Langages formels, calculabilité et complexité
TD11

12 décembre 2014

Exercice 1 Problème de correspondance de Post (*base*)

Soit Σ un alphabet fini et $P = \{(u_1, v_1), \ldots, (u_m, v_m)\}$ un ensemble fini de couples de mots sur Σ. Le **problème de correspondance de Post** (PCP) est de savoir s'il existe une suite non vide finie $(i_j)_{j=1}^{n} \in \{1, \ldots, m\}^n$ telle que $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$.

Le problème se transforme en **problème de correspondance de Post modifié** (PCPM) lorsque le premier couple est imposé : $i_1 = 1$.

1. Montrer que si Σ ne contient qu'une lettre, alors PCP est décidable.

On veut tout d'abord montrer que PCP est indécidable si et seulement si PCPM l'est. Considérons une instance de PCPM sur Σ. Posons $\Sigma' = \Sigma \cup \{\$\}$ et en introduisons les morphismes p et $s : \Sigma \to \Sigma'^*$ tels que $p(a) = \$a$ et $s(a) = a\$$.

2. Montrer que pour tout mot w, $p(w)\$ = \$s(w)$. Montrer qu'on peut trouver une instance de PCP sur Σ' qui a une solution si et seulement si l'instance de PCPM en a une.
3. En déduire l'équivalence entre PCP et PCPM.

Soit $L_\epsilon = \{(M, w) \mid \text{le mot } w \text{ est accepté par la MT } M\}$. On montre maintenant que PCPM est indécidable en réduisant le problème L_ϵ à ce problème. Soit (M, w) une instance de L_ϵ où M est une machine de Turing normalisée : elle a un unique état acceptant q_+, qui est un puits, et lorsqu'un calcul termine en q_+, la bande est vide (remplie de symboles blancs #).

4. Proposer un ensemble de couples de mots tels qu'une solution de PCPM sur ces couples existe si et seulement s'il existe un calcul acceptant. Le mot formé par la concaténation de ces couples pourra être la suite des calculs effectués par M pour accepter w.

Exercice 2 Quelques applications (*avancé*)

Soient $u_1, \ldots, u_m \in \Sigma^*$. On suppose que $\Sigma \cap \mathbb{N} = \emptyset$. On définit le langage

$$L_u = \{u_{i_1} u_{i_2} \cdots u_{i_n} i_n \cdots i_2 i_1 \mid n \geq 0 \text{ et } 1 \leq i_k \leq m \text{ pour tout } 1 \leq k \leq n\} \ .$$

1. Montrer que L_u et algébrique.

Utiliser l'indécidabilité de PCP pour montrer l'indécidabilité des problèmes suivants :

2. Décider si les langages de deux grammaires hors contexte sont disjoints.

Pour les problèmes suivants, il est utile de considérer le langage

$$L'_u = \{w i_n \cdots i_2 i_1 \mid n \geq 0, w \in \Sigma^*, 1 \leq i_k \leq m \text{ pour tout } 1 \leq k \leq n \text{ et } w \neq u_{i_1} u_{i_2} \cdots u_{i_m}\} \ .$$

3. Montrer que L'_u est algébrique et que $(\Sigma + [m])^* \setminus L_u = L'_u \cup \big((\Sigma + [m])^* \setminus \Sigma^*[m]^*\big)$ où $[m] = \{1, 2, \ldots, m\}$.
4. Décider si deux grammaires hors contexte décrivent le même langage.
5. Décider si une grammaire hors contexte est ambiguë.

Exercice 3 Séparation de langages (*base*)

Soient A et B deux langages disjoints. On dit qu'un langage C *sépare* A et B si $A \subseteq C$ and $B \subseteq \bar{C}$.

1. Si A et $\bar{A}$ ne sont séparés par aucun langage décidable, que peut-on en déduire ?
2. Montrer que A et B sont séparés par un langages décidable si A et B sont co-énumerables et disjoints.
3. Montrer qu'il existe des langages récursivement énumérables qui ne sont séparés par aucun langage décidable.

Exercice 4 Théorème de Rice (*base*)

Un état inutile d'une machine de Turing est un état qui n'est jamais visité pendant un calcul.

1. Montrer que le problème de décider si une machine de Turing a des états inutiles est indécidable.
2. Expliquer pourquoi on ne peut pas appliquer le théorème de Rice pour résoudre ce problème.