

TD 11

Exercice 1.*Mangez (encore) des réductions*

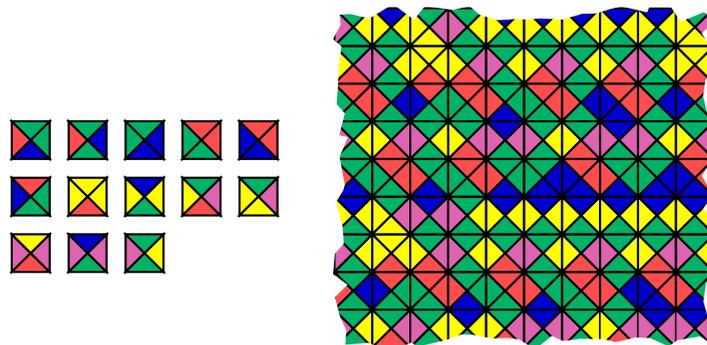
On fixe un alphabet fini Σ contenant entre autre les lettres a et b . ϵ désignera le mot vide. Posons Σ^* l’ensemble des mots finis sur Σ . On notera enfin M_σ avec $\sigma \in \Sigma^*$ la machine de Turing effectuant le programme de code σ . (Plus formellement, $M_\sigma(x) = U(\langle \sigma, x \rangle)$)

 Les ensembles suivants sont-ils récurrents ? récurrents énumérables ? de complémentaire récurrents énumérables ?

1. $\{\sigma \in \Sigma^*, M_\sigma(\epsilon) \text{ s'arrête}\}$
2. $\{\sigma \in \Sigma^*, M_\sigma(\sigma) \text{ s'arrête}\}$
3. $\{\sigma \in \Sigma^*, M_\sigma(ab) \wedge M_\sigma(ba) = aaa\}$ (avec \wedge l’opérateur de concaténation)
4. $\{\sigma \in \Sigma^*, \forall x, M_\sigma(x) = x \text{ si } M_x(x) \text{ s'arrête et } b \text{ sinon}\}$, avec $b \notin \Sigma$.
5. $\{\sigma \in \Sigma^*, \exists x, M_\sigma(x) \text{ ne s'arrête pas}\}$
6. $\{\sigma \in \Sigma^*, \exists w \in \Sigma^*, M_w(\sigma) = abb\}$
7. $\{\sigma \in \Sigma^*, M_\sigma \text{ ne s'arrête sur aucun mot dont } \sigma \text{ est un préfixe}\}$
8. $\{\sigma \in \Sigma^*, M_\sigma \text{ s'arrête sur une partie infinie de } \Sigma^*\}$
9. $\{\sigma \in \Sigma^*, \forall w \in \Sigma^* \quad M_\sigma(w) = 1 \Leftrightarrow w = \sigma\}$

Exercice 2.*Posons du carrelage*

Soit A un ensemble fini de tuiles carrées dont chaque bord est colorié. On code chaque tuile comme un 4-tuple (N, W, S, E) qui donne la couleur correspondante à chacun des bords. Un pavage est une fonction ρ de \mathbb{Z}^2 dans A telle que deux tuiles qui sont côte à côte via cette fonction ont le bord commun de la même couleur. On dit que A pave le plan s’il existe un pavage ρ . Voir la Figure ci-dessous.



 On définit le problème de domino à origine fixée comme :

$$DP_* = \{ \langle (A, a) \rangle \mid \exists x : \mathbb{Z}^2 \rightarrow A \text{ pavage}, x(0,0) = a \}$$

avec A un ensemble de tuiles de Wang et $a \in A$ une tuile de cet ensemble. Montrer que DP_* est indécidable.

(Indice : coder une machine de Turing avec des tuiles et choisir a la tuile avec l'état initial)

Exercice 3.

Moins de place

Une machine de Turing est dite *linéairement bornée* si elle n'écrit pas en dehors de l'espace utilisé par la donnée.

Comme on peut toujours supposer qu'une machine de Turing n'écrit pas le symbole blanc \sqcup (en le dupliquant éventuellement en un autre symbole \sqcup_2), on peut aussi dire qu'une machine linéairement bornée est une machine telle que si $\delta(p, \sqcup) \ni (q, b, x)$ est une transition alors $b = \sqcup$ et $x = \leftarrow$. Ceci empêche la machine de modifier les symboles blancs présents sur la bande.

1. Etant donné un mot w et une machine linéairement bornée M , peut-on décider si M accepte w ?
2. Etant donné une machine linéairement bornée M , peut-on décider si M n'accepte aucun mot, c'est-à-dire $L(M) = \emptyset$?

Indice : On pourra commencer par montrer que ce problème est indécidable pour les machines de Turing en général.

Exercice 4.

La poste par correspondance

Σ est un alphabet fini et P un ensemble fini de paires de mots sur Σ . Le Problème de Correspondance de Post associé à Σ, P est l'existence d'une suite non vide $(v_i, w_i)_i$ d'éléments de P telle que la concaténation des v_i soit égale à la concaténation des w_i . Le Problème de Correspondance de Post Modifié est celui de l'existence d'une telle suite lorsque le premier terme est fixé.

