

Théorie de la Complexité – Feuille de cours intégré n° 3

Indécidabilité : Problèmes de pavage et de correspondance de Post

1 Quizz de cours

② **Exercice 1** – ① ☆ – On note $\mathcal{L}(M)$ le langage des mots acceptés par une machine de Turing M . Si $L \subseteq \Sigma^*$ est un langage, on note $\bar{L} = \Sigma^* \setminus L$ son complément.

1. Rappeler les définitions suivantes (vous pouvez utiliser la notation $\mathcal{L}(M)$) :

- définition d'un langage décidable,
- définition d'un langage indécidable,
- définition d'un langage semi-décidable,
- définition d'un langage co-semi-décidable.

2. Un langage semi-décidable peut-il être décidable? Indécidable?

3. Si L est décidable et $K \subseteq L$, K est-il nécessairement décidable?

4. Si on sait que K se réduit à L , que peut-on en déduire si on sait en plus que :

- K est décidable?
- L est décidable?
- K est semi-décidable?
- L est semi-décidable?

5. On suppose que K se réduit à L et que L se réduit à H . Les propriétés suivantes sont-elles vraies?

- K se réduit à H .
- $\bar{K}$ se réduit à $\bar{L}$.

2 Exercices de révision

② **Exercice 2** – ① ☆ – Questions d'examen

1. (DS 2024) Est-il vrai que l'union de deux langages indécidables est toujours indécidable?

2. La fonction identité de $\mathbb{N}$ dans $\mathbb{N}$ est-elle calculable?

3. (Examen 2024–25) Le problème suivant est-il décidable ou indécidable? Justifier.

Entrée. Le code d'une machine de Turing M prenant en entrée des entiers codés en base 2.

Question. La machine M calcule-t-elle la fonction $n \mapsto n$ de $\mathbb{N}$ dans $\mathbb{N}$?

4. (DS 2024–25) Le problème suivant est-il décidable ou non? Justifier, soit en donnant un algorithme, soit en faisant une preuve.

Entrée. Une machine de Turing déterministe M .

Question. Le calcul de la machine M sur le mot vide à partir de l'état initial q_0 revient-il à nouveau sur q_0 ?

3 Problèmes de pavage du plan

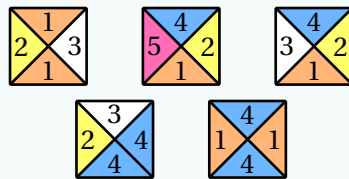
② Exercice 3 – !☆☆ – Pavage (DS 2024)

On dispose d'une grille infinie couvrant le quart du plan. Chaque « carré » de la grille est une case dans laquelle on peut placer une tuile de Wang. Une telle tuile est un carré dont chaque côté (Sud, Ouest, Nord et Est) est coloré. Dans l'exemple sur la figure ci-dessous, les couleurs sont aussi indiquées par des entiers.

Attention, les tuiles sont orientées : on ne peut pas les tourner.



La grille infinie



Un ensemble de 5 tuiles de Wang

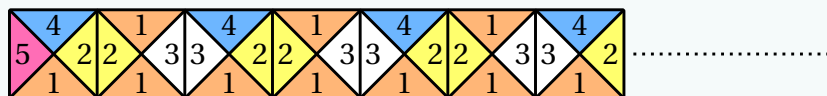
Considérons maintenant une paire (D, d_0) où D est un ensemble fini de tuiles et $d_0 \in D$ est une tuile particulière, appelée *tuile de départ*. On dispose d'un nombre infini de copies de chaque tuile de D .

Soit également E un ensemble de cases de la grille infinie, contenant la première case (celle en bas à gauche). Dans la suite, E sera soit égal à tout le quart de plan, soit égal à la première ligne (celle en bas du quart de plan).

On dit qu'on peut **paver** E **avec** (D, d_0) si on peut remplir chaque case de E avec des tuiles de 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 tuiles sont adjacentes, le côté qu'elles ont en commun doit avoir la même couleur.

Par exemple, si D est le jeu de tuiles dessiné ci-dessus et d_0 est la seconde tuile (la seule contenant la couleur 5), on peut paver la première ligne en commençant par d_0 , puis en alternant, sur les cases suivantes de cette ligne, la tuile dessinée à la gauche de d_0 et celle dessinée à sa droite. On obtient le pavage suivant de la première ligne de la grille :



Par contre, on ne peut pas paver la première ligne en choisissant comme tuile d_0 la quatrième du jeu de tuiles de Wang ci-dessus (la seule qui ne contient pas la couleur 1).

1. Montrez que le problème suivant **PAVAGE-LIGNE** est décidable :

Entrée : Une paire (D, d_0) où D est un ensemble fini de tuiles et $d_0 \in D$.

