Langage rationnels (Applications)

Exercice 1 (Mots répétés)

1. Soit Σ un alphabet. On s'intéresse à $\mathcal{L} \subset \Sigma^*$ constitué des mots u qui sont de la forme w^2, c'est à dire
$$\mathcal{L} = \{u : \exists w \in \Sigma^*, u = w^2\}$$
   (a) Si Σ est réduit à une lettre, montrer que $\mathcal{L}$ est rationnel.
   (b) Si Σ a deux lettres, montrer que $\mathcal{L}$ n'est pas rationnel
2. Soit Σ un alphabet. On s'intéresse à $\mathcal{L} \subset \Sigma^*$ constitué des mots u qui ne sont pas de la forme w^2, c'est à dire que pour tout $w \in \Sigma^*$ on a $u \neq w^2$, ainsi
$$\mathcal{L}' = \{u : \forall w \in \Sigma^*, u \neq w^2\}$$
   (a) Si Σ est réduit à une lettre, montrer que $\mathcal{L}'$ est rationnel.
   (b) Si Σ a deux lettres, montrer que $\mathcal{L}'$ n'est pas rationnel

Exercice 2 (Ensembles 2-reconnaissables) Un ensemble $\mathbb{A} \subset \mathbb{N}^*$ est 2-reconnaissable si le langage $\mathcal{L}_{\mathbb{A}} \subset \{0,1\}^*$ correspondant aux éléments de $\mathbb{A}$ écrits en base 2 est rationnel.

1. Montrer que tout sous ensemble fini de $\mathbb{N}$ est 2-reconnaissable.
2. Est ce que $\{2^n : n \in \mathbb{N}\}$ est 2-reconnaissable ?
3. On souhaite montrer par l'absurde que $\{3^n : n \in \mathbb{N}\}$ n'est pas 2-reconnaissable.
   (a) Supposons que $\{3^n : n \in \mathbb{N}\}$ soit 2-reconnaissable. Montrer qu'il existe une suite $(n_i)_{i\in\mathbb{N}}$ d'entiers et trois mots u, v, w tel que le développement binaire de 3^{n_i} soit uv^iw pour $i \geq 1$.
   (b) Montrer $\lim\limits_{i\to\infty} \frac{3^{n_{i+1}}}{3^{n_i}} = 2^{|v|}$.
   (c) En déduire que $\{3^n : n \in \mathbb{N}\}$ n'est pas 2-reconnaissable.
4. La conjecture de Mersenne dit qu'il y a une infinité de nombres premiers de la forme $2^n - 1$. Sous la condition que la conjecture de Mersenne soit vrai, est-ce que $\{n \in \mathbb{N} : n \text{ premier}\}$ est 2-reconnaissable ?

Exercice 3 (Reconnaissance d'un mot dans un texte) Dans cet exercice $\Sigma = \{a, b, c\}$.

1. Soit le mot $m = abab$ et $\mathcal{L}_m$ le langage des mots qui ont m comme suffixe. $\mathcal{L}_m$ contient les mots de la forme $v = um$, avec $u \in \Sigma^*$, autrement dit
$$\mathcal{L}_m = \{v \in \Sigma^* : \exists u \in \Sigma^* \text{ tel que } v = u.m\}.$$
   Par exemple, les mots *aaabab* et *babab* mais *caabc* n'appartient pas à $\mathcal{L}_m$. Prouvez que $\mathcal{L}_m$ est rationnel. Proposez un automate fini non-déterministe $\mathcal{A}_n$ pour $\mathcal{L}_m$.
2. Transformez $\mathcal{A}_n$ en un automate déterministe complet $\mathcal{A}_d$ équivalent.
3. Si l'alphabet était $\{a, b, c, d, e\}$, comment répercuter cette modification sur $\mathcal{A}_d$?
4. On considère l'algorithme suivant :

Data: Un mot u dans lequel on cherche le nombre d'occurrences d'un motif m
Un automate déterministe $\mathcal{A}_d = (\Sigma, Q, i, F, \delta)$ qui reconnaît le langage $\mathcal{L}_m$.
Result: Un entier c

```
q ← i;
i ← 1;
c ← 0;
while i ≤ |u| do
    q ← δ(q, u_i);
    i ← i + 1;
    if q ∈ F then
        c ← c + 1;
```

(a) Illustrez le fonctionnement de cet algorithme lorsque $u = bcababcabbababac$ et l'automate $\mathcal{A}_d$ précédent. Vous donnerez pour chaque passage dans la boucle principale la valeur de c.

(b) Est ce que cet algorithme termine ?

(c) Que calcule cet algorithme en général ? Justifiez votre affirmation.

(d) Quelle est la complexité de cet algorithme ?

(e) Que vaut c à la fin de l'exécution de l'algorithme pour $u = cabbabababababc$? Que remarquez vous ?

(f) Comment faudrait-il modifier l'automate $\mathcal{A}_d$ pour compter que le nombre maximal d'occurrences disjointes du motif au sein de la chaîne d'entrée ? Par exemple la réponse devrait être 2 dans le dernier exemple.

Exercice 4 (Barman aveugle) Tout se passe dans un bar un peu spécial. En effet, le barman est aveugle, porte des gants de boxe et est assez joueur. Il propose donc à un client de faire un petit jeu avec lui. Le client dispose 4 verres sur un plateau qui peut tourner. Le barman peut retourner autant de verres qu'il souhaite puis le client peut faire tourner le plateau à sa guise et ainsi de suite. Cependant à cause de ses gants et de sa cécité, le barman ne peut déterminer si un verre est à l'endroit ou à l'envers. Un troisième personnage joue le rôle du juge : il déclare gagnant le barman si à un moment celui-ci met tous les verres dans le même sens.
Le barman peut-il gagner de façon certaine ? Si oui en combien de coups ?

1. Pour résoudre le problème, on commencera par en donner une modélisation à l'aide d'un automate fini. On prendra pour états les configurations du plateau modulo les diverses symétries (rotations et retournement de tous les verres) afin d'en minimiser le nombre et pour actions les différents coups jouables par le barman.
2. Traduire le problème du barman en un problème sur l'automate.
3. Etant donné un automate fini $\mathcal{A} = (\Sigma, Q, i, F, \Delta)$ on construit l'automate fini $\mathcal{A}_\forall = (\Sigma, \mathcal{P}(Q), \{i\}, F, \widetilde{\Delta})$ de telle sorte que :
   — $\mathcal{P}(Q)$ est l'ensemble des sous-ensembles de Q ;
   — pour $P, P' \in \mathcal{P}(Q)$ et $a \in \Sigma$, on a $(P, a, P') \in \widetilde{\Delta}$ si et seulement si on a
   $$P' = \{q \in Q \text{ tel qu'il existe } p \in P \text{ vérifiant } (p, a, q) \in \Delta\}.$$
   Montrer que
   $$\mathcal{L}(\mathcal{A}_\forall) = \{w \in \Sigma^* \text{tel que tout chemin étiqueté par } w \text{ dans } \mathcal{A} \text{ est acceptant}\}$$
   .
4. Donner l'automate $\mathcal{A}_\forall$ correspondant à l'automate $\mathcal{A}$ de la première question.
5. Résoudre le problème du barman aveugle.