

Année universitaire 2018/19, session 1 d'automne décembre 2018

Mention : master informatique, code UE 4TIN704U

Intitulé de l'épreuve : examen Calculabilité et complexité

Date : 12/12/18      Heure : 9h00      Durée : 3h

Documents autorisés : 1 feuille A4 recto-verso

**Le barème est indicatif. On attachera une grande importance à la clarté et à la concision des justifications.**

### Exercice 1 (6 points) : décidabilité

(A) Indiquez quelles définitions sont justes parmi les suivantes.

1. Un problème  $A$  est *décidable* s'il existe un programme WHILE qui termine sur chacune de ses entrées et retourne la réponse « OUI » sur les instances positives de  $A$ .
2. Un problème  $A$  est *décidable* s'il existe un programme WHILE qui termine sur chacune de ses entrées et retourne la réponse correcte (« OUI » ou « NON ») sur chaque instance de  $A$ .

(B) Justifiez les assertions qui suivent. Vos réponses doivent être claires et concises, et la réponse « vu en cours/TD » ne sera pas acceptée.

3. Si un problème appartient à la classe **NP**, alors il est décidable.
4. Un problème  $A$  est décidable si et seulement si il est semi-décidable et son complémentaire est semi-décidable.

*Rappel* : les instances positives/négatives du complémentaire  $A^c$  de  $A$  sont les instances négatives/positives de  $A$ .

### Exercice 2 (4 points) : réductions

Soient  $A, B$  deux problèmes. On suppose que  $A$  se réduit à  $B$  :  $A \leq B$ . Pour les questions qui suivent, on demande qu'est-ce qu'on peut (ou pas) dire sur la décidabilité/complexité des problèmes en question. Justifiez vos réponses de façon précise.

1. Si  $A$  est indécidable, qu'est-ce qu'on peut déduire sur  $B$  ?
2. Si  $A$  est décidable, qu'est-ce qu'on peut déduire sur  $B$  ?
3. Si  $B = \text{SAT}$ , qu'est-ce qu'on peut déduire sur  $A$  ?
4. Si  $A = \text{SAT}$  et la réduction est polynomiale, qu'est-ce qu'on peut déduire sur  $B$  ?

### Exercice 3 (4 points) : réduction polynomiale vers SAT

Soit un graphe non-orienté  $G = (V, E)$  où  $V = \{1, \dots, n\}$  est l'ensemble des sommets, et  $E$  est l'ensemble des arêtes du graphe. Etant donné un sommet  $v \in V$  et un entier non-négatif  $r$  (« le rayon »), on note par  $B_r(v)$  l'ensemble des sommets qui sont à distance inférieure ou égale à  $r$  du sommet  $v$ . Par exemple,  $B_1(v) = \{v\} \cup \{w : (v, w) \in E\}$  consiste de  $v$  et ses voisins.

On considère le problème suivant :

#### PROXIMITE

INPUT : Un graphe orienté  $G$ , deux entiers  $k$  et  $r$ .

QUESTION : Existe-t-il des sommets  $v_1, \dots, v_k$  tels que  $V = B_r(v_1) \cup \dots \cup B_r(v_k)$ ?

Autrement dit, on cherche à savoir s'il existe  $k$  sommets distingués (non nécessairement distincts) dans le graphe tels que tout sommet est à distance inférieure ou égale à  $r$  d'un sommet distingué.

1. Justifiez que le problème PROXIMITE appartient à la classe **NP**, en appliquant la définition vue en cours. Justifiez votre réponse.
2. **Pour cette question uniquement** on suppose que  $k = 42$ , donc l'entrée du problème PROXIMITE est seulement le graphe  $G$  et le rayon  $r$ . Décrivez un algorithme polynomial pour le problème PROXIMITE, en justifiant son temps de calcul.

Le but des questions suivantes est de construire une réduction polynomiale de PROXIMITE vers CNF-SAT. A partir de  $G, k, r$  il faudra construire une formule en CNF  $\varphi_{G,k,r}$  telle que  $G, k, r$  est instance positive de PROXIMITE si et seulement si  $\varphi_{G,k,r}$  est satisfaisable.