1. Résoudre PCP pour les instances suivantes :
 1. $P = (aab, ab), (bab, ba), (aab, abab)$
 2. $P = (a, ab), (ba, aba), (b, aba), (bba, b)$
 3. $P = (ab, bb), (aa, ba), (ab, abb), (bb, bab)$
 4. $P = (a, abb), (aab, b), (b, aa), (bb, bba)$
2. Montrer que si Σ ne contient qu'une lettre le problème est décidable.
3. Montrer l'équivalence entre PCP et PCPM, c'est à dire qu'à partir d'un algorithme résolvant toute instance de PCP, vous pouvez créer un algorithme résolvant toute instance de PCPM, et inversement.

*Indication : Dans le cas PCP permet de résoudre PCPM, vous pourrez ajouter à l'alphabet deux lettres * et \$ et utiliser les deux morphismes p et s suivant :*

$$\forall a_1 \dots a_k \in \Sigma^*, p(a_1 \dots a_k) = *a_1 * \dots * a_k \text{ et } s(a_1 \dots a_k) = a_1 * \dots a_k *.$$

4. Peut-on se passer des couples de la forme (w, w) dans PCPM ?
5. Montrer que PCP est indécidable.

Indication : Pour montrer cela, vous pouvez montrer que PCP permet de résoudre l'arrêt : à une machine M et une entrée x on peut créer une instance de PCP qui est acceptée si et seulement si la machine M s'arrête sur l'entrée x .

Exercice 5.*Rationnel*

Le but de cet exercice est de montrer que toute machine de Turing à une seule bande qui fonctionne en temps $o(n \log n)$ accepte un langage rationnel. On montre d'abord que cette borne est optimale, puis on définit la notion de franchissement pour prouver le résultat. Soit M une machine de Turing et $k \geq 1$ un entier. On dit qu'une étape de calcul $C \rightarrow C'$ de M franchit la frontière entre les cases k et $k+1$ si $C = upav$, $C' = ubqv$ et $|u| = k$ en appliquant une transition $p, a \rightarrow q, b, R$ ou bien le cas symétrique. Soit γ un calcul de M , on appelle suite des franchissements de γ en k , la suite des états p_0, \dots, p_m observés juste après les franchissements de la frontière $k|k+1$.

1. Montrer que le langage $L = \{w \in \{a, b\}^* : |w|_a = |w|_b\}$ n'est pas rationnel et que L est reconnu par une machine de Turing qui fonctionne en temps $O(n \log n)$.
2. Soient $x = uy$ et $x' = u'y'$ deux contenus de bande (ie y, y' sont infinis, à support fini). Soient γ et γ' deux calculs sur x et x' tels que la suite de franchissements de γ en $|u|$ est identique à la suite de franchissement de γ' en $|u'|$. Montrer qu'il existe un calcul γ'' sur uy' qui est identique à γ sur u et à γ' sur y' .
3. En déduire que si γ est un calcul sur $x = uvv$ tel que les suites de franchissements en $|u|$ et $|uvv|$ sont identiques, il existe des calculs sur uy et uvv de même type de γ (ie acceptant ssi γ est acceptant).
4. On dit qu'une relation d'équivalence \sim sur A^* est régulière à droite si $w \sim w'$ implique $wa \sim w'a$. On rappelle qu'une relation d'équivalence \sim sur un ensemble E sature un sous-ensemble F de E si F est égal à l'union des classes qui l'intersectent. Montrer que si L est saturé par une relation d'équivalence d'indice fini qui est régulière à droite, alors L est rationnel.
5. Pour une machine de Turing M , on définit $t_M(n), r_M(n)$ le temps d'exécution maximal et la longueur maximale d'une suite de franchissements d'un calcul de M sur une entrée de longueur n .
Montrer que si M vérifie $r_M(n) \leq K$ pour tout $n \geq 0$, alors M reconnaît un langage rationnel. (Indice : on pourra supposer que M accepte uniquement en rencontrant un symbole blanc)
6. Soit M une machine de Turing telle que $t_M(n) \leq Kn$ pour une constant K . Montrer qu'il existe une machine M' équivalente à M telle que $t_{M'}(n) \leq 2Kn$ et qui n'utilise que la partie de la bande où est écrite l'entrée.
7. Soit M une machine de Turing à une seule bande telle que $t_M(n) \leq Kn$ et qui n'utilise que la partie de la bande où est écrite l'entrée. Montrer que si un calcul a une suite de franchissements de longueur supérieure à $2K|Q|^K + K$, alors il existe au moins trois suites de franchissements identiques de longueur strictement inférieure à K .
(Indice : on pourra trouver un lien entre taille des suites de franchissements et temps d'exécution de la machine puis utiliser le principe des tiroirs)
8. Montrer que toute machine de Turing M à une seule bande telle que $t_M(n) \leq Kn$ accepte nécessairement un langage rationnel.
9. (Difficile) Montrer de même que toute machine de Turing M à une seule bande telle que $t_M(n) = o(n \log n)$ accepte nécessairement un langage rationnel.