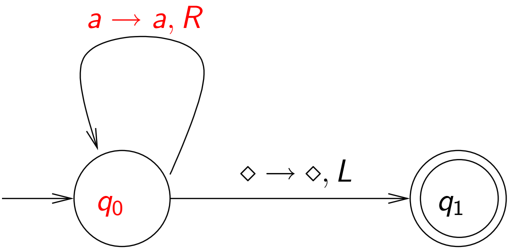


$T = 0$

Machine de Turing

Exemple de Machine de Turing

Soit la machine suivante qui accepte le langage $L = a^*$:



a	a	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

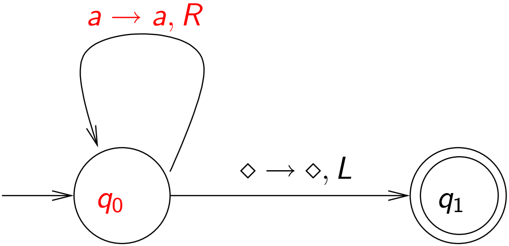
q_0

$T = 1$

Machine de Turing

Exemple de Machine de Turing

Soit la machine suivante qui accepte le langage $L = a^*$:



a	a	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

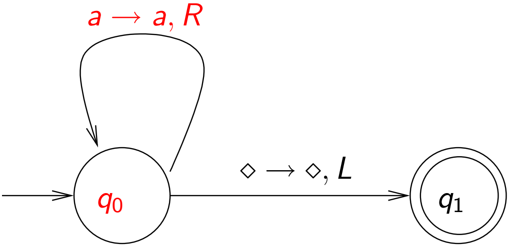
q_0

$T = 2$

Machine de Turing

Exemple de Machine de Turing

Soit la machine suivante qui accepte le langage $L = a^*$:



a	a	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

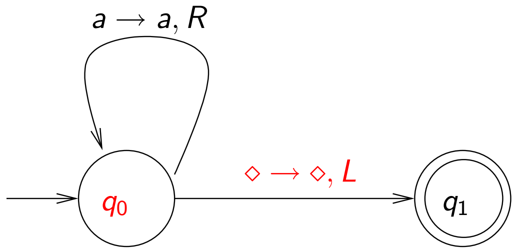
q_0

$T = 3$

Machine de Turing

Exemple de Machine de Turing

Soit la machine suivante qui accepte le langage $L = a^*$:



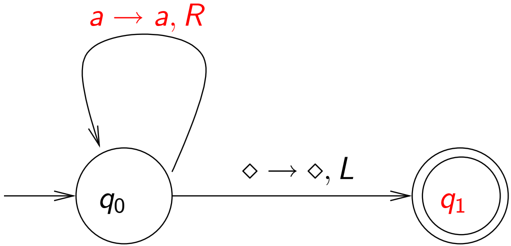
a	a	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$
			q_0				

$T = 4$

Machine de Turing

Exemple de Machine de Turing

Soit la machine suivante qui accepte le langage $L = a^*$:



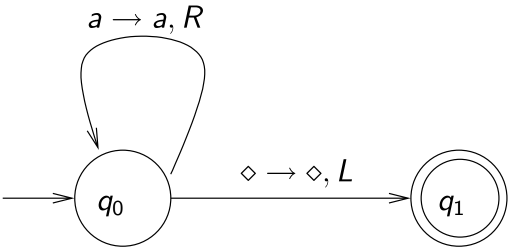
a	a	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_1

$T = 5 \cdots$: état final ; arrêt et acceptation.

Machine de Turing

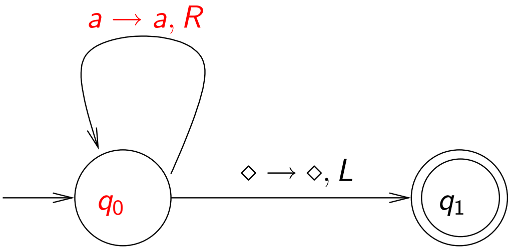
Exemple de rejet



$T = 0$

Machine de Turing

Exemple de rejet



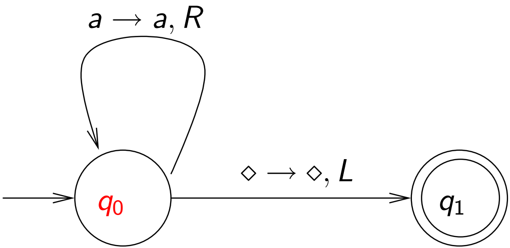
a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_0

$T = 1$

Machine de Turing

Exemple de rejet

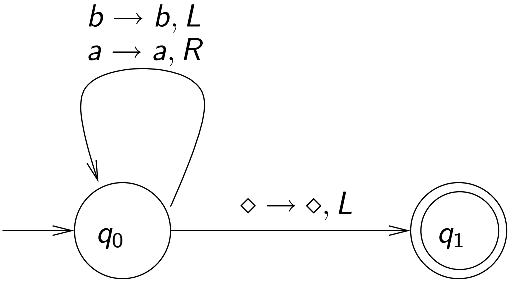


a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$
	q_0						

$T = 2$: pas de transition possible ; arrêt et rejet.

