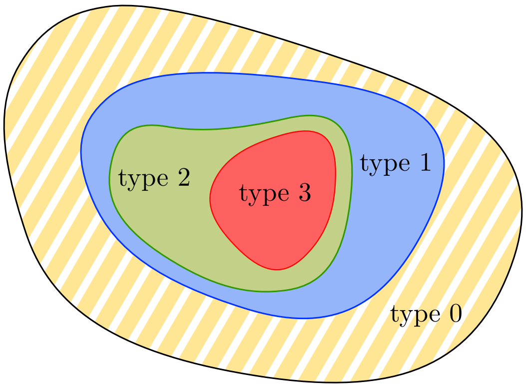
Introduction à la calculabilité

Jean-Baptiste Guet
Julien Henry

Ensimag 1A

Table des matières

Remarque. Hiérarchie de Chomsky



$$\text{type } 3 \subset \text{type } 2 \subset \text{type } 1 \subset \text{type } 0$$

Nous allons essayer de trouver des langages capables de reconnaître des langages de type 0, que l'on appelle langages récursivement énumérables :

- type 0 : langages récursivement énumérables
- type 1 : langages sous-contexte
- type 2 : langages hors contexte
- type 3 : langages réguliers

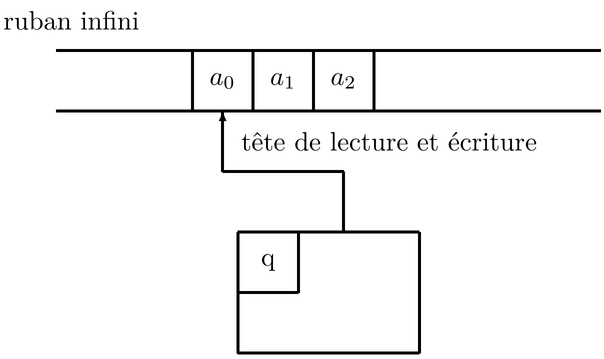
1 Modèle de calcul : la machine de Turing

1.1 Définition de la machine de Turing

L'un des modèles de calcul est la machine de Turing.

***Définition* 1.** Machine de Turing
Soit $M = (Q, V, \delta, B, q_0, F)$ avec :

Q	:	ensemble fini d'états
V	:	Vocabulaire
δ	:	fonction de transition tel que $\delta : Q \times V \longrightarrow Q \times V \times \{D, G, N\}$
B	:	le blanc $\in V$
q_0	:	état initial $\in Q$
F	:	états finaux $\subseteq Q$



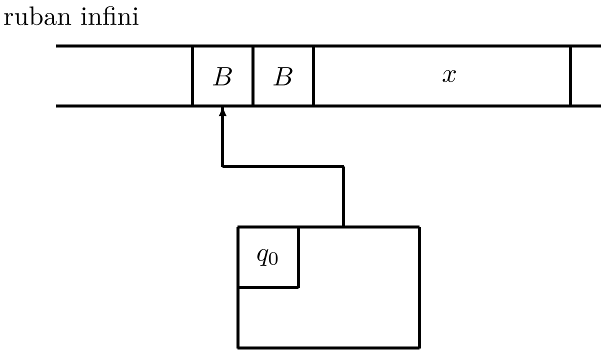
On peut expliquer la fonction de transition :

$$\delta :: Q \times V \longrightarrow Q \times V \times \{D, G, N\}$$

$$\delta(q, a) = (q', b, D)$$

CONFIGURATION : xqy

CONFIGURATION INITIALE : q_0y

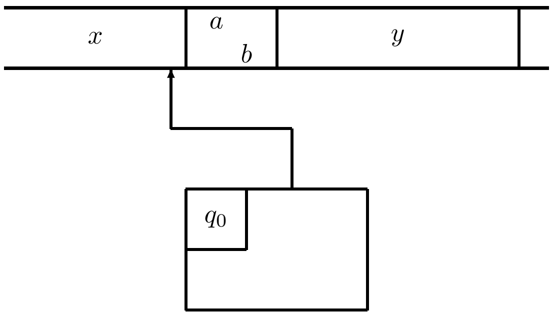


CONFIGURATION TERMINALE : $xqy \not\Rightarrow$ (ne peut pas se réécrire)

RÉÉCRITURE (TRANSFORMATION) DE CONFIGURATIONS :

