Logique et Calculabilité 2011-2012. Examen

18 Janvier 2012. Durée 3h.

Tous les documents sont autorisés. Seuls les résultats du cours peuvent être utilisés sans démonstration.

Parmi les problèmes suivants dire ceux qui sont (in)décidables. Justifier. (Ils sont indépendants et listés approximativement par ordre de difficulté croissante).

1. **Donnée :** le code d'une machine de Turing M

   Question : $L(M) \subseteq 0^*$?
2. **Donnée :** Deux formules ϕ, ψ de la logique du premier ordre, sans variable libre.

   Question : ϕ et ψ sont logiquement équivalentes ?
3. **Donnée :** le code d'une machine de Turing qui calcule la fonction (partielle) f

   Question : il existe un mot w tel que $f(w) = w$?
4. **Donnée :** Deux codes de machines de Turing M_1, M_2

   Question : $L(M_1) \cap L(M_2) = \emptyset$?
5. **Donnée :** Deux codes de machines de Turing M_1, M_2 qui s'arrêtent sur toutes les données.

   Question : $L(M_1) \cap L(M_2) = \emptyset$?
6. **Donnée :** Le code d'une machine de Turing M et un entier k

   Question : Toutes les configurations qu'atteint M lors de son calcul sur le mot vide sont de longueur inférieure ou égale à k ?

   La longueur d'une configuration (q, w, w') est la somme des longueurs de w et de w', sans compter les blancs.
7. **Donnée :** un ensemble fini de symboles de fonction unaires et de constantes $\mathcal{F}$, un ensemble fini de symboles de prédicats binaires $\mathcal{P}$ et un ensemble fini $\mathcal{S}$ de clauses monotones construites sur $\mathcal{P}, \mathcal{F}$

   Question : $\mathcal{S}$ est satisfaisable ?

   Une clause C (dont toutes les variables sont implicitement quantifiées universellement) est *monotone* si elle est de l'une des formes suivantes :
   (a) C est réduite à un littéral
   (b) $C = \neg P(x_1, \ldots, x_n) \vee P(u_1, \ldots, u_n)$ où $x_1, \ldots, x_n$ sont des variables distinctes et, pour tout i, x_i est une variable de u_i.
8. **Donnée :** Une fonction récursive primitive f à un argument

   Question : Pour tout $n \in \mathbb{N}$, $f(n) = 0$?

Solution

1. C'est indécidable, par le théorème de Rice : la propriété d'être contenu dans 0^* est en effet non triviale puisque le langage vide est récursivement énumérable, comme le langage réduit au singleton $\{1\}$.
2. C'est indécidable, par réduction de l'insatisfaisabilité en logique du premier ordre : il suffit de choisir $\bot$ pour ψ ; ϕ est logiquement équivalente à $\bot$ ssi ϕ est insatisfaisable.
3. C'est indécidable : on réduit le problème de l'arrêt sur le mot vide. À partir de la machine M (instance du problème de l'arrêt sur le mot vide), on constuit une machine M' qui, sur l'entrée w, sauvegarde w (sur un deuxième ruban par exemple), puis simule M sur le mot vide. Lorsque M est sur le point de s'arrêter, M' écrit w sur son ruban, puis efface le reste du ruban et s'arrête. $M'(w) = w$ ssi M s'arrête sur le mot vide.
4. Le probème, étant donnée M, de savoir si $L(M) = \emptyset$ est indécidable par le théorèm de Rice (il s'agit d'une propriété non triviale des langages récursivements énumérables, puisqu'il existe des langages non vides et un langage vide). On réduit ce problème à celui de l'énoncé en choisissant $M_1 = M$ et M_2 une machine qui accepte tous les mots : $L(M_1) \cap L(M_2) = \emptyset$ ssi $L(M_1) = \emptyset$. Le problème de l'énoncé est donc indécidable.
5. C'est indécidable. On réduit, comme dans la question précédente, le vide de $L(M)$. On considère pour cela une machine M_1 qui, sur la donnée (x, k) simule au plus k étapes du calcul de M sur x : si M s'arrête en moins de k étapes et accepte x, alors M_1 accepte (x, k). Si M calcule en plus de k étapes ou si M s'arrête en moins de k étapes et rejette x, alors M_1 rejette (x, k). Enfin, si la donnée de M_1 n'est pas de la forme (x, k), M_1 rejette.
   $L(M) = \emptyset$ ssi $L(M_1) = \emptyset$ et M_1 s'arrête toujours. On choisit enfin pour M_2 une machine qui accepte dès le premier mouvement, quel que soit le mot d'entrée. $L(M_1) \cap L(M_2) = \emptyset$ ssi $L(M) = \emptyset$ et M_1, M_2 s'arrêtent sur toute donnée.
6. C'est décidable. Étant donnés M, k, le nombre de configurations de M de longueur inférieure ou égale à k est fini (borné par $N = |\Sigma|^k \times |Q|$). L'algorithme suivant permet alors de répondre à la question :

