Calculabilité, Complexité,

2 – Machines de Turing vers le théorème de la Halte

P. Berthomé

INSA Centre Val de Loire
Département STI

27 septembre 2020

Mimes de machines de Turing

Objectif

- Différents modèles de Machines de Turing
- Puissance de calcul et reconnaissance de langage
- Tous ces modèles sont équivalents
- Moyennant un surcoût.

Dans quel but ?

- Utiliser la Machine de Turing la mieux adaptée
- Être capable de numéroter les MT
- Pour avoir des théorèmes généraux

Quelques variantes de Machines de Turing

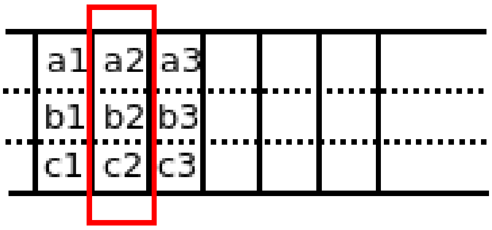
Ruban bi-infini de manière plus formelle

- Application $\rho : \mathbb{Z} \longrightarrow \Sigma$
- Convention : $\{z \mid \rho(z) \neq B\}$ est fini.
- On peut ajouter pour le départ :
  - Le mot d'entrée est sur la partie positive
  - La première lettre du mot d'entrée est sur la case 1
  - la RWH est sur la case 0

Ruban semi-infini

- Même chose si ce n'est que l'on remplace $\mathbb{Z}$ par $\mathbb{N}$.
- Attention à ne pas faire sortir la RWH du ruban

Pistes sur les rubans



Pistes

- Mettre plusieurs informations dans une seule case
- *Souder* plusieurs rubans
- Idée : Augmenter l'alphabet utilisé : Σ^k où k est le nombre de pistes.
- Contrainte : la RWH suit les trois pistes en même temps.
- Idée complémentaire : ne pas utiliser le même alphabet sur chacune des pistes.

Première équivalence

Theorem

Toute fonction calculée par une machine de Turing T à 1 ruban et une RWH sur un ruban bi-infini l'est aussi par une MT T' à un seul ruban semi-infini et 1 RWH. La réciproque est vraie.

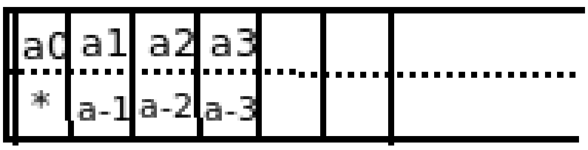
Réciproque

Preuve directe puisque $\mathbb{N} \subset \mathbb{Z}$

Idée de la preuve

- Replier le ruban à l'origine
- Rajouter des marqueurs de début
- Dupliquer les états

Définition de T'



$T' = (\Sigma', B', Q', q'_0, Q'_f, 1/2, \delta')$

- $\Sigma' = \Sigma \times (\Sigma \cup \{*\})$
- $B' = \begin{pmatrix} B \\ B \end{pmatrix}$
- $Q' = Q \times \{+, -\}$
- $Q'_f = Q_f \times \{+, -\}$
- $q'_0 = q_0^+$
- La configuration initiale place le mot d'entrée sur les cases $a_1, a_2, \ldots$ et la RWH sur la première case ($a_0 = B, *$).

Fonction de transition

Principe

- On travaille sur la piste qui correspond à l'endroit que l'on considère dans le ruban initial
- Si on est sur la partie haute de la piste (q^+), on fonctionne normalement en laissant la partie basse intacte
- si on est sur la partie basse (q^-), on inverse les directions
- on gère finement le changement de sens

Si $\delta(q, x) = (q', x', m)$ et $y \neq *$

$\delta'(q^+, (x, y)) = (q'^+, (x', y), m)$	$\delta'(q^-, (y, x)) = (q'^-, (y, x'), \bar{m})$
$\delta'(q^+, (x, *)) = (q'^+, (x', *), D)$	$\delta'(q^-, (x, *)) = (q^+, (x, *), St)$
$\delta'(q^+, (x, *)) = (q'^+, (x', *), St)$	
$\delta'(q^+, (x, *)) = (\mathbf{q}'^-, (x', *), D)$	$\longleftarrow$ Si m = G

Surcoût de la simulation

Combien de ressources supplémentaires utilise cette simulation ?

