République algérienne démocratique et populaire
Ministre de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – Biskra
Faculté des Sc.Exactes, et SNV.
Département d'Informatique

Extensions des machines de Turing et puissance de calcul

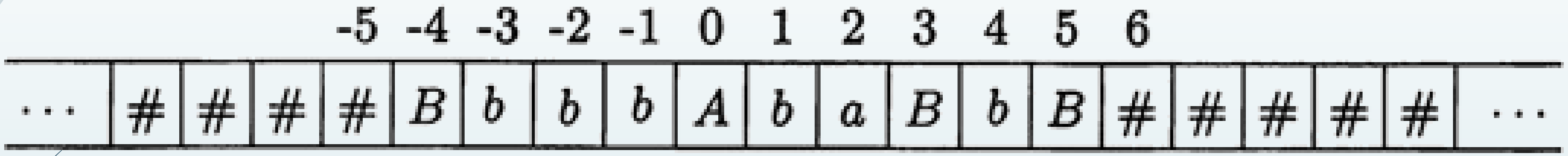
Niveau: 1er année Master
Option: Génie Logiciel et Systèmes Distribués

2023/2024

Plan

- Machine de turing à ruban doublement infini
- Machine de turing à ruban multiple
- Machine de turing à RAM
- Machine de turing non-déterministes
- Machine de turing universelle
- Thèse de Church-Turing

MT à Ruban Bi-infini



Identique à la MT étudiée sauf:

- La tête de lecture peut toujours se déplacer à gauche.

Preuve de l'équivalence (1):

MT → MT à bande bi-infinie

- Toute MT est équivalente à une machine bi- infinie…(1)

Preuve (1):

- Il suffit de marque la case -1 avec un symbole ($) n'appartenant pas à Γ.
- Comme aucune transition ne permet de lire ce symbole, la machine se bloque dès qu'elle passe dessus.

Preuve de l'équivalence (2):
MT à bande bi-infinie → MT

- Toute MT bi-infinie est équivalente à une machine de Turing… (2)

Preuve (2):

- Replier la bande en O .
- La case numéro –i se trouve en dessous de la case numéro i.
- La case sous O contient un nouveau symbole ($).
- Chaque case contient deux symboles.

0	1	2	3	4	5	6	7	
A	b	a	B	b	B	#	#	…
$	b	b	b	B	#	#	#	…
	-1	-2	-3	-4	-5	-6	-7	

Preuve de l'équivalence (2): MT à bande bi-infinie → MT

- Soit la MT à bande bi-infinie

$$M = (Q, \Gamma, \Sigma, \delta, s, B, F)$$

- La MT équivalente est :

$$M' = (Q', \Gamma', \Sigma', \delta', s', B', F')$$

Où

- $\Gamma' = \Gamma \times (\Gamma \cup \{\$\})$
- $\Sigma' = \Sigma \times \{B\}$
- $B' = (B, B)$
- $Q' = Q \times \{U, D\} \cup \{s'\}$: U (Up) partie droite, D (Down) partie gauche (du ruban bi-infini) et s' est le nouvel état initial.
- $F' = F \times \{U, D\}$

0	1	2	3	4	5	6	7	
A	b	a	B	b	B	#	#	...
$	b	b	b	B	#	#	#	...
	-1	-2	-3	-4	-5	-6	-7	

Preuve de l'équivalence (2): MT à bande bi-infinie → MT

- La fonction de transition δ' :
  - À partir de l'état initial s':

$$\delta'(s',(x,B)) = \begin{cases} ((q,U),(y,\$),R) & \text{si} \quad \delta(s,x)=(q,y,R) \\ ((q,D),(y,\$),R) & \text{si} \quad \delta(s,x)=(q,y,L) \end{cases}$$

  - Si la machine n'est pas dans l'état initial

$$\begin{array}{ll} \delta'((q,U),(x,t\neq\$)) = ((r,U),(y,t),R) & \text{si } \delta(q,x)=(r,y,R) \\ \delta'((q,U),(x,t\neq\$)) = ((r,U),(y,t),L) & \text{si } \delta(q,x)=(r,y,L) \\ \delta'((q,D),(x,t\neq\$)) = ((r,D),(x,y),R) & \text{si } \delta(q,t)=(r,y,L) \\ \delta'((q,D),(x,t\neq\$)) = ((r,D),(x,y),L) & \text{si } \delta(q,t)=(r,y,R) \end{array}$$

  - Et si t = \$?

