université de BORDEAUX	**Année Universitaire 2023–2024, Master CSI** DS, Théorie de la complexité (UE 4TCY701U) 15 novembre 2023, 9h30–12h30	**Collège Sciences et Technologies**

La rédaction doit être précise et concise. Le barème est indicatif.
Seules, les réponses correctement justifiées apporteront des points.

Exercice 1 – Applications du cours (2 points)

Répondez aux questions suivantes en quelques lignes, en justifiant clairement vos réponses.

1. Est-il vrai que l'union de deux langages indécidables est toujours indécidable ?
2. Soit Σ un alphabet fini et $L \subseteq \Sigma^*$ un langage semi-décidable. On considère le langage K formé des préfixes des mots de L, c'est-à-dire le langage suivant :

$$K = \{w \in \Sigma^* \mid \exists v \in \Sigma^* \;\; wv \in L\}.$$

   Ce langage est-il semi-décidable ?

Exercice 2 – Complexité (4 points)

Pour chacun de ces problèmes, démontrer soit qu'il est **NP**-complet, soit co-**NP**-complet (c'est-à-dire que son complémentaire est **NP**-complet), soit qu'il est dans **P**.

Remarques

- Les réductions sont *simples* en choisissant le bon problème de départ.
- Pour les réductions, vous pouvez utiliser *tout* problème **NP**-complet vu en cours ou TD.
- Pour montrer qu'un problème est **NP**-complet, pensez à vérifier qu'il est dans **NP**.

On rappelle qu'en logique propositionnelle, un littéral est une variable ou une négation de variable. Une formule est en forme normale disjonctive (DNF) si c'est une disjonction de conjonctions de littéraux. Par exemple, la formule $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_2)$ est en forme DNF.

1. **Tautologie pour formules DNF**
   Entrée : Une formule propositionnelle en forme DNF.
   Question : La formule est-elle vraie pour **toute** affectation des variables ?

2. **Couverture d'ensembles**
   Entrée : Un ensemble fini d'entiers E, des sous-ensembles $E_1, \ldots, E_k$ de E et un entier ℓ (en binaire).
   Question : Existe-t-il ℓ entiers tels que chaque E_i contienne au moins l'un de ces entiers ?

3. **Clique pour graphes de degré maximal 42**
   Entrée : Un graphe fini non-orienté G dont chaque sommet a au plus 42 voisins et un entier k (en binaire).
   Question : Le graphe G a-t-il une clique de taille k ?

4. **Circuit demi-Hamiltonnien**
   Entrée : Un graphe fini non-orienté G à $2n$ sommets.
   Question : Existe-t-il un circuit (c'est-à-dire un chemin dont le sommet de départ est celui d'arrivée) qui passe exactement 1 fois par n sommets de G et exactement 2 fois par chacun des n autres sommets ?

Problème de correspondance de Post

Dans les exercices suivants, on appelle **tuile** un couple de mots non vides sur un alphabet fini Σ. Une tuile est donc de la forme $(v, w) \in \Sigma^*$, où $v \neq \varepsilon$ et $w \neq \varepsilon$. On va considérer des problèmes dont les entrées sont des suites finies de tuiles. Une entrée est donc de la forme suivante, où les v_i et w_i sont des mots non vides :

$$(v_1, w_1),\ (v_2, w_2),\ \ldots,\ (v_k, w_k).$$

Le nombre de tuiles est un entier $k \geqslant 1$. Une suite finie d'**indices** $i_1, i_2, \ldots, i_n$ (avec $1 \leqslant i_j \leqslant k$ pour tout j) est appelée une **solution** pour la suite $(v_1, w_1),\ (v_2, w_2), \ldots,\ (v_k, w_k)$ lorsque $n \geqslant 1$ et, de plus :

$$v_{i_1} v_{i_2} \cdots v_{i_n} = w_{i_1} w_{i_2} \cdots w_{i_n}.$$

Par exemple, l'entrée $(abb, b),\ (aa, aab),\ (bb, ba)$ admet la solution $2, 3, 2, 1$ de longueur $n = 4$. En effet,

$$\begin{array}{lcll} v_2 v_3 v_2 v_1 & = & (aa)(bb)(aa)(abb) & = aabbaaabb, \\ w_2 w_3 w_2 w_1 & = & (aab)(ba)(aab)(b) & = aabbaaabb. \end{array}$$

Notez que la longueur n d'une solution n'est pas bornée a priori, et qu'un même indice peut donc être présent plusieurs fois dans une solution (c'est le cas de l'indice 2 dans l'exemple ci-dessus).

