

Contrôle continu LI 032 EX
Devoir facultatif de rattrapage
Date limite : Vendredi 21 janvier 2005
(par courrier électronique, pas de format « propriétaire »)

On demande d'implémenter l'ensemble des algorithmes qui permettent, étant donnée une grammaire algébrique quelconque, de produire une grammaire équivalente, mais **propre**, c'est-à-dire ε -libre, sans cycle, sans symbole inutile :

- Une grammaire algébrique est dite ε -libre si
 - P n'a pas d' ε -production, **ou**
 - P a exactement une ε -production $S \rightarrow \varepsilon$, où S est l'axiome, et S est inaccessible, i.e. n'apparaît pas dans un membre droit de règle.
- Une grammaire algébrique est dite *sans cycle* si elle ne permet aucune dérivation $A \xrightarrow{+} A$ pour tout $A \in V$. En particulier, une grammaire sans production singulière est nécessairement sans cycle.
- Un symbole non-terminal X sera dit *inutile* si X est sans contribution, ou inaccessible.
 - Un symbole non-terminal X est dit *sans contribution* s'il n'existe pas de dérivation $X \xrightarrow{*} u$, $u \in X^*$.
 - Un symbole non-terminal X est dit *inaccessible* s'il n'existe pas de dérivation $S \xrightarrow{*} wXy$ ($w, y \in (X \cup V)^*$).

On demande de fournir le code de l'implémentation, *généreusement commenté*, ainsi qu'une trace de l'exécution pour (au moins) deux exemples de grammaire.
