TD 7 – Indécidabilité

1 Indécidabilité et non-reconnaissabilité

Un état $q \in Q$ d'une MT $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$, différent de q_{acc} et q_{rej}, est dit inutile lorsqu'aucun calcul d'une configuration initiale $q_0 w$ pour $w \in \Sigma^*$ ne visite l'état q.

Exercice 1. Montrer que les langages suivants ne sont pas décidables.

1. $\{\langle\mathcal{M}\rangle \mid \mathcal{M}$ MT normalisée ayant un état inutile$\}$.
2. $\left\{\langle\mathcal{M}\rangle \;\middle|\; \begin{array}{l}\mathcal{M} \text{ MT normalisée qui écrit le symbole blanc au cours de son exécution} \\ \text{sur le mot vide}\end{array}\right\}$.
3. $\left\{\langle\mathcal{M}\rangle \;\middle|\; \begin{array}{l}\mathcal{M} \text{ MT normalisée s'arrêtant sur le mot vide et étant l'une de celles, à états et} \\ \text{alphabets fixés, effectuant le nombre maximal d'étapes avant arrêt}\end{array}\right\}$.

Solution 1. On note

$$\mathrm{HE}_{\mathrm{TM}} = \{\langle\mathcal{M}\rangle \mid \mathcal{M} \text{ MT normalisée s'arrêtant sur le mot vide}\}\ ,$$

qui est un langage indécidable.

1. Soit $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, q_2, q_3)$ une MT normalisée telle que $Q = \{q_1, \ldots, q_n\}$ avec $n \in \mathbb{N}, n \geq 3$, $\Sigma = \{a_1, \ldots, a_l\}$ avec $l \in \mathbb{N}, l \geq 1$ et $\Gamma = \{a_1, \ldots, a_k\}$ avec $k \in \mathbb{N}, k \geq l+1$, où a_k correspond au symbole blanc.
   Construisons la MT normalisée $\mathcal{M}' = (Q', \Sigma, \Gamma', \delta', q_1, q_2, q_3)$ où $Q' = Q \cup \{q_{n+1}, q_{n+2}\}$, $\Gamma' = \Gamma \cup \{a_{k+1}\}$ (où a_{k+1} correspond au symbole blanc) et $\delta' \colon (Q' \setminus \{q_2, q_3\}) \times \Gamma' \to Q' \times \Gamma' \times \{L, R\}$ est telle que :
   — pour tous $i \in [\![1, n]\!] \setminus \{2, 3\}$ et $j \in [\![1, k-1]\!]$, $\delta'(q_i, a_j) = (p', b', D)$ avec, si $(p, b, D) = \delta(q_i, a_j)$, alors $p' = \begin{cases} p & \text{si } p \notin \{q_2, q_3\} \\ q_{n+1} & \text{sinon} \end{cases}$ et $b' = \begin{cases} b & \text{si } b \neq a_k \\ a_{k+1} & \text{sinon} \end{cases}$;
   — pour tout $i \in [\![1, n]\!] \setminus \{2, 3\}$, $\delta'(q_i, a_{k+1}) = (p', b', D)$ avec, si $(p, b, D) = \delta(q_i, a_k)$, alors $p' = \begin{cases} p & \text{si } p \notin \{q_2, q_3\} \\ q_{n+1} & \text{sinon} \end{cases}$ et $b' = \begin{cases} b & \text{si } b \neq a_k \\ a_{k+1} & \text{sinon} \end{cases}$;
   — pour tous $j \in [\![1, k+1]\!]$, $\delta'(q_{n+1}, a_j) = (q_{n+2}, a_k, R)$ et $\delta'(q_{n+2}, a_j) = (q_1, a_k, L)$;
   — si $n = 3$, $\delta'(q_1, a_k) = (q_2, a_k, L)$, sinon $\delta'(q_1, a_k) = (q_4, a_k, R)$, $\delta'(q_n, a_k) = (q_2, a_k, L)$ et pour tout $i \in [\![4, n-1]\!]$, $\delta'(q_i, a_k) = \begin{cases} (q_i, a_k, L) & \text{si } i \bmod 2 = 0 \\ (q_i, a_k, R) & \text{sinon} \end{cases}$.

