Examen du cours de L3: Calculabilité

9 novembre 2017, durée: 3 heures

Tous les documents sont autorisés, les appareils électroniques sont interdits. Les résultats vus en TD, s'ils sont utilisés, doivent être redémontrés. Pour chacun des problèmes suivants, dire s'ils sont décidables ou non. Justifier. (NB: les 3 dernières questions sont plus difficiles).

1. **Donnée:** (Le code d')une machine de Turing M
   Question: M accepte l'ensemble des mots de la forme $\{w \cdot w \mid w \in \{0,1\}^*\}$.

2. **Donnée:** (Les codes de) deux machines de Turing M_1, M_2
   Question: pour tout x, si M_1 s'arrête sur x, alors M_2 s'arrête sur x.

3. **Donnée:** Deux suites finies de mots $u_1, \ldots, u_n$, $v_1, \ldots, v_n$
   Question: Il existe une suite finie d'indices, *de longueur paire* $i_1, \ldots, i_{2k}$ telle que
   $$u_{i_1} \cdots u_{i_{2k}} = v_{i_1} \cdots v_{i_{2k}}$$

4. **Donnée:** (Les code de) machines de Turing M_1, M_2, M_3
   Question: $L(M_1) \cup L(M_2) \subseteq L(M_3)$ ou $L(M_2) \cup L(M_3) \subseteq L(M_1)$

5. **Donnée:** `pas de donnée`
   Question: Le problème de l'arrêt sur le mot vide pour les machines de Turing a au plus 5 états et sur un alphabet d'au plus 7 symboles est décidable.

6. **Donnée:** Une machine de Turing M
   Question: Sur toute donnée $\underbrace{0 \cdots 0}_{n}$, M calcule $\underbrace{0 \cdots 0}_{f(n)}$ où f est récursive primitive.

7. **Donnée:** Une machine de Turing M et un entier n
   Question: Pour toute donnée x, M s'arrête sur x après au plus n étapes de calcul.

8. **Donnée:** Une machine de Turing M (à un seul ruban)
   Question: Il existe un mot w tel que M accepte w après au plus $|w|$ étapes de calcul

9. **Donnée:** Une suite finie de mots $u_1, \ldots, u_n \in \Sigma^*$
   Question: Il existe une suite d'entiers $i_1, \ldots, i_k \in \{1, \ldots, n\}$ telle que
   $$u_{i_1} \cdots u_{i_k} = u_{1+i_1} \cdots u_{1+i_k}$$

10. **Donnée:** Un ensemble de paires d'entiers S, dont la fonction caractéristique est primitive récursive, un entier n
    Question: il existe $m \in \mathbb{N}$ tel que $(m, n) \in S$

Solution

1. Indécidable. On utilise le théorème de Rice: la propriété $\{w \cdot w \mid w \in \{0,1\}^*\} \subseteq L(M)\}$ est une propriété non triviale des langages récursivement énumérables puique le langage vide ne la vérifie pas et le langage Σ^* la vérifie.

   Taux de réussite: 83%. Remarque: l'énoncé était ambigü et la question pouvait être comprise comme: $L(M) = \{w \cdot w \mid w \in \{0,1\}^*\}$. Dans ce cas, on peut aussi appliquer le théorème de Rice, mais il faut montrer que $\{w \cdot w \mid w \in \{0,1\}^*\}$ est récursivement énumérable. Ceux qui l'ont fait (correctement) ont obtenu un petit bonus pour compenser le temps perdu.
   Bien sûr, des réductions directes étaient aussi possibles.

2. Indécidable. On réduit le problème de l'arrêt universel (prouvé indécidable en cours). À la donnée $< M >$ on associe $< M_1 >, < M_2 >$ où M_1 est une machine qui s'arrête sur toute donnée et $M_2 = M$. Alors (M s'arrête sur toute donnée) ssi $\forall x.(M_1$ s'arrête sur $x \Rightarrow$ M_2 s'arrête sur $x)$.

   Taux de réussite: 79%. Les erreurs principales sont des erreurs logiques.

3. Indécidable. On réduit PCP. La fonction de réduction est l'identité: $u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}$ ssi $u_{i_1} \cdots u_{i_k} \cdot u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k} \cdot v_{i_1} \cdots v_{i_k}$. En effet, pour tous mots u, v, $u \cdot u = v \cdot v$ implique $2|u| = 2|v|$, donc $|u| = |v|$. Les préfixes de longueur $|u|$ de $u \cdot u$ et $v \cdot v$ sont identiques, et donc $u = v$. Par conséquent, PCP a une solution ssi il a une solution dans laquelle la suite d'indices est de longueur paire.

   Taux de réussite:74%. Les principales erreurs viennent de codages complexes qui sont parfois incorrects

4. Indécidable. On réduit le problème: étant donnée $< M >$, est ce que $L(M) = \emptyset$? Ce dernier problème a été montré indécidable en cours comme application du théorème de Rice.

