Machines de Turing

Laure Danthony

Table des matières

1 Définitions **2**
1.1 Définition heuristique 2
1.2 Définition formelle 2

2 Résultats **4**
2.1 Equivalence entre machines de Turing 4
2.2 Codage de Kleene 4

3 Théorèmes importants **6**
3.1 Machine universelle 6
3.2 Théorème s-n-m 6

1 Définitions

1.1 Définition heuristique

On réalise ici la description d'une MT à k rubans.

DÉFINITION 1 (ORDINATEUR?)

- On dispose de k rubans avec une infinité de cases et une seule tête de lecture par ruban. Les cases des k rubans contiennent des lettres d'un alphabet Γ, ou des caractères spéciaux comme \$ ou B (blanc) qui servent de délimiteurs ou d'informations.

- Une **configuration** est la donnée des lettres des rubans, de la position des têtes de lecture, ainsi que de leur état *commun* $q \in Q$. On passe d'une configuration à une autre grâce à la donnée d'une **fonction de transition** δ qui contient les nouvelles lettres à écrire, le nouvel état $\widetilde{q}$ et les mouvements des têtes effectués.

- Une configuration **code** pour un mot u si le mot est sur le premier ruban, les autres rubans sont "blancs", les têtes sont en position 0.

- Le calcul de u **s'arrête** sur la MT si, partant d'une configuration c on arrive à une configuration d'état $q \in Q_{accept}$ (acceptant) qui code pour un mot v qui est sur le premier ruban à la place 1.

- Une MT **calcule** une fonction ssi elle s'arrête sur $v = f(u)$ si u appartient au domaine de définition, et sinon elle ne s'arrête pas.

DÉFINITION 2

Une fonction est **calculable** si il existe une MT qui la calcule.

1.2 Définition formelle

DÉFINITION 3 (MT)

Une machine de Turing à k rubans est un 6-uplet :

$$\mathcal{M} = (\mathcal{A}, k, B, Q, Q_{accept}, \delta)$$

où :

- $\mathcal{A}$ est l'alphabet de travail ;

- $B \in \mathcal{A}$ (blanc) est la première lettre de $\mathcal{A}$;

- Q est l'ensemble des états ;

- $Q_{accept} \subset Q$ est l'ensemble des états d'acceptation ;

- δ "fonction de transition" : $\mathcal{A}^k \times Q \to \mathcal{A}^k \times Q \times \{-1, 0, +1\}^k$: $\mathcal{A}^k$ représentent les k lettres présentes dans les têtes, ces têtes étant dans un état $q \in Q$, il y a écriture de k lettres, passage dans un nouvel état, et k mouvements des têtes (arrière, sur place ou avant).

DÉFINITION 4 (RUBAN)
Une **configuration** de la machine de Turing $\mathcal{M}$ est une application :

$$\begin{array}{rccc} c: & \mathbb{Z} & \longrightarrow & \big(A \times (Q \cup \$)\big)^k \\ & j & \longmapsto & \Big(\big(a_1(j), q_1(j)\big), \dots, \big(a_k(j), q_k(j)\big)\Big) \end{array}$$

telle que :($ signifie "il n'y a pas de tête à cette case")

- $\forall i \in [1, k], \exists! j, q_i(j) \neq \$$(les têtes sont à une seule position).
- et $q_i(j) = q \Rightarrow \forall i, j \big(q_i(j) = \$$ ou $q_i(j) = q$ (les têtes sont au même état q)

Plus simplement :

- $a_i(j)$ est la j-ième case du i-ème ruban.
- $q_i(j)$ est l'état de la j-ième *case* du i-ème ruban : si $q_i(j) = \$$, alors la tête n'est pas dessus ; sinon $q_i(j) = q$ et l'état de la machine est q.

