Corrigé TD 3
Calculabilité

Luc Lapointe
luc.lapointe@ens-paris-saclay.fr
home.lmf.cnrs.fr/LucLapointe/

Exercice 1 - Diagonalisations

Montrer par un argument diagonal (et donc sans réduction) que les langages suivants sont indécidables :

$$L_{\text{accepte}} = \left\{(\langle M\rangle, w)^{\text{bin}} \mid M \text{ accepte } w\right\}$$

$$L_{\text{arrêt}} = \left\{(\langle M\rangle, w)^{\text{bin}} \mid M \text{ termine avec l'entrée } w\right\}$$

Solution

- Supposons par l'absurde que L_{accepte} soit décidable. Soit `accepte(M, w)` la fonction qui renvoie *Vrai* si et seulement si M est le code d'une machine qui accepte w. On considère la fonction suivante :

  ```
  paradoxe(x):
    si accepte(x, x):
      refuser x
    sinon:
      accepter x
  ```

  Cette fonction est calculable, car accepte l'est. Soit M_{paradoxe} la machine qui la décide. Considérons l'exécution de `paradoxe` sur le code de M_{paradoxe}.
  - Si cette exécution accepte, c'est que `accepte` renvoie *Faux* sur $(M_{\text{paradoxe}}, M_{\text{paradoxe}})$: contradiction.
  - Si cette exécution refuse, c'est que `accepte` renvoie *Vrai* sur $(M_{\text{paradoxe}}, M_{\text{paradoxe}})$: contradiction.
- Pour $L_{\text{arrêt}}$, raisonnement identique, mais avec une fonction `paradoxe` qui teste `arret(x, x)` au lieu de `accepte(x, x)`, et qui tourne indéfiniment plutôt que refuser `x` en cas de réponse positive.

Exercice 2 - Vers la réduction (1)

En supposant que le langage $L_{\text{arrêt}}$ est indécidable (et seulement celui-ci), montrez que

$$L_{\emptyset} = \{\langle M\rangle^{\text{bin}} \mid \mathcal{L}(M) = \emptyset\}$$

l'est également. Vous raisonnerez par l'absurde et supposerez l'existence d'une machine de Turing $M_\emptyset$ décidant $L_\emptyset$. Vous construirez ensuite une machine M décidant $L_{\text{arrêt}}$ en utilisant $M_\emptyset$.

Solution

Supposons par l'absurde qu'il existe une machine $M_\emptyset$ décidant $L_\emptyset$. Je définis une machine $M'_{M,x}$ qui, sur toute entrée y:

- Commence par remplacer y par des caractères blancs,
- puis écrit x à gauche du ruban,
- puis déplace le curseur à gauche du ruban,
- et imite M, sauf que toute transition vers `reject` est remplacée par une transition vers `accept`.

La fonction qui à $(\langle M\rangle, x)^{\text{bin}}$ associe $\langle M'_{M,x}\rangle^{\text{bin}}$ est calculable. Par ailleurs, on a que $M'_{M,x}$ accepte au moins un mot si et seulement si M termine sur x.

Soit l'algorithme suivant qui prend en entrée $(\langle M\rangle, x)^{\text{bin}}$, et :

1. Calcule $\langle M'_{M,x}\rangle^{\text{bin}}$,
2. Décide si le langage de $M'_{M,x}$ est vide.

Cet algorithme est calculable car $M_\emptyset$ existe, et il décide $L_{\text{arrêt}}$. Contradiction.

Exercice 3 - Vers la réduction (2)

En supposant que le langage L_{accepte} est indécidable (et seulement celui-ci), montrez que

$$L_{\text{arrêt}}$$

l'est également. Vous raisonnerez par l'absurde et supposerez l'existence d'une machine de Turing $M_{\text{arrêt}}$ décidant $L_{\text{arrêt}}$. Vous construirez ensuite une machine M décidant L_{accepte} en utilisant $M_{\text{arrêt}}$.

Solution

Supposons par l'absurde qu'il existe une machine $M_{\text{arrêt}}$ décidant $L_{\text{arrêt}}$. Je définis une machine une machine M'_M qui est identique à M, sauf que toute transition vers `reject` est remplacée par une transition vers un état qui est une boucle infinie dont il est impossible de sortir. La fonction qui à $\langle M\rangle^{\text{bin}}$ associe $\langle M'_M\rangle^{\text{bin}}$ est calculable. Par ailleurs, pour tout mot x, M accepte x ssi M' termine sur x.

Soit l'algorithme suivant qui prend en entrée $(\langle M\rangle, x)^{\text{bin}}$, et :

1. Calcule $\langle M'_M\rangle^{\text{bin}}$,
2. Décide si M'_M s'arrête sur x.

Cet algorithme est calculable car $M_{\text{arrêt}}$ existe, et il décide L_{accepte}. Contradiction.

Les preuves des exercices 2 et 3 ont toutes un point commun expliqué ci-dessous. On dit que ce sont des preuves **par réduction**.

Rappel (ou non) : Soit Σ un alphabet. Un *problème* A sur Σ est un langage sur Σ. Les mots de $\mathcal{A}$ sont les *instances acceptantes* du problème, et les mots de Σ^* sont les *instances* du problème.