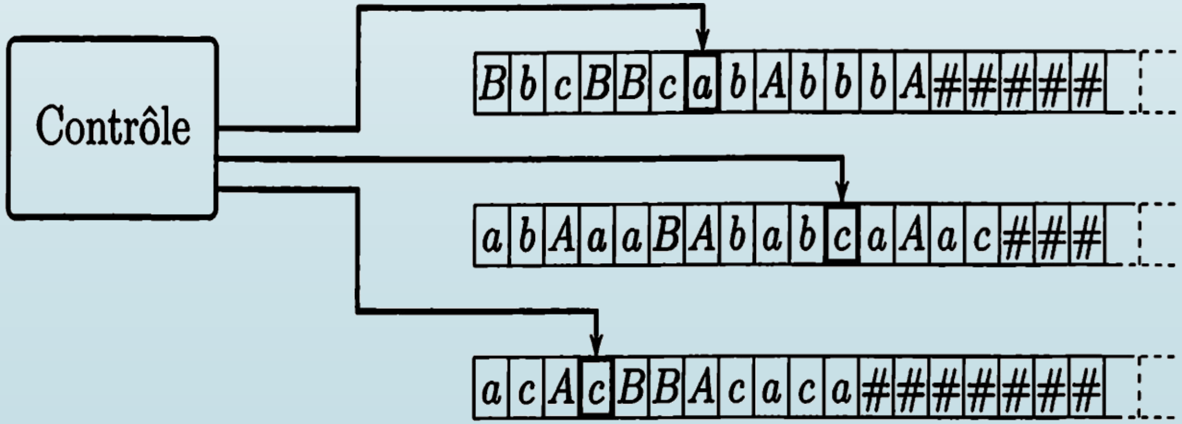
Conclusion 1

Ce qui peut être calculé par une machine à ruban bi-infini, peut l'être par une machine ordinaire.

Le ruban bi-infini n'augmente pas la puissance de calcul.

MT à rubans multiples

- La machine contient plusieurs rubans et une tête de lecture pour chaque ruban.
- Une tête peut être laissée sans mouvement après une transition.
- Une configuration est caractérisée par :
  - L'état,
  - Le contenu de chaque ruban et
  - La position de chaque tête.



MT à rubans multiples:

- La transition spécifie:
  - Les symboles à écrire par chaque tête.
  - Le sens de déplacement de chaque tête (qui peut être nul « N »)
- La fonction de transition:

