Grammaires et Analyse

Exercice 1 Lesquel parmis les langages suivant sont algebriques ? Justifiez vos reponses.

1. $\{a, b, aa, bb\}$
2. $a^*b^*c^*$
3. $\{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$
4. $\{a^m b^n \ : \ m \neq n\}$
5. $\{a^n b^n c^m : n, m > 0\}$
6. $\{a^{n^2} : n > 0\}$
7. $\{a^n b^m : m = n^2\}$
8. le miroir d'un langage algebrique L
9. $\{v\widetilde{v}w\widetilde{w} : v, w \in \{a, b\}^* \ et \ \widetilde{z} \ est \ l'image \ miroir \ de \ z\}$
10. $L = \{a^n b^m c^n d^m \ : \ n, m > 0\}$
11. MIN($\mathcal{L}$) $= \{x \in \mathcal{L}, x = yz$ et $y \in \mathcal{L}$ impliquent $z = \epsilon\}$, lorsque $\mathcal{L}$ est algebrique
    (indication : $\mathcal{L} = \{0^i 1^j 2^k, i \leq k$ ou $j \leq k\}$) ;
12. MAX($\mathcal{L}$) $= \{x \in \mathcal{L}, xy \in \mathcal{L}$ implique $y = \epsilon\}$.

Exercice 2 On considère la grammaire suivante qui reconnait un langage $\mathcal{L}$ sans epsilon :

$$\begin{array}{lcl} S & \longrightarrow & MS \mid c \\ M & \longrightarrow & Ac \mid AbV \\ V & \longrightarrow & S \\ A & \longrightarrow & a \end{array}$$

- Proposez le plus directement possible un automate à pile pour cette grammaire
- Mettez ensuite cet automate sous forme normale de Greibach et proposez à nouveau un automate à pile pour ce langage.
- Comparez l'interêt des deux automates pour obtenir un outil efficace qui reconnait $\mathcal{L}$
- Dans le cas de l'exemple précédent, le non-déterminisme est t'il réellement présent dans la grammaire ?
- On appelle k-forme normale de greibach une grammaire dont toutes les règles sont, soit de la forme $X \longrightarrow w$ avec $w \in \Sigma^*$ et $|w| \leq k$, soit de la forme $X \longrightarrow w.\gamma$ avec $w \in \Sigma^*, |w| = k$ et $\gamma \in (N + \Sigma)^*$. Montrer que pour tout entier k, un langage algébrique admet une k-forme normale de greibach.
- On considère à présent la grammaire

  $$\begin{array}{lcl} S & \longrightarrow & MNP \mid NP \\ M & \longrightarrow & NM \mid P \\ P & \longrightarrow & b \mid bP \\ N & \longrightarrow & a \end{array}$$

  Comment reconnaitre le mot aabab
  Mettre cette grammaire sous 2-forme normale de greibach
  Comment reconnait-on le mot aaab avec cette grammaire ?

Exercice 3 Donner des expressions rationnelles pour les langages suivants :
- L'ensemble des identificateurs dans PASCAL (ou C)
- L'ensemble des chaînes de caractères pour
  - Les entiers (par exemple 100 ou -42 mais pas 007),
  - Les nombres réelles avec exponent (ou non), par exemple $1.02E-3$.
- toutes les suites de 0 et de 1 qui ne contiennent pas la sous suite 011
- toutes les suites de 0 et de 1 qui ne contiennent pas la sous chaine 011
- toutes les suites de chiffres sans chiffre répété

Exercice 4 Soit $\Sigma = \{a, b\}$. Donner un automate non-détérministe qui réconnait le langage $\Sigma^* ababaa \Sigma^*$. Détérminiser cette automate. Minimiser l'automate. En déduire un algorithme pour tester si $ababaa$ est un facteur d'un mot w.

Exercice 5 En lex les unités lexicales sont décrites à l'aide d'expressions régulières. Etant donné une expression régulière quelconque, peut-on se passer de l'opérateur de complémentation ?

Que peut on dire de :
- $\overline{a.(b|c)^*.d}$
- $\overline{(a.b|a.a.b)^*.a.a}$

Exercice 6 *Lex construit les automates deterministes minimaux assocés aux expressions régulières. Que cela donne t'il sur les expressions :*
- $(a|b)^*.a.(a|b)$
- $(a|b)^*.a.(a|b).(a|b)$
- $(a|b)^*.a.(a|b).(a|b).(a|b)$
- *Montrer que tout automate fini pour* $(a|b)^*.a.(a|b).(a|b).\ldots.(a|b)$, *où il y a* $n-1$ *occurrences de* $(a|b)$ *à la fin, doit avoir au moins* 2^n *etats.*

Peut t'on s'attendre à un tel phénomène dans l'utilisation de Lex pour un compilateur ?