École polytechnique — X2022 — INF412 Fondements de l'informatique

Calculabilité

(corrigé)

1 Questions de décidabilité

Parmi les problèmes suivants, lesquels sont décidables, lesquels sont indécidables ? (Argumentez votre réponse.)

Question 1.1. *Déterminer si le langage accepté par une machine de Turing M ne contient que des palindromes (un palindrome est un mot dont l'ordre des lettres reste le même qu'on le lise de gauche à droite ou de droite à gauche, comme "anna").*

Solution : C'est une application directe du Théorème de Rice : dans le cas où l'alphabet a plus d'une lettre, la propriété du langage est non-triviale puisque le langage réduit au mot a la possède, alors que le langage réduit au mot ab ne la possède pas. Dans le cas où l'alphabet a une (ou zéro) lettre, la propriété est triviale. □

Question 1.2. *Déterminer si une machine de Turing M et un mot w sont tels que M accepte l'entrée w en n'utilisant (= écrivant) aucune autre case que celles qui contiennent initialement le mot w.*

Solution : Tout d'abord, on peut observer que le théorème de Rice ne s'applique pas : la propriété donnée est une propriété de la machine et non du langage qu'elle reconnaît.

La réponse est que ce problème est décidable. Il est facile en simulant M de détecter si M essaye d'accéder à une case du ruban hors de celles qui contiennent initialement w (et alors on refuse), ou si M accepte (et alors on accepte). Reste à détecter les cas où M ne termine pas et les refuser. Pour cela, il suffit au simulateur de garder trace de toutes les configurations rencontrées lors de l'exécution (c'est-à-dire les triplets (case du ruban, état de la machine, lettre lue)). Comme le nombre de configurations est fini, une machine qui ne termine pas doit nécessairement passer deux fois par la même configuration, ce que le simulateur peut détecter, et terminer alors en refusant. □

Question 1.3. *Déterminer si une machine de Turing M et un mot w sont tels que M accepte l'entrée w en n'utilisant aucune des cases initialement blanches à gauche de celles qui contiennent initialement w.*

Solution : À nouveau, il faut bien comprendre que le théorème de Rice ne s'applique pas, puisque la propriété porte sur la machine et non sur le langage qu'elle reconnaît.

Ce problème est indécidable. Pour le prouver, on construit une réduction du problème de l'arrêt vers ce problème (et non l'inverse !). Soit M une machine de Turing, et w une entrée. Considérons la machine M' qui simule M en décalant systématiquement d'une position vers la droite le contenu du ruban de M avant chaque étape. Par construction, M' vérifie la propriété de la question 3. si et seulement si M accepte w. □

2 Décidabilité et langages rationnels

Comme il a été vu à la PC 4, les machines de Turing dont toutes les transitions sont de la forme $\delta(q_1, \ell) = (q_2, \ell, \rightarrow)$ sont classiquement appelées automates finis déterministes (noté DFA pour *deterministic finite automaton*) et les langages qu'elle reconnaissent sont appelés rationnels. Les questions qui suivent abordent ces machines et ces langages du point de vue de la décidabilité.

Question 2.1. *Montrer que le langage* $\mathsf{All}_{\mathsf{DFA}} := \{\langle A \rangle \mid A \text{ est un DFA et } L(A) = \Sigma^*\}$ *est décidable.*

Solution : Pour montrer que ce langage est décidable, il suffit de montrer qu'on peut écrire une machine de Turing qui le reconnaît. Une telle machine est obtenue en composant des machines qui reconnaissent des langages dont l'intersection est $\mathsf{All}_{\mathsf{DFA}}$:

1. les mots codant une machine de Turing tel que décrit dans le poly (§8.1.2). Une telle machine n'est pas difficile à écrire : le langage est "presque rationnel" (il l'est pour un nombre d'états et de symboles donné) ;
2. les mots correspondant à un DFA :
   — aucune des transitions ne contient le déplacement vers la gauche ($\leftarrow$) ou l'absence de déplacement ($|$), ce qui revient à vérifier que le mot ne contient des suites 1^k que pour $k \in \{0, 1, 5\}$. Il s'agit donc d'un sous-ensemble du langage rationnel $(0 + (1|11111))^*$;
   — la lecture d'un B, et elle seule, envoie vers l'état de rejet ou d'acceptation : les transitions $\delta(q_i, X_j) = (q_k, X_l, m)$ avec $q_k \in \{q_a, q_r\}$ vérifient $X_j = B$, ce qui est facile à tester ;
   — la machine n'écrit pas sur son ruban, ou plus exactement écrit la même lettre qu'elle lit ;
3. $L(A) = \Sigma^\star$: l'état de rejet n'est pas accessible. Une première approche consiste à tester que dans une transition $\delta(q_i, X_j) = (q_k, X_l, m)$ on doit avoir $q_k \neq q_r$, ce qui est encore facile à tester, mais ce n'est pas tout à fait suffisant : il se pourrait que l'état q_i ne soit jamais atteint. Le test à réaliser est donc plutôt la question de savoir s'il existe un chemin depuis l'état initial vers l'état de rejet (un cas particulier du problème REACH), ce qui peut être testé par un parcours de graphe.

