Partiel - Durée 2 heures

Les documents manuscrits (ou polys de cours) sont autorisés.
Les exercices sont indépendants et de difficulté très variable. Ne vous obstinez pas inutilement sur un exercice, mais ne vous éparpillez pas trop non plus (oui, je sais, c'est dur).
L'énoncé est trop long, nous en tiendrons compte dans la notation.
Sauf mention contraire, dans tout le sujet Σ désigne l'alphabet à deux lettres $\{a, b\}$ (lorsque l'alphabet n'est pas spécifié, c'est $\{a, b\}$ par défaut).
Les résultats vus en cours ou TD peuvent être utilisés directement (en mentionnant éventuellement que ça a été vu en cours ou TD, pour les résultats un peu obscurs).
N'hésitez pas à faire des schémas pour expliquer vos réponses, ne vous perdez pas en explications trop complexes, allez à l'essentiel (sans négliger les points importants).

Exercice 1 *Introduction*

Les langages suivants sont-ils rationnels ? algébriques ? Dans le cas où ils sont rationnels, donner le nombre d'états de l'automate minimal. Justifier les réponses.

- $L_1 = \{a^{51n} \mid n \in \mathbb{N}\}$
- $L_2 = \{x \in \{0,1\}^* \mid x$ est l'écriture binaire d'un nombre pair $\}$
- $L_3 = \{x \in \{0,1\}^* \mid x$ est l'écriture binaire d'une puissance de 2 $\}$
- $L_4 = \{a^n \mid$ il existe n chiffres "2" consécutifs dans le développement décimal de π $\}$
- $L_5 = \{w \mid \overline{w}$ est un facteur de $ww\}$ ($\overline{w}$ est le miroir de w)
- $L_6 = \{x \mid xx$ est un palindrome $\}$
- $L_7 = \{xx \mid x$ est un palindrome $\}$

Exercice 2 *Concaténations*

Soit L un langage rationnel. Soient X et Y deux langages sur Σ.

1. Montrer que si $L \subseteq X.Y$, alors

$$L = \bigcup_{i \in I} X_i.Y_i$$

Où I est un ensemble fini et $\forall i \in I$, $X_i \subseteq X$ et $Y_i \subseteq Y$.

2. Soit une fonction totale $f : \mathbb{N} \to \mathbb{N}$ telle que le langage $\{a^n b^{f(n)} \mid n \in \mathbb{N}\}$ soit rationnel. Déduire de la question précédente que $f(\mathbb{N})$ est fini (ou bien le faire de manière directe, sans utiliser la question précédente).

3. Montrer que pour tout n, l'ensemble $f^{-1}(n)$ est semi-linéaire (union finie de suites arithmétiques).

Exercice 3 *Mince alors !*

Définition 1 (rayon). *On appelle* rayon *un ensemble de mots de la forme uv^*w, où u, v et w sont des mots de Σ^* (si $v = \varepsilon$, le rayon est réduit à $\{uw\}$).*

Définition 2 (mince). *Un langage est* mince *si c'est une union finie de rayons.*

Définition 3 (croissance bornée). *Un langage L est dit* à croissance bornée *si le nombre de mots de longueur n dans L est borné par une constante indépendante de n pour tout n.*

On se propose de montrer la proposition suivante :

Proposition 1. *Un langage est mince si et seulement si il est rationnel et à croissance bornée.*

1. Montrer que si un langage est mince, alors il est rationnel et à croissance bornée.

On considère maintenant un langage L rationnel. Soit $\mathcal{A}$ l'automate minimal reconnaissant L.

2. Montrer que si $\mathcal{A}$ a deux cycles distincts qui ont au moins un état en commun, alors L n'est pas à croissance bornée.

3. Montrer que si $\mathcal{A}$ a deux cycles disjoints reliés par un chemin, alors L n'est pas à croissance bornée.

4. Conclure.

Exercice 4 *Une grammaire*

On considère la grammaire algébrique

$$S \to aSS \mid b$$

Montrer que le langage engendré par cette grammaire est exactement l'ensemble des mots w vérifiant les deux propriétés suivantes :

1. $|w|_a = |w|_b - 1$
2. tout préfixe propre u de w vérifie $|u|_a \geq |u|_b$

Exercice 5

1. Soit L un langage rationnel. Les langages $\mathrm{DEMI}(L) = \{w \in \Sigma^* \mid ww \in L\}$ et $\mathrm{DOUBLE}(L) = \{ww \mid w \in L\}$ sont-ils nécessairement rationnels ?

2. On suppose maintenant que L est algébrique. Les deux langages précédents sont-ils nécessairement algébriques ?

Exercice 6 *Automates boustrophédons*

1. (cette question ne vaut pas de points... juste un cookie) En français, que signifie le mot "boustrophédon" ?

