

**Toute réponse non justifiée n'apportera pas de point.**  
Le barème est indicatif.

**Exercice 1 : Questions de cours**

**1.5 Points**

Répondre aux questions suivantes en **justifiant brièvement** (une réponse non justifiée vaut 0 point).

- 1) Soit le langage  $\text{PREM} = \{\langle M \rangle \mid M \text{ a un nombre premier d'états}\}$ . Ce langage est-il décidable ?
- 2) Soit le langage  $\text{FACT} = \{\langle M, w \rangle \mid M \text{ écrit les facteurs premiers de l'entier codé par } w \text{ puis s'arrête}\}$ . Ce langage est-il décidable ?
- 3) Une machine de Turing est *exponentiellement bornée* si pour toute entrée  $w$ , la machine  $M$  utilise un espace borné par  $2^{|w|}$ . Le problème suivant est-il décidable ?

**Entrée** : Une machine  $M$  exponentiellement bornée et un mot  $w$ .

**Question** :  $M$  s'arrête-elle sur  $w$  ?

**Exercice 2 : Décidabilité et indécidabilité**

**4 Points**

On se place sur l'alphabet  $\Sigma = \{0, 1\}$  qui permet de coder à la fois les machines de Turing et leurs entrées. Étant donnée une telle machine  $M$ , on note  $\langle M \rangle \in \Sigma^*$  le codage de  $M$  du cours et  $\langle M, w \rangle$  le codage de  $M$  suivi par le mot  $w$ . Si  $X$  est un langage, on note  $\bar{X}$  son complémentaire, c'est-à-dire  $\bar{X} = \Sigma^* \setminus X$ . Enfin, on rappelle qu'un langage  $X \subseteq \Sigma^*$  est appelé *co-semi-décidable* quand  $\bar{X}$  est semi-décidable. On considère un langage  $L \subseteq \Sigma^*$ . Soit  $K \subseteq \Sigma^*$  le langage suivant :

$$K = \{\langle M \rangle \mid L(M) = L\}.$$

Répondre aux questions suivantes, en **justifiant** (une réponse mal justifiée ne rapporte pas de point).

- 1) On suppose que  $L$  n'est pas semi-décidable. Le langage  $K$  est-il décidable ?
- 2) On suppose que  $L = \emptyset$ . Le langage  $K$  est-il co-semi-décidable ? Est-il semi-décidable ?

On rappelle que le langage  $\text{UNIV} \subseteq \Sigma^*$  est défini de la façon suivante :

$$\text{UNIV} = \{\langle M, w \rangle \mid w \in L(M)\}.$$

Dans les questions 3, 4 et 5 suivantes, on suppose que  $L$  est semi-décidable.

- 3) Si  $L \neq \emptyset$  et  $L$  est semi-décidable, montrer que  $\text{UNIV}$  se réduit à  $K$ . Que peut-on en déduire sur  $K$  ? Pour les questions 4 et 5, vous pouvez vous inspirer de la preuve d'indécidabilité de l'arrêt universel.
- 4) Si  $L$  est semi-décidable et infini, montrer que  $\overline{\text{UNIV}}$  se réduit à  $K$ . Que peut-on en déduire sur  $K$  ?
- 5) On suppose que  $L \neq \emptyset$  et que  $L$  est fini. Montrer à nouveau que  $\overline{\text{UNIV}}$  se réduit à  $K$ . Que peut-on en déduire sur  $K$  ?

**Exercice 3 : 2-Sat et accessibilité****3.5 Points**

Étant donné un ensemble de variables propositionnelles, on rappelle qu'un *littéral* est soit une variable, soit la négation d'une variable. Une *2-clause* est de la forme  $\ell \vee \ell'$ , où  $\ell$  et  $\ell'$  sont des littéraux. Enfin, une formule propositionnelle est en forme *2-CNF* si elle est conjonction de 2-clauses.

On s'intéresse au problème 2-SAT suivant :

**Entrée** : Une formule 2-CNF  $\varphi$ .

**Question** : La formule  $\varphi$  est-elle satisfaisable ?

À partir d'une formule 2-CNF  $\varphi$ , on construit un graphe orienté  $G(\varphi)$  de la façon suivante :

- Les sommets de  $G(\varphi)$  sont les variables  $\{x_1, \dots, x_n\}$  de  $\varphi$  ainsi que leurs négations  $\{\neg x_1, \dots, \neg x_n\}$ .
- Les arêtes de  $G(\varphi)$  sont de la forme  $\neg \ell \rightarrow \ell'$  et  $\neg \ell' \rightarrow \ell$  pour chaque 2-clause  $\ell \vee \ell'$  de  $\varphi$ .

