École polytechnique — X2021 — INF412 Fondements de l'informatique

PC6 notée

Cet énoncé comporte quatre parties indépendantes et qui pourront être résolues dans n'importe quel ordre. Dans chaque partie, on pourra, pour répondre à une question, admettre les résultats dont on demande la démonstration aux questions *précédentes*. Il n'est pas nécessaire de traiter toutes les questions pour avoir la note maximale. Les correcteurs vous remercient d'avance d'écrire lisiblement.

1 Quelques questions de calculabilité

Les problèmes de décision suivants sont-ils décidables ? Justifier.

Question 1.1. *Déterminer si le langage reconnu par une machine de Turing est fini.*

Question 1.2. *Déterminer si une machine de Turing M accepte une entrée w après un nombre pair de transitions.*

Question 1.3. *Déterminer si le langage reconnu par une machine de Turing est aussi reconnu par une machine avec plus de* 10^{42} *états.*

2 Machines de Turing et mots binaires

On considère des machines de Turing sur l'alphabet $\Sigma = \{0, 1\}$. L'objectif est d'implémenter certaines opérations sur les entiers codés en binaire. On rappelle par exemple que le codage binaire de 13 est 1101, ou bien 001101 (on autorisera des codages qui débutent par le bit 0, et le codage de 0 sera toujours non vide).

On demande des machines de Turing explicitement décrites. Pour chaque machine, on donnera quelques phrases expliquant leur fonctionnement. Toute réponse de type « une machine avec 50 états sans explication » sera refusée.

Question 2.1. *Proposer une machine de Turing qui détermine si un entier est nul.*

Question 2.2. *Proposer une machine de Turing qui détermine si un entier est pair.*

Question 2.3. *Proposition une machine de Turing qui multiplie un entier par deux.*

Question 2.4. *Proposer une machine de Turing qui calcule le successeur d'un entier.*

Question 2.5. *Proposer une machine de Turing qui calcule le prédécesseur d'un entier supposé non nul.*

Question 2.6. *Proposer une machine de Turing qui calcule la somme de deux entiers séparés par un caractère spécial « # ». Par exemple, le résultat sur l'entrée* 110#101 *sera* 1011 *(car* $6 + 5 = 11$*). On s'autorisera à utiliser des machines auxiliaires qu'on introduira explicitement si elle ne l'ont pas déjà été à des questions précédentes.*

3 Groupes sans torsion

On considère la signature avec une constante « 1 », une fonction d'arité 2 « $\times$ » et l'égalité comme unique symbole de relation, d'arité 2. On notera $\mathcal{T}$ la théorie des groupes sur cette signature. Pour n entier, on s'autorisera la notation x^n pour $((x \times x) \times \ldots) \times x$, avec n occurrences de x.

On rappelle qu'il faut justifier les réponses aux questions suivantes. En particulier, veilliez à bien nommer explicitement les théorèmes utilisés.

Question 3.1. *Rappeler les axiomes de la théorie des groupes.*

Question 3.2. *La formule* $\exists x.x^5 = 1$ *est-elle prouvable dans cette théorie ?*

Question 3.3. *La formule* $\exists x.\neg(x = 1) \land x^5 = 1$ *est-elle prouvable dans cette théorie ?*

Question 3.4. *La formule* $\neg(\exists x.\neg(x = 1) \land x^5 = 1)$ *est-elle prouvable dans cette théorie ?*

Étant donné un entier n, un élément a d'un groupe est dit d'*ordre* n lorsque $a^n = 1$. Un élément a d'un groupe est dit d'*ordre fini* lorsqu'il est d'ordre n pour un certain entier $n > 0$. Un groupe est *sans torsion* lorsqu'il ne contient aucun élément d'ordre fini à l'exception de l'unité.

Question 3.5. *Donner une théorie du premier ordre dont les modèles sont les groupes sans torsion. On notera* $\mathcal{U}$ *cette théorie.*

Question 3.6. *Cette théorie est-elle cohérente ?*

Question 3.7. *Étant donné un entier* n*, proposer une théorie* $\mathcal{U}_n$ *dont les modèles sont les groupes qui contiennent au moins un élément d'ordre* n*, et aucun élément d'ordre* k *pour* $1 < k < n$*.*

On admettra dans la suite que, pour tout entier *premier* n, la théorie $\mathcal{U}_n$ admet le groupe additif $\mathbb{Z}/n\mathbb{Z}$ comme modèle.

Nous cherchons maintenant à montrer qu'on ne peut pas donner de théorie avec un nombre fini d'axiomes dont les modèles sont les groupes sans torsion. Nous allons raisonner par l'absurde et supposer qu'il existe une telle théorie $\mathcal{V}$.

Question 3.8. *Montrer qu'on peut supposer que* $\mathcal{V}$ *est réduite à une formule, c'est-à-dire qu'elle est de la forme* $\mathcal{V} = \{\phi\}$ *pour une certaine formule* ϕ*.*

On considère la théorie $\mathcal{W} = \mathcal{U} \cup \{\neg\phi\}$ (on rappelle que $\mathcal{U}$ est la théorie de la question 3.5 et ϕ est la formule de la question précédente).