DÉFINITION 5
Une configuration c **conduit** directement à une configuration c' (noté $c \vdash c'$) si :

- $\forall i, j$, si $q_i(j) = \$$, alors $a_i'(j) = a_i(j)$ (il n'y a pas de changement si la tête n'est pas dessus) ;

Soit j_i l'unique entier tel que $q_i(j_i) \neq \$$ (j_i est la case du ruban i où est la tête).
Alors si :

$$\delta\Big(\big(a_1(j_1), a_2(j_2), \dots, a_k(j_k)\big), q\Big) = \big((\widetilde{a_1}, \widetilde{a_2}, \dots \widetilde{a_k}), \widetilde{q}, mv_1, mv_2, \dots, mv_k)\big)$$

- $\forall i, a_i(j_i) = \widetilde{a_i}$
- $\forall i, q_i'(j) \neq \$$ ssi $j = j_i + mv_i$
- $\forall i$, si $q_i'(j) \neq \$$ alors $q_i'(j) = \widetilde{q}$

Les deux derniers points illustrent la position et l'état des nouveaux curseurs.

DÉFINITION 6 (CLÔTURE)
c **conduit** à c^* ssi il existe m et des configurations c_i, pour $i \in [1, m]$, telles que : $c = c_0$, $c_m = c^*$, et $\forall i, c_i \vdash c_{i+1}$.

NOTATION On note alors $c \overset{*}{\vdash} c^*$

DÉFINITION 7
Soit Γ un alphabet fini avec $\Gamma \subset \mathcal{A} \setminus B$. Une configuration c de $\mathcal{M}$ **code** pour le mot $u = u_0 u_1 \dots u_{l-1} \in \Gamma^*$ ssi :

- $a_1(0) = B$ et $\forall j \in [1, l], a_1(j) = u_{i-1}$ et $\forall j > l, a_1(j) = B$ (le mot est sur le premier ruban, il démarre en position $a_1(1)$)
- $\forall i \neq 1, \forall j, a_i(j) = B$ (les autres rubans sont blancs)
- $\forall i, q_i(0) = q_0$ (les curseurs sont en position 0)

- $\forall i, \forall j \neq 0, q_i(j) = \$$

DÉFINITION 8
Le calcul de u sur $\mathcal{M}$ **s'arrête** si, partant de c codant u, il existe c^* avec $c \overset{*}{\vdash} c^*$ tel que :

- $\forall i, j, q_i(j) \neq \$ \Rightarrow q_i(j) \in Q_{accept}$ (l'état de c^* est dans Q_{accept}) ;
- c^* code pour un mot v (de taille p) qui est sur le premier ruban) , i.e. $a_1(0) = B$ et $\forall j \in [1, p], a_i(j) = v_{j-1}$ et $\forall j > p, a_1(j) = B$.

DÉFINITION 9
Soit f une fonction définie sur une partie de Γ^* et à valeurs dans Γ^*. $\mathcal{M}$ **calcule** f si :

- pour tout mot $u \in \Gamma^*$, $u \in \mathcal{D}(f) \Leftrightarrow$ le calcul de u sur $\mathcal{M}$ s'arrête.
- si le calcul de $\mathcal{M}$ s'arrête, alors le(s) résultat(s) v est(sont) égal(égaux) à $f(u)$.

DÉFINITION 10
Une fonction f est dite **calculable** s'il existe une machine de Turing qui la calcule.

REMARQUE 1 On représente le fonctionnement d'une machine de Turing par un graphe de transitions.

2 Résultats

2.1 Equivalence entre machines de Turing

PROPOSITION 1 *Une machine de Turing à k rubans telle qu'on l'a définie est équivalente à :*

1. *une machine à 1 ruban semi-infini ;*
2. *avec délimiteurs (G et D);*
3. *sur l'alphabet $\{0, 1, B\}$.*

2.2 Codage de Kleene

DÉFINITION 11 (CODAGE D'UNE MT)
A une machine de Turing on fait correspondre un mot sur l'alphabet $\{0, 1, E, F, \$\}$ en codant l'alphabet (on numérote les lettres, et on utilise le changement de base vers la base 2), les états (idem), les mouvements (code sur 2 bits : 01 ou 00 10), la fonction de transition (pour une transition donnée, \$ délimite la transition, E segmente, F si si l'état codé est acceptant). Ensuite on concatène les transitions obtenues et on code chacune des lettres de l'alphabet $\{0, 1, E, F, \$\}$ sur l'alphabet $\{0, 1\}$, ainsi on récupère un mot codant pour la machine de Turing considérée.