Machine de Turing

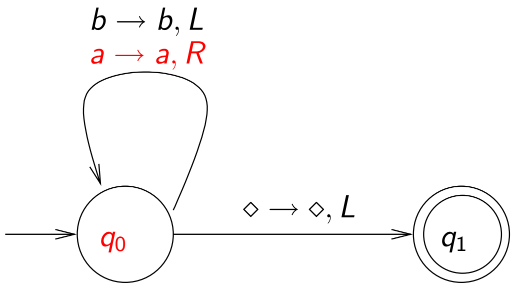
Exemple de boucle infinie



$T = 0$

Machine de Turing

Exemple de boucle infinie



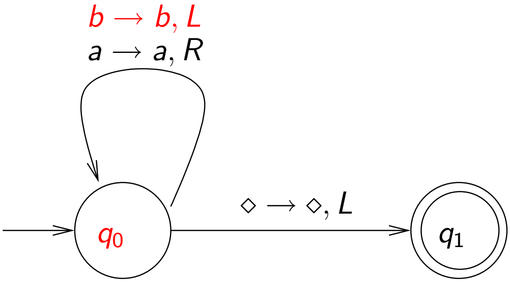
a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_0

$T = 1$

Machine de Turing

Exemple de boucle infinie



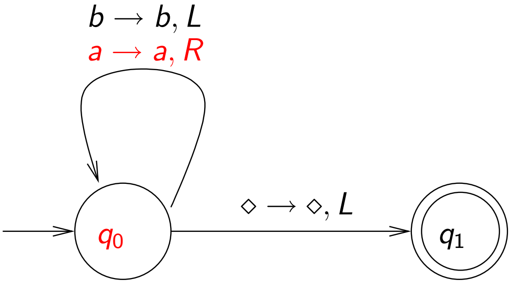
a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_0

$T = 2$

Machine de Turing

Exemple de boucle infinie



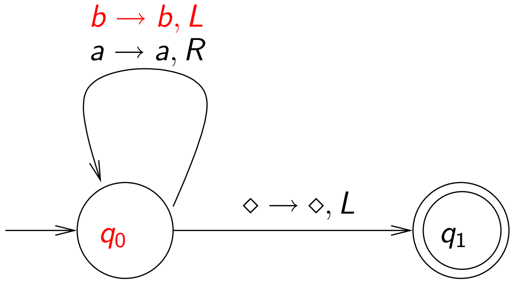
a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_0

$T = 3$

Machine de Turing

Exemple de boucle infinie



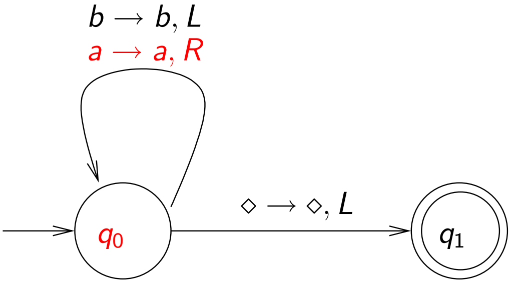
a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_0

$T = 4$

Machine de Turing

Exemple de boucle infinie



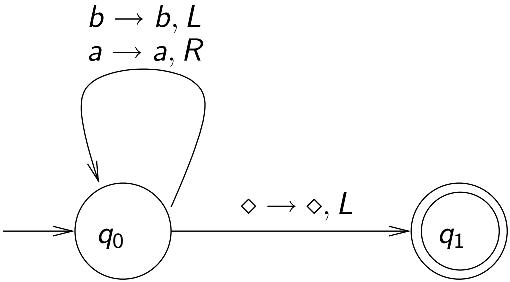
a	b	a	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\cdots$

q_0

$T = 5 \cdots$: boucle inifie.

