1 Transformations de grammaire (encore)

1. Appliquer l'algorithme de dé-récursivation (gauche) à la grammaire suivante, grammaire de la liste.
   $S \to (\ L\ ) \mid a$
   $L \to L, S \mid S$
2. Dé-récursiver la grammaire suivante $S \to aS \mid BS \mid \varepsilon$
   $B \to bAb \mid SaS$
   $A \to a \mid Sa$

2 Automates à pile

1. Soit la grammaire de l'exercice 1.1 de la feuille 2 : la grammaire sur l'alphabet $X = \{+, =, a\}$ pour le langage L dont chaque mot représente une addition correcte de deux suites de caractères a. Par exemple L contient le mot $aa + aaaa = aaaaaa$. Donner un automate à pile qui reconnaît le même langage.
2. Donner la suite de configurations de l'automate A_1 (donné ci-dessous) qui correspond à la reconnaissance du mot $aabb$.

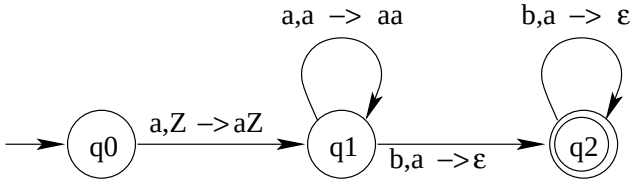

FIG. 1 – Un automate à pile pour $a^n b^n$ (fond de pile noté Z)

Soit l'automate A_2 ci-dessous, reconnaissant $a^i b^j c^k$ avec $i = j$ ou $i = k$. Donner l'ensemble des suites de configurations pour le mot $aabcc$, puis pour le mot $aabbc$.

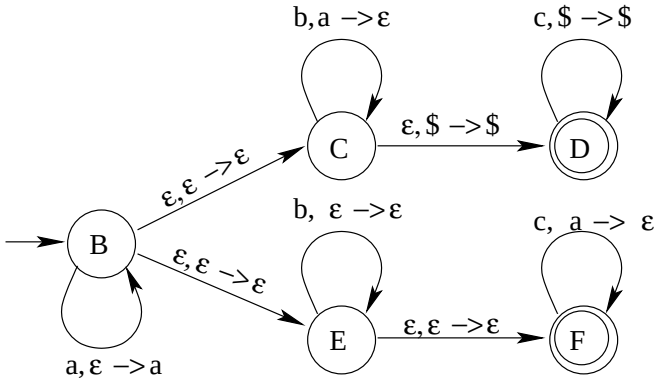

FIG. 2 – Automate à pile reconnaissant $a^i b^j c^k$ avec $i = j$ ou $i = k$ (fond de pile noté \$)

3. Trouver un automate à pile qui accepte le langage $\{w\overline{w} \,/\, w \in X^*\}$, où $\overline{w}$ désigne le mot miroir de w.
4. Proposer un automate à pile pour le langage $\{a^n b^p \,/\, 0 < n \leq p \leq 2n\}$.

5. Proposer un automate à pile pour le langage des mots sur $X = \{a, b\}$ qui contient autant d'occurrences de a que d'occurrences de b.
6. **Equivalences** des différents types de conditions d'arrêt. On redonne ci-dessous trois définitions pour les automates à pile, qui diffèrent uniquement dans les conditions d'acceptation de l'automate. **Prouver** que ces trois définitions sont équivalentes. (Pour cela définir un algorithme pour à partir d'un automate d'un des trois types, construire un automate d'un autre type, reconnaissant le même langage.

   Mixte Soit un automate A, défini par le sextuplet : $\langle Q, X, \Gamma, \delta, q_0, Z_0, F\rangle$:
   - Q est un ensemble fini d'états
   - X est l'alphabet d'entrée (souvent noté Σ)
   - Γ est l'alphabet de pile (pas nécessairement disjoint de X)
   - δ est l'application[1] de transition $\delta : Q \times (X \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \longrightarrow Q \times \Gamma^*$
   - $q_0 \in Q$ est l'état initial
   - $Z_0 \in \Gamma$ est le symbole de fond de pile
   - $F \subset Q$ est l'ensemble des états d'acceptation

   On définit le langage reconnu comme l'ensemble des chaînes reconnues en partant d'une configuration (état initial / le mot à reconnaître / pile réduite à Z_0) vers une configuration (un état d'acceptation / chaîne vide / pile réduite à Z_0).
   Soit formellement : le **langage reconnu** par A, noté $L(A)$ est l'ensemble des mots reconnus par A :

   $$L(A) = \{m \in X^* \,/\, (q_0, m, Z_0) \overset{*}{\vdash} (q, \varepsilon, Z_0) \text{ avec } q \in F\}$$

   Acceptation par état final Même définition formelle de l'automate, mais changement pour la formalisation du langage reconnu : on supprime la contrainte de pile réduite à Z_0 dans la configuration finale : il suffit que l'état atteint soit un état d'acceptation.
   I.e. formellement : le **langage reconnu** par A, noté $L(A)$ est l'ensemble des mots reconnus par A :

   $$L(A) = \{m \in X^* \,/\, (q_0, m, Z_0) \overset{*}{\vdash} (q, \varepsilon, \alpha) \text{ avec } q \in Fet\alpha \in \Gamma*\}$$

   Acceptation par pile vide La définition formelle est simpifiée en supprimant la notion d'état d'acceptation. On modifie la contrainte sur la configuration finale pour une chaîne appartenant au langage : peu importe le type d'état atteint, l'acceptation est atteinte lorsque la pile est vide (ϵ et pas pile réduite à Z_0).
   I.e. formellement : le **langage reconnu** par A, noté $L(A)$ est l'ensemble des mots reconnus par A :

   $$L(A) = \{m \in X^* \,/\, (q_0, m, Z_0) \overset{*}{\vdash} (q, \varepsilon, \varepsilon)\}$$

[1] i.e. pas nécessairement déterministe.