Question : Peut-on paver la **première ligne** du quart de plan avec (D, d_0) ?

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

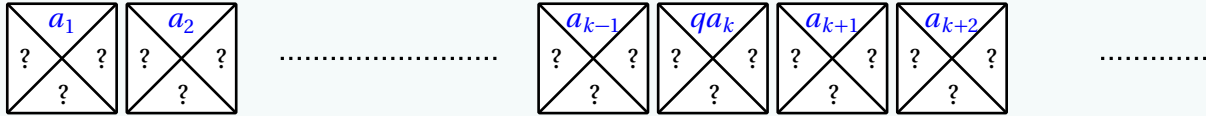
Entrée : Une paire (D, d_0) où D est un ensemble fini de tuiles et $d_0 \in D$.

Question : Peut-on paver le quart de plan avec (D, d_0) ?

2. Décrivez ce qu'est une telle réduction, et en particulier les propriétés qu'elle doit satisfaire.

On veut maintenant construire la réduction. Soit M une machine de Turing **déterministe**. L'idée est de forcer l'ensemble D de tuiles 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 tuiles sur la grille **code** C si elle est de la forme suivante :



(où deux tuiles adjacentes respectent toujours la condition que leur côté commun a la même couleur). Notez qu'on utilise des mots comme couleurs. Ces mots ont une ou deux lettres pour l'instant (a_1 a une lettre, qa_k en a deux). Vous pourrez utiliser plus de couleurs si nécessaire dans la suite, l'important étant d'en avoir un **nombre fini**.



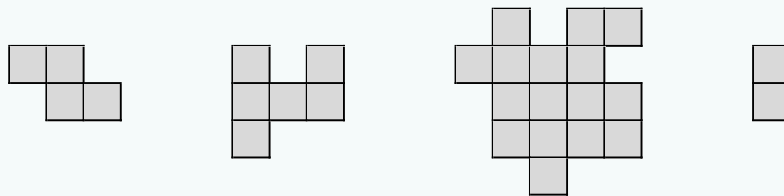
Remarque



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.

3. Soit C' la configuration obtenue en appliquant une transition de M depuis C . Construisez un ensemble fini de tuiles forçant la ligne au dessus d'une ligne qui code C à coder C' . Vous pouvez distinguer plusieurs cas selon le type de transition effectué.
4. Complétez cet ensemble de tuiles pour qu'en fixant la tuile en bas à gauche, la ligne 0 code la configuration initiale de M sur le mot vide.
5. Récapitulez la réduction et justifiez qu'elle fonctionne.

🔗 **Exercice 4 – ★★ – Pavage par pièces de Tetris (d'après le DS 2015)** On considère le problème TETRIS. Une entrée du problème TETRIS est un ensemble fini P de *polyominos* de formes différentes et sans couleur. Un polyomino est une forme connexe obtenue en réunissant des dominos de même taille (*dominos unitaires*) par leurs côtés. Un exemple d'entrée est l'ensemble de polyominos suivants :



Étant donnée une entrée P , le problème TETRIS consiste à déterminer si on peut couvrir, entièrement et sans chevauchements, le plan infini entier avec des polyominos de P . Pour cela, on peut utiliser plusieurs fois le même polyomino. On peut faire tourner chaque polyomino d'un angle multiple de $\pi/2$, mais les symétries verticales ou horizontales sont interdites. Sur l'exemple, la réponse est positive, en utilisant par exemple uniquement le dernier polyomino.

- 1) On admet que le problème WANG de pavage par tuiles de Wang reste indécidable pour le plan entier, même sans fixer la tuile en coordonnées $(0, 0)$. On veut montrer que TETRIS est aussi indécidable. Pour cela, faut-il réduire WANG à TETRIS ou TETRIS à WANG? Justifier votre réponse.
- 2) Décrire en français :
 - a) une réduction de WANG à TETRIS,
 - b) une réduction de TETRIS à WANG.

4 Problème de correspondance de Post

Definition 1 - Problème de correspondance de Post.

Le problème de correspondance de Post (**PCP** en abrégé) est le suivant. Une entrée de ce problème est composée d'un entier $k \geq 1$ et deux suites $v_1, v_2, \dots, v_k$ et $w_1, w_2, \dots, w_k$ de chacune k mots sur un alphabet Σ . Une solution au problème est une suite finie d'indices $i_1, i_2, \dots, i_n$, avec $n \geq 1$ et $1 \leq i_j \leq k$ pour tout j , telle que,

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

Le problème consiste à déterminer si une solution existe ou non.

