

Correction du Partiel Compilation : Langages et Grammaires

EPITA – Promo 2005

Mars 2003

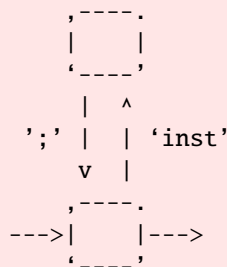
1 Hiérarchie de Chomsky

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).

1. $P \rightarrow P \text{ inst } ;'$
 $P \rightarrow \varepsilon$

- a. Régulière
- b. Hors Contexte
- c. Ambigüe
- d. Reconnaisable par un automate fini déterministe
- e. Produit un langage non vide

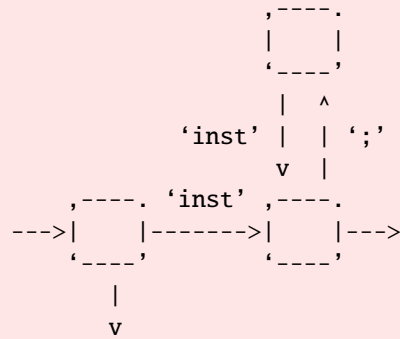
Correction: Linéaire à gauche, donc régulière. Elle est non ambigüe, et engendre une suite de zéro ou plusieurs *inst terminés* par des ;'. Ce langage est infini, régulier (puisqu'engendré par une grammaire régulière) : type 3. Il existe donc un automate, comme par exemple :



2. $P \rightarrow P1$
 $P \rightarrow \varepsilon$
 $P1 \rightarrow P1 ;' \text{ inst}$
 $P1 \rightarrow \text{inst}$

- a. Régulière
- b. Hors Contexte
- c. Ambigüe
- d. Reconnaisable par un automate fini déterministe
- e. Produit un langage non vide

Correction: Linéaire à gauche, donc régulière. Elle est non ambiguë. P1 engendre une liste de une ou plusieurs *inst séparés* par des ;. Donc, cette grammaire engendre une liste de zéro ou plusieurs *inst séparés* par des ;. Ce langage est infini, régulier (puisque engendré par une grammaire régulière) : type 3. Il existe donc un automate, comme par exemple :



3. $P \rightarrow P1$
 $P \rightarrow \varepsilon$
 $P1 \rightarrow P1 \text{ ' ; ' } P1$
 $P1 \rightarrow \text{inst}$
 - a. Régulière
 - b. Hors Contexte
 - c. Ambigüe
 - d. Reconnaisable par un automate fini déterministe
 - e. Produit un langage non vide

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

4. $S \rightarrow P$
 $P \rightarrow p P Q R$
 $P \rightarrow p q R$
 $R Q \rightarrow Q R$
 $q Q \rightarrow q q$
 $q R \rightarrow q r$
 $r R \rightarrow r r$
 - a. Régulière
 - b. Hors Contexte
 - c. Ambigüe
 - d. Reconnaisable par un automate fini déterministe
 - e. Produit un langage non vide

Correction: Cette grammaire est visiblement monotone, non hors contexte. Bien qu'il ne soit pas simple de le montrer formellement, une "exécution" de cette grammaire à la main montre qu'elle n'est pas ambiguë.

On reconnaît l'exemple de grammaire engendrant $a^n b^n c^n$, i.e., le langage des mots commençant par un certain nombre (non nul) de a, puis d'autant de b, et enfin autant de c. Mais avec p, q et r. Ce langage est bien connu comme l'exemple type des langages sensibles au contexte (et non hors-contexte), comme vu en cours.

Bien entendu, il est impossible de trouver un automate fini (le langage n'est pas régulier), ni même un d'automate à pile (le langage n'est pas hors-contexte).

2 Compilation

1. Le métier d'un parser est de :
 - a. segmenter un flux de caractères en un flux de tokens
 - b. s'assurer que les types sont bien utilisés
 - c. faire de l'imagerie médicale
 - d. éliminer les récursions terminales
 - e. faire de l'analyse syntaxique
2. Le langage Tiger dispose de :
 - a. la structure de blocs
 - b. des types génériques
 - c. des fonctions polymorphes définissables par l'utilisateur
 - d. des constructeurs de types pointeurs
 - e. une gestion des exceptions
3. En compilation, AST signifie
 - a. Adjunction of Sub Types
 - b. Abstract Syntax Tree
 - c. Atlantic Standard Time
 - d. for All Such That
 - e. Absolutely Strict Typing
4. Quelles fonctionnalités pouvez ajouter à votre compilateur Tiger sans devoir changer le backend :
 - a. Support d'un type `time` et calcul de l'Atlantic Standard Time
 - b. Inlining
 - c. Surcharge des fonctions (*overloading*)
 - d. Compilation vers IA-64
 - e. Chargement dynamique de modules
5. La classe `Symbol` permet :
 - a. économie de mémoire
 - b. comparaisons rapides
 - c. calcul numérique symbolique
 - d. de reconnaître les mots-clés
 - e. de représenter la ponctuation