TD n° 11

Machines de Turing et problème de l'arrêt

FONCTIONS ET LANGAGES NON RECURSIFS

Exercice 11.1. Que pensez-vous des énoncés suivants.

1. $\nexists M_{halt} : \forall \langle M \rangle, w : M_{halt}(\langle M \rangle, w) = halt(\langle M \rangle, w)$
2. $\forall \langle M \rangle, w : \nexists M_{halt} : M_{halt}(\langle M \rangle, w) = halt(\langle M \rangle, w)$

Exercice 11.2. Extrait du cours :

On dit qu'un problème A se réduit à un problème B si, connaissant une réponse à B, on peut trouver une réponse à A. Plus formellement, on dira qu'un langage A est **Turing-réductible** à un langage B si il existe une machine de Turing avec **oracle** B, qui résout le problème A. Une machine de Turing avec oracle B est une machine qui peut, au cours de son calcul, obtenir autant de réponses qu'elle souhaite sur des questions d'appartenance au langage B : est-ce que tel $w \in B$? est-ce que tel autre $w' \in B$? Et l'oracle pour le langage B lui donne des réponses oui/non instantanément.

1. Comment intégrer formellement ce concept aux machines de Turing ? C'est-à-dire, comment définir les appels à l'oracle, et comment définir la façon dont une MT obtient une réponse de l'oracle ?
2. Donner le diagramme d'une machine de Turing avec oracle $B = \{w \in \{0,1\}^* \mid w \text{ est un palindrome}\}$ qui reconnait le langage $A = \{aw \in \{0,1\}^* \mid w \text{ est un palindrome}\}$.
3. Est-ce utile d'autoriser une machine à avoir plusieurs oracles ?
4. Le langage $A = \{\langle M \rangle \# w \mid M \text{ accepte } w\}$ est-il Turing-réductible au langage

$$B = \{\langle M \rangle \mid M \text{ s'arrête si on la lance sur l'entrée } abba\} \text{ (justifier) ?}$$

Que peut-on en déduire ?

Exercice 11.3. Utiliser le théorème de Rice pour étudier la récursivité des propriétés suviantes.

1. $\{L \mid L = a^*\}$.
2. $\{L \mid aa \in L, \text{ et } \forall k \neq 2 : a^k \notin L\}$.
3. $\{L \mid ab \notin L, \text{ ou } \exists k : ab^k \in L\}$.
4. $\{\{\langle M \rangle \mid M \text{ accepte } \langle M \rangle\}\}$.
5. Plus généralement, que diriez-vous, pour un langage $L_{objectif} \subseteq \Sigma^*$ que l'on souhaite reconnaître, de la propriété $\{L_{objectif}\}$?

EXPRESSIVITE DES MACHINES DE TURING

Exercice 11.4. Le but de cet exercice est de montrer que le modèle des machines de Turing à plusieurs rubans et plusieurs têtes de lecture (une tête de lecture indépendante par ruban, et un seul état pour toute la machine) est équivalent au modèle des machines de Turing.

En fonction de l'état et du symbole lu sur chacun des rubans, la machine peut

— changer d'état,
— écrire un symbole sur chaque ruban,
— déplacer chaque tête de lecture vers la droite ou la gauche indépendament les unes des autres.

1. Donner le type de la fonction de transition δ, et donner un exemple de transition.

Dans l'état initial, l'entrée est écrite sur le premier ruban et tous les autres rubans sont vides. Un mot est accepté si et seulement si la machine entre dans l'état final q_F au cours du calcul.

2. Comment simuler une MT *multi-tape* avec une MT *mono-tape* ?

Exercice 11.5. Le but de cet exercice est de montrer que le modèle des machines de Turing non-déterministes est équivalent au modèle des machines de Turing.

Une machine de Turing non-déterministe peut, à une étape de temps donnée (c'est à dire dans un état et pour un symbole lu), avoir plusieurs transitions possibles. Un calcul d'une machine de Turing est une succession de choix parmi l'ensemble des transitions possibles. Un mot est accepté si et seulement si il existe au moins un calcul qui mène à l'état final.

1. Donner le type de la fonction de transition δ, et donner un exemple de transition.
2. Donner une machine de Turing non déterministe qui reconnaisse le langage $L = \{ww \mid w \in \{a,b\}^*\}$.
3. Soit $r = \max\{|\delta(q,x)| \mid q \in Q \text{ et } x \in \Gamma\}$. Que représente q ?
4. Que pourrait représenter une suite finie de lettres sur l'alphabet $R = \{1,\ldots,r\}$?
5. Peut-on construire une machine de Turing qui énumère (c'est-à-dire écrit un à un et se place dans un état particulier q_e à chaque fois que son ruban contient un mot à énumérer) tous les mots finis sur l'alphabet R ?
6. Montrer que tout langage reconnu par une MT non-déterministe est reconnu par une MT déterministe (indication : trois rubans avec chacun leur tête semble un bon choix).