3. **Pour cette question on suppose que  $r = 1$ .**

Les variables de  $\varphi_{G,k,1}$  seront de la forme  $x(v, i)$ , où  $v \in V$  et  $1 \leq i \leq k$ . La variable  $x(v, i)$  sera vraie si et seulement si  $v$  est le  $i$ -ème sommet distingué.

- (a) Ecrivez une formule  $\varphi_1$  en CNF qui garantit que pour tout  $1 \leq i \leq k$  il existe un unique  $v \in V$  t.q.  $x(v, i)$  est vrai.
  - (b) Ecrivez une formule  $\varphi_2$  en CNF qui garantit que tout sommet de  $V$  est à distance  $\leq 1$  d'un des sommets distingués.
  - (c) Décrivez  $\varphi_{G,k,1}$  et justifiez que vous avez une réduction polynomiale de PROXIMITE vers CNF-SAT.
4. Comment faut-il modifier  $\varphi_2$  pour obtenir  $\varphi_{G,k,r}$  pour  $r > 1$  quelconque ? Justifiez votre réponse.

### Exercice 4 (6 points) : programmes WHILE

Dans cet exercice : programme = programme WHILE.

Rappels :

- A. Les variables  $x, y, \dots$  d'un programme prennent seulement des valeurs dans  $\mathbb{N} = \{0, 1, \dots\}$ .

- B. Les seules instructions que vous pouvez utiliser sont **les instructions élémentaires**  $x:=y+c$ ,  $x:=c$ ,  $x:=y-c$ , **skip**, ainsi que les boucles **WHILE** :
- $x:=y+c$  ou  $x:=c$  : où  $c \in \mathbb{N}$  est une constante,  $x$  et  $y$  sont des variables (potentiellement la même),
  - $x:=y-c$  : avec les mêmes conditions (attention, ceci calcule la différence tronquée, c'est à dire  $\max(y - c, 0)$ ),
  - skip** : ne fait rien,
  - WHILE** ( $x > 0$ ) **DO**  $P$  **OD** : qui exécute le programme **WHILE**  $P$  tant que  $x$  est non-nul.
- C. Chaque programme a une variable appelée **res** dans laquelle se trouve à la fin du programme le résultat du calcul (s'il termine). La valeur initiale de **res** (et de toute variable autre que les variables contenant les entrées du programme) est 0.

**Vous n'avez pas le droit d'utiliser d'autres instructions que celles listées ci-dessus, sauf si c'est précisé autrement dans les questions qui suivent.**

1. Soit  $P, Q$  deux programmes. Complétez le programme qui suit, pour qu'il soit équivalent à l'instruction **IF** ( $x > 0$ ) **THEN**  $P$  **ELSE**  $Q$  **FI** ( $y, z$  sont des variables qui n'apparaissent pas dans  $P, Q$ ) :

```

y ← 1 - x
z ← 1 - y
while ( $y > 0$ ) do ...
end while
while (...) do ...
end while

```

2. Soit  $P$  un programme qui prend un entier comme entrée. Pour cette question vous devez construire à partir de  $P$  un programme  $R$  qui calcule la fonction  $R : \mathbb{N} \rightarrow \mathbb{N}$  suivante :

$$R(n) = \begin{cases} 1 & \text{si le nombre d'instructions élémentaires que } P \text{ exécute} \\ & \text{sur entrée } 0 \text{ est } \leq n \\ 0 & \text{sinon} \end{cases}$$

**Le programme  $R$  peut utiliser l'instruction if-then-else en dehors des instructions élémentaires et des boucles WHILE.**

- (a) Expliquez de manière informelle comment modifier  $P$  pour obtenir  $R$  conforme à la description.
  - (b) Précisez quelles sont les variables additionnelles de  $R$ , et comment vous modifiez chaque instruction de  $P$ , afin d'obtenir  $R$ .
3. Montrez une réduction de  $\text{HALT}_0$  (arrêt d'un programme sur entrée 0) au problème : **INFINI**  
**INPUT** : Programme  $R$ .  
**QUESTION** : Est-ce que l'ensemble des entrées de  $R$  sur lesquelles  $R$  s'arrête et retourne 1, est infini?
  4. Que déduisez-vous pour la décidabilité du problème **INFINI** ? Justifiez.
  5. (bonus) Donnez une justification alternative pour la question précédente.