PROPOSITION 2 *Il existe une machine de Turing qui calcule la fonction* $\{0,1\}^* \to \{0,1\}$ *suivante :*

$$u \longmapsto \begin{cases} 1 & \textit{si } u \textit{ est le codage de Kleene d'une MT} \\ 0 & \textit{sinon} \end{cases}$$

PROPOSITION 3 *Il existe une MT qui calcule la fonction* $\{0,1\}^* \to \{0,1\}^*$ *suivante :*

$$u \longmapsto \begin{cases} x & \textit{si } u = \overline{n} \textit{ donne le codage } x \textit{ de Kleene de la } n\textit{-ième MT dans l'ordre lexicographique} \\ 0 & \textit{sinon} \end{cases}$$

REMARQUE 2 $\overline{n}$ désigne ici le codage en base 2 de n.

NOTATION On note φ_n la machine de Turing (ou son codage), résultat de la machine précédente sur l'entrée $\overline{n}$. φ_n est la n-ième machine de Turing.

REMARQUE 3 Cette numérotation des MT dépend du codage utilisé. On prendra toujours le même.

DÉFINITION 12 (CODAGE D'UNE CONFIGURATION)
On code de la même façon les configurations en ajoutant dans un premier temps des lettres à l'alphabet, puis en les codant ensuite sur l'alphabet $\{0,1\}$.

PROPOSITION 4 *Il existe une MT qui, sur l'entrée* $u \in \{0,1\}$*, s'arrête :*

- *sur* 1 *si* u *code pour une configuration de support fini pour une MT.*
- *sur* 0 *sinon.*

PROPOSITION 5 *Il existe une MT qui, sur l'entrée* $v \in \{0,1\}$*, s'arrête sur :*

- *le* n*-ième codage possible de configurations à support fini* ($v = \overline{n}$)
- 0 *sinon.*

PROPOSITION 6 *Il existe une MT qui sur l'entrée* $u \in \{0,1\}$*, s'arrête ssi* u *se termine par un 1 et si elle s'arrête, elle fait correspondre le codage de la configuration initiale de toute MT sur l'entrée* u.

PROPOSITION 7 *Il existe une MT qui, sur l'entrée* $u \in \{0,1\}$ *:*

- *ne s'arrête pas si* u *ne code pas pour un entier.*
- *s'arrête sur :* 1 *si* u *code pour* (n,m) *et la configuration codée par* $\overline{m}$ *convient à la MT codée par* $\overline{n}$*., et* 0 *sinon.*

3 Théorèmes importants

3.1 Machine universelle

DÉFINITION 13
Une machine de Turing est dite **universelle** si, quels que soient x et y de $\mathbb{N}$, sur l'entrée $z = \overline{<x, y>}$, elle s'arrête ssi la x-ième machine de Turing (φ_x) :

- d'une part s'arrête sur la configuration initiale codant l'entrée y ($\varphi_x(y)$ "s'arrête").
- et ne n'arrête pas sinon.

THÉORÈME 1 *Il existe une machine de Turing universelle*

3.2 Théorème s-n-m

Dans cette section, nous établissons l'existance d'une fonction qui nous sera bien utile dans la suite.

THÉORÈME 2 *Pour tous entiers m et n, il existe une fonction s_n^m définie sur $\mathbb{N}^{m+1}$ et à valeurs dans $\mathbb{N}$, croissante en chacune de ses variables, calculable par MT, telle que, $\forall x_1, \ldots, x_m, y_1, \ldots, y_n, a \in \mathbb{N}$:*

$$\varphi_a(<x_1, \ldots, x_m, y_1, \ldots, y_n>) = \varphi_{s_n^m(a, x_1, \ldots x_m)}(<y_1, \ldots, y_n>)$$