Langages Formels, Calculabilité, Complexité: Travail de rédaction. Pavages et indécidabilité

Matthieu SOLNON

Vendredi 18 Janvier 2007

Table des matières

Table des figures

1 Introduction

Nous nous posons le problème du pavage d'un plan : étant donné certaines tuiles de base, et certaines règles d'adjacence, peut-on paver un quart de plan entier avec ces tuiles et ces règles ? On cherche à démontrer la décidabilité (ou l'indécidabilité) de ce problème.

2 Définition du problème

2.1 Approche intuitive

On veut paver le quart de plan supérieur droit avec des tuiles carrées de même taille, et toutes placées sur un quadrillage du plan, uniforme, et dont les cases sont aussi des carrés.
On se donne un jeu fini de tuiles, dont on dispose pour chacune d'un nombre infini de copies. On oblige une certaine tuile à être placée à l'origine, et seules certaines tuiles peuvent être adjacentes horizontalement et verticalement. On ne peut bien sûr pas retourner ou pivoter les tuiles.
Existe t'il un algorithme qui permet de déterminer si, étant donné le jeu de tuile, les règles d'adjacence et la tuile initiale, permet de décider si on peut paver le plan ou non ?

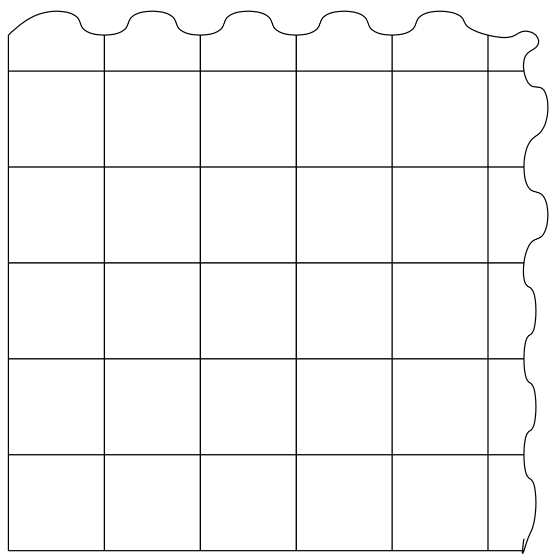

Fig. 1 – Un exemple de quadrillage

2.2 Définition formelle

On formalise le problème comme suit : un système de pavage est un quadruplet : $\mathcal{D} = (D, d_0, H, V)$ où D est un ensemble fini, $d_0 \in D$, et $H, V \subseteq D \times D$. Un pavage est une fonction $f : \mathbb{N} \times \mathbb{N} \to D$ telle que :

$$\begin{array}{ll} f(0,0) = d_0 & \\ (f(n,m), f(n+1,m)) \in H & \forall n,m \in \mathbb{N} \\ (f(n,m), f(n,m+1)) \in V & \forall n,m \in \mathbb{N} \end{array}$$

3 Démonstration de l'indécidabilité du problème de pavage

3.1 Résultat préliminaire

On désignera par ε le mot vide.

Lemme 1. *Le langage* $L_\varepsilon = \{< M > \mid M \textit{ s'arrête sur } \varepsilon\}$ *est indécidable.*

Démonstration. Réduisons polynomialement (le polynomial sera implicite dans la suite du texte lorsque l'on parlera de réduction) le problème de l'arrêt $L_{Halt} = \{< M, w > \mid \text{M s'arrête sur w}\}$ à L_ε. Nous savons déja que ce problème est indécidable. Supposons qu'il existe une machine de Turing qui décide L_ε. Construisons alors une machine M qui décide L_{Halt}. On sait que l'on peut construire à partir du mot w et d'une machine de Turing M une machine M_w qui simule M sur w. Si le calcul sur M_w sur ε s'arrête, alors celui de M sur w aussi, et réciproquement. Ainsi L_{Halt} serait décidable, c'est absurde. □

3.2 L'indécidabilité du problème

Théorème 2. *Le problème de déterminer si, étant donné un système de pavage, il existe un pavage correspondant est indécidable.*

