



Quelques machines de Turing

Préparation au CAPES de Mathématiques, option Informatique

Fev. 2017

Sources

- Cours de complexité M1 info FST 2013-16
- Cours d'informatique fondamentale 4A polytech Lille 2012.

EXERCICE 1 ► Test

D'après <http://www.computing.dcu.ie/~josef/CA215/Exercices/turingqns.html> Soit \mathcal{M} la machine de Turing avec le vocabulaire $\Sigma = \{a, b\}$, le vocabulaire de pile $\Gamma = \Sigma \cup \{B, Y\}$, d'états $\{q_0, \dots, q_8\}$, d'état final q_8 , et de fonction de transition donnée par la table suivante :

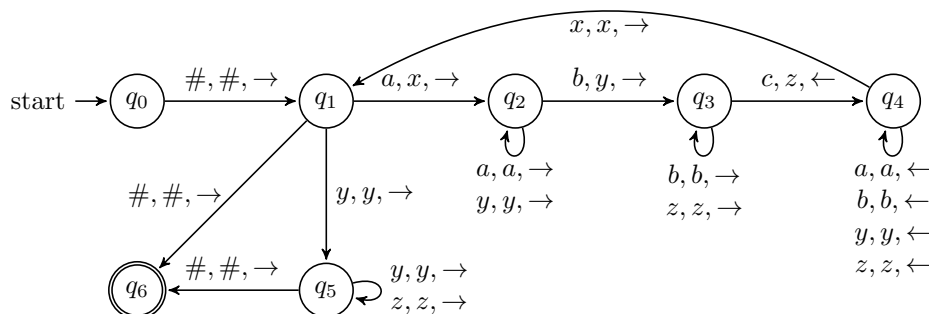
| état | Symbole | (état suivant, symbole, mouvement) |
|-------|---------|------------------------------------|
| q_0 | B | (q_1, B, D) |
| q_0 | B | (q_1, B, D) |
| q_1 | a | (q_1, a, D) |
| q_1 | b | (q_1, b, D) |
| q_1 | B | (q_2, B, G) |
| q_2 | a | (q_3, B, D) |
| q_2 | b | (q_5, B, D) |
| q_2 | B | (q_8, B, I) |
| q_3 | B | (q_4, a, D) |
| q_4 | a | (q_4, a, D) |
| q_4 | b | (q_4, b, D) |
| q_4 | B | (q_7, a, G) |
| q_5 | B | (q_6, b, D) |
| q_6 | a | (q_6, a, D) |
| q_6 | b | (q_6, b, D) |
| q_6 | B | (q_7, b, G) |
| q_7 | a | (q_7, a, G) |
| q_7 | b | (q_7, b, G) |
| q_7 | B | (q_2, B, G) |

1. Sur l'entrée $BabBBBBBBBBBBBBBBBBBBBBBBBB$, que retourne la machine ?
2. Sur l'entrée : $BbaaBBBBBBBBBBBBBBBBBBBBBBBB$ que retourne cette machine ?

1 Reconnaissance de langages

EXERCICE 2 ► MT mystère

Quel langage reconnaît la machine de Turing (sur l'alphabet d'entrée $\{a, b, c\}$) ci-dessous ?



Solution. Le langage reconnu est $a^n b^n c^n$. Explication informelle : Chaque lecture de a est remplacée par un x , puis est suivi d'une séquence ($G \rightarrow D$) qui remplace un b par y puis un c par z , et retourne vers la gauche pour chercher le x initial. Par construction donc chaque cycle élimine un $a - - - b - - - c$, puis retourne à gauche jusqu'à rencontrer un x (élimination du a). Une fois que l'on n'a plus de a , la seule façon de terminer est alors d'aller vers la droite en ne rencontrant que des y , puis que des z .

Tentative de rédaction plus formelle (avant on a fait des exemples, on s'est convaincus que $\{a^n b^n c^n | n \in \mathbb{N}\}$ est le langage reconnu). On rédige par double inclusion, of course :

- Si on a un entrée un mot w de la forme $a^i b^j c^i$, alors il est accepté : le ruban contenant $\#a^i b^j c^i\#$ en entrée, le chemin $(q_0, \text{chemin}_i, q_5(\dots), q_6)$ est acceptant, avec chemin_k dénotant une boucle dans la MT :

$$\text{chemin}_k = (q_1, q_2, b_2^{k-1}, q_3, b_3^{k-1}, q_4, b_4^{2k-2}, q_1)$$

(b_i dénote la boucle autour du noeud i) Il est clair que chemin_k élimine le k -ième a du mot, ainsi que le k -ième b et le k -ième c .

- Si un mot est accepté, alors clairement il est de la forme $a^i b^j c^j$: Tout chemin acceptant termine par un certain nombre de y suivis d'un certain nombre de z puis un $\#$. Pour générer un y il faut forcément avoir lu un a , donc fait au moins un cycle du haut, donc lu des a , puis des b puis des c . De plus, si on a un b qui excède, on est bloqué dans l'état q_3 . Si on a un b en moins, on est bloqué dans l'état q_2 . Le nombre de c est égal au nombre de a et de b de la même manière. Donc tout mot accepté est de la forme $a^i b^j c^j$.

TODO : faire un dessin. □

EXERCICE 3 ► Sous-mot et co

Écrire une machine de Turing acceptant les mots sur l'alphabet $\Sigma = \{a, b\}$ qui contiennent le sous-mot aab et se terminent par un b , et refusant¹ tous les autres mots sur Σ .

2 Calculs

EXERCICE 4 ► Forward/Modif

Écrire une machine de Turing $\text{Modif}_{a,b}$ qui transforme les a en b , les b en a , de gauche à droite, retourne à la position initiale et s'arrête.

EXERCICE 5 ► Addition unaire

Écrire une machine de Turing qui calcule la fonction prenant en entrée deux entiers n et m et renvoyant $n + m$.

EXERCICE 6 ► Fois 2

Construire une machine de Turing calculant la fonction $x \mapsto 2 * x$ (en binaire).

EXERCICE 7 ► Successeur binaire

Écrire une machine de Turing qui calcule la fonction prenant en entrée un entier n (en base 2) et renvoyant $n + 1$.

1. Cette machine doit donc terminer sur toutes les entrées. Chaque mot du langage doit mener vers un état acceptant, mettons q_a , et les mots qui ne sont pas dans le langage doivent mener à un état de refus, mettons, q_r .