Les exercices suivants étudient des variations et applications du problème « PCP » décrit ci-dessous. Ils sont indépendants les uns des autres : vous pouvez faire un exercice en admettant les questions des exercices précédents. Il est cependant conseillé de traiter en premier l'exercice 3, qui est simple, pour bien comprendre le problème.

Problème de correspondance de Post

Le problème de correspondance de Post (**PCP** en abrégé) est le suivant.

Entrée : Une suite finie de tuiles $(v_1, w_1),\ (v_2, w_2), \ldots,\ (v_k, w_k)$ sur un alphabet fini Σ.

Question : Existe-t-il une solution $i_1, i_2, \ldots, i_n$ pour la suite $(v_1, w_1),\ (v_2, w_2), \ldots,\ (v_k, w_k)$?

Exercice 3 – Compréhension du problème (1,5 points)

1. Pour chacune des 3 entrées suivantes du PCP, donnez une solution ou justifiez qu'il n'y en a pas.

i	1	2	3
v_i	b	aab	ba
w_i	a	ba	aba

i	1	2	3
v_i	ab	aab	ab
w_i	a	ab	ab

i	1	2	3
v_i	aab	$cdabdc$	d
w_i	aa	a	$bdcd$

2. Dans cette question seulement, on suppose que l'alphabet Σ sur lequel sont écrits les mots de l'entrée est $\Sigma = \{a\}$. Donner un algorithme pour tester si le PCP a une solution sur une telle entrée.
3. Dans cette question seulement, on suppose que l'entrée vérifie $|v_i| \leqslant |w_i|$ pour chaque $i \leqslant k$. Donner un algorithme pour tester si le PCP a une solution sur une telle entrée.

Exercice 4 – Une variation : le problème d'inclusion de Post (1,5 points)

Soit v, w deux mots. On écrit $v \subseteq w$ quand v peut s'obtenir à partir de w en effaçant des lettres (par exemple $abca \subseteq \underline{a}\,\underline{b}ab\underline{c}d\underline{a}$). On considère le problème d'inclusion de Post qui est une variante du PCP :

Entrée : Une suite finie de tuiles $(v_1, w_1),\ (v_2, w_2), \ldots,\ (v_k, w_k)$ sur un alphabet fini Σ.

Question : Existe-t-il une suite finie non vide d'indices $i_1, i_2, \ldots, i_n$ tels que $v_{i_1} \cdots v_{i_n} \subseteq w_{i_1} \cdots w_{i_n}$?

1. Soient v, v', w et w' des mots, montrez que si $v\,v' \subseteq w\,w'$, alors $v \subseteq w$ ou $v' \subseteq w'$.
2. Montrez que si le problème d'inclusion de Post a une solution, alors il a une solution de longueur 1.
3. Que peut-on en déduire sur le problème d'inclusion de Post ?

Exercice 5 – Indécidabilité du PCP (8 points)

1. Le PCP est-il semi-décidable ? Justifiez votre réponse.

On veut montrer que le PCP est **indécidable**. Pour cela, on définit la variante suivante de ce problème.

Problème de correspondance de Post modifié

Le problème de correspondance de Post modifié (**PCPM** en abrégé) est le suivant.

Entrée : Une suite finie de tuiles $(v_1, w_1), (v_2, w_2), \ldots, (v_k, w_k)$ sur un alphabet fini Σ.

Question : Existe-t-il une solution $i_1, i_2, \ldots, i_n$ pour la suite $(v_1, w_1), (v_2, w_2), \ldots, (v_k, w_k)$ **avec $i_1 = 1$** ?

