Notes de cours Informatique théorique - III
(version 1.1, 2.4.2013)

Machines de Turing

Nous allons agrandir la classe de langages reconnus par des machines en en définissant une nouvelle sorte, plus puissante. L'idée de base est toujours celle qui accompagnait les automates finis déterministes, ou non, et les automates a piles (non déterministes).
Si on se fixe un alphabet Σ, on peut écrire un mot $w \in \Sigma^*$ sur un ruban infini dans un sens que l'on imagine divisé en cases (formaté) numérotées par des naturels de 0 à l'infini. La tête de la machine de Turing commencera son exécution en son état initial sur la première case du ruban. L'état actuel de la machine et le symbole de Σ lu par la tête dans la case actuelle détermineront l'état suivant de la machine, le symbole qui doit remplacer le symbole lu, et la direction (droite, gauche) du déplacement de la tête d'une case (au cas où la tête devrait aller à gauche de la première case, elle restera sur place). La machine acceptera ou refusera le mot w si elle arrive dans son état acceptant ou refusant. De cette idée vient la description formelle.

Définition 1 *Une machine de Turing (MT)* $M = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$ *est définie par*

- Q*, un ensemble fini détats;*
- Γ*, un alphabet du ruban;*
- $\Sigma \subset \Gamma$*, un alphabet d'entrée;*
- $B \in \Gamma \setminus \Sigma$*, le symbole "blanc";*
- $s \in Q$*, l'état initial;*
- $q_A \in Q$*, l'état acceptant;*
- $q_R \in Q$*, l'état refusant;*
- $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{D, G\}$*, une fonction de transition.*

Le ruban est initialisé à B *partout sauf les premières* $|w|$ *cases qui contiennent* w*, le mot d'entrée. Si la machine entre l'état* q_A *ou* q_R*, elle s'arrête; pour tout autre état la fonction de transition doit lui permettre de continuer.*

Afin de définir précisément le langage accepté par un machine de Turing, nous devons immiter la définition de $\hat{\delta}$ des automates finis.

Définition 2 *Soit $M = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$ une machine de Turing. Une* configuration *de M est un triplet (u, q, v), $u, v \in \Gamma^*$, $q \in Q$ exprimant le fait que le mot uv dont le dernier symbole n'est pas B est sur les $|uv|$ premières cases du ruban, les cases après la $|uv|-$ème contiennet des B, et la tête de la machine est sur le premier symbole de v.*

Définition 3 *Soit $M = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$ une machine de Turing et soit $u = u_1 \dots u_m$, $v = v_1 \dots v_n$, $u_i, v_j \in \Gamma$, $1 \leq i \leq m$ et $1 \leq j \leq n$.*

1. *Une configuration (u, q, v)* dérive *ou* donne *la configuration (u, q, v) en 0 étapes.*

2. *Une configuration (u, q, v)* dérive *ou* donne *une configuration (u', q', v')* directement *(ou* en une étape*), si*

   - *soit $u' = uc$, $v' = v_2 \dots v_n$ et $\delta(q, v_1) = (q', c, D)$,*
   - *soit $u' = u_1 \dots u_{m-1}$, $v' = u_m c v_2 \dots v_n$ et $\delta(q, v_1) = (q', c, G)$.*

   On écrit $(u, q, v) \overset{1}{\longmapsto} (u', q', v')$ ou même $(u, q, v) \longmapsto (u', q', v')$.

3. *Une configuration (u, q, v)* dérive *ou* donne *une configuration (u', q', v')* en k étapes, *$k > 0$ – noté $(u, q, v) \overset{k}{\longmapsto} (u', q', v')$ – s'il existe une configuration (α, p, β) de M telle que*

$$(u, q, v) \overset{1}{\longmapsto} (\alpha, p, \beta) \overset{k-1}{\longmapsto} (u', q', v').$$

4. *Une configuration (u, q, v)* dérive *ou* donne *une configuration (u', q', v') – noté $(u, q, v) \overset{*}{\longmapsto} (u', q', v')$ – s'il existe un $k \in \mathbb{N}$ tel que $(u, q, v) \overset{k}{\longmapsto} (u', q', v')$.*

Notons que la partie 2 de la définition suit de la partie 3 si on modifie cette dernière légérement. Toutefois c'est plus facile à comprendre écrit comme nous venons de le faire.