$$Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, N\}^k$$

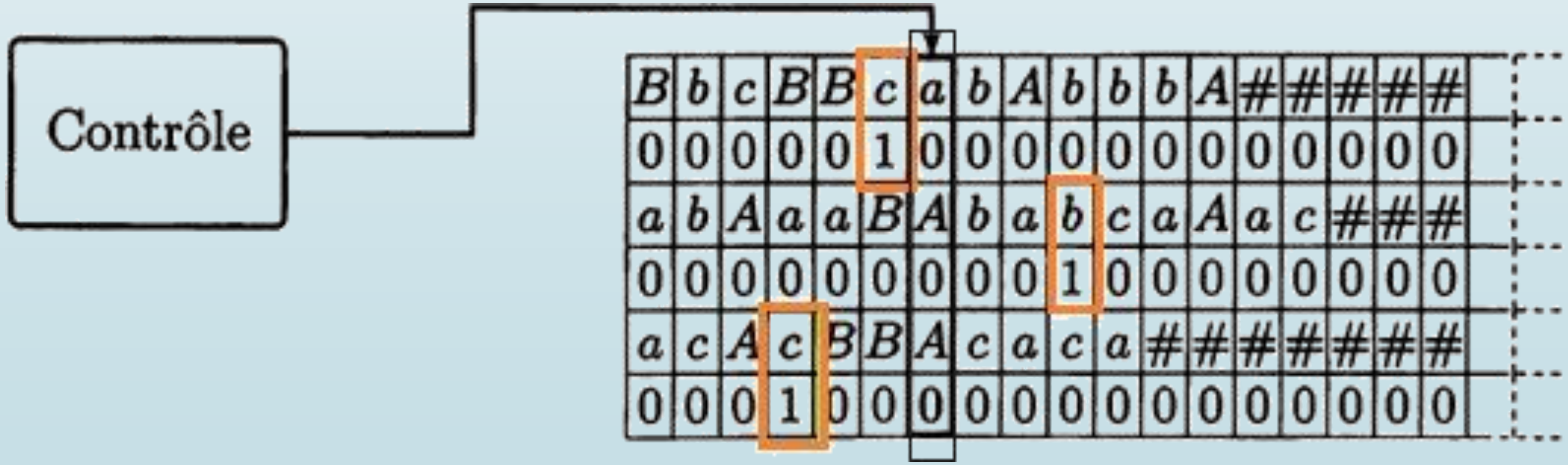
où k est le nombre de rubans

- On suppose que tous les rubans ont le même alphabet pour simplifier la notation (l’inverse n’aura pas un grand intérêt).

Equivalence:

MT rubans multiples → MT

- Utiliser des symboles composés de:
  - **Contenu de chaque ruban,**
  - **Positions des têtes de lectures (0/1).**
- Le contenu de chaque case est un symbole de
- $(\Gamma \times \{0,1\})^k$ où k est le nombre de rubans.



Equivalence:

MT rubans multiples → MT (suite)

- Pour simuler une étape de l'exécution d'une machine à rubans multiples, la machine à ruban unique doit faire deux parcours:
  1. Trouver la position des têtes de lectures et identifier les symboles correspondants.
  2. Modifier ces symboles, déplacer les têtes et changer l'état.
- La construction exacte est assez complexe, l'important est de se convaincre qu'elle est possible!

Conclusion 2

Ce qui peut être calculé par une machine à rubans multiple, peut l'être (plus difficilement) par une machine ordinaire.

Les rubans multiples n'augmentent pas la puissance de calcul.

Mais offrent une plus grande souplesse dans la programmation

Machine à mémoire à accès direct (RAM)

- La machine contient :
  - Une mémoire à accès direct (RAM)
  - Un certain nombre de registres dont le CO (compteur ordinal)
- Elle peut être simulée par une machine de Turing à rubans multiples :
  - un ruban pour la RAM
  - un ruban pour chaque registre
- Le contenu de la RAM est représenté par des paires de mots (adresse,contenu)
  - Les mots de chaque paire sont séparés par *;
  - Les paires sont séparées par #.

Simulation de la machine à RAM

- À chaque étape de l'exécution:
  - Parcourir le ruban RAM (représentant la RAM) jusqu'à trouver le mot correspondant à l'adresse qui se trouve dans le ruban du CO.
  - Lire et décoder l' instruction se trouvant à cette adresse.
  - Le cas échéant, trouver les opérandes de cette instruction.
  - Exécuter l'instruction, ce qui implique éventuellement le changement de la RAM et/ou des registres.
  - Incrémenter le CO (ou branchement) et passer au cycle suivant.

- La construction complète et détaillée de cette machine et très lourde. Il est seulement nécessaire de se convaincre que c'est possible.

Conclusion 3

Ce qui peut être calculé par une machine à RAM, peut l'être (plus difficilement) par une machine ordinaire.

La RAM n'augmente pas la puissance de calcul.

