

# Compilation : Langages et Grammaires

EPITA – Promo 2005

Janvier 2003

Dans cette épreuve, les non-terminaux sont écrits en majuscules, les terminaux en minuscules ou entre guillemets, et  $\varepsilon$  désigne le mot vide. Lorsque l'on demande le type (de Chomsky) d'une grammaire ou d'un langage on demande **évidemment** le type le plus précis. On considère la hiérarchie de Chomsky « non stricte », c'est-à-dire que la grammaire  $S \rightarrow \varepsilon$  est considérée hors-contexte.

Les résultats ne doivent pas être balancés comme « évidents », sous peine de disqualification pour tentative de bluffage. Néanmoins, une argumentation informelle mais convaincante, sera souvent suffisante.

Une copie synthétique (= sans verbiage ou mauvaise traduction d'Appel), bien orthographiée, dont les résultats sont clairement affichés, sera toujours mieux notée qu'une copie qui aura demandé une quelconque forme d'effort de la part du correcteur.

## 1 Hiérarchie de Chomsky

Pour chacune des grammaires suivantes, préciser (i) son type dans la hiérarchie de Chomsky, (ii) si elle est ambiguë, (iii) le langage qu'elle engendre, (iv) le type du langage dans la hiérarchie, (v) un automate fini déterministe (sous forme d'un diagramme) qui reconnaisse le même langage.

Justifier vos réponses.

1.  $P \rightarrow P \text{ inst } ' ; '$   
 $P \rightarrow \varepsilon$
2.  $P \rightarrow P1$   
 $P \rightarrow \varepsilon$   
 $P1 \rightarrow P1 ' ; ' \text{ inst}$   
 $P1 \rightarrow \text{inst}$
3.  $P \rightarrow P1$   
 $P \rightarrow \varepsilon$   
 $P1 \rightarrow P1 ' ; ' P1$   
 $P1 \rightarrow \text{inst}$
4.  $S \rightarrow P$   
 $P \rightarrow p P Q R$   
 $P \rightarrow p q R$   
 $R Q \rightarrow Q R$   
 $q Q \rightarrow q q$   
 $q R \rightarrow q r$   
 $r R \rightarrow r r$

## 2 Parsage LL(1)

Soit le langage de la logique (dite propositionnelle) composée de deux symboles  $t$  (vrai) et  $f$  (faux), de l'opération unaire  $\neg$  (non), des opérations binaires  $\vee$  (ou) et  $\wedge$  (et), et des parenthèses. Ce langage inclut des mots tels que :

$$t \wedge t \quad t \vee f \quad \neg\neg(t \wedge t) \vee (f \wedge f)$$

Noter que l'on parle de tout ce langage, et non seulement du sous-langage des formules « vraies » ; il comprend donc aussi des mots tels que :

$$f \quad \neg t \quad \neg t \wedge f \wedge f \wedge t$$

1. Écrire une grammaire hors contexte naïve de la logique propositionnelle. On cherchera une formulation abstraite, courte et très lisible, au prix de l'ambiguïté.
2. Désambigüiser cette grammaire en considérant les règles suivantes :
  - (a)  $\wedge$  et  $\vee$  sont associatives à gauche ;
  - (b)  $\neg$  est prioritaire sur  $\wedge$  ;
  - (c)  $\wedge$  est prioritaire sur  $\vee$ .c'est-à-dire que  $f \vee t \vee f \wedge \neg\neg t \wedge f$  se lit  $(f \vee (t \vee ((f \wedge (\neg(\neg t))) \wedge f)))$ .
3. Expliquer pourquoi cette grammaire ne peut pas être LL(1).
4. Transformer cette grammaire en une grammaire susceptible d'être LL(1).
5. Quelle critique formuler sur la grammaire obtenue ?
6. Récrire cette grammaire en s'autorisant les extensions de l'EBNF. Par exemple,  
 $A \rightarrow a^*$ .
7. Écrire en pseudo code un parseur LL(1) avec les bonnes priorités et associativités pour cette grammaire. Il suffira d'écrire une et une seule des routines de parsage, à condition qu'elle soit significative (comprendre que eat, aussi appelée accept, n'est pas demandée).