   Il n'est pas trop difficile de voir que $\mathcal{M}$ s'arrête sur le mot vide si et seulement si $\mathcal{M}'$ n'a pas d'état inutile. De plus, la transformation de $\langle\mathcal{M}\rangle$ en $\langle\mathcal{M}'\rangle$ est calculable par une MT, d'où il s'ensuit que le langage considéré ne peut être décidable, autrement $\mathrm{HE}_{\mathrm{TM}}$ serait lui-même décidable.
2. Soit $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, q_2, q_3)$ une MT normalisée telle que $Q = \{q_1, \ldots, q_n\}$ avec $n \in \mathbb{N}, n \geq 3$, $\Sigma = \{a_1, \ldots, a_l\}$ avec $l \in \mathbb{N}, l \geq 1$ et $\Gamma = \{a_1, \ldots, a_k\}$ avec $k \in \mathbb{N}, k \geq l+1$, où a_k correspond au symbole blanc.
   Construisons la MT normalisée $\mathcal{M}' = (Q, \Sigma, \Gamma', \delta', q_1, q_2, q_3)$ où $\Gamma' = \Gamma \cup \{a_{k+1}\}$ (où a_{k+1} correspond au symbole blanc) et $\delta' \colon (Q \setminus \{q_2, q_3\}) \times \Gamma' \to Q \times \Gamma' \times \{L, R\}$ est telle que :
   — pour tous $i \in [\![1, n]\!] \setminus \{2, 3\}$ et $j \in [\![1, k]\!]$, $\delta'(q_i, a_j) = (p, b', D)$ avec, si $(p, b, D) = \delta(q_i, a_j)$, alors $b' = \begin{cases} b & \text{si } p \notin \{q_2, q_3\} \\ a_{k+1} & \text{sinon} \end{cases}$;

— pour tout $i \in [\![1,n]\!] \setminus \{2,3\}$, $\delta'(q_i, a_{k+1}) = (p, b', D)$ avec, si $(p, b, D) = \delta(q_i, a_k)$, alors $b' = \begin{cases} b & \text{si } p \notin \{q_2, q_3\} \\ a_{k+1} & \text{sinon} \end{cases}$.

Il n'est pas trop difficile de voir que $\mathcal{M}$ s'arrête sur le mot vide si et seulement si $\mathcal{M}'$ écrit le symbole blanc au cours de son exécution sur le mot vide. De plus, la transformation de $\langle\mathcal{M}\rangle$ en $\langle\mathcal{M}'\rangle$ est calculable par une MT, d'où il s'ensuit que le langage considéré ne peut être décidable, autrement $\mathrm{HE}_{\mathrm{TM}}$ serait lui-même décidable.

3. On observe qu'il y a un nombre fini de MT normalisées à états et alphabets fixés.

   Supposons qu'il existe une MT $\mathcal{D}$ décidant le langage considéré. On suppose avoir une MT $\mathcal{V}$ décidant le langage des mots sur $\{0,1\}$ correspondant à des encodages avec $\langle\cdot\rangle$ de MT normalisées. On construit maintenant une MT $\mathcal{D}'$ qui, étant donné donné un mot $w \in \{0,1\}^*$ en entrée, se comporte selon l'algorithme suivant :

   si $\mathcal{V}(w)$ rejette **alors**
       rejeter
   fin si ▷ À partir d'ici on suppose que $w = \langle\mathcal{M}\rangle$
   $v \leftarrow \varepsilon$
   pour $\mathcal{M}'$ MT normalisée différente de $\mathcal{M}$ ayant les mêmes états et alphabets **faire**
       si $D(\langle\mathcal{M}'\rangle)$ accepte **alors**
           $v \leftarrow \langle\mathcal{M}'\rangle$
       fin si
   fin pour
   si $v = \varepsilon$ **alors**
       rejeter
   fin si ▷ À partir d'ici on suppose que $v = \langle\mathcal{M}_{\max}\rangle$
   exécuter $\mathcal{M}_{\max}(\varepsilon)$ en comptant le nombre d'étapes t avant arrêt
   si $\mathcal{M}(\varepsilon)$ s'arrête en au plus t étapes **alors**
       accepter
   sinon
       rejeter
   fin si.

   Il n'est pas trop difficile de voir que $\mathcal{D}'$ décide $\mathrm{HE}_{\mathrm{TM}}$, ce qui contradictoire avec le fait que ce langage soit indécidable. Le langage considéré ne peut donc être décidable.

Exercice 2. Soit $L \subseteq \Sigma^*$ un langage non décidable sur un alphabet Σ contenant $\{0,1\}$. Soit le langage $K = 0L \cup 1\bar{L}$. Montrer que le langage K et son complémentaire $\bar{K}$ ne sont pas reconnaissables.

Solution 2. Supposons que K est reconnaissable. Puisque $0\Sigma^*$ et $1\Sigma^*$ sont tous les deux trivialement reconnaissables, il s'ensuirait que $L = K \cap 0\Sigma^*$ et $\bar{L} = K \cap 1\Sigma^*$ sont tous les deux reconnaissables, ce qui impliquerait que L est en fait décidable.