- $xqay \Longrightarrow xbq'y$ si

$$\delta(q, a) = (q', b, D)$$



- $xqay \Longrightarrow xq'by$ si

$$\delta(q, a) = (q', b, N)$$

- $xaqby \Longrightarrow xq'acy$ si

$$\delta(q, b) = (q', c, G)$$

1.2 Langage accepté par une machine de Turing

Soit $M = (Q, V, \delta, B, q_0, F)$ une machine de Turing. Le langage sur le vocabulaire $W \subseteq V$ accepté par M est défini par :

$$L(M) = \{\omega \in W^* /\ \exists\ q \in F,\ \exists\ \alpha_1, \alpha_2 \in V^*,\ q_0\omega \Rightarrow^* \ \alpha_1 q \alpha_2 \ \not\Rightarrow\}$$

Exemple. Soit la machine de Turing qui accepte le langage $\{a^nb^n/n \geq 1\}$. On note :
- $Q = \{q_i, i = 0..4\}$
- $V = \{a, b, B, X\}$
- $F = \{q_4\}$
- $W = \{a, b\}$

$$BBq_0XX$$

$$\begin{array}{rcl}
\delta(q_0, a) & = & (q_1, B, D) \\
\delta(q_0, X) & = & (q_3, X, D) \\
\delta(q_1, a) & = & (q_1, a, D) \\
\delta(q_1, b) & = & (q_2, X, G) \\
\delta(q_1, X) & = & (q_1, X, D) \\
\delta(q_2, X) & = & (q_2, X, G) \\
\delta(q_2, Q) & = & (q_2, a, G) \\
\delta(q_2, B) & = & (q_0, B, D) \\
\delta(q_3, X) & = & (q_3, X, D) \\
\delta(q_3, B) & = & (q_4, B, N)
\end{array}$$

1.3 Fonction calculée par une machine de Turing

On note :

$$f: \begin{array}{rcl} D & \longrightarrow & A \\ x & \longmapsto & f(x) \end{array}$$

CONFIGURATION INITIALE : $q_0 x'$, où x' est une représentation de x.
RÉSULTAT : $f(x) : q_0 x' \Rightarrow^* \alpha_1 q \alpha_2 \not\Rightarrow$. $\alpha_1\alpha_2$ est une représentation de $f(x)$.

Exemple.

$$f: \begin{array}{rcl} \mathbb{N} & \longrightarrow & \mathbb{N} \\ i & \longmapsto & 2i \end{array}$$

i est représenté par $\underbrace{11\ldots1}_{i \text{ fois}}$.

$$\begin{array}{rcl}
q_0 & \Rightarrow^* & |||q_1B \\
 & \Rightarrow & ||q_2|\# \\
 & \Rightarrow^* & q_2B|||\# \\
 & \Rightarrow & q_3|||\# \\
 & \Rightarrow & Bq_4||\# \\
 & \Rightarrow & ||\#q_4B \\
 & \Rightarrow & ||\#|q_5B \\
 & \Rightarrow & ||\#|q_6| \\
 & \Rightarrow^* & q_6B||\#|| \\
 & \Rightarrow & q_3||\#|| \\
 & \Rightarrow^* & q_3|\#|||| \\
 & \Rightarrow^* & q_3\#|||||| \\
 & \Rightarrow & qB||||||
\end{array}$$

$$\begin{array}{rcl}
\delta(q_0, B) & & \text{Non défini} \\
\delta(q_0, |) & = & (q_1, |, D) \\
\delta(q_1, |) & = & (q_1, |, D) \\
\delta(q1, B) & = & (q_2, \#, G) \\
\delta(q_2, |) & = & (q_2, |, G) \\
\delta(q_2, B) & = & (q_3, B, D) \\
\delta(q_3, \#) & = & (q, B, N) \\
\delta(q_3, |) & = & (q_4, B, D) \\
\delta(q_4, |) & = & (q_4, |, D) \\
\delta(q_4, \#) & = & (q_4, \#, D) \\
\delta(q_4, B) & = & (q_5, |, D) \\
\delta(q_5, B) & = & (q_6, |, D) \\
\delta(q_6, |) & = & (q_6, |, G) \\
\delta(q_6, \#) & = & (q_6, \#, G) \\
\delta(q_6, B) & = & (q_3, B, D)
\end{array}$$

Définition 2. Une fonction partielle est dite calculable si et seulement si il existe une machine de Turing qui la réalise.

Définition 3. Une fonction totale est dite calculable si et seulement si il existe une machine de Turing qui la réalise et qui s'arrête toujours.

