

**Seules, les réponses correctement justifiées apporteront des points.
Le barème est indicatif.**

Exercice 1 : Union de langages (3 points)

Pour chacune des affirmations suivantes, dites si elle est vraie ou fausse. Si vous répondez « vrai », donnez une justification. Si vous répondez « faux », donnez un contre-exemple.

1. L'union de deux langages semi-décidables est toujours un langage semi-décidable.
2. L'union de deux langages indécidables est toujours un langage indécidable.
3. L'union de deux langages de la classe **NP** est toujours un langage de la classe **NP**.

On veut maintenant répondre à la même question pour des langages **NP**-complets. On se place sur un alphabet $\Sigma = \{a, b\}$.

4. Que peut-on dire d'un langage de Σ^* qui se réduit à \emptyset ? À Σ^* ? Justifiez.
5. Soit $L \subseteq \Sigma^*$ un langage **NP**-complet. Montrer que $aL \cup b\Sigma^*$ est aussi **NP**-complet.
6. L'union de deux langages **NP**-complets est-il toujours un langage **NP**-complet? Justifier.

Exercice 2 : Satisfaisabilité (5 points)

Dans cet exercice, on s'intéresse à des variations du problème SAT. On rappelle qu'un *littéral* est une variable ou une négation de variable, qu'une *clause* est une disjonction de littéraux, qu'une formule *en forme CNF* est une conjonction de clauses, et enfin, qu'une formule est (*en forme*) *k*-CNF si elle est en forme CNF et chaque clause comporte au plus *k* littéraux.

1. Donnez la plus petite classe de problèmes que vous connaissez auquel appartient le problème suivant, en justifiant.

Entrée : Une formule 3-CNF φ .

Question : Existe-t-il une affectation des variables de φ qui rend φ fausse?

Pour tout entier $k \geq 1$, on considère maintenant le problème $3SAT_{\leq k}$ suivant.

Entrée : Une formule 3-CNF dans laquelle chaque variable apparaît au plus *k* fois.

Question : Existe-t-il une affectation des variables de φ qui rend φ vraie?

On s'intéresse d'abord au cas $k = 3$.

2. Justifier que le problème $3SAT_{\leq 3}$ est dans **NP**.
3. On veut montrer que ce problème est **NP**-complet.
 - (a) Rédiger ce qu'il faut faire pour cela.
 - (b) Montrer que $3SAT_{\leq 3}$ est effectivement **NP**-complet.



Indication



Après avoir rédigé clairement l'objectif en question 3a, vous pouvez considérer les occurrences d'une variable qui apparaît 4 fois ou plus, et vous inspirer de la réduction de SAT vers $3SAT$ vue en cours.

- Soit k un entier supérieur ou égal à 3. Que peut-on dire du problème $3SAT_{\leq k}$? Justifier.
- Montrer que le problème SAT où chaque variable apparaît au plus deux fois est **NP**-complet.



Indication



Vous pouvez distinguer plusieurs cas, selon qu'une variable apparaissant trois fois dans une formule apparaît toujours sous la forme du même littéral, ou non. Par symétrie, vous pouvez ne traiter que deux cas : celui où une telle variable apparaît trois fois positivement, et celui où elle apparaît une fois positivement et deux fois négativement.

Pour $k \geq 1$, on considère maintenant le problème NAE- $kSAT$ suivant :

Entrée : Une formule propositionnelle φ en forme k -CNE.

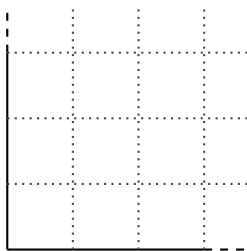
Question : Existe-t-il une affectation des variables de φ pour laquelle chaque clause de φ comporte au moins un littéral vrai, et au moins un littéral faux?

- Montrer que NAE-4SAT est **NP**-complet, en utilisant le fait que 3SAT est **NP**-complet.
- En déduire que NAE-3SAT est aussi **NP**-complet.

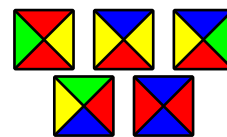
Exercice 3 : Pavage (6 points)

On dispose d'une grille infinie couvrant le quart du plan. Chaque « carré » de la grille est une case dans laquelle on peut placer un domino de Wang. Un tel domino est un carré dont chaque côté (Sud, Ouest, Nord et Est) est coloré (voir l'exemple sur la figure ci-dessous).

Attention, les dominos sont orientés : on ne peut pas les tourner.