□

Question 2.2. *Le problème* $\mathsf{Eq}_{\mathsf{DFA}}$ *a pour donnée les codages* $\langle A \rangle$ *et* $\langle B \rangle$ *de deux DFAs, et consiste à déterminer si* $L(A) = L(B)$*. Montrer que* $\mathsf{Eq}_{\mathsf{DFA}}$ *est décidable. [Indication : le complémentaire, l'union et l'intersection de langages rationnels sont rationnels, et les (codages d')automates correspondants sont calculables.]*

Solution : Il suffit de tester que $(\overline{L(A)} \cup L(B)) \cap (L(A) \cup \overline{L(B)}) = \Sigma^*$. La construction de l'automate correspondant au membre gauche est calculable d'après l'énoncé, et la conclusion découle de la question précédente. □

Question 2.3. *Montrer que le langage* Rat *des* $\langle M\rangle$ *tels que* M *est une machine de Turing et* $L(M)$ *est reconnu par un DFA est indécidable. [Indication : le langage* $\Lambda = \{0^m 1^m \mid m \in \mathbb{N}\}$ *est un exemple de langage qui n'est pas reconnu par un DFA.]*

Solution : C'est une conséquence du théorème de Rice. Il suffit de prouver qu'il existe une machine de Turing qui reconnaît un langage rationnel (et c'est le cas de tous les DFAs) et une autre qui reconnaît un langage non-rationnel (une telle machine reconnaissant Λ a été vue dans la PC sur les machines de Turing) et le théorème de Rice conclut. □

Question 2.4. *Si* $A \leq_m B$ *et* B *est reconnu par un DFA, est-ce que* A *est reconnu par un DFA ? Justifier.*

Solution : Le résultat du cours (Prop. 8.2) donne une réponse positive à la variante de cette question où "reconnu par un DFA" est remplacé par "décidable". Pourtant ici, la réponse est négative : par exemple, si $A = \Lambda$ le langage de la question précédente, on considère la fonction f qui envoie le mot w sur 1 si $w \in \Lambda$ et sur 0 sinon. Cette fonction est calculable, et l'ensemble d'arrivée $\{1\}$ est reconnu par un DFA trivial. □

Question 2.5. *Montrer que* A *est décidable si et seulement si* $A \leq_m 0^*1^*$.

Solution : Si $A \leq_m 0^*1^*$ alors clairement A est décidable, puisque le langage 0^*1^* est rationnel. Dans l'autre sens, il suffit de procéder comme à la question précédente, en prenant la fonction f qui vaut 10 sur un $w \notin A$ et 01 sinon. Cette fonction est calculable puisque A est décidable, et on a bien la propriété que $f(w) \in 0^*1^* \Leftrightarrow w \in A$. □

3 Facultatif : unions et intersections de langages

Question 3.1. *Soient* A *et* B *deux langages semidécidables (= récursivement énumérables). Montrez que leur union et leur intersection le sont aussi.*

Solution : Notons a et b les machines de Turing qui reconnaissent A et B. Pour l'intersection : sur l'entrée x, simuler $a(x)$ puis simuler $b(x)$. Pour l'union : simuler simultanément les calculs de $a(x)$ et $b(x)$ (faire $0, 1, 2\ldots$ pas des deux calculs), terminer dès que l'un des deux calculs termine. □

Nous pouvons nous poser la même question concernant des unions et intersections infinies.

Question 3.2. *Soit* A_k *une famille de langages semidécidables, indexée par* $k \in \mathbb{N}$, *donnés par une fonction calculable* a *telle que* $a(k)$ *désigne le codage de la machine de Turing qui reconnaît* A_k. *Montrez que leur union est également semidécidable.*

Solution : On a $x \in \bigcup_k A_k$ si et seulement s'il existe (y, z) tels que le calcul « calculer $a(y)$ puis exécuter la machine $a(y)$ sur l'entrée x » termine en au maximum z pas, ce qui se teste effectivement. Il suffit donc d'énumérer tous les couples (y, z). □

On admettra que l'ensemble suivant n'est pas semidécidable :

$$T = \{x \mid \forall y \text{ La machine de codage } x \text{ termine sur l'entrée } y\}.$$

Question 3.3. *Qu'en est-il d'une intersection infinie de langages semidécidables ?*

Solution : Soit A_k l'ensemble des machines de Turing qui terminent sur l'entrée k. Cet ensemble est clairement semidécidable. L'intersection de tous ces ensembles est l'ensemble T des machines de Turing qui terminent sur toute entrée, qui n'est pas semidécidable, comme admis. □