- $|\Sigma'| = |\Sigma|(|\Sigma| + 1)$
- $|Q'| = 2|Q|$
- Temps de calcul : entre 1 et 2 fois plus longtemps.
- Facteur **polynomial**

Autres mimes

Plusieurs rubans semi-infinis sur un seul

- Faire n pistes sur le même ruban
- Comment repérer les RWH ?
- Pour chaque ruban initial, on ajoute une piste sur l'alphabet $\Sigma_p = \{B, *\}$. Cette piste ne contiendra qu'un seul caractère $*$ qui marquera l'emplacement de la RWH sur le ruban simulé.
- On ajoute une piste pour signaler le début du ruban semi-infini

Autres mimes

Plusieurs rubans semi-infinis sur un seul (suite)

- Simulation :
  - Faire des parcours entre le début et chacune marque de RWH (ruban par ruban) pour récupérer l'état des RWH
  - Faire la transition
  - Refaire des parcours pour faire les écritures et les déplacements ruban par ruban.
- Surcoût en temps : **quadratique**
- Surcoût total : **polynomial**

Autres mimes

Mimes polynômiaux

On peut effectuer les mimes suivants en surcoût polynomial :

- 1 ruban semi-infini/n RWH par 1 ruban semi-infini/**1** RWH
- 1 ruban semi-infini/1 RWH/Alphabet à n lettres par 1 ruban semi-infini/1 RWH/Alphabet à **2** lettres
- On peut réduire le nombre de mouvements à 2 : $\{D, G\}$

Theorem

Toute machine de Turing peut être mimée avec un surcoût polynomial par une machine de Turing possédant :

- *1 ruban semi-infini avec 1 RWH*
- *Alphabet à deux lettres*
- *sans état stationnaire*

Énumération de Kleene

But

- Trouver toutes les MT avec un seul ruban, une seule RWH
- Donc, parcourir toutes les MT
- Les numéroter . . .
- . . . en utilisant une MT
- . . . qui aura son propre numéro

Codage

- D'une machine de Turing (*programme*)
- et d'une description instantanée (*état de la mémoire*)

Codage d'une MT

Alphabet de codage

- Alphabet propre
- $Z = \{0, 1, F, *, T, |, \$, E\}$

Codage d'un état

- Codage binaire du numéro d'état
- $q_0 \longrightarrow 0$; $q_1 \longrightarrow 1$; $q_2 \longrightarrow 10$
- ...
- Si q_i est final : $q_i \longrightarrow Fi_2$

Codage d'une MT (suite)

Codage de la transition $\delta(a_i, q_j) = (a_l, q_k, mvt)$

- $$i_2 * j_2 * l_2 * k_2 * mvt$$
- Si q_k est final, on remplace k_2 par Fk_2
- mvt peut être codé sur 2 caractères (e.g., 00, 10, 01)

Codage de la MT

- On code toutes les transitions les unes après les autres séparées par des caractères T
- On les range dans l'ordre lexicographique
- $T\ \delta(a_0, q_0)\ T\ \delta(a_1, q_0)\ T \ldots$

Reconnaissance d'une MT

Theorem

Soit x un mot de Z^*. Il existe une MT T_1 qui accepte x si et seulement si x code une Machine de Turing

Idée de preuve

- Il suffit de vérifier que le mot d'entrée possède la bonne syntaxe
- La MT T_1 peut être décrite avec toutes les possibilités en nombre de rubans, de RWH, ...
- La MT s'arrête toujours soit sur un état d'acceptation, soit sur un état de refus

Représentation d'un ruban

Codage d'une case

- Un ruban ne contient que des mots de taille finie
- Description de chaque case contenant la lettre a_i : i_2
- Si la RWH est sur la case codée et la MT est dans l'état q_j : $i_2\$j_2$

Codage d'un ruban semi-infini

- On code l'ensemble des cases $y_1, y2, \ldots, y_n$ non vides ($y_i \in \Sigma$)
- $|y_1|y_2|\ldots|y_n|$
- Reconnaissable par une MT T_2

Calcul

Vérification initiale d'une machine de Turing

Il existe une MT T_3 qui étant donné un mot $z \in Z^*$, vérifie que ce mot est de la forme $z = xEy$ où x représente une MT et y une description instantanée qui est l'état initial du ruban de la machine x.

Étape de calcul

Il existe une MT T_4 qui à partir d'un mot $z \in Z^*$ représentant une machine de Turing T et une description instantanée DI, effectue une transition T sur le ruban. I.e., $z = xEy$ est transformé par T_4 en $z' = xEy'$, où y' code la description instantanée après une étape de calcul.