La grille infinie



Un ensemble de 5 dominos de Wang

Considérons une paire (D, d_0) où D est un ensemble fini de dominos et $d_0 \in D$ est un domino spécial, appelé *domino de départ*. On dit qu'on **peut paver le quart de plan avec** (D, d_0) si on peut remplir chaque case de la grille infinie avec des dominos de D (on a un nombre infini de copies de chaque domino dans D), tout en satisfaisant les deux conditions suivantes :

- Une copie de d_0 doit être placée dans la première case (celle en bas à gauche).
- Si deux dominos sont adjacents, le côté qu'ils ont en commun doit avoir la même couleur.

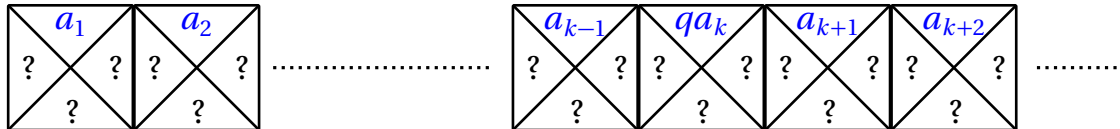
On veut montrer que le problème suivant **PAVAGE-1/4-PLAN** est indécidable, par réduction depuis le complémentaire du problème de l'arrêt sur mot vide.

Entrée : Une paire (D, d_0) où D est un ensemble fini de dominos et $d_0 \in D$.
Question : Peut-on paver le quart de plan avec (D, d_0) ?

1. Expliquez ce que l'on doit faire pour cela (sans expliquer comment pour l'instant).

On veut maintenant construire la réduction. Soit M une machine de Turing **déterministe**. L'idée est de forcer l'ensemble D de dominos construit à partir de M à représenter la suite C_0, C_1, \dots de configurations de M lancée sur le mot vide. Pour cela, on utilise une ligne de la grille pour représenter chaque configuration : la configuration C_0 sera représentée dans la ligne 0 (la plus basse), la configuration C_1 (si elle existe) sera représentée sur la ligne 1, etc.

Supposons que la machine M est dans une configuration C , avec $a_1 a_2 a_3 \dots$ comme contenu (infini) de la bande, q comme état courant, et k comme position de la tête (elle lit donc a_k). On dit qu'une ligne infinie de dominos sur la grille **code** C si elle est de la forme :



Notez qu'on utilise des mots comme couleurs (ces mots ont une ou deux lettres pour l'instant, mais vous pourrez utiliser plus de couleurs si nécessaire dans la suite).

 **Remarque**

⚡ L'intuition sur la réduction est visuelle. Pour *chaque* question, il peut être utile de faire un dessin de la situation, au moins au brouillon, pour la comprendre.

2. Soit C' la configuration obtenue en appliquant une transition de M depuis C . Construisez un ensemble fini de dominos forçant la ligne au dessus d'une ligne qui code C à coder C' .
3. Complétez cet ensemble de dominos pour qu'en fixant le domino en bas à gauche, la ligne 0 code la configuration initiale de M sur le mot vide.
4. Récapitulez la réduction et justifiez qu'elle fonctionne.
5. On s'intéresse maintenant au pavage d'un carré **fini** du plan. À nouveau, on dispose d'une infinité de copies de chaque domino. Montrez que le problème suivant est **NP**-complet.

Entrée : Une paire (D, d_0) où D est un ensemble fini de dominos, **un** domino $d_0 \in D$, et un entier k codé **en unaire** (c'est-à-dire en base 1).

Question : Peut-on paver le carré $k \times k$ avec D et d_0 dans le coin en bas à gauche?

 **Rappel**

⚡ La définition d'un problème **NP**-complet comprend *deux* propriétés. Pour l'une d'entre-elles, on pourra s'inspirer de la réduction ci-dessus pour le pavage infini.

6. Que pensez-vous de la complexité du problème lorsque k est donné en binaire? Expliquez en particulier si le problème est encore dans **NP**, en justifiant.

Exercice 4 : Automates enrichis (6 points)

Un *automate enrichi* manipule une mémoire. Il possède des transitions spéciales qui permettent de lire soit le début, soit la fin de la mémoire, et d'écrire au début de la mémoire. Formellement, un *automate enrichi* est un tuple $\mathcal{A} = (\Sigma, \Gamma, Q, I, F, \delta, \pi, \rho, \beta)$, où Σ est l'alphabet d'entrée, Γ est l'alphabet de mémoire, Q est un ensemble d'états, $I \subseteq Q$ est l'ensemble d'états initiaux et $F \subseteq Q$ est l'ensemble d'états initiaux finaux. De plus $\delta \subseteq Q \times \Sigma \times Q$ contient les *transitions de lecture*, $\pi \subseteq Q \times \Gamma \times Q$ est les *transitions de filage*, $\rho \subseteq Q \times \Gamma \times Q$ les *transitions de défilage* et $\beta \subseteq Q \times \Gamma \times Q$ les *transitions de dépilage*.