   Soit $\gamma = \gamma_0$ et $L = \emptyset$.
   Tant que $\gamma \notin L$ et $|\gamma| \leq k$ faire
   Ajouter γ a L
   Remplacer γ par la configuration suivante de la machine.
   Fin Tant Que
   Si $|\gamma| > k$ repondre non, sinon repondre oui.

   L'algorithme s'arrête en au plus N étapes. Si $|\gamma| > k$, la réponse donnée est trivialement correcte. Sinon, il existe deux entiers m_1, m_2 tels que $m_2 \geq 1$ et $m_1 + m_2 \leq N$ et une configuration γ_{m_1} telle que $\gamma_0 \vdash_M^{m_1} \gamma_{m_1} \vdash_M^{m_2} \gamma_{m_1+m_2} = \gamma_{m_1}$ et toutes les configurations intermédiaires sont de longueur inférieure ou égale à k. Dans ce cas, pour tout n, si $\gamma_0 \vdash_M^n \gamma_n$, alors $\gamma_n \in \{\gamma_0, \ldots, \gamma_{m_1+m_2-1}\}$ et donc $|\gamma_n| \leq k$.
7. C'est indécidable : on réduit le probème de correspondance de Post modifié. Soient $(u_0, u_1, \ldots, u_n), (v_0, v_1, \ldots, v_n)$ où $u_i, v_i \in \Sigma^*$ une instance du problème de correspondance de Post. On construit $\mathcal{F} = \Sigma \uplus \{0\}$, tous symboles supposés unaires sauf 0 qui est une constante et $\mathcal{P} = \{P\}$. Si $u \in \Sigma^*$, $\overline{u}(x)$ est défini par récurrence par $\overline{\epsilon}(x) = x$ et $\overline{ua}(x) = a(\overline{u}(x))$. L'ensemble de clauses est défini par :

(a) $P(\overline{u_0}(0), \overline{v_0}(0))$

(b) $\neg P(x,y) \vee P(\overline{u_i}(x), \overline{v_i}(y))$ pour $i = 1, .., n$

(c) $\neg P(x,x)$

Cet ensemble de clauses est bien fini et monotone.

Montrons que PCP modifié a n'a pas de solution ssi l'ensemble de clauses $\mathcal{S}$ ci-dessus est satisfaisable.

Par le théorème de Herbrand, $\mathcal{S}$ est satisfaisable ssi il a un modèle de Herbrand. Montrons, par récurrence sur k que, pour toute séquence d'indices $0 = i_0, i_1, \ldots, i_k$, tout modèle de Herbrand $\mathcal{H}$ de $\mathcal{S}$ contient $P(\overline{u_{i_0} \cdots u_{i_k}}(0), \overline{v_{i_0} \cdots v_{i_k}}(0))$.

Pour $k = 0$, $P(\overline{u_0}(0), \overline{v_0}(0))$ est dans $\mathcal{H}$.