❓ **Exercice 5** – ☆ – Pour chacune des quatre entrées suivantes, dire si PCP a une solution.

i	1	2	3
v_i	<i>b</i>	<i>aab</i>	<i>ba</i>
w_i	<i>a</i>	<i>ab</i>	<i>a</i>

i	1	2	3
v_i	<i>ab</i>	<i>aab</i>	<i>a</i>
w_i	<i>a</i>	<i>ab</i>	<i>a</i>

i	1	2	3
v_i	<i>a</i>	<i>aa</i>	<i>ba</i>
w_i	<i>aaa</i>	<i>ab</i>	<i>a</i>

i	1	2	3
v_i	<i>abb</i>	<i>b</i>	<i>a</i>
w_i	<i>a</i>	<i>abb</i>	<i>bb</i>

Le problème PCP est **indécidable**. Pour le montrer, on définit la variante suivante de ce problème.

Definition 2 - Problème de correspondance de Post modifié.

Le problème de correspondance de Post modifié (**MPCP** en abrégé) est le suivant. une entrée de ce problème est identique à l'une de PCP. Une solution au problème est une suite finie d'indices $i_1, i_2, \dots, i_n$, avec $n \geq 1$ et $1 \leq i_j \leq k$ pour tout j , telle que,

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

avec $i_1 = 1$. Le problème consiste à déterminer si une solution existe ou non.

❓ **Exercice 6** – ★ – **Problème de correspondance de Post modifié**

1. Quelles sont instances de l'exercice 5 qui sont solutions de MPCP?
2. Montrer que le problème MPCP se réduit au problème PCP.
3. Que peut-on en déduire si on prouve que PCP est indécidable?
4. Que peut-on en déduire si on prouve que MPCP est indécidable?

❓ **Exercice 7** – ① ★ ★ ★ ☆ – **Indécidabilité du problème de correspondance de Post** On veut réduire le problème de l'arrêt sur mot vide au problème MPCP.

1. Écrire ce qu'il faut faire pour cela.
2. L'idée de la réduction est de forcer les solutions minimales éventuelles de PCP sur l'entrée construite commençant par $(C_0 \# C_1 \# C_2 \# \dots \# C_n, C_0 \# C_1 \# C_2 \# \dots \# C_n \# C_{n+1})$, où les C_k sont les configurations successives de la machine de départ. Proposer des paires de mots à produire, pour construire la réduction voulue. Quel est l'intérêt d'être passé par le problème intermédiaire MPCP?
3. Terminer la réduction, et conclure.

❓ **Exercice 8** – 🏠 ☆ – Montrer que PCP restreint à l'alphabet unaire $\{1\}$ est décidable, mais que PCP restreint à l'alphabet binaire $\{0, 1\}$ reste indécidable.

❓ **Exercice 9 – 🏠 ★★ – Problème d’inclusion de Post** 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 $aaa \subseteq ababca$). On considère le problème d’inclusion de Post qui est une version légèrement modifiée du problème de correspondance :

ENTRÉE : $(v_1, w_1), \dots, (v_k, w_k)$
QUESTION : $\exists n \geq 1, i_1, \dots, i_n$ 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. Montrez que le problème d’inclusion de Post est décidable.

❓ **Exercice 10 – 🏠 ① ★★ – Transducteurs** On considère un alphabet fini A ayant au moins 2 lettres. On appelle *transducteur* un tuple $T = (Q, q_I, F, \delta)$ où Q est un ensemble fini d’états, $q_I \in Q$ un état initial, $F \subseteq Q$ un ensemble d’états finaux et $\delta \subseteq Q \times A^* \times A^* \times Q$ un ensemble fini de transitions. Un tel transducteur $T = (Q, q_I, F, \delta)$ accepte des paires de mots $(u, u') \in A^* \times A^*$. Une telle paire $(u, u') \in A^* \times A^*$ est acceptée par T quand il existe une séquence de la forme suivante :

$$q_1 \xrightarrow{(x_1, x'_1)} q_2 \xrightarrow{(x_2, x'_2)} \cdots q_{n-1} \xrightarrow{(x_{n-1}, x'_{n-1})} q_n \quad \text{avec } n \geq 1 \text{ et,}$$

- $q_1, \dots, q_n \in Q$ sont des états tels que $q_1 = q_I$ et $q_n \in F$.
- Pour tout i , (x_i, x'_i) est une paire dans $A^* \times A^*$ qui satisfait $(q_i, x_i, x'_i, q_{i+1}) \in \delta$.
- $u = x_1 \cdots x_n$ et $u' = x'_1 \cdots x'_n$.

1. Montrer que le problème suivant est indécidable.

Entrée : Un transducteur $T = (Q, q_I, F, \delta)$.

Question : Existe-t-il un mot $u \in A^*$ **non vide** tel que la paire (u, u) est acceptée par T ?

2. Même question pour le problème dans lequel on supprime la restriction que le mot u est non vide.