   On choisit alors pour $M_1 = M_3$ une machine qui refuse toujours et $M_2 = M$. Cette réduction est calculable et $((L(M_1) \cup L(M_2)) \subseteq L(M_3)) \vee ((L(M_2) \cup L(M_3)) \subseteq L(M_1))$ ssi $\emptyset \cup L(M) \subseteq \emptyset \vee L(M) \cup \emptyset \subseteq \emptyset$ ssi $L(M) = \emptyset$.

   Taux de réussite: 77%. Commentaire: le théorème de Rice est un énoncé qui concerne les langages récursivement énumérables, pas les paires ou les triplets de langages récursivement énumérables.

5. Décidable. Comme il n'y a pas de données, la machine qui accepte tout ou la machine qui refuse toujours répond à la question (sans qu'on sache laquelle est la bonne).
   Taux de éussite: 83%

6. Indécidable. On réduit le problème de l'arrêt sur le mot vide. À la machine M, on associe la machine M_1 qui

   (a) Simule sur un ruban auxilliaure M sur le mot vide

   (b) Si (la simulation de) M s'arrête, efface le ruban, écrit 0 et s'arrête.

Si M s'arrête sur le mot vide, alors M_1, sur la donnée $0 \cdots 0$, renvoie 0. M_1 ne s'arrête pas si M ne s'arrête pas sur le mot vide.

Ainsi M_1 calcule la fonction nulle sur les données $0 \cdots 0$ si et seulement si M s'arrête sur le mot vide. Comme la fonction nulle est primitive récursive, on obtient l'indécidabilité du problème.

Taux de réussite: 60%. Commentaires: Certains élèves on interprété l'énoncé avec une fonction f fixée (donc une question paramétrée par f). Ceci ne change presque rien à la solution: au lieu d'effacer $0 \cdots 0$ et écrire 0 à létape 2, il suffit de vérifier que la donnée est de la forme $0 \cdots 0$, puis de simuler une machine qui implémente f. Une telle machine existe puisque les fonctions récursives primitives sont calculables.

7. Décidable. On commence par remarquer que (pour tout mot w tel que $|w| \leq n$, M s'arrête sur w en moins de n étapes) si et seulement si (pour tout mot w, M s'arrête sur w en moins de n étapes). En effet, pour un mot $w = w_1 \cdot w_2$ avec $|w_1| = n$, $q_0\$w \vdash_M^n w_1' q w_2' \cdot w_2$ où $w_2' \neq \epsilon$, puisque la tête de lecture de M s'est déplacée d'au plus n symboles vers la droite. Si M s'arrête sur w_1 en moins de n étapes, elle s'arrête donc sur $w_1 \cdot w_2$ en moins de n étapes.

   Pour décider ensuite de l'arrêt en moins de n étapes sur tous les mots de longueur au plus n, on énumère tous ces mots et on simule M dessus. On fixe l'énumération, par longueur croissante puis lexicographiquement pour les mots de même longueur. Le succcesseur d'un mot dans cette énumération est calculable: soit M_s la machine qui effectue ce calcul.

   La machine qui décide du problème de lénoncé est ainsi une machine à plusieurs rubans, qui conserve le mot courant w et sa longueur ℓ sur un ruban auxilliaire. Elle utilise la machine universelle M_u ainsi que la machine M_s.

   (a) Initialement $w = \epsilon$ et $\ell = 0$, $b = 1$
   (b) Tant que $\ell \leq n$ et b faire
      i. Simuler (en utilisant M_u) le calcul de M sur w jusqu'à ce que M s'arrête ou que le nombre d'étapes de calcul simulées s soit strictement supérieur à n.
      ii. Si $s > n$, alors $b = 0$, sinon $w = M_s(w)$ et $\ell = |w|$
   (c) Répondre b.

   Taux de réussite: 26%. Commentaire: il faut faire la remarque qu'il suffit de considérer les mots de longueur inférieure ou égale à n, sans quoi la simulation sur tous les mots ne termine pas. Lénumération des mots dans l'ordre lexicographique ne marche pas puisque $b > a \cdots a$ dans cet ordre, et on ne considérera donc jamais les mots commençant par b. Enfin, et c'est le seul endroit où c'est nécessaire dans cet examen, il faut utiliser la machine universelle puisqu'ici $< M >$ est une donnée dela machine que l'on construit (celle-ci ne peut pas dépendre de M !). C'est différent dans les réductions des autres questions, dans lesquelles la machine que l'on construit dépend du code de M.

8. Indécidable. On réduit le problème de l'arrêt de M_0 sur ϵ. On construit d'abord la machine M_1 à deux rubans qui sur la donnée w, simule $|w|$ étapes du calcul de M_0 sur ϵ et accepte si M_0 s'arrête: les états de M_1 sont les mêmes que ceux de M_0 et $\delta_1(q, a, b) = q', (a, \rightarrow), (b', d)$ si $\delta_0(q, b) = q', b', d$ et $a \neq B$. $\delta_1(q, B, b) = \textbf{reject}$.