Supposons que $\bar{K}$ est reconnaissable. On a $\bar{K} = 0\bar{L} \cup 1L \cup (\Sigma \setminus \{0,1\})\Sigma^* \cup \{\varepsilon\}$, donc il s'ensuirait que $L = \bar{K} \cap 1\Sigma^*$ et $\bar{L} = \bar{K} \cap 0\Sigma^*$ sont tous les deux reconnaissables, ce qui impliquerait que L est en fait décidable.

2 Castors affairés

Le problème des « castors affairés » (« busy beavers ») a été introduit par Radó, dans le but de définir « simplement » une fonction non calculable. Le modèle de MT considéré est le suivant : on suppose les machines déterministes, possédant un ruban bi-infini, un alphabet réduit à $\{0,1\}$ (on ne distingue pas 0 du symbole blanc). On suppose de plus que les machines possèdent un unique état final, que nous appellerons *état d'arrêt*, duquel aucune transition ne sort, et qui n'est pas compté parmi les états de la machine. Enfin, on considérera toujours un ruban initialement complètement blanc.

La fonction du castor affairé $\Sigma\colon \mathbb{N} \to \mathbb{N}$ est définie comme donnant, pour tout $n \in \mathbb{N}$, le nombre maximum de 1 écrits sur la bande (pas nécessairement consécutifs) après qu'une MT telle que décrite ci-dessus à $n+1$ états (n états d'opération et un état d'arrêt) atteigne l'état d'arrêt.

Exercice 3. Justifier que $\Sigma(n)$ est bien définie pour tout $n \in \mathbb{N}$.

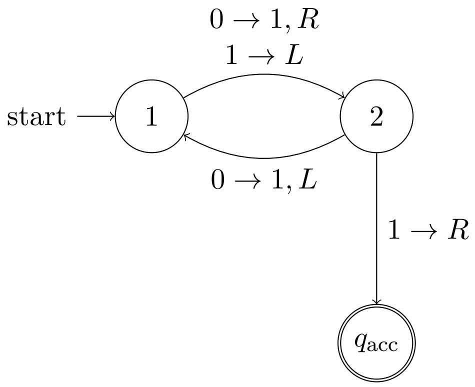
Solution 3. Il y a un nombre fini de MT telles que décrites ci-dessus (que l'on nommera *adaptées*) avec $n \in \mathbb{N}$ états, à renommage des états prêt. On observe aussi qu'il existe toujours au moins une de ces machines qui s'arrête (c'est-à-dire, atteigne l'état d'arrêt). $\Sigma(n)$ est donc bien définie.

Exercice 4. Que valent $\Sigma(0)$, $\Sigma(1)$, $\Sigma(2)$? (Plus simplement, montrer que $\Sigma(2) \geq 4$.)

Solution 4. $\Sigma(0) = 0$, $\Sigma(1) = 1$, $\Sigma(2) = 4$.

Pour $\Sigma(0)$ et $\Sigma(1)$, c'est évident.

Pour $\Sigma(2)$, on constate que la MT ci-dessous s'arrête en écrivant bien 4 lettres 1 :



.

Pour montrer qu'on ne peut pas faire mieux, il faut énumérer les MT adaptées de façon intelligente.

Exercice 5. Montrer que la fonction Σ est strictement croissante.

Solution 5. Pour toute MT adaptée à $n \in \mathbb{N}$ états qui écrit $k \in \mathbb{N}$ lettres 1 avant de s'arrêter, on peut construire une MT adaptée à $n+1$ états qui écrit $k+1$ lettres 1 avant de s'arrêter : toute transition allant vers l'état d'arrêt est envoyée vers le nouvel état, dans lequel on boucle en allant vers la droite tant que l'on lit 1, et dans lequel on écrit un 1 en passant à l'état d'arrêt dès que l'on rencontre un 0.

Considérons la définition suivante de fonction sur les entiers naturels calculable : $f\colon \mathbb{N} \to \mathbb{N}$ est calculable si et seulement s'il existe une MT qui, sur un ruban contenant initialement n symboles 1 consécutifs à droite du symbole blanc de départ, s'arrête sur ce symbole après un nombre fini d'étapes avec un bloc de $f(n)$ symboles 1 consécutifs à droite du symbole blanc de départ.

Exercice 6. Montrer que la fonction Σ du castor affairé croit strictement plus vite que toute fonction calculable f (c'est-à-dire, $f(n) \in \mathrm{o}(\Sigma(n))$).