***Définition* 4.** Une fonction est dite partielle récursive si et seulement si elle est partielle calculable.

***Définition* 5.** Une fonction est dite récursive si elle est totale et calculable.

2 Extensions de la machine de Turing

On pourrait essayer de faire une extension de la machine de Turing avec par exemple :
- avec plusieurs têtes de lecture et d'écriture.
- avec plusieurs rubans et têtes de lecture.
- non déterministe

On peut imaginer n'importe quelle extension, on pourra toujours réaliser une telle machine avec une machine de Turing de base. De même, si l'on imagine une machine de Turing avec un ruban semi-infini, on pourra la reproduire à l'aide d'une machine de Turing classique. De plus, on constate que la machine de Turing initiale est toujours meilleure. Un type de machine de Turing important est les machines de Turing sur le vocabulaire $\{1, B\}$.

2.1 Machines RAM

***Définition* 6.** Une machine RAM (Random Access Memory) est une succession d'instructions

$$p_1, p_2 \dots p_n, n \leq 0$$

de l'une des formes suivantes :

$$\begin{array}{lccc} R_i & \longleftarrow & k & k \in \mathbb{N} \\ R_i & \longleftarrow & R_i + 1 & \\ R_i & \longleftarrow & R_i - 1 & Si\ R_i > 0 \\ R_i & \longleftarrow & R_j & \end{array}$$

Si $R_i = 0$, aller à j.
Aller à j

2.2 Machine de Turing Universelle (MTU)

2.2.1 Définition

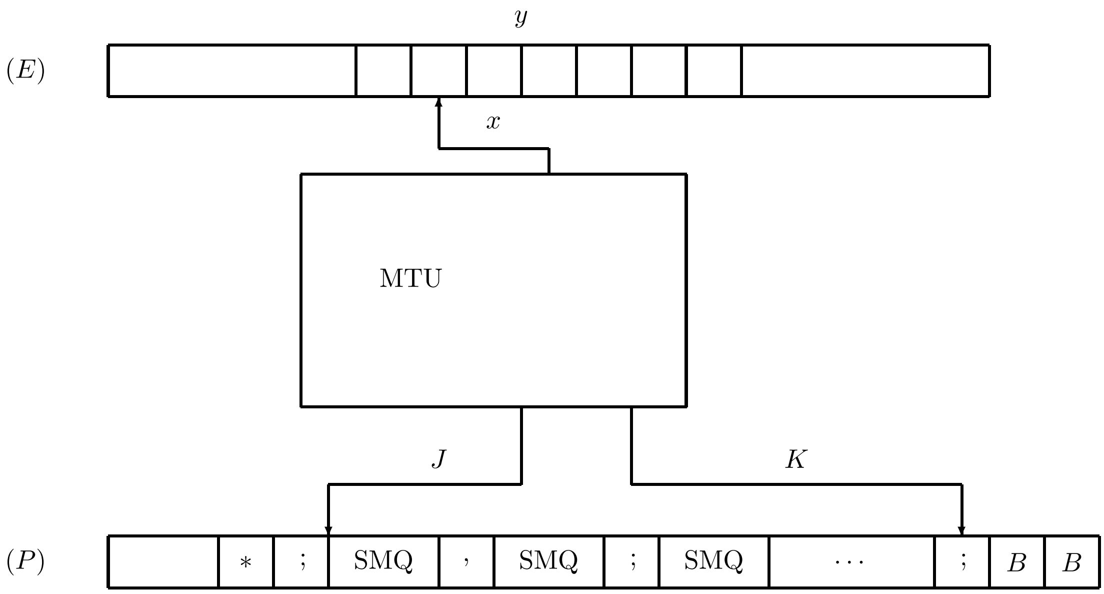
On voudrait définir une machine de Turing qui pourrait simuler n'importe quelle machine de Turing, y compris elle même. On la formalise ainsi :
- Soit $MTU(x, y)$ la machine qui simule la MTx sur la donnée y, où $(x, y) \in \mathbb{N}^2$.
- L'ensemble des chaînes possibles est dénombrable, d'où l'existence d'une bijection entre ces chaînes et l'ensemble des naturels $\mathbb{N}$ On peut donc utiliser un codage de ces chaînes avec des naturels.
- y est la donnée de la MTx.
- On suppose la machine MTx sur le vocabulaire $\{1, B\}$

$$\begin{array}{lcl} \delta(q, 1) & = & (q', S, M) \quad S \in \{', B\}, M \in \{D, N\} \\ \delta(q, B) & = & (q, S, M) \end{array}$$