Mais offre une plus grande souplesse dans la programmation

Machines de Turing non-déterministes

- MT déterministe : Il y a **au plus** une configuration qui peut être dérivée **en une étape**, à partir d'une configuration donnée.
- MT non-déterministe:
  - Même définition que MT déterministe (vue précédemment)

Sauf:

- La définition de la fonction de transition:

➔ est une relation de la forme:

$$\Delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R\})$$

- Une transition est donc un quintuplet de la forme:

$$((q, a), (q', x, X)) \in \Delta$$

Exécution

- Une MT non-déterministe n’a pas une exécution unique.
- Il suffit qu’une seule de ces exécutions aura un état accepteur pour que la machine accepte

Elimination du non-déterminisme

Théorème: Tout langage accepté par une machine de Turing non-déterministe M est aussi accepté par une machine de Turing déterministe M'.

Preuve:

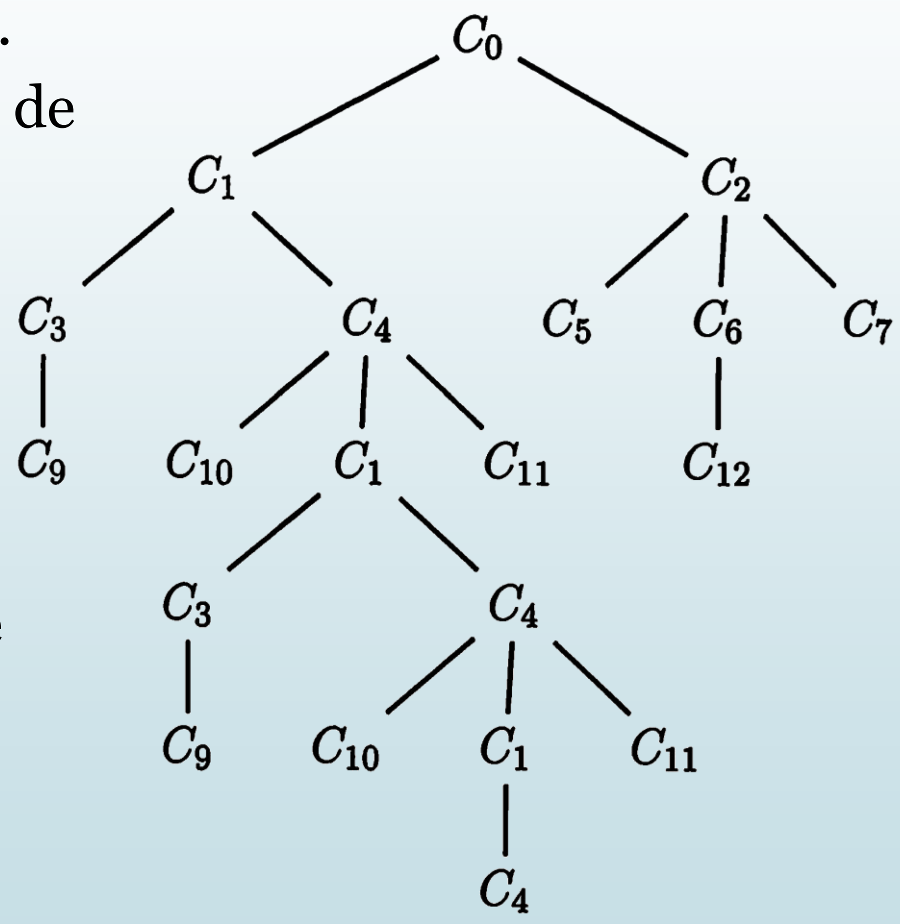
- Il faut que la MT déterministe simule toutes les exécutions de la MT non-déterministe.

Problème →

  - Les exécutions ne peuvent pas être simulées simultanément (**un seul ruban**)
  - Simuler les exécutions une par une (**il y a certaines exécutions infinies**)

Un arbre de calcul (non-déterministe)

- Soit $w \in \Sigma^*$ une entrée de la machine M.
- L'arbre représente l'ensemble des calculs de M sur w.
- Chaque branche représente un calcul valide de M.
- Les nœuds sont étiquetés par les configurations.
- Les fils d'un calcul γ sont tous les calculs obtenus en prolongeant γ d'une étape de calcul.
- Le nombre de fils est borné par le nombre de transitions de la machine.