M_1 effectue bien au plus $|w|$ étapes de calcul, mais utilise un ruban auxilliaire. On considère alors la machine M a un seul ruban qui simule M_1, comme vu dans le cours. Cette machine M calcule, a priori, en temps $O(|w|^2)$. Mais, si M_0 s'arrête après N étapes de calcul sur la donnée ϵ, M s'arrêtera sur tout w tel que $|w| \geq N$. Si n est le temps de calcul de M sur un mot de longueur N, alors, pour $|w| \geq n$, le temps de calcul de M sur w est borné par $|w|$ et M accepte w. Réciproquement, si M_0 ne s'arrête pas sur ϵ, M n'accepte aucun mot.

On a ainsi réduit l'arrêt de M_0 sur ϵ au problème de l'énoncé.

Taux de réussite 13%. Commentaire: plusieurs élèves ont proposé une autre solution assez jolie, qui consiste à construire une machine qui ignore son entrée, mais sans l'effacer, et simule M_0 sur ϵ. Cette solution est correcte, à condition de bien expliquer le fonctionnement de la machine quand M_0 doit faire une transition sur un blanc. Seules deux copies l'ont bien fait.

9. Indécidable. On réduit PCP. Si $u_1, \ldots, u_n, v_1, \ldots, v_n$ est la donnée de PCP, on définit $w_1 =< \#\overline{u_1}$, $w_2 =< \widetilde{v_1}$, $w_{2i+1} = \overline{u_i}$, $w_{2i+2} = \widetilde{v_i}$, $w_{2n+3} =>$, $w_{2n+4} = \# >$. $\overline{a \cdot u} = a\# \cdot \overline{u}$ et $\widetilde{a \cdot v} = \#a \cdot \widetilde{v}$. Il faut montrer que si $w_{i_1} \cdots w_{i_k} = w_{1+i_1} \cdots w_{1+i_k}$, alors le PCP original a une solution. Pour cela, on remarque que $i_1 = 1$ (comme dans la réduction de PCP modifié à PCP), puis, par récurrence, que i_j est impair.

   En effet, $i_1, \ldots, i_j$ sont impairs, $w_{i_1} \cdots w_{i_j} =< \#\overline{u_1} \cdots \overline{u_{m_j}}$ où $m_j = \frac{i_j - 1}{2}$ et $w_{1+i_1} \cdots w_{1+i_j} =< \widetilde{v_1} \cdots \widetilde{v_{m_j}}$. En particulier, la dernière lettre de $w_{i_1} \cdots w_{i_j}$ est $\#$ (sauf si $j = k$) et une lettre sur deux de ce mot est $\#$. De même, la dernière lettre de $w_{1+i_1} \cdots w_{1+i_j}$ n'est pas $\#$ et une lettre sur deux est $\#$. Si $j \neq k$, $w_{1+i_{j+1}}$ doit donc commencer par $\#$ ou bien $w_{i_{j+1}}$ doit commencer par une lettre $a \notin \{<, >, \#\}$. Dans les deux cas, i_{j+1} doit être impair.

   La séquence $i_1, \ldots, i_k$ est donc une solution du PCP de départ (en effaçant les symboles $\#, >, <$ des w_i).

   Réciproquement, si $u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}$, alors

   $$w_1 \cdot w_{2i_1+1} \cdots w_{2i_k+1} \cdot w_{2n+3} = w_2 \cdot w_{2i_1+2} \cdots w_{2i_k+2} \cdot w_{2n+4}$$

   Nous donne une solution de notre problème.

   Taux de réussite: 9%. Plusieurs élèves ont eu des bonnes idées, mais la solution correcte de bout en bout n'apparait que dans une copie.

10. Indécidable. On réduit le problème de l'arrêt de M sur ϵ (en supposant de plus que, quand M s'arrête, le ruban est vide). Pour une configuration γ de M, on note $\overline{\gamma}$ son codage dans les entiers. γ_0 est la configuration initiale de M.

    On choisit $S = \{(m, \overline{\gamma}) | \gamma_0 \vdash_M^{\leq m} \gamma\}$. D'après le cours, la fonction qui à m associe $\overline{gamma}$ telle que $\gamma_0 \vdash_M^{\leq m} \gamma$ est récursive primitive. Donc S est primitif récursif et, si on choisit $n = \overline{\gamma_f}$ (configuration finale), l'existence de m telle que $(m, n) \in S$ est équivalente à l'arrêt de M.

    Taux de réussite: 16%. Comme l'ont remarqué plusieurs copies, on pouvait aussi choisir $n = 0$ et considérer un prédicat différent.