Ainsi, on a deux fonctions qui sont codées comme suit :

q_i	B	q'	S_2	M	,	q_i	1	q''	S_i	M

avec $M \in \{D, G, N, H\}$ On remarque que l'on peut se passer des états initiaux. En effet, on sait que l'on a deux fonctions, et l'on sait où elles sont définies. De plus, on sait que la première fonction est définie pour un blanc, et que la deuxième est définie pour un 1, donc on peut s'en passer.



fonction de transition de la MTx

2.2.2 Simulation

Principe de l'algorithme

1. – *Si* "I lit "|" ", *alors* déplacer J à droite jusqu'au premier symbole après ","
   – J lit le symbole à écrire sur le ruban (E)
2. I écrit sur E le caractère lu par J
3. Déplacer J à droite (J lit le mouvement que I doit exécuter)
4. – Si le symbole lu par J appartient à $\{G, D, N\}$, alors I exécute le mouvement lu par J
   – Si le symbole lu par J est H, la MTU s'arrête.
5. Déplacer J à droite (J lit le premier "1"de l'état suivant)
6. Tant que "J lit "|" " , faire déplacer J et K à droite.
7. Déplacer J à gauche jusqu'à "∗"
8. tant que "K lit un "B" " faire
   – déplacer K à gauche
   – déplacer J juqu'au prochain j
9. Déplacer J à droite

Problème de l'arrêt. (halting problem)

$$\begin{array}{rcl} h : \mathbb{N} \times \mathbb{N} & \longrightarrow & \mathbb{N} \\ (x, y) & \longmapsto & h(x, y) = \begin{cases} 1 & \textit{si } MTU(x, y) \textit{ existe} \\ 0 & \textit{sinon} \end{cases} \end{array}$$

h est une fonction. On se demande si h est calculable.

Démonstration. On suppose que h est calculable. Dans ce cas, la fonction g suivante est calculable :

$$\begin{array}{rcl} g : \mathbb{N} \times \mathbb{N} & \longrightarrow & \mathbb{N} \\ (x, y) & \longmapsto & g(x, y) = \begin{cases} 1 & \text{si } h(x, x) = 0 \\ \text{indéfini} & \text{sinon} \end{cases} \end{array}$$

Soit $z \in \mathbb{N}$ une machine de Turing qui réalise g.

$$g(z) = 1 \Leftrightarrow MTU(z, z) \text{ boucle} \Leftrightarrow g(z) \text{ indéfini}$$

$$g(z) \text{ indéfini} \Leftrightarrow MTU(z, z) \text{ s'arrête} \Leftrightarrow g(z) \text{ défini}$$

□

***Définition* 7.** L'ensemble A est dit récursif si et seulement si sa fonction caractéristique :

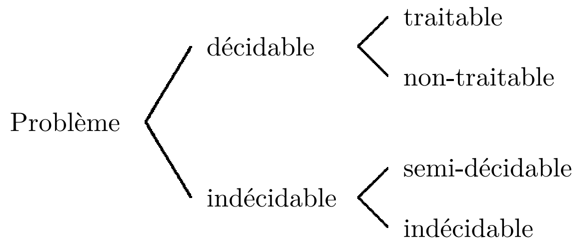
$$\begin{cases} \chi(x) = 1 \ si \ x \in A \\ \chi(x) = 0 \ sinon \end{cases}$$

est totale récursive.

***Définition* 8.** Un ensemble A est dit récursivement énumérable si et seulement si il est le domaine d'unr fonction partielle récursive.

***Définition* 9.** DÉCIDABILITÉ

1. Un problème est décidable si et seulement si il exdiste une machine de Turing qui résout le problème et qui s'arrête toujours.
2. Un problème est dit indécidable si il n'est pas décidable.
3. Un problème est semi-décidable si il existe une machine de Turing qui le résout. Il n'y a aucune condition sur l'arrêt.



