Problèmes décidables (voir Kozen § 32, pages 235–236)

Soit $n \in \mathbb{N}$. Est-il décidable si une MdT M donnée :

1. prend plus de n étapes pour le mot d'entrée ϵ ?
2. prend plus de n étapes pour ***au moins un*** mot d'entrée ?
3. prend plus de n étapes pour ***tous*** les mots d'entrée ?

Prendre une étape signifie "exécuter une transition de la machine".

Prendre un certain nombre n d'étapes de M pour traiter w signifie que M atteint un état accepteur ou rejettant après exactement n transitions.

Définition 1 (Nombre d'étapes) *Soit M une MdT et w un mot d'entrée pour M.*

Soit $M(w)$ la sequence de transitions entre configurations de M en lisant le mot w

$$C_0 \rightarrow^1_M C_1 \rightarrow^1_M \cdots \rightarrow^1_M C_{m-1} \rightarrow^1_M C_m \rightarrow^1_M \cdots$$

qui représente l'exécution déterministe de M pour w, alors $C_0 = (s_M, \vdash w \llcorner\!\lrcorner^\omega, 0)$.

Notons $C_i = (p_i, \vdash w_i \llcorner\!\lrcorner^\omega, n_i)$.

Le nombre d'étapes $\#(M, w)$ est défini par la sequence $M(w)$ par

$$\#(M,w) \stackrel{\text{def}}{=} \begin{cases} m & si \ \ \forall 0 \leq i < m : p_i \notin \{\, t, r \,\} \ \ et \ \ p_m \in \{\, t, r \,\} \\ \infty & si \ \ \forall 0 \leq i : p_i \notin \{\, t, r \,\} \end{cases}$$

Observation 2 *Pour tout w avec $|w| = k$, toute machine a au moins besoin de $k+1$ étapes pour détecter la fin du mot w car l'on ne peut pas lire plus d'un symbole par étape/transition. Cependant il se peut que la machine n'y arrive jamais, dans le cas où elle s'arrête avant d'avoir atteint la fin du mot, ou elle boucle à l'intérieur d'un préfixe.*

La longueur d'un mot est ainsi étroitement liée au nombre de transitions qu'une machine execute pour le traiter. Si $\#(M, w) \leq |w|$, alors la MdT M s'est arretée avant d'avoir atteint la fin de w. Cela s'applique aussi à tous les mots w' dont w est un préfixe. De la même manière, si une MdT prend déjà plus de $|w|$ étapes pour traiter un préfixe w de w', elle prendra plus de $|w|$ étapes pour w'.

Lemme 3 *Soit $|w| = k$ et $w' = w \cdot x$.*

- *Si $\#(M, w) \leq k$, alors $\#(M, w') \leq k$ et même, $\#(M, w') = \#(M, w)$.*
- *Si $\#(M, w) \geq k$, alors $\#(M, w') \geq k$.*

Preuve. Comme M est déterministe, et comme aucune machine peut avoir atteint la fin du mot w en moins de $k+1$ étapes, les projection des séquence $M(w')$ et $M(w)$ sur les états p_i (et aussi sur les positions de la tête n_i) sont égales de la configuration C_0 jusqu'à C_k.

Plus formellement :

$$\begin{array}{llllllll} \text{si} & (s_M, \vdash w \llcorner\!\lrcorner^\omega, 0) & \rightarrow^{k-1}_M & (p_{k-1}, \vdash w_{k-1} \llcorner\!\lrcorner^\omega, n_{k-1}) & \rightarrow^1_M & (p_k, \vdash w_k \llcorner\!\lrcorner^\omega, n_k) & \rightarrow^1_M & \cdots \\ \text{alors} & (s_M, \vdash wx \llcorner\!\lrcorner^\omega, 0) & \rightarrow^{k-1}_M & (p_{k-1}, \vdash w_{k-1}x \llcorner\!\lrcorner^\omega, n_{k-1}) & \rightarrow^1_M & (p_k, \vdash w_k x \llcorner\!\lrcorner^\omega, n_k) & \rightarrow^1_M & \cdots \end{array}$$

Notons que la partie x du mot w' reste inchangée durant les k premières étapes de l'exécution $M(w')$. ■

On suppose que les trois MdT suivantes possèdent la capacité — similaire à celle d'une MdT universelle — de pouvoir simuler l'exécution d'une MdT quelconque sur un mot donné. Nous ne détailons pas de manière formelle comment elle en sont capable, nous décrivons juste comment on peut utiliser cette capacité pour répondre aux trois questions de décision de la page précédente.

Machine M_1

- De la même manière qu'une machine universelle, pour une méthode de codage fixée, M_1 sait déchiffrer ses entrées, c'est à dire le codage de n'importe quelle machine M.
- M_1 rajoute sur le ruban le codage du mot ϵ.
- Sur un deuxième ruban, M_1 prépare le comptage des étapes de la simulation de M pour ϵ.
- M_1 rejette si la simulation prend plus de n étapes ; M_1 accepte si la simulation s'arrête avant n étapes.

Machine M_2

- De la même manière qu'une machine universelle, pour une méthode de codage fixée, M_1 sait déchiffrer ses entrées, c'est à dire le codage de n'importe quelle machine M.
- M_1 rajoute sur le ruban le codage de tous les mots dans l'ensemble
  $\Sigma^{\leq n} = \{\, w \in \Sigma_M^* \mid |w| \leq n \,\}$.
  Notons que ceci est un ensemble fini, cette tâche est donc à la portée d'une MdT.
- Sur un deuxième ruban, M_2 prépare le comptage des étapes de la simulation de M pour les mots contenu dans $\Sigma^{\leq n}$.
- M_2 s'exécute comme suit :
  1. M_2 simule l'exécution de M pour les éléments de $\Sigma^{\leq n}$ l'un après l'autre.
  2. Si, pendant ces simulations, il trouve un mot pour lequel M prends plus de n étapes, alors M_2 accepte.
  3. Sinon, cela implique que $\forall w \in \Sigma^{\leq n}$ cela prend moins de n étapes. Ceci étant le cas pour tous les mots w tels que $|w| = n$, on peut donc appliquer la première partie du lemme 3 pour déduire que toutes les prolongations des mots de longueur n prennent moins de n étapes, et la machine rejette.

Machine M_3

Cette machine est très similaire à M_2. On remplace juste les partie 2 et 3.

- De la même manière qu'une machine universelle, pour une méthode de codage fixée, M_1 sait déchiffrer ses entrées, c'est à dire le codage de n'importe quelle machine M.
- M_1 rajoute sur le ruban le codage de tous les mots dans l'ensemble
  $\Sigma^{\leq n} = \{\, w \in \Sigma_M^* \mid |w| \leq n \,\}$.
  Notons que ceci est un ensemble fini, cette tâche est donc à la portée d'une MdT.
- Sur un deuxième ruban, M_3 prépare le comptage des étapes de la simulation de M pour les mots contenu dans $\Sigma^{\leq n}$.
- M_3 exécute comme suit :
  1. M_3 simule l'exécution de M pour des éléments de $\Sigma^{\leq n}$ l'un après l'autre.
  2. Si, pendant ces simulations, il trouve un mot pour lequel M prends moins de n étapes, alors M_3 rejette.

3. Sinon, cela implique que $\forall w \in \Sigma^{\leq n}$ cela prend plus de n étapes. Ceci étant le cas pour tous les mots w tels que $|w| = n$, on peut donc appliquer la deuxième partie du lemme 3 pour déduire que toutes les prolongations des mots de longueur n prennent aussi plus de n étapes, et la machine accepte.