Machine universelle

Théorème

Il existe une machine de Turing T_u avec un état d'acceptation qui lancée sur une entrée $z \in Z^*$:

1. Elle entre dans un état d'acceptation ssi :
   - z code une MT T_x et une description instantanée dans laquelle le mot a est sur le ruban
   - T_x lancée sur le mot a s'arrête
   - Le ruban de T_u à la fin contient $xE\tilde{y}$ où $\tilde{y}$ est le code de la DI de T_x à la fin de son calcul
2. Elle s'arrête dans un état de refus si z ne code pas de MT où que la machine codée bloque.
3. Si T_x lancée sur a ne s'arrête pas, alors T_u lancée sur a ne s'arrête pas.

Énumération

Théorème

Il existe une fonction $f : \mathbb{N} \longrightarrow Z^*$ telle que :

1. $f(n)$ est le code d'une MT
2. La suite $(f(n))_{n \in \mathbb{N}}$ est triée dans l'ordre lexicographique
3. Si $f(n) < x < f(n+1)$, alors x ne code pas de MT
4. f est calculable par une MT
5. f est une bijection de $\mathbb{N}$ dans $\{x \in Z^* | x \text{ code une MT}\}$

Énumération des rubans initiaux

De même, on peut énumérer les rubans initiaux par une fonction g : $g(i)$ est le i-ème ruban semi-infini où la RWH est positionné sur la première case.

D'autres machines assez universelles

Où on utilise une machine et un ruban

On note $\widetilde{T}_u$ la MT qui :

- prend deux entiers en paramètres $(i, j) \in \mathbb{N}^2$
- calcule $x = f(i)$ le codage de la i-ème MT
- calcule $y = g(j)$ le codage du j-ème ruban
- lance T_u sur l'entrée xEy

Aparté : Énumération de $\mathbb{N}^2$

- On rappelle que $\mathbb{N}^2$ est dénombrable
- Il existe donc une bijection entre $\mathbb{N}^2$ et $\mathbb{N}$
- Soit χ une énumération de $\mathbb{N}^2$; π_1 et π_2 les fonctions inverses (Si $\chi(i, j) = k$ alors $\pi_1(k) = i$ et $\pi_2(k) = j$)

D'autres machines tordues

φ_u

On note φ_u la machine de Turing qui prend un entier binaire en entrée et telle que

$$\varphi_u(i) = \widetilde{T}_u(\pi_1(i), \pi_2(i))$$

Pour en finir avec les définitions

- Soit φ_i la MT de code de Kleene $f(i)$
- $\varphi_i(j)$ est donc la machine $\varphi_u(\chi(i, j))$
- en d'autres termes $\widetilde{T}_u(i, j)$

Théorème de la Halte

Théorème de la Halte

Il n'existe pas de MT T prenant deux entiers (x, y) en entrée telle que :

- T s'arrête sur toute entrée
- Elle s'arrête avec le résultat 1 si $\varphi_x(y)$ s'arrête
- Elle s'arrête avec le résultat 0 si $\varphi_x(y)$ ne s'arrête pas

Preuve par l'absurde

- On suppose que T existe
- On construit $\widetilde{T}$ la MT qui :
  - prend en entrée deux entiers (x, y)
  - s'arrête si $T(x, y)$ s'arrête avec le résultat 0
  - sinon ne s'arrête pas

Preuve Théorème de la Halte

Suite de la preuve

- $\widetilde{T}$ existe, une fois que l'on a T, c'est facile
- $\widetilde{T}(x, x)$ est une MT dépendant de x : $\widetilde{\widetilde{T}}$
- Cette dernière machine possède un numéro : a
- $\forall x \; \widetilde{\widetilde{T}}(x) = \varphi_a(x)$
- Que fait $\varphi_a(a)$?

preuve Théorème de la Halte (fin)

Suite d'implications contradictoires

Chaque assertion est la conséquence directe de la précédente

1. Supposons que $T(a, a) = 1$
2. $\varphi_a(a)$ s'arrête
3. $\widetilde{T}(a, a)$ s'arrête
4. $T(a, a) = 0$

De même :

1. Supposons que $T(a, a) = 0$
2. $\varphi_a(a)$ ne s'arrête pas
3. $\widetilde{T}(a, a)$ ne s'arrête pas
4. $T(a, a) = 1$