***Définition* 10.** Un problème P_1 se réduit au problème P_2 si il existe une fonction récursive f_1 qui réduit toute instance I de P_1 en une instance $f(I)$ de P_2 tel que :

I admet une solution si et seulement si $f(I)$ admet une solution.

$$P_1 \overset{f}{\curvearrowright} P_2 \tag{1}$$

$$I \rightsquigarrow f(I) \tag{2}$$

Si P_1 se réduit en P_2, alors :

$$\begin{aligned} P_2 \text{ décidable} &\Longrightarrow P_1 \text{ décidable} \\ P_2 \text{ indécidable} &\Longrightarrow P_1 \text{ indécidable} \end{aligned}$$

2.2.3 Problème de correspondance de Post

Soient deux listes de k chaînes sur le vocabulaire V

$$\begin{aligned} A &= \omega_1, \ldots, \omega_k \\ B &= x_1, \ldots, x_k \end{aligned}$$

On se demande s'il existe une suite d'indices $i_1, \ldots, i_n$ avec $n \geq 1$ et $1 \leq i_j \leq k, j = 1..n$ telle que :

$$\omega_{i_1}\omega_{i_2}\ldots\omega_{i_n} = x_{i_1}x_{i_2}\ldots x_{i_n}$$

Exemple. Soit $V = \{0, 1\}$ et $K = 3$ On pose :

$$\begin{aligned} A : &\quad \omega_1 = 1, \ \omega_2 = 10111, \ \omega_3 = 10 \\ B : &\quad x_1 = 111, \ x_2 = 10, \ x_3 = 0 \end{aligned}$$

Solution $n = 4$:

$$i_1 = 2, \ i_2 = 1, \ i_3 = 1, \ i_4 = 3$$

$$\omega_2\omega_1\omega_1\omega_3 = x_2x_1x_1x_3$$

$$101111110 = 101111110$$

Théorème 1. *Le problème de l'arrêt peut être réduit en un problème de Post Modifié, qui lui même peut être réduit en un problème de Post.*

2.2.4 Problème de Post Modifié (PPM)

Soient deux listes de k chaînes sur le vocabulaire V. On cherche s'il existe une suite d'indices $i_1 \dots i_m,\ m \geq 0$ telle que :

$$\omega_1 \omega_{i_1} \omega_{i_2} \dots \omega_{i_m} = x_1 x_{i_1} x_{i_2} \dots x_{i_m}$$

Lemme 1. *Le problème de Post Modifié se réduit en problème de Post.*

Démonstration.

$$\begin{array}{rl} A: & \omega_1,\ \omega_2, \dots,\ \omega_k \\ B: & x_1,\ x_2, \dots,\ x_k \end{array}$$

On transforme les données A et B en A' et B' telles que $PPM(A,B)$ admette une solution si et seulement si $PP(A',B')$ admet une solution.

Soient $\Box$ et $\Diamond$ deux nouveaux symboles $\notin V$. On construit deux nouvelles listes :

$$\begin{array}{rl} A': & y_0\ y_1 \dots\ y_{k+1} \\ B': & z_1\ z_2 \dots\ z_{k+1} \end{array}$$

Avec :
- $y_0 = \Box y_1$.
  Pour $1 \leq i \leq k$, y_i est obtenu à partir de ω_i en insérant $\Box$ après chaque sumbole $y_{k+1} = \Diamond$.
- $z_0 = z_1$.
  Pour $1 \leq i \leq k$, z_i est obtenu à partir de x_i en insérant $\Box$ après chaque sumbole $z_{k+1} = \Box\Diamond$.

On montre maintenant que :

$$PPM(A,B) \text{ admet une solution} \Leftrightarrow \text{si } PP(A',B') \text{ admet une solution.}$$

$\Longrightarrow$ Si $i_1, \dots, i_m$ est une solution pour $PPM(A,B)$

$$\omega_1 \omega_{i_1} \omega_{i_2} \dots \omega_{i_k} = x_1 x_{i_1} x_{i_2} \dots x_{i_k}$$

Alors $0, i_1, \dots, i_m, k+1$ est une solution pour $PP(A',B')$.

$\Longleftarrow$ Si $i_1, \dots i_m$ est une solution de $PP(A',B')$. Alors $i_1 = 0$ et $i_m = k+1$.

En effet, y_0 et z_0 sont les seules chaînes de même indice qui commencent avec le même symbole. De même, y_{k+1} et z_{k+1} sont les seules chaînes de même indice qui se terminent avec le même symbole $\Diamond$.

Soit j le plus petit entier tel que $i_j = k+1$. Alors $i_1, i_2, \dots i_j$ est aussi une solution de $PP(A',B')$. On en déduit que $i_2 \dots i_{j+1}$ est une solution de $PPM(A,B)$.

□

Lemme 2. *Le problème de l'arrêt des machines de Turing se réduit au PPM.*