Définition 4 *Soit $M = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$ une machine de Turing. Le* langage de M *est l'ensemble $L(M) = \{w \in \Sigma^* :$ il existent $u, v \in \Gamma^*$ tels que $(\varepsilon, s, w) \overset{*}{\longmapsto} (u, q_a, v)\}$.*

La machine de Turing reconnaît *son langage. Elle* décide *son langage si elle s'arrête sur toute entrée, c'est-à-dire, si pour tout $w \in \Sigma^*$, soit $(\varepsilon, s, w) \overset{*}{\longmapsto} (u, q_a, v)$, soit $(\varepsilon, s, w) \overset{*}{\longmapsto} (u, q_r, v)$ (pour certains $u, v \in \Gamma^*$). En d'autre mots, M soit accepte, soit refuse tout mot $w \in \Sigma^*$.*

Cette définition est très importante pour la suite.

Comme pour les automates finis, on peut définir des variantes des machines de Turing et prouver qu'elles sont équivalentes. Regardons-en une en un peu de détail.

Une machine de Turing $M = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$ à k rubans est définie comme la machine de Turing ordinaire mais, comme le nom l'indique, avec k rubans, numérotés $1, \ldots, k$, et k têtes indépendantes. La fonction de transition est une fonction $\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{G, D\}^k$ avec l'interprétation que M, en lisant $(a_1, \ldots, a_k)$ en état q remplace a_i par b_i sur le ruban i et déplace la i-ème tête dans la direction d_i. Une telle machine de Turing commence avec un mot sur le premier ruban, les autres rubans vides, et les têtes sur la première case de leur ruban.

On voudrait simuler une telle machine de Turing $M = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$ par une machine de Turing $\hat{M} = (\hat{Q}, \Sigma, \hat{\Gamma}, \hat{s}, \hat{q_A}, \hat{q_R})$ avec un seul ruban. Soit $w = a_1 \ldots a_n \in \Sigma^*$ le mot d'entrée de M. Voici UNE façon de faire. On suppose que l'on ait à notre disposition une machine de Turing D_p qui déplace le contenu du ruban d'une case à droite à partir de la position de la tête et puis revient à la case libérée en l'état p. La machine de Turing du devoir peut facilement être adaptée pour devenir D_p; il en faudra plusieurs, suivant p.

La machine de Turing $\hat{M}$ commencera en état $\hat{s}$ lisant le premier symbole de w sur son ruban. La simulation se déroule en deux étapes, prétraitement (le ruban est préparé pour faciliter la suite) et simulation.

Prétraitement. On décale w de deux cases à droite, on met $\$\#$ dans les deux cases de gauche libérées, et on met après le dernier symbole de w $k-1$ segments entourés de $\#$ à utiliser pour simuler les $k-1$ rubans restants. On indique également l'emplacement de chaque tête et l'indication de quel ruban il s'agit. On aura donc besoin d'avoir $\{T_{q,a,i} : q \in Q,\ a \in \Gamma,\ i \in \{1, \ldots, k\} \subseteq \hat{\Gamma}$. Chaque $T_{q,a,i}$ indique que la tête du ruban i est sur le symbole a en l'état q dans M. Le prétraitement laissera sur le ruban

$$\$\#T_{s,a_1,1}a_2 \ldots a_n\#T_{s,B,2}\# \ldots \#T_{s,B,k}\#$$

suive des blancs. Voici les détails (les états utilisés seront ajoutés à $\hat{Q}$).

1. L'état initial $\hat{s}$ déclenche un appel à D_{p_1} quelque soit le symbole lu. Ensuite, la tête sera en l'état p_1 sur le B que D_{p_1} aura écrit sur la case libérée; p_1 déclenche un appel à D_{p_2} qui va se terminer en l'état $p_\$$ sur la première case du ruban. Ensuite
$\hat{\delta}(p_\$, B) = (p_{\#,1}, \$, D)$
$\hat{\delta}(p_{\#,1}, B) = (p_{T,1}, \#, D)$
$\hat{\delta}(p_{T,1}, x) = (p_{r,1}, T_{s,x,1}, D)$ pour tout $x \in \Sigma$ (on indique la position initiale de la première tête sur le premier ruban).
$\hat{\delta}(p_{r,1}, x) = (p_{r,1}, x, D)$ pour tout $x \in \Sigma$ (on cherche la fin de w)
$\hat{\delta}(p_{r,1}, B) = (p_{b,2}, \#, D)$ (on marque la fin du premier ruban)
$\hat{\delta}(p_{b,2}, B) = (p_{r,2}, T_{s,B,2}, D)$ (on marque la position de la deuxième tête sur le deuxième ruban ainsi que le blanc qu'elle lit dans l'état initial)
$\hat{\delta}(p_{r,i}, B) = (p_{b,i+1}, \#, D)$ pour tout $i \in \{2, \ldots, k\}$ (marque la fin du i-ème ruban)
$\hat{\delta}(p_{b,i}, B) = (p_{r,i+1}, T_{s,B,i}, D)$ pour tout $i \in \{3, \ldots, k-1\}$ (marque la position de la i-ème tête sur le i-ème ruban ainsi que le blanc qu'elle lit dans l'état initial)
$\hat{\delta}(p_{b,k+1}, x) = (p_{b,k+1}, x, G)$ pour $x \neq \$$(on a fini, on retourne)
$\hat{\delta}(p_{b,k+1}, \$) = (m_1, \$, D)$ (on va sur le premier # et on commence la simulation).