Démonstration. L'idée est de réduire le problème L_ε à ce problème. Ainsi, ce problème devient clairement indécidable.
Si M_0 décide de L_ε, alors on représente une configuration du calcul de M_0 sur ε par une ligne du pavage (qui représentera la bande de M_0 avec quelques informations supplémentaires), la suite des configurations s'ajoutant suivant l'axe croissant des ordonnées du quadrillage. La k^{ieme} configuration se retrouve sur la k^{ieme} ligne.
Ainsi, un pavage infini équivaut à un calcul infini (et pas d'arrêt), alors qu'une impossibilité de pavage équivaut à un arrêt du calcul.
On peut considérer les bords des tuiles comme étiquetés par des mots, seules les tuiles ayant des bords avec la même étiquette pouvant être adjacentes.
Les étiquettes des bords haut et bas permettent de donner la position de la tête de lecture, de la lettre écrite sur la case ainsi que l'état dans lequel se trouve l'automate dans la configuration.
Les étiquettes latérales permettent de forcer la première ligne à représenter le bonne configuration initiale, ainsi que de simuler le mouvement de la tête de lecture.
Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \#)$ une machine de Turing, δ étant la fonction de transition, qui devrait décider L_ε.

Détaillons maintenant les tuiles nécessaires à la résolution du problème.
.

- Pour chaque $a \in \Sigma \cup \Gamma \cup \{\#\}$, on a la tuile suivante, qui communique uniquement une lettre d'une configuration à une autre, sur la même case de la bande, sans la changer :

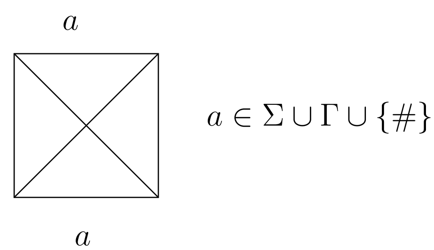

Fig. 2 – Tuile T_1

- Pour chaque $q \in Q$ et chaque $a \in \Sigma \cup \Gamma \cup \{\#\}$, tels que $\delta(q, a) = (p, b, \rightarrow)$ (écriture de b sur la bande, passage dans l'état p et déplacement à droite de la tête de lecture), on utilise les tuiles suivantes :

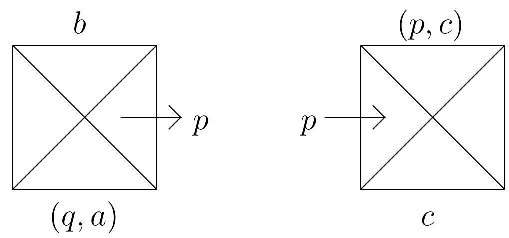

Fig. 3 – Tuiles T_2 et T_3

-

  De même, si $\delta(q, a) = (p, b, \leftarrow)$, on a les tuiles suivantes

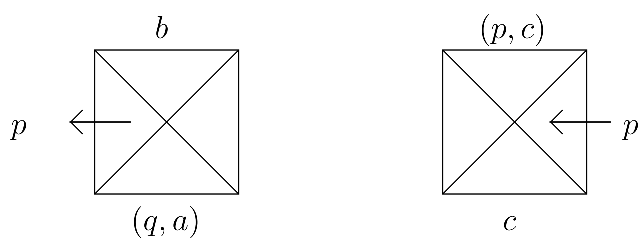

Fig. 4 – Tuiles T_4 et T_5

– On a effectué grâce à ces tuiles la simulation du calcul : propagation des lettres non modifiées, écriture d'une lettre sur une bande, changement d'état, déplacement de la tête de lecture.
Il ne reste plus qu'à initialiser la simulation.

– Il faut, pour cela, placer une tuile à l'origine, et placer une tuile qui comblera le pavage à droite des zones écrites (avec le symbole #).