Simulation de M par M'

- L'idée générale est de faire parcourir l'arbre de calcul M par M'.
- Pour éviter les branches infinies, il faut faire un parcours en largeur.
- La suite de transitions effectuées par un calcul peut être considérée comme un mot formé par les libellés de ces transitions, $x = \tau_1 \cdots \tau_n$

où $\tau_1, \ldots, \tau_n$ sont des transitions.

- Tout mot ne correspond pas forcément à un calcul valide.

Simulation de M par M’ (suite)

- La machine M’ va parcourir les nœuds de l’arbre dans l’ordre des mots correspondants ordonnés par:
  - longueur du mot
  - pour la même longueur , par ordre lexicographique.
- Les transitions de M sont numérotées de 0 à k-1 où k est le nombre des transitions.
- Les mots formés de l’alphabet {0,…, k-1} correspondent donc à une exécution des transitions qui forment le mot.

Construction de la machine M’

- La machine M’ est construite en utilisant trois rubans (équivalente à une machine à ruban unique).
- 1er ruban: contient toujours l’entrée w (n’ai jamais modifié).
- *2ème* ruban: contient une copie du ruban de M au cours de son calcul
- *3ème ruban:* contient un mot sur l’alphabet {0,..., k-1} correspondant à ....

Exécution de M’

1. Initialement, le 1er ruban contient l’entrée w et les deux autres sont vides.
2. Placer le premier mot (transitions) dans l’ordre hiérarchique défini.
3. Copier le 1er ruban sur le 2ème .
4. Utiliser le 2ème ruban pour simuler M sur l’entrée w le long du mot spécifier par le 3ème ruban.

   À chaque étape de l’exécution le 3ème ruban indique le numéro de la transition à appliquer.
5. Remplacer la chaîne du 3ème ruban par la chaîne qui suit dans l’ordre hiérarchique défini. La machine M’ retourne à l’étape 3 pour simuler ce nouveau calcul.

Conclusion 4

Ce qui peut être calculé par une machine de Turing non-déterministe, peut l'être par une machine ordinaire.

Le non-déterminisme n'augmente pas la puissance de calcul.

Mais offre une plus grande souplesse dans la programmation

Machine de Turing universelle

C'est une MT qui serai capable de simuler n'importe quelle machine de Turing.

- Soit M une machine de Turing, à laquelle on fournirait:
  1. La description d'une machine de Turing M'.
  2. Un mot d'entrée w.

La machine M doit simuler l'exécution de M' sur w.

- En informatique, M est considérée comme un interpréteur de machine de Turing.
- Cette idée est considérée comme l'origine de l'ordinateur à programme enregistré de John Von Neumann.

Fonctions calculables par MT

- Nous avons considéré jusqu'à maintenant le rôle d'acceptation d'une MT.
- Une MT peut aussi calculer une fonction:
  - L'argument de la fonction→ la valeur d'entrée
  - La valeur de la fonction → contenu du ruban à la fin du calcul.

Définition : Une MT calcule une fonction $f : \Sigma^* \rightarrow \Sigma^*$ si pour tout mot d'entrée w, elle s'arrête toujours dans une configuration où $f(w)$ est sur le ruban.

Fonction calculable

- Une fonction est calculable s’il existe une MT qui la calcule.
- Décider un langage est calculer une fonction dont le codomaine est $\{0, 1\}$
- On énonce souvent la thèse de Church-Turing en terme de fonctions calculables.

Thèse de Church-Turing

- Les différentes variantes des machines de Turing conduisent à la même notion de calculable.

- Beaucoup d'autres modèles de calcul ont été introduits: fonctions récursives, λ-calcul,… et se sont avérés équivalents (ou plus faibles) que les machines de Turing.

Tous ce qui est effectivement calculable est calculable par une machine de Turing

Thèse de Church-Turing

- Une *thèse* consiste à affirmer, sans le démontrer, qu'une notion intuitive correspond exactement à une notion formelle.

- La thèse de Church ne peut être démontrée mais elle est largement admise