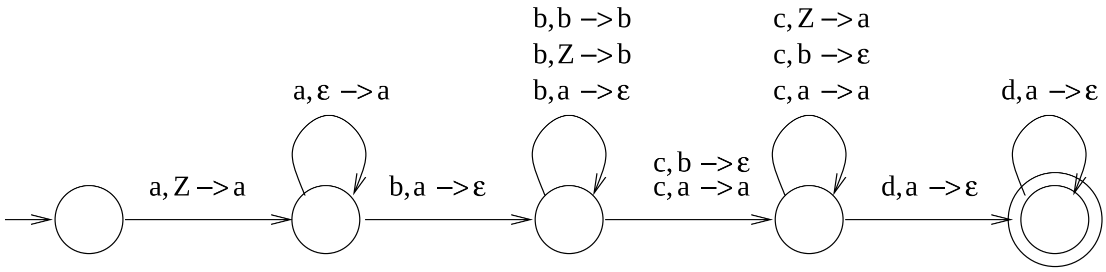
Contrôle continu Langages formels (LI324)
Aucun document autorisé.
Durée : 1 heure 30.

1. **Grammaires simples**
   Une grammaire algébrique $G = \langle X, V, S, P\rangle$ est dite *simple* si G vérifie les deux conditions :
   – $P \subset V \times XV^*$
   – $\forall A \in V, \forall x \in X, \forall u, u' \in V^*, ((A \to xu) \wedge (A \to xu') \Rightarrow (u = u'))$
   Un langage algébrique est un *langage simple* s'il existe un grammaire simple qui l'engendre.
   (a) Trouver une grammaire simple pour le langage $\{a^n b^{n+1}, n \geq 0\}$
   (b) Trouver une grammaire simple pour le langage $\{a^n b^n, n > 0\}$
   (c) Soit L le langage engendré par : $S \longrightarrow aSS \mid b$. Construire une grammaire algébrique qui engendre le langage Lc^*d.
   (d) Montrer que la concaténation de deux langages simples est un langage simple. On demande une explication rigoureuse, pas nécessairement une démonstration mathématique.
   (e) Donner un exemple simple de langage non simple.
2. **Automates à pile**
   Proposer un automate à pile qui reconnaît le langage
   $\{a^n b^p c^q d^r$ avec $n, p, q, r > 0$ et $n + q = p + r\}$
3. **Langages réguliers**
   Proposer un automate fini déterministe qui reconnaît le langage sur l'alphabet $X = \{a, b, c\}$ qui comprend tous les mots qui ne comprennent pas le motif ba^*cb. On pourra procéder en plusieurs étapes, qu'il conviendra d'ordonner soigneusement.

1. (a) $S \longrightarrow aSb \mid b$
   (b) $S \longrightarrow sSB \mid B$; $B \longrightarrow b$
   (c) $S \longrightarrow aSSX \mid bX$; $X \longrightarrow cX \mid d$
   (d) Soient L_1 et L_2 deux langages simples, engendrés par deux grammaires G_1 et G_2 simples. Soient S_1 et S_2 les axiomes respectifs de ces deux grammaires. Soient $S_1 \to u_1 \mid u_2 \dots u_k$ l'ensemble des règles d'origine S_1 dans G_1. Alors le langage L_1L_2 est engendré par une grammaire G, d'axiome S, comprenant les règles $S \to u_1S_2 \mid u_2S_2 \dots u_kS_2$, ainsi que les règles de G_1 d'origine autre que S_1, et toutes les règles de G_2.

   Toutes les règles qui proviennent des grammaires d'origine sont simples par hypothèse, et les nouvelles règles d'origine S sont simples car la concaténation d'un non terminal à droite d'un mot de XV^* est un mot de XV^*, et les préfixes ne sont pas modifiés. La grammaire proposée est donc simple.

   Il reste à démontrer que la grammaire ainsi définie engendre bien le langage L_1L_2, ce qui ne pose pas de difficulté.
   (e) Etant donnée la définition stricte donnée des grammaires simples, aucun langage comprenant ε n'est simple.
2. Idée : on empile les a rencontrés, puis on les dépile en rencontrant les b. Si on arrive au fond, on empile des b (qui représentent en quelque sorte un manque de a). Les c sont empilés sur les a, ou compensent les b, et enfin, les d doivent trouver des a dans la pile (en nombre égal).



3. Méthode : on crée l'automate (non déterministe) qui reconnaît $(a|b|c)^*ba^*cb(a|b|c)^*$. Puis on le déterminise, on le complète, et on prend son complément.