TD 05 – Grammaires ambiguës, factorisation, récursivité gauche

Exercice 1. *Derivation*

Soit la grammaire suivante :

$$\begin{array}{lcl} S & \longrightarrow & (\ L\ )\ S \mid a \\ L & \longrightarrow & L\ ,\ S \mid S \end{array}$$

(a) Quels sont les symboles non terminaux et les symboles terminaux ?

(b) Les mots « $(\ a\ )\ a$ », « $(\ a\ )\ (\ a\ )$ » et « $(\ a\ ,\ a\ )\ a$ » sont-ils engendrés par la grammaire ?

(c) Donnez les dérivations gauche et droite du mot « $(\ a\ ,\ (\ a\ ,\ a\ )\ a\ )\ a$ ».

Exercice 2. *Grammaires ambigües*

Soit la grammaire des expressions logiques suivantes :

$$S \quad \longrightarrow \quad 0 \mid 1 \mid S \vee S \mid S \wedge S \mid (S)$$

1. Quels sont les symboles non terminaux et les symboles terminaux ?
2. Donnez deux arbres de dérivations pour l'expression $1 \vee 1 \wedge 0$.
3. Est-ce que cette grammaire est ambiguë ? Dans ce cas particulier, quelle pourrait être une conséquence si un compilateur utilisait cette grammaire pour le calcul d'expression logique ?
4. En sachant que l'opérateur $\wedge$ est prioritaire sur l'opérateur $\vee$, proposez une grammaire non ambiguë équivalente.
5. Donner l'arbre de dérivation (normalement unique...) du mot $1 \vee 1 \wedge 0$ avec votre nouvelle grammaire.

Exercice 3. *Ambigüité (2)*

Soit la grammaire :

$$P \quad \longrightarrow \quad \epsilon \mid (\ P\ ) \mid P\ P$$

1. Quel est le langage défini par cette grammaire ?
2. Montrez qu'elle est ambigüe.
3. Proposez une grammaire équivalente qui ne l'est pas.

Exercice 4. *Factorisation.*

Définition. Une grammaire G, avec N (resp. T) l'ensemble de ses symboles non-terminaux (resp. terminaux), est *non factorisée* si elle admet deux règles, utiles et distinctes, de la forme suivante :

$$\begin{array}{lcl} X & \longrightarrow & \alpha\, \beta \\ X & \longrightarrow & \alpha\, \gamma, \end{array}$$

où α, β et γ appartiennent à $(N \cup T)^*$ et où α ne se dérive pas en ϵ.

N.B. Des règles de la forme $A \longrightarrow \alpha\,\beta \mid \alpha \mid \gamma$, dont une partie droite est préfixe d'une autre, peuvent être modifiées (factorisées) facilement en $A \longrightarrow \alpha\,A' \mid \gamma$, $A' \longrightarrow \beta \mid \epsilon$.

Soit la grammaire suivante :

$$
\begin{array}{lcl}
A & \longrightarrow & a\,B\,c \mid a\,C\,B\,d \\
B & \longrightarrow & a\,A\,b\,c \mid c\,b \mid \epsilon \\
C & \longrightarrow & b\,C\,a\,A\,b\,c \mid b\,C\,a\,B
\end{array}
$$

1. Cette grammaire est-elle factorisée ?
2. Si la réponse est négative, quelles sont la(es) règle(s) qu'il conviendrait de factoriser pour la simplifier ? Opérez les opérations de factorisation adéquates.
3. Soit maintenant un grand classique, avec l'échantillon de grammaire suivant :

$$
\begin{array}{lcl}
\textit{Instruction} & \longrightarrow & \text{if } \textit{Condition Instruction} \\
 & & \mid \text{if } \textit{Condition Instruction} \text{ else } \textit{Instruction} \\
 & & \mid \text{while } \textit{Condition Instruction}
\end{array}
$$

   Que peut-on dire de ce morceau de grammaire ? Comment l'« arranger » ?

Exercice 5. *Récursivité à gauche.*

Définition. Une grammaire G, avec N (resp. T) l'ensemble de ses symboles non-terminaux (resp. terminaux), est *récursive gauche directe* si elle admet une règle de la forme suivante :

$$X \longrightarrow X\,\alpha,$$

où α appartient à $(N \cup T)^*$.

N.B. On passe facilement d'une récursivité gauche directe à une récursivité droite directe (et inversement). En effet, si on a $A \longrightarrow A\,\alpha \mid \beta$, on voit que $A \stackrel{+}{\Longrightarrow} \beta\,\alpha^*$. Et cette dérivation peut être produite par $A \longrightarrow \beta\,A'$, $A' \longrightarrow \alpha\,A' \mid \epsilon$.

Soit la grammaire suivante, permettant d'écrire des expressions arithmétiques du type $id + id + \ldots + id$:

$$
\begin{array}{lcl}
E & \longrightarrow & E\,+\,T \mid T \\
T & \longrightarrow & \text{id}
\end{array}
$$

(a) Supprimez la récursivité gauche directe de cette grammaire.

Considérons à présent une nouvelle grammaire, notée G_1, définissant un langage d'expressions arithmétiques plus complexes :

$$E \longrightarrow E\,+\,E \mid E\,\times\,E \mid (\,E\,) \mid \text{id}$$

1. La grammaire G_1 est-elle ambigüe ?
2. Proposez une grammaire équivalente qui ne serait pas ambigüe.
3. Est-ce que votre nouvelle grammaire est sans récursivité gauche directe et factorisée ? Trouvez une grammaire équivalente factorisée et sans récursivité gauche directe tout en restant non ambiguë.

Exercice 6. *Langage $\mathcal{L}_1 = \{a^*b\}$*

Soit l'alphabet $\Sigma = \{a, b\}$ et le langage $\mathcal{L}_1 = \{a^*b\}$ sur Σ. Écrivez une grammaire de $\mathcal{L}_1$.

Exercice 7. *Langage* $\mathcal{L}_2 = \{a^n b^n \mid n \in \mathbb{N}\}$

Soit l'alphabet $\Sigma = \{a, b\}$ et le langage $\mathcal{L}_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ sur Σ. Écrivez une grammaire de $\mathcal{L}_2$.

Exercice 8. *Langage* $\mathcal{L}_3 = \{a^n b^p \mid n > p$ *où* $(n, p) \in \mathbb{N} \times \mathbb{N}\}$

Soit l'alphabet $\Sigma = \{a, b\}$ et le langage $\mathcal{L}_3 = \{a^n b^p \mid n > p$ où $(n, p) \in \mathbb{N} \times \mathbb{N}\}$ sur Σ. Écrivez une grammaire de $\mathcal{L}_3$.