A.1 Transformations de grammaires

1. Proposer, pour chacun des langages suivants, une grammaire algébrique.
   - langage des expressions arithmétiques (complètement) parenthésées $(((4 + -3) * 2), (3 + (4 - (6 * 6)))...)$
   - langage des expressions arithmétiques postfixes `5 4 + 2 *`, `3 4 6 6 * - +`...
   - langage des palindromes : *aabaa*, *abccba*, *aa*...
   - $\{w \in X^*/w = a^n b^{2n}; n > 0\}$
   - langage des fonctions d'un langage de programmation, dont les arguments peuvent être des constantes (a, b, c) ou des fonctions (f, g, h) d'arité variable : $f(g(a, b), c)$, a, $g(a, g(a), a)$...
   - langage des expressions bien formées de la logique des propositions
   - langage des expressions bien formées de la logique des prédicats
   - $\{w \in X^*/w = a^p b^n c^n a^p; n, p > 0\}$
   - $\{w \in X^*/w = a^n b^p \,; n \geq p \geq 0\}$
2. Soit la grammaire suivante : $\mathcal{G}_1 = \langle \{a, b\}, \{S\}, S, \{\ S \ \rightarrow \ aSbS|bSaS|\varepsilon\ \}\rangle$.
   (a) Quel est le langage reconnu ?
   (b) Proposer une grammaire $\mathcal{G}_2$ ε-libre qui reconnaît le même langage.
   (c) Dessiner les deux arbres de dérivation qui correspondent à l'analyse du mot *aabbaabbbaab* au moyen des deux grammaires $\mathcal{G}_1$ et $\mathcal{G}_2$.
3. Soit l'alphabet $X = \{+, =, a\}$. (1) Donner une grammaire algébrique pour le langage L dont chaque mot représente une addition correcte de deux suites de caractères a. Par exemple L contient le mot $aa + aaaa = aaaaaa$. (2) Donner un automate à pile qui reconnaît le même langage.
4. Dé-récursiver la grammaire suivante
   $S \rightarrow aS \mid BS \mid \varepsilon$
   $B \rightarrow bAb \mid SaS$
   $A \rightarrow a \mid Sa$
5. Règles simples (ou productions singulières). Transformer la grammaire suivante en grammaire sans règles simples.
   $S \rightarrow AB \mid A\ ;\ A \rightarrow aB \mid bA \mid aSb\ ;\ B \rightarrow S \mid b$
6. Proposer une grammaire algébrique ambigüe qui reconnaisse le langage $a^n b^n$.
7. Trouver une grammaire quadratique équivalente à la grammaire suivante :
   $S \rightarrow aAB \mid B\,A$
   $A \rightarrow BBB \mid a$
   $B \rightarrow AS \mid b$
8. Ambiguïté de grammaires
   (a) Montrer que la grammaire suivante est ambiguë :
   $S \rightarrow TU; T \rightarrow ST \mid a; U \rightarrow US \mid b$
   (b) La grammaire suivante est-elle ambiguë ? $S \rightarrow aSSb \mid ab$
9. Transformer en forme normale de Greibach la grammaire suivante (qui est déjà non récursive gauche) :
   $S \rightarrow Aa \mid b$
   $A \rightarrow bdC \mid cC$
   $C \rightarrow cC \mid adC \mid c \mid ad$

10. Proposer une grammaire pour le langage consistant de toutes les chaînes contenant seulement des a et des b, et dont le nombre de a est différent du nombre de b.
11. Un groupe nominal est soit un déterminant suivi d'un nom, soit un groupe nominal coordonné avec un groupe nominal, soit un groupe nominal suivi d'un adjectif[1].
    (a) Ecrire la grammaire G du groupe nominal
    (b) Ecrire les deux arbres de dérivation que G associe au mot `D N E D N A` où les symboles `D`, `N`, `E`, `A` se lisent comme les symboles associés aux catégories « déterminant », « nom », « coordonnant », et « adjectif », respectivement.
    (c) Parmi les deux arbres de la question précédente, lequel choisiriez vous pour représenter la structure du groupe nominal "les femmes et les enfants malades" ?
    (d) Modifiez la grammaire G de sorte qu'elle n'associe au mot `D N E D N A` que l'analyse que vous avez identifiée dans la question précédente.
12. Soit le langage L_1 sur le vocabulaire $V = \{$*l', homme, ours, qui, a, vu*$\}$ formé de l'ensemble des phrases finies de la forme *l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu . . . qui a vu l'ours*.
    (a) Donner une grammaire algébrique (*context-free*) engendrant L_1.
    (b) Donner une grammaire régulière engendrant L_1.

    Soit L_2 le langage engendré par la grammaire $\mathcal{G}_2$:

S	→	NP Rel
NP	→	l'homme
	|	l'ours
Rel	→	qui a vu NP Rel
	|	qui a vu l'ours

    (c) Mettre la grammaire $\mathcal{G}_2$ sous forme normale de Chomsky.
    (d) Proposer une grammaire $\mathcal{G}_3$ qui engendre le langage L_3, sur-ensemble de L_2 mais dans lequel les symboles *ours* et *homme* sont **strictement** interchangeables.
    (e) Décrire informellement les différences entre les langages L_1 et L_2.
    (f) Donner une grammaire algébrique du langage $L_2 \setminus L_1$.
13. Mettre sous forme normale de Chomsky la grammaire définie par les règles de production suivantes
    $S \to AB \mid aS \mid a$
    $A \to Ab \mid \varepsilon$
    $B \to AS$
14. Eliminer la récursivité gauche de la grammaire ETF.
15. Appliquer l'algorithme de dé-récursivation (gauche) vu en cours à la grammaire suivante, grammaire de la liste.
    $S \to (\, L\,) \mid a$
    $L \to L, S \mid S$

[1] Cette règle n'est pas très orthodoxe, mais nous l'adopterons pour cet exercice.