

Les Machines de Turing

Épisode 1

Définition

CMDR Charlotte THOMAS
USS Versailles, R9

6 juin 2022

Résumé

Les termes de "Machine de Turing" ou encore de "Turing-complet" sont maintenant commun sur internet, même lors de simples articles pour le grand public. Ces termes réfèrent à des concepts en *informatique théorique* qui est la science de l'étude de l'informatique au sens mathématique (et non au sens physique comme construire des ordinateurs) plus particulièrement dans la discipline qu'on appelle la *calculabilité*.

Cependant, ces termes ne sont pas toujours bien défini, cette série d'articles va plonger dans l'univers de l'informatique théorique et de la calculabilité, mais tout d'abord définissons ce qu'est une *Machine de Turing*..

Le niveau de lecture de cet article est difficile, surtout la partie formelle, mais est intéressante au niveau de l'histoire du concept, une présentation plus en détail, avec des exemples et une implémentation dans de nombreux langages (python, OCaml, B#, etc.) sera faite lorsque ce sera le tour avec la liste des articles sur l'informatique théorique depuis 0.

Table des matières

1 Histoire	2
2 Définition Intuitive	3
3 Définition formelle	4

1 Histoire

Avant de définir intuitivement et formellement une telle machine, il faut en donner le contexte historique, tout d'abord une Machine de Turing n'est pas une machine au sens réel, ce n'est pas un ordinateur ou une calculatrice, (bien que de nombreux chercheurs ont depuis construits de telles machines purement mécaniquement) c'est un *concept mathématique*, un outil de calcul inventé par Turing dans son papier de 1936 sur la calculabilité, une machine de Turing, (qu'il appelait *automatic machine*) est un modèle permettant d'exécuter un *algorithme*. Un algorithme étant simplement une suite d'instruction

Algorithme 1 : Un exemple d'algorithme simple pour servir des céréales
Entrée : Céréales, Lait, Bol
Résultat : Bol de Céréales
1 Mettre les céréales dans le bol;
2 Verser du lait jusqu'à hauteur des céréales;

Un algorithme est une suite d'instructions finie (i.e il n'y a pas un nombre infini d'instructions), *l'algorithme* est l'étude des algorithmes, s'ils finissent ou pas, comment les exécuter, leur rapidité d'exécution (dite *complexité temporelle*), la place qu'ils prennent en mémoire (dite *complexité spatiale*).

Une fonction qui peut être représentée à l'aide d'un algorithme (comme ci-dessus pour créer un bol de céréale ou une fonction pour compter jusqu'à 1000) est dite *calculable*, l'étude de ces fonctions et leur application en informatique théorique est *la calculabilité* et c'est ce domaine qui a donné naissance aux Machines de Turing.

2 Définition Intuitive

Avant de définir formellement un tel objet, il est utile d'en faire une définition dites "intuitive" en effet, on peut la dessiner et elle est bien plus facile à comprendre que la définition qui va suivre.

Une Machine de Turing peut être représenté par un ruban qui s'étend à l'infini à droite, avec des cases, ce ruban représente à la fois la mémoire vive de la machine et l'entrée/sortie du programme, l'entrée étant l'état de la mémoire vive *avant* de "démarrer" la machine et la sortie étant l'état de la mémoire vive à la *sortie* du programme (quand, et si, il s'arrête)