Solution 6. Soit f une fonction calculable. La fonction $p\colon n \mapsto 2^n$ est aussi calculable. On peut donc calculer la composée $p \circ f$ avec une MT à k états. De plus, pour tout $x \in \mathbb{N}_{>0}$, la fonction $c_x\colon n \mapsto x$ peut se calculer avec une MT à $\lfloor \log_2(x) \rfloor + l$ états.

Pour tout $x \in \mathbb{N}_{>0}$, on peut donc construire une MT à $\lfloor \log_2(x) \rfloor + l + k$ états calculant la fonction $p \circ f \circ c_x$, c'est-à-dire, écrivant $2^{f(x)}$ sur la bande lorsque lancée sur le mot vide (ou tout autre mot). Il s'ensuit donc que $\Sigma(\lfloor \log_2(x) \rfloor + l + k) \geq 2^{f(x)}$.

Par conséquent, pour tout $x \in \mathbb{N}_{>0}$ suffisamment grand, puisque Σ est strictement croissante, on a que $\Sigma(x) \geq \Sigma(\lfloor \log_2(x) \rfloor + l + k) \geq 2^{f(x)}$, ce qui prouve que $f(n) \in \mathrm{o}(\Sigma(n))$.

3 Quines

Pour deux MT A et B on note $A \cdot B$ une MT qui exécute B après avoir exécuté A.

Étant donné un alphabet normalisé Σ (alphabet d'entrée d'une MT normalisée), on suppose avoir une fonction d'encodage $\langle\cdot\rangle\colon \Sigma^* \to \{0,1\}^*$ pour les mots sur Σ associée à la fonction d'encodage pour les MT normalisées. Pour chaque mot $w \in \Sigma^*$, il existe P_w une MT normalisée avec Σ pour alphabet d'entrée qui écrit le mot w sur le ruban en utilisant $|w|$ états non finaux, l'état $i \in [\![1, |w|\,]\!]$ permettant d'écrire w_i.

Exercice 7. Étant donné un alphabet normalisé Σ, expliquer pourquoi la fonction $q\colon \{0,1\}^* \to \{0,1\}^*$ telle que

$$q(u) = \begin{cases} \langle P_w \rangle & \text{s'il existe } w \in \Sigma^* \text{ tel que } u = \langle w \rangle \\ \varepsilon & \text{sinon} \end{cases},$$

pour tout $u \in \{0,1\}^*$ est calculable.

Solution 7. q peut être calculée en déterminant, pour une entrée u, le mot w tel que $u = \langle w \rangle$ s'il existe, puis en écrivant $\langle P_w \rangle$.

Exercice 8. Expliquer pourquoi la fonction $s_2\colon \{0,1,\#\}^* \to \{0,1\}^*$ telle que

$$s_2(u) = \begin{cases} \langle A \cdot B \rangle & \text{s'il existe } A, B \text{ MT normalisées telles que } u = \langle A \rangle \# \langle B \rangle \\ \varepsilon & \text{sinon} \end{cases}$$

pour tout $u \in \{0,1,\#\}^*$ est calculable.

Solution 8. s_2 peut être calculée en déterminant, pour une entrée u, les MT normalisées A et B telles que $u = \langle A \rangle \# \langle B \rangle$ si elles existent, puis en écrivant $\langle A \cdot B \rangle$.

Exercice 9. En déduire qu'il existe une MT $\mathcal{M}$ qui, sur le mot vide, s'arrête après avoir écrit $\langle \mathcal{M} \rangle$ sur la bande.

Solution 9. Soit $s\colon \{0,1\}^* \to \{0,1\}^*$ la fonction $x \mapsto s_2(q(x)\#x)$, calculée par une MT normalisée $\mathcal{S}$. On considère $\mathcal{M}$ une MT normalisée qui calcule $P_{\langle \mathcal{S} \rangle} \cdot \mathcal{S}$, et qui donc vérifie $\langle \mathcal{M} \rangle = \langle P_{\langle \mathcal{S} \rangle} \cdot \mathcal{S} \rangle$.

L'exécution de $\mathcal{M}$ sur le mot vide donne :

$$\varepsilon \xrightarrow{P_{\langle \mathcal{S} \rangle}} \langle \mathcal{S} \rangle \xrightarrow{\mathcal{S}} s(\langle \mathcal{S} \rangle) = s_2(q(\langle \mathcal{S} \rangle)\#\langle \mathcal{S} \rangle) = s_2(\langle P_{\langle \mathcal{S} \rangle} \rangle \# \langle \mathcal{S} \rangle) = \langle P_{\langle \mathcal{S} \rangle} \cdot \mathcal{S} \rangle \ .$$