Machine de Turing

Exemple de boucle infinie



$T =$

Machine de Turing

Définition formelle

$$M = (\mathcal{Q}, \Sigma, \Gamma, \delta, q_0, \diamond, F)$$

avec

- $\mathcal{Q}$: Etats
- Σ : Alphabet d'entrée
- Γ : Alphabet du ruban
- δ : fonction de transition (ex : $\delta(q_1, a) = (q_2, b, R)$)
- q_0 : état initial
- $\diamond$: blanc
- F : Etat final

Machine de Turing

Langage accepté

Pour toute machine de Turing M,

$$L(M) = \{w \ : \ q_0 \ w \mapsto^* x_1 \ q_f \ x_2\}$$

avec

- $q_0 \ w$:configuration initiale (état q_0 et tête sur première lettre de w
- $q_1 \ xv \mapsto x \ q_2 v$: déplacement de la tête en lisant la lettre x et en passant de l'état q_1 à l'état q_2
- $\mapsto^*$: occurrence multiple du déplacement $\mapsto$

Machine de Turing

Décidabilité

- Langage décidable : il existe un algorithme qui permet de reconnaître en un temps fini si un mot w appartient ou non à L
- Un langage L est décidé par une machine de Turing M si
  - M accepte L
  - M n'a pas d'exécution infinie

Grammaires générales

Exemple

$$\begin{array}{lll} S \rightarrow TZ & T \rightarrow 0T1C & T \rightarrow \varepsilon \\ C1 \rightarrow 1C & CZ \rightarrow Z2 & 1Z \rightarrow 1 \end{array}$$

Langage engendré :

Grammaires générales

Exemple

$$
\begin{array}{lll}
S \rightarrow TZ & T \rightarrow 0T1C & T \rightarrow \varepsilon \\
C1 \rightarrow 1C & CZ \rightarrow Z2 & 1Z \rightarrow 1
\end{array}
$$

Langage engendré : $0^i 1^i 2^j$

Grammaires générales

Exemple

$$\begin{array}{lll} S \to TZ & T \to 0T1C & T \to \varepsilon \\ C1 \to 1C & CZ \to Z2 & 1Z \to 1 \end{array}$$

Langage engendré : $0^i 1^i 2^j$
Langage non algébrique, non analysable via les automates à piles.

Grammaires générales

Théorème (Chomsky 1959)

Le langage L est engendré par une grammaire générale si et seulement si il est accepté par une machine de Turing (automate à deux piles).

Utilisation pratique

Nous verrons l'an prochain (cours de décidabilité), que malheureusement, ces langages sont en général indécidables, donc inexploitables dans la pratique.

Grammaires contextuelles

Restriction des grammaires générales, en contraignant les parties droites des règles à être au moins aussi long que les parties gauches. Ceci exclu évidemment le mot vide.

Exemple

$$
\begin{array}{lll}
S \rightarrow TZ & T \rightarrow 0U1 & T \rightarrow 01 \\
U \rightarrow 0U1C & U \rightarrow 01C & \\
C1 \rightarrow 1C & CZ \rightarrow Z2 & 1Z \rightarrow 12
\end{array}
$$

Langage engendré :

Grammaires contextuelles

Restriction des grammaires générales, en contraignant les parties droites des règles à être au moins aussi long que les parties gauches. Ceci exclu évidemment le mot vide.

Exemple

$$\begin{array}{lll} S \to TZ & T \to 0U1 & T \to 01 \\ U \to 0U1C & U \to 01C & \\ C1 \to 1C & CZ \to Z2 & 1Z \to 12 \end{array}$$

Langage engendré :$0^i 1^i 2^j$

Grammaires contextuelles

Restriction des grammaires générales, en contraignant les parties droites des règles à être au moins aussi long que les parties gauches. Ceci exclu évidemment le mot vide.

Exemple

$$\begin{array}{lll} S \rightarrow TZ & T \rightarrow 0U1 & T \rightarrow 01 \\ U \rightarrow 0U1C & U \rightarrow 01C & \\ C1 \rightarrow 1C & CZ \rightarrow Z2 & 1Z \rightarrow 12 \end{array}$$

Langage engendré :$0^i1^i2^j$

Propriété des grammaires contextuelles

Propriétés

- Soit une grammaire contextuelle et un mot de longueur n. Il est possible de vérifier si ce mot est engendré par la grammaire : fabrication de toutes les dérivations à partir de l'axiome.
- Reconnaissance des langages contextuels via les machine de Turing linéairement bornées : machine de Turing non déterministe qui n'utilise de sa mémoire infinie qu'une portion dépendant linéairement de la taille du mot testé.

Propriété des grammaires contextuelles (suite)

Théorème

Un langage est contextuel si et seulement si il est accepté par une machine de Turing linéairement bornée.

Limitations

- On ne sait pas si l'on peut se passer de l'hypothèse de non déterminisme
- Beaucoup de problèmes indécidables, notamment pour déterminer si un langage contextuel est vide

Hiérarchie de Chomsky

Outils algorithmique

Manipulation des langages de type 0 et 1 via les machines de Turing :

Langages	Grammaires	Reconnaissance
Langages récursivement énumérables	Type 0	Machine de Turing
Langages contextuels	Type 1	Machine de Turing linéairement borné
Langages hors contexte	Type 2	Automates à Pile déterministe
Langages rationnels	Type 3	Automates d'états finis