

Modèles de calcul - TD6

Rappels : On a vu les langages suivants en cours :

- $\{\langle M \rangle \mid M \text{ n'accepte pas } \langle M \rangle\}$ ($\notin RE$)
- $\{\langle M \rangle \mid M \text{ accepte } \langle M \rangle\}$ ($\in RE \setminus R$)
- $\{\langle M, w \rangle \mid M \text{ n'accepte pas } w\}$

Exercice 1 Quelques langages

Les langages suivants sont-ils décidables ? Récursivement énumérables ?

- $L_1 = \{\langle M, k, w \rangle \mid M \text{ s'arrête en } k \text{ étapes sur } w\}$.
- $L_2 = \{\langle M \rangle \mid M \text{ ne s'arrête pas sur } \langle M \rangle\}$.
- $L_3 = \{\langle M, w \rangle \mid M \text{ s'arrête sur } w\}$.

Solution de l'exercice 1

- C'est un langage décidable (donc aussi RE). La machine de Turing que l'on va construire va simuler la machine M , en maintenant en plus un compteur du nombre de pas de simulation effectué. Elle va vérifier à chaque pas simulé si le compteur a dépassé k ou pas. Si la machine dépasse k , la sortie est non. Tant que ce n'est pas arrivé, si la machine s'arrête, on répond oui.
- Pas décidable, ni RE par réduction depuis $\{\langle M \rangle \mid M \text{ n'accepte pas } \langle M \rangle\}$. Étant donnée une machine M , on construit M' qui lance M sur $\langle M \rangle$, si ça renvoie non, boucle, si ça renvoie oui, accepte. M n'accepte pas $\langle M \rangle$ ssi M' ne s'arrête pas, on a donc bien une réduction donc L_2 pas RE.
- C'est un langage récursivement énumérable mais pas décidable. La machine pour L_3 va simuler le calcul M sur w . Si M s'arrête, elle répond oui. Cette machine satisfait la définition de récursivement énumérable (elle accepte L_3). Cependant L_3 n'est pas décidable (faire une petite réduction depuis L_2 vers L_3 + utiliser $R = RE \cap co-RE$ vu en cours et dans exo4).

Exercice 2 Quelques questions

Supposons qu'on a une réduction calculable d'un problème P_1 vers un problème P_2 .

1. Si P_1 est décidable, est-ce que P_2 est décidable ?
2. Si P_2 est décidable, est-ce que P_1 est décidable ?
3. Si P_1 est récursivement énumérable, est-ce que P_2 est récursivement énumérable ?
4. Si P_2 est récursivement énumérable, est-ce que P_1 est récursivement énumérable ?
5. Si P_2 est récursivement énumérable, est-ce que P_1 est décidable ?
6. Si P_2 est décidable, est-ce que P_1 est récursivement énumérable ?

Solution de l'exercice 2

1. Non, pas forcément.
2. Oui, on peut décider P_1 en faisant la réduction, et en reprenant la décision pour P_2 .
3. Non, pas forcément.
4. Oui : on prend une machine qui calcule la réduction puis qui lance le calcul sur P_2 . Si la réponse le mot est dans le langage, ce calcul termine avec oui. Si la machine ne s'arrête pas, le mot n'est pas dans le langage.
5. Non, pas forcément.
6. D'après la question précédente, P_2 décidable implique P_1 décidable, et on sait que P_1 décidable implique P_1 récursivement énumérable.

Exercice 3 Énumérateurs

Un *énumérateur* est une MT déterministe qui liste des mots de Σ^* séparés par un symbole spécial \$.

1. Montrer qu'un langage L est décidable si et seulement s'il existe un énumérateur qui écrit les mots de L par taille croissante (et aucun mot n'appartenant pas à L).
2. Montrer que L est récursivement énumérable si et seulement s'il existe un énumérateur M qui n'écrit que des mots de L et tel que tout mot de L sera écrit à un moment de l'exécution de M .

Solution de l'exercice 3

1. Si L est décidable, on teste les mots par ordre de taille croissant, et on écrit seulement ceux qui sont bon. Dans l'autre sens, si on a un énumérateur, il suffit d'énumérer les mots jusqu'à, ou bien trouver le mot que l'on cherche, et sortir oui, ou bien commencer à énumérer des mots plus grands, et alors on répond non.
2. Si on a un énumérateur, c'est facile de construire une MT qui accepte les mots du langage : sur l'entrée w , on simule l'énumérateur jusqu'à obtenir w . Si on y arrive on accepte, sinon, on continue la simulation. C'est une MT qui accepte exactement les mots de L mais qui ne s'arrête pas si $w \notin L$.

Dans l'autre sens, c'est un peu plus technique. Soit M une MT pour L . On va tester pour $i = 1, 2, \dots$, pour chaque mot w de longueur au plus i , si M termine sur w en i étapes. Si c'est le cas, et que la machine répond oui, on l'écrit. Ainsi tout mot, de longueur ℓ sur lequel la machine s'arrête après k pas, avec un accept, sera listé lors de l'itération $\max(\ell, k)$. Et aucun autre mot ne sera listé.

Exercice 4 Quelques questions un peu plus difficiles

Supposons qu'on a une réduction calculable d'un problème P_1 vers un problème P_2 .

1. Si P_1 et P_2 sont décidables, est-ce qu'il existe une réduction calculable de P_2 vers P_1 ?
2. Si P_1 et P_2 ne sont pas décidables, est-ce qu'il existe une réduction calculable de P_2 vers P_1 ?

Solution de l'exercice 4

1. Il y a une construction artificielle. Disons que l'on connaît un mot accepté et un mot rejeté de P_1 . La réduction commence par calculer la réponse à P_2 (c'est possible puisque P_2 est décidable), puis si la réponse est oui, elle sort le mot accepté que l'on connaît pour P_1 , et sinon le mot rejeté pour P_1 . Donc oui quasi-tout le temps, sauf si P_1 renvoie toujours oui (ou toujours non) et pas P_2 .
2. Non, c'est faux, il y a différent niveau de difficulté pour les problèmes non-décidables. Prenons un problème P_1 qui n'est pas récursivement énumérable, et un problème P_2 qui est récursivement énumérable mais pas décidable. On a une contradiction. (Voir Exo 1.)