1) La formule  $\varphi_0 = (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_1 \vee \neg x_3)$  est-elle satisfaisable ? Construire le graphe  $G(\varphi_0)$ .

Pour deux sommets  $\ell, \ell'$  de  $G(\varphi)$ , on note  $\ell \xrightarrow{*} \ell'$  si  $\ell'$  est accessible depuis  $\ell$  dans  $G(\varphi)$ .

2) Soit  $\varphi$  une formule 2-CNF. Montrer que  $\varphi$  est satisfaisable si et seulement si dans  $G(\varphi)$ , il n'existe aucun sommet  $x$  tel que  $x \xrightarrow{*} \neg x$  et  $\neg x \xrightarrow{*} x$ .

3) Que peut-on dire de la complexité du problème 2-SAT ? Justifier la réponse.

On considère maintenant le problème DIGRAPHE-INACCESSIBILITÉ suivant :

**Entrée** : Un graphe orienté  $G$  et deux sommets  $s, t$  de  $G$ .

**Question** : Dans  $G$ , le sommet  $t$  est-il inaccessible depuis le sommet  $s$  ?

4) Montrer que DIGRAPHE-INACCESSIBILITÉ se réduit à 2-SAT et que la réduction ne nécessite qu'un espace logarithmique.

**Exercice 4 : P vs. NP-complet****4 Points**

Dans cet exercice, on suppose que  $\mathbf{P} \neq \mathbf{NP}$ . Pour chacun des problèmes suivants, dites si il est **NP-complet** ou dans **P**. Attention, les réponses doivent être **prouvées**. Si vous répondez qu'un problème est **NP-complet** vous devez prouver qu'il est dans **NP** et qu'il est **NP-difficile**. Si vous répondez qu'un problème est dans **P** vous devez donner un algorithme polynomial qui le résout.

*Remarque : les réductions sont faciles, la principale difficulté est de trouver le bon problème à réduire.*

1) CHEMIN PRESQUE HAMILTONIEN

**Entrée** : Un graphe orienté  $G$  à  $n \geq 43$  sommets.

**Question** : Le graphe  $G$  a-t-il un chemin qui visite  $(n - 42)$  sommets distincts en passant une seule fois par chacun de ces  $(n - 42)$  sommets ?

2) PRESQUE CLIQUE

**Entrée** : Un graphe orienté  $G$  à  $n \geq 43$  sommets.

**Question** : Le graphe  $G$  a-t-il comme sous-graphe une clique à  $(n - 42)$  sommets ?

3) DOUBLE SAT

**Entrée** : Une formule 3-CNF  $\varphi$ .

**Question** : Y a-t-il **au moins deux** affectations des variables qui rendent  $\varphi$  vraie ?

4) AFFECTATION FAUSSE

**Entrée** : Une formule 3-CNF  $\varphi$ .

**Question** : Y a-t-il une affectation des variables rendant la formule  $\varphi$  **fausse** ?

Étant donné un alphabet  $A$  et deux mots  $u, v \in A^*$  de même longueur  $n$ , on définit la distance  $d(u, v) \in \mathbb{N}$  comme suit. Soit  $a_1, \dots, a_n, b_1, \dots, b_n \in A$  les lettres telles que  $u = a_1 \cdots a_n$  et  $v = b_1 \cdots b_n$ . On définit  $d(u, v)$  comme le nombre d'indices  $i \leq n$  tels que  $a_i \neq b_i$ . Par exemple, si  $A = \{0, 1, \square\}$  :

$$d(011\square 0, 110\square\square) = 3$$

$$d(000111, 000110) = 1$$

$$d(0\square 0, 1\square\square) = 2$$

On s'intéresse au problème MOT LOINTAIN suivant :

**Entrée** : Un alphabet  $A$ , des entiers  $k, m, n \in \mathbb{N}$  et  $m$  mots  $u_1, \dots, u_m \in A^*$  de longueur  $n$ .

**Question** : Existe-t-il un mot  $v \in A^*$  tel que  $d(u_i, v) \geq k$  pour tout  $i \leq m$  ?

1) Montrer que MOT LOINTAIN est dans **NP**.

On veut maintenant montrer que le problème est **NP**-complet. On cherche à réduire 3-SAT. Soit donc  $\varphi$  une entrée de 3-SAT. On note  $\{C_1, \dots, C_m\}$  l'ensemble de clauses et  $\{x_1, \dots, x_n\}$  les variables de  $\varphi$ .