Question 3.9. *Montrer que l'existence de la théorie* $\mathcal{V}$ *implique que théorie* $\mathcal{W}$ *n'a pas de modèle.*

Question 3.10. *Montrer qu'il existe un modèle de* $\mathcal{W}$*.*

Question 3.11. *Conclure qu'il n'existe pas de théorie avec un nombre fini d'axiomes dont les modèles sont les groupes sans torsion.*

4 Problème de correspondance de Post et langages non contextuels

Le *problème de correspondance de Post* est le problème de décision suivant :

Donnée : Une liste $(s_1, t_1), \ldots, (s_n, t_n)$ de n paires de mots sur un alphabet Σ.

Réponse : Décider s'il existe une liste $i_1, \ldots, i_k$ d'indices telle que $s_{i_1} s_{i_2} \cdots s_{i_k} = t_{i_1} t_{i_2} \cdots t_{i_k}$.

Par exemple, étant données les paires suivantes de mots sur l'alphabet $\Sigma = \{a, b\}$

$$(aba, ab), (ba, aaba), (b, bab), (baba, ab)$$

la réponse est « oui », car la liste des indices $1, 4, 1, 2, 3$ donne une correspondance :

$$\begin{array}{|c|c|c|c|c|} \hline aba & baba & aba & ba & b \\ ab & ab & ab & aaba & bab \\ \hline \end{array} = \begin{array}{c} abababaababab \\ abababaababab \end{array}$$

Question 4.1. *Montrer que le problème de correspondance de Post est décidable dans le cas d'un alphabet unaire* $|\Sigma| = 1$*.*

Nous admettrons dans la suite que le problème de correspondance de Post est indécidable si $|\Sigma| \geq 2$.

Comme mentionné en PC4, les langages *non contextuels* sont une généralisation des langages rationnels, qui incluent par exemple le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ et le langage des mots bien parenthésés. Un langage non contextuel est généré par une *grammaire non contextuelle* $G = (\Sigma, N, S, P)$ composée :

— d'un ensemble fini Σ dont les éléments sont appelés *terminaux* ;

— d'un ensemble fini N (disjoint de Σ) dont les éléments sont appelés *non-terminaux* ;

— d'un non-terminal distingué $S \in N$ appelé le *symbole initial* ;

— et d'une liste P de *productions* de la forme $R \to \alpha$ où $R \in N$ est un non-terminal et $\alpha \in (\Sigma \cup N)^*$ est un mot sur les terminaux et les non-terminaux.

On définit une relation $\Rightarrow$ entre mots dans $(\Sigma \cup N)^*$ comme suit : $\beta \Rightarrow \beta'$ s'il existe des mots $\gamma, \delta \in (\Sigma \cup N)^*$ et une production $R \to \alpha$ dans P telle que $\beta = \gamma R \delta$ et $\beta' = \gamma \alpha \delta$. Le langage engendré par la grammaire G est l'ensemble des mots

$$L(G) = \{w \mid w \in \Sigma^*, S \Rightarrow^+ w\}$$

où $\Rightarrow^+$ est la fermeture transitive de $\Rightarrow$.

Exemple 1. *Le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ est généré par une grammaire avec un seul non-terminal S et deux productions*

$$S \to aSb \qquad\qquad S \to \epsilon$$

où ϵ est le mot vide.

Exemple 2. *Le langage des mots bien parenthésés est généré par une grammaire avec un non-terminal S et deux productions*

$$S \to (S)S \qquad\qquad S \to \epsilon$$

Le *problème d'intersection des langages non contextuels* est le problème de décision suivant :

Donnée : Deux grammaires non contextuelles $G_1 = (\Sigma, N_1, S_1, P_1)$ et $G_2 = (\Sigma, N_2, S_2, P_2)$ sur le même alphabet de terminaux Σ.

Réponse : Décider s'il existe un mot $w \in \Sigma^*$ telle que $w \in L(G_1) \cap L(G_2)$.

Question 4.2. *Montrer que le problème d'intersection des langages non contextuels est indécidable si on ne met pas de contrainte sur l'alphabet Σ, par réduction à partir du problème de correspondance de Post.*

Question 4.3. *Utiliser le résultat précédent pour conclure que les langages non contextuels ne sont pas clos par intersection de façon calculable, c'est-à-dire qu'il n'existe pas d'algorithme qui, étant données deux grammaires non contextuelles G_1 et G_2, renvoie une grammaire non contextuelle G telle que $L(G) = L(G_1) \cap L(G_2)$. (En fait, il existe des langages non contextuels L_1 et L_2 tels que $L_1 \cap L_2$ n'est pas non contextuel, mais nous ne vous demandons pas de le montrer.)*

Une grammaire est dite *linéaire* si pour chaque production $R \to \alpha$, le membre droit α contient au plus un non-terminal.

Question 4.4. *Montrer que le problème d'intersection est décidable pour les langages non contextuels linéaires sur un alphabet unaire $|\Sigma| = 1$.*