Si $P(\overline{u_{i_0} \cdots u_{i_{k-1}}}(0), \overline{v_{i_0} \cdots v_{i_{k-1}}}(0)) \in \mathcal{H}$, comme $\mathcal{H} \models \neg P(x,y) \vee P(\overline{u_{i_k}}(x), \overline{v_{i_k}}(y))$, $P(\overline{u_{i_k}}(\overline{u_{i_0} \cdots u_{i_{k-1}}}(0)), \overline{v_{i_k}}(\overline{v_{i_0} \cdots v_{i_{k-1}}}(0))) \in \mathcal{H}$. Or $\overline{uv}(x) = \overline{v}(\overline{u}(x))$ pour tous mots u, v par récurrence sur la longueur de v. On obtient donc bien le résultat souhaité.

Donc, si $\mathcal{H} \models \mathcal{S}$, en particulier $\mathcal{H} \models \neg P(x,x)$ et donc il n'existe aucune séquence d'indices $0 = i_0, i_1, \ldots, i_k$ telle que $u_{i_0} \cdots u_{i_k} = v_{i_0} \cdots v_{i_k}$.

Réciproquement, si $\mathcal{S}$ est insatisfaisable, la structure de Herbrand $\mathcal{H}$ qui consiste en l'ensemble des $P(\overline{u_{i_0} \cdots u_{i_k}}(0), \overline{v_{i_0} \cdots v_{i_k}}(0))$ pour les séquences $0 = i_0, \ldots, i_k$, n'est pas un modèle de $\mathcal{S}$. Or cette structure est un modèle de $P(\overline{u_0}(0), \overline{v_0}(0))$ et des clauses $\neg P(x,y) \vee P(\overline{u_i}(x), \overline{v_i}(y))$ pour $i = 1, .., n$. Donc $\mathcal{H} \not\models \neg P(x,x)$, autrement dit $\mathcal{H} \models \exists x.P(x,x)$: il existe une séquence d'indices $0 = i_0, i_1, \ldots, i_k$ telle que $u_{i_0} \cdots u_{i_k} = v_{i_0} \cdots v_{i_k}$.

8. On montre que le problème est indécidable, par réduction du problème de l'arrêt sur le mot vide.

   Soit M une machine de Turing. Les configurations de la machine sont vues comme des triplets d'entiers non nuls en base $|\Sigma| + |Q| + 2$. La fonction $f_M : \mathbb{N} \to \mathbb{N}$ telle que $f(0) = 0$ et qui associe au code $J(w, J(q, w'))$ d'une configuration $\gamma = (w, q, w')$ le code de la configuration suivante de la machine (ou bien 0 si la machine est dans une configuration d'arrêt) est une fonction récursive primitive (vu en cours). On considère alors la fonction $g_M : \mathbb{N} \to \mathbb{N}$ qui, à $J(n,m)$ associe n si $m = 0$ et $J(f_M(n), m-1)$ si $m > 0$. g_M est récursive primitive car les fonctions J, K, L, f_M sont récursives primitives.

   La fonction g_M appliquée à $J(c_0, m)$, où c_0 est le code de la configuration initiale de M, retourne ainsi :

   $$\begin{cases} 0 \text{ Si la machine } M \text{ s'arrête en moins de } m \text{ étapes} \\ c_m \text{ Code de la configuration de } M \text{ après } m \text{ étapes de calcul sinon} \end{cases}$$

   Soit alors la fonction $h : \mathbb{N} \to \mathbb{N}$ telle que $h(m) = 1$ si $g_M(J(c_0, m)) = 0$ et $h(m) = 0$ sinon. h est récursive primitive et $h(m) = 0$ si la machine M ne s'est pas arrêtée après m étapes de calcul. Donc M ne s'arrête pas sur le mot vide ssi $\forall m \in \mathbb{N}.h(m) = 0$. Ce qui prouve l'indécidabilité du problème proposé.