

# Correction du QCM Compilation : Langages et Grammaires

EPITA – Promo 2006

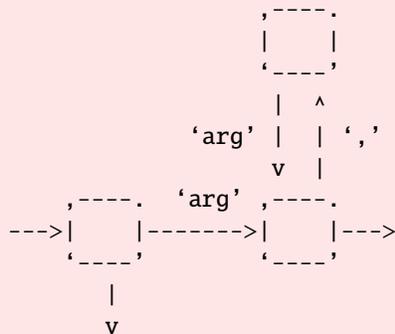
27 Novembre 2003

À quelle(s) classe(s) appartiennent chacune des grammaires suivantes ? (NB : si une grammaire est de type  $A$  et  $A \subset B$ , cocher  $A$  et  $B$ ).

- a. Régulière
- b. Hors Contexte
- c. Non ambiguë
- d. Engendre un langage rationnel
- e. Engendre un langage non vide

1.  $\langle \text{args} \rangle ::= \langle \text{args1} \rangle \mid \varepsilon$   
 $\langle \text{args1} \rangle ::= \langle \text{args1} \rangle \text{ , } \text{ 'arg' }$   
 $\quad \quad \quad \mid \text{ 'arg' }$

**Correction:** Linéaire à gauche, donc régulière. Elle est non ambiguë.  $\langle \text{args1} \rangle$  engendre une liste de une ou plusieurs *arg séparés* par des *,*. Donc, cette grammaire engendre une liste de zéro ou plusieurs *arg séparés* par des *,*. Ce langage est infini, régulier (puisque engendré par une grammaire régulière) : type 3. Il existe donc un automate, comme par exemple :



2.  $\langle \text{args} \rangle ::= \langle \text{args1} \rangle \mid \varepsilon$   
 $\langle \text{args1} \rangle ::= \langle \text{args1} \rangle \text{ , } \text{ 'arg' } \langle \text{args1} \rangle$   
 $\quad \quad \quad \mid \text{ 'arg' }$

**Correction:** Cette grammaire est très visiblement une version ambiguë de la grammaire précédente. On pourrait dire que dans la grammaire précédente l'opérateur *,* est associatif à gauche, ici il est associatif (i.e., à droite et à gauche) : une phrase comme *arg , arg , arg* peut se lire comme *(arg , arg) , arg* ou *arg , (arg , arg)*. Le langage, lui, reste évidemment de type 3, et reconnu par le même automate.

3.  $A \rightarrow a B$   
 $B \rightarrow A b \mid c$

**Correction:**  $a^n cb^n$ ... Ça aurait été étonnant qu'il ne nous la colle pas celle-là vu comme il est créatif... Pas régulière, hors-contexte, pas ambiguë, pas reconnaissable (ce langage est l'archétype du non rationnel). Et oui : non  $a^n cb^n$  n'est pas vide.

4.  $X \rightarrow x X Y Z$   
 $X \rightarrow X y Z$   
 $Z Y \rightarrow Y Z$   
 $y Y \rightarrow y y$   
 $y Z \rightarrow y z$   
 $z Z \rightarrow z z$

**Correction:** Cette grammaire est visiblement monotone, non hors contexte. On voit rapidement qu'il n'y a aucun moyen de se "débarrasser" de  $X$  : cette grammaire ne produit aucun mot ! Elle ne peut donc être ambiguë, et le langage qu'elle engendre,  $\emptyset$ , est bien sûr reconnaissable par un automate fini.