Remarque: les états s'appellent p pour prétraitemnt. *Il serait peut être mieux d'éliminer le trop d'indices et les appeler, par exemple,* r_1, r_i, b_i, t_i *etc au prix de ne pas voir le p. Dites-moi ce qui serait plus compréhensible.*

La simulation est maintenant relativement simple. Soit $\delta(q, (x_1, \ldots, x_k)) = (p, (y_1, \ldots, y_k), (d_1, \ldots, d_k))$ une transition de M. Dans $\hat{M}$ elle devient :

Premier essai

pour pour tout $i \in \{1, \ldots, k\}$
$\hat{\delta}(m_i, x) = (m_i, x, D)$ pour $x \neq T_{q,x_i,i}$, (on cherche la i-ème tête)
$\hat{\delta}(m_i, T_{q,x_i,i}) = (m_{i,p}, y_i, d_i)$ (on remplace x_i par y_i et on va dans la direction d_i se souvenant de p)
$\hat{\delta}(m_{i,p}, a) = (m_{i+1}, T_{p,a,i}, D)$ (on note où se trouve la tête, dans quel état, et sur quel symbole)
$\hat{\delta}(m_{k+1}, x) = (m_{k+1}, x, G)$ pour $x \neq \$$ (on revient au début du ruban)
$\hat{\delta}(m_{k+1}, \$) = (m_1, \$, D)$ (on recommence)

Ceci marche à un détail près : quand on ècrit sur le ruban i sur le dernier blanc, on doit en ajouter un, en déplaçant le contenu du ruban à droite de ce B vers la droite. Il faut donc modifier notre premier essai.

Deuxième essai

pour pour tout $i \in \{1, \ldots, k\}$
$\hat{\delta}(m_i, x) = (m_i, x, D)$ pour $x \neq T_{q,x_i,i}$, (on cherche la i-ème tête)
$\hat{\delta}(m_i, T_{q,x_i,i}) = (m_{i,p,d_i}, y_i, d_i)$ (on remplace x_i par y_i et on va dans la direction d_i
se souvenant de p et de d_i)
$\hat{\delta}(m_{i,p,G}, a) = (m_{i+1}, T_{p,a,i}, D)$ (on note où se trouve la tête, dans quel état, et sur quel symbole)
$\hat{\delta}(m_{i,p,D}, a) = (m_{i+1}, T_{p,a,i}, D)$ si $a \neq B$ (on note où se trouve la tête, dans quel état, et sur quel symbole)
$\hat{\delta}(m_{i,p,D}, B) = (m^{\#}_{i+1}, T_{p,a,i}, D)$ (on note où se trouve la tête, dans quel état, et sur quel symbole et on va vérifier si le B était le dernier)
$\hat{\delta}(m^{\#}_{i+1}, x) = (m_{i+1}, x, D)$ pour $x \neq \#,\ i \in \{1, \ldots, k\}$
$\hat{\delta}(m^{\#}_{i+1}, \#)$ appelle $D_{m_{i+1}}$ pour $i \in \{1, \ldots, k\}$
$\hat{\delta}(m_{k+1}, x) = (m_{k+1}, x, G)$ pour $x \neq \$$ (on revient au début du ruban)
$\hat{\delta}(m_{k+1}, \$) = (m_1, \$, D)$ (on recommence)

Exercice 1 *Donnez exactement $\hat{Q}$ et $\hat{\Gamma}$ (en utilisant Γ, Σ, Q de M).*

Exercice 2 *Les cas où une tête tomberait à gauche du début du ruban dans M n'est pas traité ici. Ecrivez les transitions manquantes.*