On utilise l'alphabet  $A = \{0, 1, \square\}$  et on construit  $m + 9$  mots  $u_1, \dots, u_{m+9}$ , chacun de longueur  $n + 2$  :

- Pour  $i \leq m$ , le mot  $u_i$  de longueur  $n + 2$  est obtenu à partir de la clause  $C_i$  : ses  $n$  premières lettres sont déterminées par  $C_i$  et les deux dernières sont égales à " $\square$ ". Plus précisément, on a  $u_i = a_1 \cdots a_n \square \square$  tel que pour tout  $j \leq n$ ,

$$a_j = \begin{cases} 0 & \text{si la variable } x_j \text{ apparaît positivement dans } C_i \\ 1 & \text{si la variable } x_j \text{ apparaît négativement dans } C_i \\ \square & \text{si la variable } x_j \text{ n'apparaît pas dans } C_i \end{cases}$$

Par exemple si  $C_i = x_2 \vee \neg x_5 \vee x_3$ , alors,

$$u_i = \square 0 0 \square 1 \square \square \square \cdots \square$$

- Les mots  $u_{m+1}, \dots, u_{m+9}$  sont les neuf mots de longueur  $n + 2$  dont les  $n$  premières lettres sont égales à " $\square$ " et les 2 dernières parcourent les 9 combinaisons de 2 lettres de  $\{0, 1, \square\}$ .

2) Montrer que  $\{C_1, \dots, C_m\}$  est satisfaisable si et seulement si il existe un mot  $v \in A^*$  de longueur  $n + 2$  tel que  $d(v, u_i) \geq n$  pour tout  $i \leq m + 9$ . Que peut-on en conclure pour MOT LOINTAIN ?

On s'intéresse maintenant au problème MOT PROCHE suivant :

**Entrée** : Un alphabet  $A$ , des entiers  $k, m, n \in \mathbb{N}$  et  $m$  mots  $u_1, \dots, u_m \in A^*$  de longueur  $n$ .

**Question** : Existe-t-il un mot  $v \in A^*$  tel que  $d(u_i, v) \leq k$  pour tout  $i \leq m$  ?

3) Pour cette question, on admettra que MOT LOINTAIN reste **NP**-complet quand on restreint ses entrées possibles aux alphabets à deux lettres. En utilisant cette hypothèse, montrer que MOT PROCHE est également **NP**-complet.

Exercice 6 sur la page suivante

Dans cet exercice, on considère des langages **finis** représentés par des expressions (+, •). Les expressions (+, •) sur un alphabet  $A$  sont définies récursivement :

- Pour tout  $a \in A$ , l'expression  $a$  représente le langage  $\{a\}$ .
- Si  $E_1$  représente le langage  $L_1$  et  $E_2$  représente le langage  $L_2$ , alors,
  - l'expression  $E_1 + E_2$  représente le langage  $L_1 \cup L_2$ .
  - l'expression  $E_1 \bullet E_2$  représente le langage  $L_1 L_2$  défini par

$$L_1 L_2 = \{u_1 u_2 \mid u_1 \in L_1 \text{ et } u_2 \in L_2\}.$$

1) Quel est le langage représenté par l'expression  $(a + b + c) \bullet (d + e + f)$  ?

On appelle *alphabet d'un mot  $u$*  l'ensemble des lettres apparaissant dans le mot  $u$ , c'est-à-dire, le plus petit sous-alphabet  $B$  de l'alphabet  $A$  tel que  $u \in B^*$ . Par exemple, l'alphabet du mot  $baadb$  est  $\{a, b, d\}$ , et l'alphabet du mot  $abbabbbb$  est  $\{a, b\}$ .

On s'intéresse au problème ALPHABET COMMUN suivant :

**Entrée** : Un alphabet  $A$  fini et deux expressions (+, •)  $E_1$  et  $E_2$  sur  $A$  représentant des langages  $L_1$  et  $L_2$  de  $A^*$ .

**Question** : Existe-t-il des mots  $u_1 \in L_1$  et  $u_2 \in L_2$  qui ont le même alphabet ?

- 2) Montrer que le problème ALPHABET COMMUN est dans **NP**.
- 3) Montrer que le problème ALPHABET COMMUN est **NP**-complet.



#### Indication

Faire une réduction à partir du problème 3-SAT, l'une des expressions codant les clauses de la formule, et l'autre codant les assignations des variables.

— FIN —