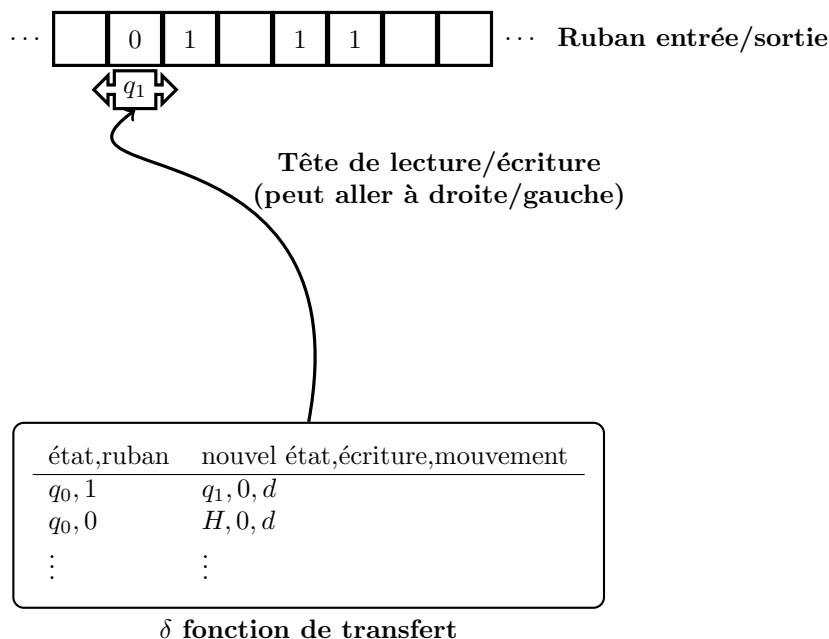


FIGURE 1 – Représentation d'une Machine de Turing [1]

Au fur et à mesure du programme (qui est la fameuse fonction δ représenté par le tableau de transfert) la tête de lecture/écriture va bouger et modifier le ruban en fonction du programme et de l'entrée, une fois arrivé dans l'état de fin (par exemple l'état H pour HALTE) le programme s'arrête et le ruban devient alors la sortie du programme. Un problème dont on reviendra plus tard est savoir si oui ou non un programme s'arrête pour une certaine entrée voir pour toute, ce problème nommé "problème de l'arrêt" a été grandement discuté par Turing lui même dans plusieurs papiers, grâce aux outils qu'il a développés il a pu prouver en 1937 que le problème était *indécidable*, c'est-à-dire que dans nos systèmes de logique usuels il n'existe pas de preuve qu'il soit vrai, ou de preuve qu'il soit faux. [2]

3 Définition formelle

Voici enfin la partie où l'on a défini formellement ce qu'est qu'une machine de Turing, il y a de nombreuses définitions formelles développées par-dessus le concept de Turing, la définition communément utilisée est celle de Lewis dans son livre de 1982, puis la réédition en 1998, *Elements of the Theory of Computation* au chapitre 4, définition 4.1.1, page 181 [3] que suit

Il définit une machine de Turing comme un 5-uplet $(K, \Sigma, \delta, s, H)$ avec K l'ensemble fini d'états possible de la machine, Σ est l'alphabet qui contient le caractère nul et le caractère marquant le début du ruban, $s \in K$ est l'état initial de la machine, δ est la fonction de transition des états formellement $(K - H) \times \Sigma \rightarrow K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ (elle prend en entrée un couple formé d'un état non sortant et d'un état du ruban lu et renvoie un couple du nouvel état de la machine et du nouvel état à écrire ou s'il faut bouger à gauche ou à droite, ce qui revient à un nouvel état à écrire ET s'il faut bouger à gauche/droite), et enfin $H \subset K$ est l'ensemble des états finaux dit "de halte".

Pour une machine de Turing dite, *binaire*, avec le caractère 2 comme caractère vide, il y a un seul état sortant dénoté \bar{H} (pour éviter de le confondre avec le H de la définition, ils sont ensuite confondus) les autres états sont des entiers. Les caractères $r, l, *$ respectivement signifient aller à droite, aller à gauche, et ne pas bouger. Enfin la fonction δ est découpé en trois fonctions (pour pouvoir l'implémenter avec uniquement des matrices normales) $\delta_1, \delta_2, \delta_3$ donnant respectivement le nouvel état, ce qu'il faut écrire sur le ruban, et comment il faut bouger la tête on a donc quelque chose comme ci-dessous.

$$\begin{aligned} K &= H \cup Q & Q &\subset \mathbb{N} \\ \Sigma &= \{0, 1, 2\} \\ s &= 0 \\ H &= \{\bar{H}\} \\ \delta_1 : Q \times \Sigma &\rightarrow K \\ \delta_2 : Q \times \Sigma &\rightarrow \Sigma \\ \delta_3 : Q \times \Sigma &\rightarrow \{r, l, *\} \end{aligned}$$

Références

- [1] Inspiré de TeX Examples. <https://texample.net/tikz/examples/turing-machine-2/>.
- [2] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1) :230–265, 1937.
- [3] Harry R Lewis and Christos H Papadimitriou. Elements of the theory of computation. *ACM SIGACT News*, 29(3) :62–78, 1998.