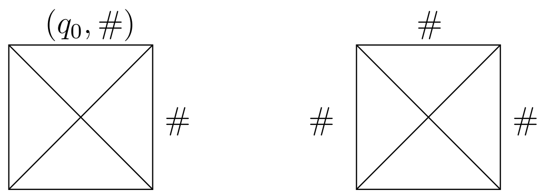

Fig. 5 – Tuiles T_6 et T_7

– La première ligne est donc forcée d'être la représentation de configuration (s,#)### ...

On a donc bien construit la réduction, comme souhaité. □

4 Exemple

Construisons un exemple de la simulation du calcul d'une machine de Turing sur ε (entrée vide) :

On prend comme machine de Turing la machine :

$$M = (\{1, 2, 3\}, (a, b), \{\}, \delta, \{1\}, \{\}, \#)$$
$$\delta(1, \#) = (2, a, \rightarrow)$$
$$\delta(2, \#) = (3, b, \rightarrow)$$

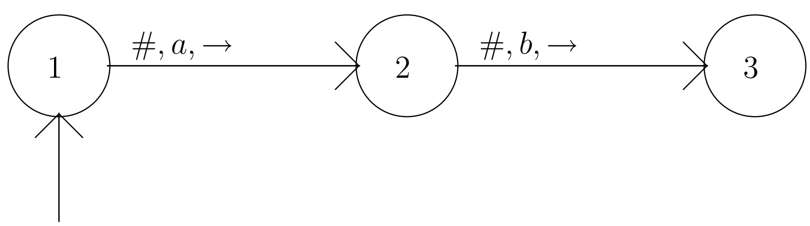

Fig. 6 – Exemple : machine de Turing

Cette machine, lorsqu'elle est lancée sur ε, écrit tout simplement ab et s'arrête.

Voici le pavage (incomplet) associé :

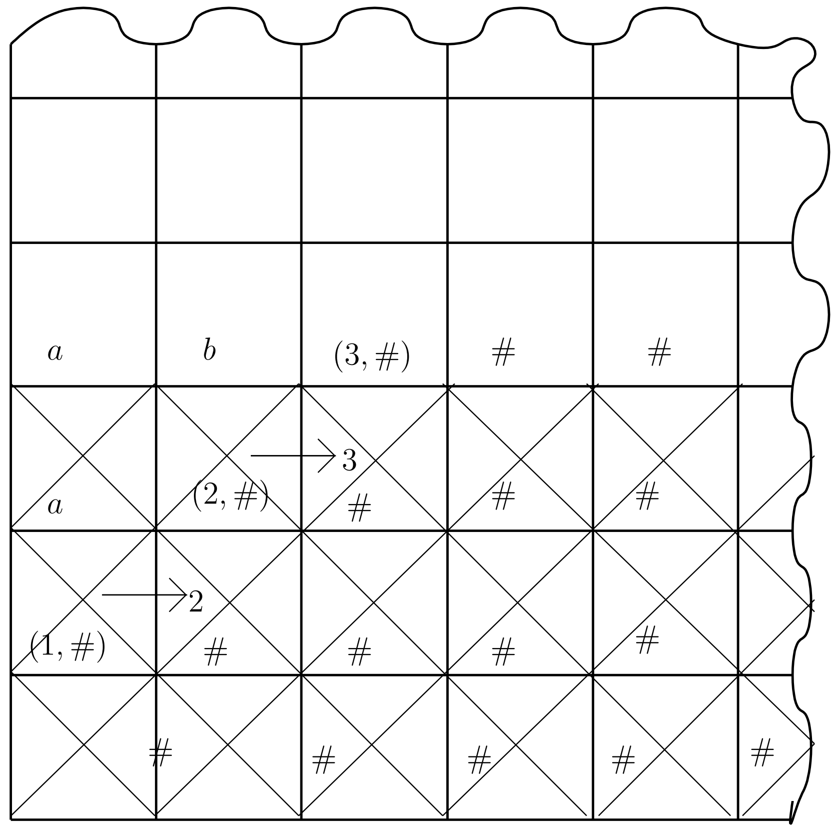

Fig. 7 – Exemple de pavage

Références

[1] Harry R. Lewis, Christos H. Papadimitriou : *Elements of the Theory of Computation*, Prentice Hall International Editions, 1981.