Il reste à définir le langage reconnu par un automate enrichi \mathcal{A} . Une *configuration* de \mathcal{A} est un triplet $(q, u, x) \in Q \times \Sigma^* \times \Gamma^*$. Intuitivement, u est le mot restant à lire, et x est le contenu courant de la mémoire. On définit une relation " \rightarrow " entre les configurations :

- **Lecture** : Si $(q, a, r) \in \delta$ alors pour tout $u \in \Sigma^*$ et tout $x \in \Gamma^*$, on a $(q, a u, x) \rightarrow (r, u, x)$.
- **Filage** : Si $(q, d, r) \in \pi$ alors pour tout $u \in \Sigma^*$ et tout $x \in \Gamma^*$, on a $(q, u, x) \rightarrow (r, u, d x)$.
- **Défilage** : Si $(q, d, r) \in \rho$ alors pour tout $u \in \Sigma^*$ et tout $x \in \Gamma^*$, on a $(q, u, x d) \rightarrow (r, u, x)$.
- **Dépilage** : Si $(q, d, r) \in \beta$ alors pour tout $u \in \Sigma^*$ et tout $x \in \Gamma^*$, on a $(q, u, d x) \rightarrow (r, u, x)$.

Enfin, on note $\xrightarrow{*}$ la clôture reflexive transitive de \rightarrow . C'est-à-dire que $(q, u, x) \xrightarrow{*} (r, v, y)$ si et seulement si il existe $n \in \mathbb{N}$ et $(q_0, u_0, x_0), \dots, (q_n, u_n, x_n) \in Q \times \Sigma^* \times \Gamma^*$ tels que,

$$(q, u, x) = (q_0, u_0, x_0) \rightarrow (q_1, u_1, x_1) \rightarrow \dots \rightarrow (q_n, u_n, x_n) = (r, v, y).$$

Le langage reconnu par \mathcal{A} , noté $L(\mathcal{A}) \subseteq \Sigma^*$ contient exactement les mots $w \in \Sigma^*$ tels qu'il existe $q \in I$ et $r \in F$ qui satisfont $(q, w, \varepsilon) \xrightarrow{*} (r, \varepsilon, \varepsilon)$.

1. Soit $\Sigma = \{a, b\}$. Donnez un automate enrichi \mathcal{A} tel que $L(\mathcal{A}) = \{w w \mid w \in \Sigma^*\}$.
2. Soit $\Sigma = \{a, b\}$. Soit $K \subseteq \Sigma^*$ le langage des mots qui contiennent autant de copies de la lettre "a" que de copies de la lettre "b". Donnez un automate enrichi \mathcal{A} tel que $L(\mathcal{A}) = K$.
3. Un *automate à file* est un automate enrichi $\mathcal{F} = (\Sigma, \Gamma, Q, I, F, \delta, \pi, \rho, \beta)$ tel que $\beta = \emptyset$ (il n'y a pas de transitions de dépilage). Montrer que pour tout automate enrichi \mathcal{A} , il existe un automate à file \mathcal{F} tel que $L(\mathcal{A}) = L(\mathcal{F})$.
4. Soit Σ un alphabet quelconque et \mathcal{M} une machine de Turing déterministe ayant Σ pour alphabet d'entrée. Montrez qu'il existe un automate enrichi \mathcal{A} tel que $L(\mathcal{A}) = L(\mathcal{M})$.
5. Est-il vrai que tout langage défini par une grammaire hors-contexte est également reconnu par un automate enrichi?
6. Un *automate à pile* est un automate enrichi $\mathcal{P} = (\Sigma, \Gamma, Q, I, F, \delta, \pi, \rho, \beta)$ tel que $\rho = \emptyset$ (il n'y a pas de transitions de défilage). Montrer que pour tout automate à pile \mathcal{P} , il existe une grammaire hors-contexte \mathcal{G} telle que $L(\mathcal{G}) = L(\mathcal{P})$.



Indication



Pour toute paire d'états $(q, r) \in Q^2$ de \mathcal{P} , \mathcal{G} contient une variable $X_{q,r}$, telle qu'un mot $w \in \Sigma^*$ peut être dérivé depuis celle-ci si et seulement si $(q, w, \varepsilon) \xrightarrow{*} (r, \varepsilon, \varepsilon)$.

7. On considère le problème du vide : étant donné un automate \mathcal{A} pris en entrée, est-il vrai que $L(\mathcal{A}) = \emptyset$? Ce problème est-il décidable pour les automates enrichis? Pour les automates à file? Pour les automates à pile? On justifiera précisément chaque réponse.