Corrigé TD 4
Calculabilité

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

Exercice 1 - Reprise du TD précédent

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 : Est-ce qu'il existe un polynôme p tel que pour tout mot w, M termine en moins de $p(|w|)$ étapes sur w ?

Solution

cf corrigé du TD précédent.

Exercice 2 - Théorème de Rice : subtilités

1. Trouver une propriété non triviale $\mathcal{P}$ telle que le problème suivant est décidable :
   Donnée : Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : $\langle M \rangle^{\text{bin}} \in \mathcal{P}$?
2. On considère les énoncés suivants en français:

   Si $\mathcal{P}$ est une propriété non triviale des langages récursifs,
   alors $\mathcal{P}$ est indécidable.

   Si $\mathcal{P}$ est une propriété non triviale des langages,
   alors $\mathcal{P}$ est indécidable.

   Pour chacun :

- Essayer de le formaliser sous la forme d'un problème de décision au format Entrée/Sortie,
- Prouver, ou faire mentir, l'énoncé formalisé.

Solution

1. Comparé au théorème de Rice, cet énoncé demande une propriété *qui ne porte pas nécessairement sur les langages*. On peut donc par exemple considérer la propriété « Termine en moins de 10 étapes de calcul », qui peut être décidée notamment grâce à l'aide d'une machine universelle.
2. Plusieurs solutions sont acceptables. Je présente ici des solutions proches du cours.

   Tout d'abord, je rappelle que lorsqu'un problème de décision est de la forme :
   Donnée : x selon un format f
   Question : $x \in \mathcal{P}$?,

   le problème est dit *décidable* s'il existe une machine qui reconnaît le langage

   $$\{w \in \Sigma^* \mid w \text{ est de la forme } x \text{ et satisfait au format } f, \text{et } x \in \mathcal{P}\}$$

   - **Langages récursifs**

     Il est tentant d'écrire le problème de décision suivant :
     Donnée : Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing qui s'arrête toujours.
     Question : $\langle M \rangle^{\text{bin}} \in \mathcal{P}$?

     Il faut cependant bien avoir en tête que ce problème de décision est le même que le suivant :
     Donnée : Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
     Question : Est-ce que $\langle M \rangle^{\text{bin}} \in \mathcal{P}$ **et** M s'arrête toujours ?

     Considérons maintenant la définition de la non-trivialité dans les langages récursifs. On a envie de définir comme : « Il existe une machine s'arrêtant toujours qui la vérifie, et une machine s'arrêtant toujours qui ne la vérifie pas. ». Le fait que la propriété soit *sur les langages* peut se définir de même que pour le théorème de Rice.

     Avec toutes ces définitions, le problème est indécidable, avec la même preuve que pour le théorème de Rice. Notamment, le fait que M doive toujours s'arrêter n'entrave pas la preuve, car la machine transformée ne s'arrête pas ssi M n'a pas la propriété.

   - **Langages**

     Ici, il y a un problème pour décrire le langage en entrée. Notamment, l'entrée est un mot de Σ^*, qui est un ensemble dénombrable, car Σ est fini;

alors que l'ensemble des langages est indénombrables. Le problème semble donc impossible à poser avec le formalisme du cours.

Exercice 3 - Théorème de Rice : applications, non applications

Dire si les problèmes suivants sont décidables ou non. On se forcera à utiliser le théorème de Rice lorsque c'est possible.

1. **Donnée :** Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : Est-ce que $\mathcal{L}(M) = \emptyset$?
2. w est un mot fixé.
   Donnée : Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : Est-ce que M s'arrête sur w ?
3. **Donnée :** Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : Est-ce que la tête de lecture passe sur \$ lors du calcul de M sur ε ?
4. **Donnée :** Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : Est-ce que le complémentaire de $L(M)$ est récursivement énumérable ?
5. **Donnée :** Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : Existe-t-il deux mots différents w_1, w_2 de même longueur tels que $w_1, w_2 \in \mathcal{L}(M)$?
6. **Donnée :** Le code de deux machines de Turing $\langle M_1 \rangle^{\text{bin}}$ et $\langle M_2 \rangle^{\text{bin}}$
   Question : $\mathcal{L}(M_1) \cap \mathcal{L}(M_2) = \emptyset$?
7. **Donnée :** Le code $\langle M \rangle^{\text{bin}}$ d'une machine de Turing.
   Question : Est-ce que $\mathcal{L}(M) = \{w \in \Sigma^* \mid |w| \text{ est pair}\}$?

Solution

1. Application du théorème de Rice. C'est bien une propriété sur les langages, il existe une machine qui a un langage vide, et une machine qui a un langage non vide. *(Exhibez-les !)*
2. Ce n'est **pas** une application du théorème de Rice, car ce n'est pas une propriété sur les langages.

   Réduction depuis le problème de l'arrêt sur le mot vide. À partir de M, considérer M'_M qui, sur tout y en entrée, commence par effacer y, puis écrit w, puis place le curseur à gauche puis simule M.
3. Ce n'est **pas** une propriété sur les langages. Notamment, si une machine passe par \$ dans son calcul, on peut en déduire une machine qui commence

par décaler son entrée d'un cran, utilise un caractère £ qui est une imitation de \$, et fait les mêmes calculs. Cette machine a le même langage.

Réduction à partir du problème de l'arrêt sur le mot vide. On construit la machine définie dans le paragraphe précédent, et on rajoute qu'avant toute acceptation ou rejet, la machine se déplace jusqu'au \$.

4. Application du théorème de Rice. C'est une propriété sur les langages, et on sait qu'il existe des langages dont le complémentaire n'est pas récursif, tout comme des langages dont il l'est. *(Exhibez-les !)*
5. Application du théorème de Rice. C'est une propriété sur les langages. Exhibez une machine qui est dedans, et une machine qui ne l'est pas.
6. Application indirecte du théorème de Rice, via une réduction depuis le premier problème de l'exercice, en considérant M_2 une machine qui accepte tout. **Attention**, le théorème de Rice n'est pas applicable tel quel, car il y a deux codes en entrée !
7. Application du théorème de Rice. C'est une propriété sur les langages. Exhibez une machine qui est dedans, et une machine qui ne l'est pas.