2. Quelles sont les entrées données en question 1 de l'exercice 3 qui sont solutions du PCPM (justifiez) ?
3. Que pouvez-vous dire à propos du PCP si l'entrée contient une tuile de la forme (u, u) ? La même conclusion est-elle valable pour le PCPM ? Justifiez la réponse à cette dernière question.
4. Montrer que le PCPM se réduit au PCP.

Indication pour la réduction

Pour forcer une solution à commencer par l'indice 1, faites en sorte qu'il soit impossible de commencer par les autres. Pour cela, ajoutez une nouvelle lettre #, et utilisez des mots d'une ou plusieurs des formes suivantes :

- $u^{\#}$, qui est le mot de $(\Sigma \cup \{\#\})^*$ obtenu à partir du mot u de Σ^* en insérant un « # » entre chaque paire de lettres consécutives de u. Par exemple, $(a\,b\,c)^{\#} = a\#b\#c$.
- $\#u^{\#}$, $u^{\#}\#$ et $\#u^{\#}\#$. Par exemple, $\#(a\,b\,c)^{\#} = \#a\#b\#c$.

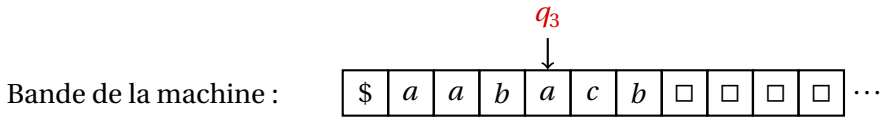
On va maintenant réduire le problème de l'arrêt d'une machine de Turing déterministe $\mathcal{M}$ sur le mot vide au PCPM.

5. Écrivez l'objectif à réaliser, c'est-à-dire (1) ce qu'une telle réduction doit prendre en entrée, (2) ce qu'elle doit construire et (3) quelles sont les propriétés que cette réduction doit satisfaire.

On rappelle le codage vu en cours des configurations d'une machine de Turing à une bande par des mots. On suppose que $Q \cap \Gamma = \emptyset$ (où Q est l'ensemble d'états de la machine et Γ est son alphabet de travail). On code par le mot $u\,q\,v \in (Q \cup \Gamma)^*$ la configuration où :

- la machine est dans l'état q,
- le mot $u\,v \in \Gamma^*$ est écrit sur sa bande (avec des blancs $\square$ dans toutes les cases à droite de v),
- la tête de lecture pointe sur la première lettre qui suit u.

Par exemple, la configuration suivante est codée par le mot $\$a\,a\,b\,q_3\,a\,c\,b$.



On note $C_0 \vdash C_1 \vdash C_2 \vdash \cdots$ le calcul de la machine $\mathcal{M}$ sur ε (ce calcul peut être fini ou non). Ici, C_i est la configuration obtenue après i pas de calcul de $\mathcal{M}$. Une idée de réduction est de faire en sorte que les tuiles construites forcent des solutions à « passer » par des couples de mots de la forme suivante, où la nouvelle lettre # n'appartient pas à $Q \cup \Gamma$:

$$\#C_0\#C_1\#C_2\#\ldots\#C_\ell\#$$
$$\#C_0\#C_1\#C_2\#\ldots\#C_\ell\#C_{\ell+1}\#$$

Autrement dit, si la configuration $C_{\ell+1}$ du calcul de $\mathcal{M}$ sur ε existe, il doit exister une solution et un entier r tel que la concaténation des mots w_i des tuiles des r premiers indices de la solution est le mot $\#C_0\#C_1\#C_2\#\ldots\#C_\ell\#C_{\ell+1}\#$ et la concaténation des mots v_i des tuiles des r premiers indices de la solution est le mot $\#C_0\#C_1\#C_2\#\ldots\#C_\ell\#$. La concaténation des w_i est ainsi « en avance » d'une configuration sur la concaténation des v_i.