Si on considère deux problèmes $\mathcal{A}$ e $\mathcal{B}$, on dit que

$$\mathcal{A} \text{ se réduit à } \mathcal{B} \text{ (noté } \mathcal{A} \preccurlyeq \mathcal{B})$$

si on peut exhiber une fonction **calculable** f qui **pour toute** instance a de $\mathcal{A}$, renvoie une instance $b = f(a)$ de $\mathcal{B}$ telle que

$$\begin{gathered} a \textit{ est une instance acceptante de } \mathcal{A} \\ \textbf{\textit{si et seulement si}} \\ b \textit{ est une instance acceptante de } \mathcal{B}. \end{gathered}$$

Attention ! Il n'est pas nécessaire que **toutes** les instances de $\mathcal{B}$ soient dans l'image de f.

Si on a $\mathcal{A} \preccurlyeq \mathcal{B}$ alors :
- si $\mathcal{A}$ est indécidable, alors $\mathcal{B}$ aussi;
- si $\mathcal{B}$ est décidable, alors $\mathcal{A}$ aussi.

Exercice 4 - Est-ce décidable ?

Dire si les problèmes suivants sont décidables ou non. Si c'est le cas, donner l'idée de la machine de Turing décidant le langage, et si non, faire une preuve **par réduction**.

1. **Donnée :** le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : M s'arrête-t-elle sur le mot vide ?
2. **Donnée :** le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : M s'arrête-t-elle sur au moins une donnée ?
3. **Donnée :** les codes $\langle M \rangle^{\text{bin}}$ et $\langle M' \rangle^{\text{bin}}$ de deux machines de Turing.
   Question : $L(M) = L(M')$?
4. **Donnée :** les codes $\langle M \rangle^{\text{bin}}$ d'une machine de Turing et d'un mot w et un entier n en base 2.
   Question : M accepte-t-elle w après au plus n transitions ?
5. **Donnée :** le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : M termine-t-elle en temps polynomial ?

Solution

Je note $\mathcal{P}_i$ le problème numéro i de l'exercice.

1. Indécidable. Réduction depuis $L_{\text{arrêt}}$. Soit $(\langle M \rangle, x)^{\text{bin}}$ une entrée de $L_{\text{arrêt}}$. Je définis une machine $M'_{M,x}$ qui, sur toute entrée y :
   - Commence par remplacer y par des caractères blancs,
   - puis écrit x à gauche du ruban,
   - puis déplace le curseur à gauche du ruban, et imite M.

   La fonction r qui à $(\langle M \rangle, x)^{\text{bin}}$ associe $\langle M'_{M,x} \rangle^{\text{bin}}$ est calculable. Par ailleurs, on a que $M'_{M,x}$ s'arrête sur ε si et seulement si M termine sur x.

   r est donc une réduction de $L_{\text{arrêt}}$ à $\mathcal{P}_1$, qui est donc indécidable.

2. Indécidable. Considérons encore r de la question précédente. On a que $M'_{M,x}$ s'arrête sur ε si et seulement si M termine sur au moins une entrée x.

   r est donc une réduction de $L_{\text{arrêt}}$ à $\mathcal{P}_2$, qui est donc indécidable.

3. Indécidable. Réduction depuis $L_\emptyset$. Soit $M_\perp$ une machine à 1 état qui rejette tous les mots.

   La fonction r qui à $\langle M \rangle^{\text{bin}}$ associe $\langle M \rangle^{\text{bin}}$ et $\langle M_\perp \rangle^{\text{bin}}$ est calculable. Par ailleurs, on a que M reconnaît $\emptyset$ si et seulement si $\mathcal{L}(M) = \mathcal{L}(M_\perp)$.

   r est donc une réduction de $L_\emptyset$ à $\mathcal{P}_3$, qui est donc indécidable.

4. Décidable. Considérons la fonction qui à $(\langle M \rangle, n)^{\text{bin}}$ associe le code d'une nouvelle machine à une bande M_n exécutant l'algorithme suivant :

   ```
   M_n(x):
     Déplacer x de log2(n) + 1 cases vers la droite.
     Écrire n en binaire au début du ruban, puis un caractère £
     Tant que le mot avant £ n'encode pas 0:
       Exécuter une étape de M après £, en assimilant £ à $
       Décrémenter l'entier avant £
     Refuser x
   ```

   Cette transformation nécessite une machine universelle. La fonction logarithme binaire est calculable. La première ligne de la boucle tant que peut se faire en ajoutant une lettre indiquant l'état actuel de M sur la bande. Cet état est clairement identifiable et permet de passer de la partie à gauche de £ à la partie à droite de £ dans la bande.

   On a que $M_{n(x)}$ accepte si et seulement si $M(x)$ accepte en au plus n transitions. De plus, la fonction qui à $(\langle M \rangle, n)^{\text{bin}}$ associe $\langle M_n \rangle^{\text{bin}}$ est calculable.

On a donc un algorithme de résolution de $\mathcal{P}_4$ qui consiste à transformer l'entrée en M_n, puis à exécuter une machine universelle U sur w.

5. Indécidable. Réduction depuis $\mathcal{P}_1$. Soit $\langle M \rangle^{\text{bin}}$ i,e entrée de $\mathcal{P}_1$. Je définis la machine M'_M qui, sur toute entrée y :
   - Commence par remplacer y par des caractères blancs,
   - puis déplace le curseur à gauche du ruban, et imite M.

   La fonction r qui à $\langle M \rangle^{\text{bin}}$ associe $\langle M'_M \rangle^{\text{bin}}$ est calculable.

   M'_M termine sur toute entrée si et seulement si M termine sur ε. Par ailleurs, si M'_M termine, son temps d'exécution en fonction de la taille de son entrée est **constant** – donc, entre autres, polynomial.

   r est donc une réduction de $\mathcal{P}_1$ à $\mathcal{P}_5$, qui est donc indécidable.