

TD6 - Grammaires (2)

Exercice 1

Déterminer les langages engendrés par les grammaires suivantes.

Question 1. $S \rightarrow AB \quad A \rightarrow a \quad B \rightarrow b$

Question 2. $S \rightarrow AB \quad A \rightarrow aA \mid a \quad B \rightarrow bB \mid b$

Exercice 2

Soit la grammaire non contextuelle définie par les règles de production suivantes.

$$E \rightarrow E + E \mid E \times E \mid \text{id}$$

Question 1. Écrire les dérivations gauche et droite de : $\text{id} + \text{id} \times \text{id}$.

Question 2. Cette grammaire est-elle ambiguë ?

Exercice 3

Déterminer la ou les grammaires ambiguës par les suivantes.

Question 1. $S \rightarrow SS \mid aSb \mid bSa \mid \varepsilon$

Question 2. $S \rightarrow aSbS \mid bSaS \mid \varepsilon$

Question 3. $S \rightarrow aAB \quad A \rightarrow bBb \quad B \rightarrow A \mid \varepsilon$

Exercice 4

Question 1. Qu'est la mise en forme normale de Chomsky ?

Question 2. Déterminer une CNF pour la grammaire suivante.

$$S \rightarrow AaB \mid b \quad A \rightarrow S \mid \varepsilon \mid AB \quad B \rightarrow bbb \mid ASA$$

Exercice 5

On considère le fragment simplifié de la grammaire d'un langage de programmation hypothétique.

$$\begin{aligned} \textit{Instr} &\rightarrow \textit{foo} \mid \textit{bar} \mid \textit{qux} \mid \textit{Cond} \mid \textit{begin InstrList end} \\ \textit{Cond} &\rightarrow \textit{if Expr then Instr else Instr} \mid \textit{if Expr then Instr} \\ \textit{InstrList} &\rightarrow \textit{Instr} \mid \textit{Instr InstrList} \\ \textit{Expr} &\rightarrow \textit{true} \mid \textit{false} \mid \textit{happy} \mid \textit{trippy} \end{aligned}$$

Les *lettres* ou *tokens* y sont écrits comme des mots ; par exemple *foo* est une *lettre*. Les *terminaux* sont écrits en police à espacement fixe ; les *non-terminaux* sont en italique et commencent par une majuscule. L'axiome est *Instr*.

Question 1. Donner l'arbre de dérivation du *mot* suivant.

`if happy then if trippy then foo else bar else qux`

Pourquoi n'en existe-t-il qu'un seul ?

Question 2. Donner deux arbres de dérivation distincts de `if happy then if trippy then foo else bar`. Que dire de la grammaire présentée ?

Question 3. En supposant que, dans ce langage, `begin I end`, où *I* est une instruction, a le même effet que *I* seul, comment un programmeur peut-il récrire l'instruction considérée à la question précédente pour obtenir un comportement équivalent à l'une ou l'autre des deux interprétations ?

Question 4. Modifier la grammaire proposée pour obtenir une grammaire faiblement équivalente dans laquelle seul l'un des arbres de dérivation obtenus à la question 2. est possible (i.e., une grammaire qui force cette interprétation là par défaut). On pourra être amené à introduire des nouveaux non-terminaux pour des variantes de *Inst* et *Cond* qui interdisent récursivement les conditionnelles sans *else*.