6. Proposer la première tuile du PCPM (celle d'indice 1).
7. Pourquoi n'est-il pas possible de compléter la construction avec les tuiles $(C_0\#, C_1\#)$, $(C_1\#, C_2\#)$, $(C_2\#, C_3\#)$, etc. ?
8. En combien de positions, au plus, deux codages de configurations C et C' telles que $C \vdash C'$ peuvent-ils différer ?
9. Proposez des tuiles dépendant des transitions de $\mathscr{M}$ pour poursuivre la réduction avec l'objectif ci-dessus. Expliquez l'utilité de chaque tuile proposée (indication : les mots des tuiles doivent avoir longueur 3 au maximum).
10. Pour compléter la réduction, on veut permettre à la concaténation des mots v_i de « rattraper » le retard qu'elle a sur la concaténation des mots w_i si (et seulement si) une configuration produite $C_{\ell+1}$ contient un état acceptant. Proposer des tuiles (qui ne dépendent pas des transitions de $\mathscr{M}$) pour réaliser cet objectif, et conclure.
11. Montrer que le PCP reste indécidable si l'alphabet sur lequel sont écrits les mots d'entrée est $\Sigma = \{0,1\}$.

Exercice 6 – Problème d'intersection vide pour les langages hors-contexte (3 points)

1. Montrer que le problème suivant est indécidable :

   Entrée : Deux grammaires hors-contexte G, H.

   Question : Les langages des mots engendrés par G et H sont-ils disjoints, c'est-à-dire, a-t-on $\mathscr{L}(G) \cap \mathscr{L}(H) = \emptyset$?
2. Ce problème est-il semi-décidable ? Justifiez la réponse.

Exercice 7 – Machines à deux files (2 points)

Une machine à deux files $\mathscr{M}$ est donnée par un quadruplet (Σ, Q, q_0, δ) où Σ est un alphabet fini, Q est un ensemble fini d'états, $q_0 \in Q$ est l'état de départ, et δ est l'ensemble des transitions. Les transitions permettent à la machine de changer le contenu de deux files F_1 et F_2 qu'elle manipule. Ces transitions sont de deux sortes :

- $(q, \text{enfiler}(a,i), q')$ où $q \in Q$, $q' \in Q$, $a \in \Sigma$ et $i \in \{1,2\}$. Intuitivement, l'instruction enfile la lettre a dans F_i.
- $(q, \text{défiler}(a,i), q')$ où $q \in Q$, $q' \in Q$, $a \in \Sigma$ et $i \in \{1,2\}$. Intuitivement, l'instruction défile la lettre a de F_i.

Une *configuration de la machine* $\mathscr{M}$ est un triplet $(q, u_1, u_2) \in Q \times \Sigma^* \times \Sigma^*$. Ici, q est l'état courant, et u_i le contenu actuel de la file F_i (pour $i = 1,2$). On définit une relation $\vdash$ entre configurations : $C \vdash C'$ signifie qu'on passe de C à C' en appliquant une transition. On a $C \vdash C'$ si l'une des quatre situations suivantes s'applique :

- il existe une transition $(q, \text{enfiler}(a,1), q') \in \delta$ et on a $C = (q, u_1, u_2)$ et $C' = (q', a\,u_1, u_2)$,
- il existe une transition $(q, \text{enfiler}(a,2), q') \in \delta$ et on a $C = (q, u_1, u_2)$ et $C' = (q', u_1, a\,u_2)$,
- il existe une transition $(q, \text{défiler}(a,1), q') \in \delta$ et on a $C = (q, u_1 a, u_2)$ et $C' = (q', u_1, u_2)$,
- il existe une transition $(q, \text{défiler}(a,2), q') \in \delta$ et on a $C = (q, u_1, u_2 a)$ et $C' = (q', u_1, u_2)$.

En utilisant une réduction à partir du PCP ou du PCPM, montrer que le problème suivant est indécidable :

Entrée : Une machine à deux files $\mathscr{M} = (\Sigma, Q, q_0, \delta)$ et un mot $w \in \Sigma^*$.

Question : Existe-t-il une suite de configurations de $\mathscr{M}$ de la forme $C_0 \vdash C_1 \vdash \cdots \vdash C_n$ avec $C_0 = (q_0, w, \varepsilon)$ et $C_n = (q, \varepsilon, \varepsilon)$ (où $q \in Q$ est quelconque) ? Autrement dit, y a-t-il une exécution de la machine $\mathscr{M}$ qui,

- depuis la configuration partant de l'état initial et du mot d'entrée w sur la file 1 (la file 2 étant vide),
- arrive à une configuration où les deux files sont vides ?