Un automate *boustrophédon* (ou *bilatère* ou encore *two-way* pour les anglophiles) est un quintuplet $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ où Q, Σ, q_0 et F sont respectivement les états, l'alphabet, l'état initial et les états finals de l'automate (tout pareil que pour un automate fini), et δ est la fonction de transtion de l'automate :

$$\delta : Q \times \Sigma \to Q \times \{\leftarrow, \rightarrow\}$$

La capacité supplémentaire d'un automate boustrophédon par rapport à un automate fini classique, est de pouvoir se déplacer sur le mot vers la droite et la gauche (là où un automate fini ne peut qu'avancer vers la droite).
Une *configuration instantanée* de $\mathcal{A}$ est un élément uqv de $\Sigma^*.Q.\Sigma^*$. Elle décrit la situation de l'automate où le mot $uv \in \Sigma^*$ est le mot en entrée et où l'automate est dans l'état q en

train de lire la première lettre du mot v. La configuration initiale de l'automate sur le mot w est donc q_0w.
Chaque étape du calcul de $\mathcal{A}$ est une succession de deux configurations instantanées :

$$\forall q \in Q, \forall a, b \in \Sigma, \forall u, v \in \Sigma^*, \quad \begin{array}{ll} u.aqb.v \vdash u.abq'.v & \text{si} \quad \delta(q,b) = (q', \rightarrow) \\ u.aqb.v \vdash u.q'ab.v & \text{si} \quad \delta(q,b) = (q', \leftarrow) \end{array}$$

Si l'automate se trouve sur la lettre la plus à gauche du mot et qu'il effectue une transition qui l'amène encore plus à gauche, le calcul est interrompu et le mot n'est pas reconnu. Un mot est reconnu si le calcul de l'automate sort par la droite en étant dans un état final :

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid q_0w \vdash^* wq,\ q \in F\}$$

On se propose de montrer le résultat suivant (de deux manières différentes) :

Proposition 2. *Tout langage L reconnu par un automate boustrophédon est rationnel.*

2. Montrer que pour tout automate boustrophédon $\mathcal{A}$, il existe une constante k telle que pour tout mot w accepté par l'automate, l'automate ne passe pas plus de k fois sur chaque lettre de w.

Etant donné un automate boustrophédon $\mathcal{A}$ et un mot $w = x_1 \dots x_n$ accepté par $\mathcal{A}$, on s'intéresse pour chaque lettre x_i de w à la suite des q_j^i, état dans lequel se trouve l'automate quand il passe sur la lettre x_i pour la j-ième fois.

3. Expliquer comment, étant données deux suites finies d'états (q_n) et (q'_m) et deux lettres α et β, on peut vérifier si elles sont "compatibles", c'est-à-dire s'il existe un mot w contenant le facteur $\alpha\beta$ tel que la suite d'états rencontrés en α est (q_n) et la suite d'états rencontrés en β est (q'_m).

4. Etant donné un automate boustrophédon $\mathcal{A}$, déduire des questions précédentes la construction d'un automate fini non-déterministe qui simule le fonctionnement de $\mathcal{A}$, c'est-à-dire qu'il "devine" les suites d'états en chaque lettre, et vérifie qu'elles sont compatibles. Conclure.

5. Majorer le nombre d'états de l'automate fini déterministe qui reconnaît le même langage qu'un automate boustrophédon à n états obtenu à l'aide de la construction de la question précédente (après déterminisation).

On va maintenant montrer le même résultat mais d'une manière différente (en construisant un automate bien plus petit).
Soit $\mathcal{A}$ un automate boustrophédon, $w = w'a$ un mot ($a \in \Sigma$), on définit la fonction $f_w : Q \to Q \cup \{\#\}$ (ou $\#$ est un élément qui n'est pas dans Q) par :
– $f_w(q) = \#$ si pour tout mot u, partant de la configuration $w'qau$ (l'automate est sur la dernière lettre de w et dans l'état q) l'automate ne ressort jamais par la droite de w (soit il fait un cycle, soit il sort par la gauche et le calcul s'arrête) ;
– $f_w(q) = q'$ si partant de la configuration $w'qau$ l'automate est dans l'état q' la première fois qu'il sort à droite du mot w (donc on arrive dans la configuration wqu).

6. Soient $w \in \Sigma^*$ et $a \in \Sigma$, expliquer comment calculer f_{wa} à partir de f_w et δ.

7. A l'aide des fonctions f_w, déduire une construction d'un automate fini déterministe reconnaissant le même langage que $\mathcal{A}$ n'utilisant que $|Q|^{O(|Q|)}$ états.