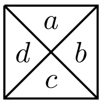
TD 8 : Pavages et degrés Turing

{lionel.rieg,paolo.tranquilli}@ens-lyon.fr

Exercice 1. *Pavage de Wang*

Étant donné un alphabet fini Σ, une *tuile de Wang à couleurs dans* Σ est une figure de la forme



où a, b, c et d sont dans Σ. Étant fixé un ensemble S de tuiles de Wang, un *pavage* du plan par S consiste à recouvrir le plan par des tuiles de S (sans pouvoir les faire pivoter) de façon que les côtés adjacents aient la même couleur. Un *pavage partiel* est défini de façon analogue, mais en ne recouvrant que des portions du plan. Un *pavage fini* est un pavage partiel qui recouvre une portion finie du plan. Possiblement un pavage partiel peut être *complété* en un pavage du plan entier.

1. Formaliser les notions décrites ci-dessus.

On s'intéresse au problème suivant :

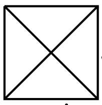
Données : Un ensemble S de tuiles de Wang sur Σ et un pavage fini ρ.

Question : Est-ce que ρ peut-être complété en un pavage du plan ?

On va découvrir que ce problème est indécidable !

2. Prenons une machine de Turing $M = (Q, \Sigma, \Gamma, \Delta, q_0, \{q_+\})$. Construire un alphabet de couleurs Σ_M, un ensemble de tuiles S_M et un pavage fini $\rho(w)$ qui dépend du mot initial w, tels que $\rho(w)$ admet une complétion ssi M ne s'arrête pas sur w.

Suggestions :

- Inclure dans Σ_M une couleur vide, et dans S_M la tuile .
- Penser chaque ligne infinie de tuiles comme une configuration de la machine, une tuile par case du ruban.
- Ne pas oublier d'avoir une (et une seule !) tête de lecture.

Exercice 2. *Degrés Turing*

Lorsqu'on atteint des problèmes indécidables, il devient difficile de comparer la difficulté de deux problèmes. Pour pallier cela, on ajoute aux machines de Turing (unaires, c.-à-d. sur l'alphabet $\{1, \_\}$) un *oracle*, c.-à-d. un second ruban en lecture seule sur lequel peut se trouver n'importe quelle partie A de $\mathbb{N}$ (donc n'importe quel problème) : la i^{e} case du ruban d'oracle contient 1 si $i \in A$ et contient $\_$ sinon. On parle alors de machine de Turing *relativisées à* A. Par exemple, on peut ainsi fournir la solution du problème de l'arrêt comme oracle d'une machine de Turing.
On définit deux relations de comparaison :

- la *réduction Turing* : un problème A est aussi difficile qu'un problème B s'il existe une fonction Turing-calculable $f : \mathbb{N} \to \mathbb{N}$ telle que $x \in B$ ssi $f(x) \in A$, ce qu'on note $B \leq A$;

– la *décidabilité relativisé* : on dira que B est A-décidable (noté $B \preceq A$) s'il existe une machine de Turing relativisée à A qui décide B.

1. Montrer que les deux relations définissent des pré-ordres sur les parties de $\mathbb{N}$: montrer qu'elle sont transitives et réflexives. Sont-elles des ordres ? (Sont-elles antisymétriques ?)

2. Quelle est la relation entre les deux pré-ordres ?

On définit la relation d'équivalence $A \equiv B$ par $A \preceq B \wedge B \preceq A$ et on pose $\mathfrak{D} = \mathcal{P}(\mathbb{N})/\equiv$ l'ensemble des degrés Turing. La relation $\preceq$ induit alors un ordre sur $\mathfrak{D}$.

3. $\mathfrak{D}$ est-il dénombrable ?

4. Montrer que $\mathfrak{D}$ admet un élément minimum noté $\emptyset$. À quels problèmes cela correspond-il ?

5. Montrer que $(\mathfrak{D}, \preceq)$ est un semi-trellis supérieur, c.-à-d. que pour tous oracles A et B il existe un plus petit majorant pour $\preceq$ (unique à $\equiv$ près). En d'autres termes, il existe un oracle $A \vee B$ tel que que A et B sont $A \vee B$-décidable et tel que si A et B sont L-décidables alors $A \vee B$ l'est aussi. Faire de même pour $\leq$.

Pour A fixé, on peut s'intéresser au problème de l'arrêt des machines de Turing relativisées à A. On l'appelle le *jump* de A et on le note A'.

6. Montrer que le *jump* est croissant. En déduire que l'on peut définir le *jump* d'un degré Turing.

7. À quels problèmes correspond $\emptyset'$?

8. Montrer que pour tout problème A, $A \leq A'$ mais $A' \not\preceq A$. Que peut-on en déduire sur $\mathfrak{D}$?

La théorie de la complexité logique s'intéresse notamment aux propriétés de la hiérarchie arithmétique définie par $\prec$. On sait par exemple qu'il existe des degrés incomparables ou caractériser logiquement les énoncés $\preceq \emptyset^{(n)}$.

Exercice 3. *Récursion généralisée*

Étant données des fonctions $g : \mathbb{N} \to \mathbb{N}$, $h : \mathbb{N}^3 \to \mathbb{N}$ et $s : \mathbb{N}^2 \to \mathbb{N}$, soit $f = \texttt{rec}^+(g, h, s) : \mathbb{N}^2 \to \mathbb{N}$ la seule fonction définie par

$$f(0, y) = g(y) \qquad\qquad f(x+1, y) = h(x, f(x, s(x, y)), y)$$

(récursion structurelle sur x). On va prouver que ce schéma est définissable par récursion primitive.

1. Écrire explicitement le déploiement de $f(3, y)$. Quel genre de termes écrits avec la seule fonction s voit-on apparaître ?

Supposons g, h et s récursives primitives.

2. Écrire la fonction récursive primitive $s^*(i, j, y)$ qui itère i fois la fonction s avec premier argument croissant, telle que l'application de s la plus profonde ait comme argument $j \dot{-} 1$. Par exemple $s^*(3, 5, y) = s(2, s(3, s(4, y)))$. Que vaut $s^*(1, i+1, y)$?

3. Prouver que si $j > i$ alors $s^*(j - i, j, x) = s^*(1, i+1, s^*(j - i - 1, j, x))$.

4. Proposer une fonction récursive primitive $f^*(i, j, x)$ telle que $f(x, y) = f^*(x, x, y)$ soit la fonction cherchée, en le prouvant.
(Suggestion : s'assurer que de $i \leq j$ découle $f^*(i, j, x) = f^*(i, i, s^*(j - i, j, x))$ en prouvant cette propriété intermédiaire.)

5. Déduire que le cas général avec $g : \mathbb{N}^k \to \mathbb{N}$, $h : \mathbb{N}^{k+2} \to \mathbb{N}$, $s_i : \mathbb{N}^{k+1} \to \mathbb{N}$ et $f(x + 1, y_1, \ldots, y_k) = h(f(x, s_1(x, \vec{y}), \ldots s_k(x, \vec{y})), x, \vec{y})$ est également récursif primitif.