

## Langages Formels

**Exercice 1 (Sinon quoi?)**

On considère la grammaire suivante :

$$\begin{aligned} S &\rightarrow \text{if } c \text{ then } S \text{ else } S \\ S &\rightarrow \text{if } c \text{ then } S \\ S &\rightarrow a \end{aligned}$$

Montrer qu'elle est ambiguë mais que le langage engendré ne l'est pas.

**Exercice 2 (Applications du lemme d'Ogden)**

1. Montrer que le langage  $L_{\text{cross}} \stackrel{\text{def}}{=} \{a^n b^m c^n d^m \mid n, m \geq 0\}$  n'est pas algébrique.
2. Montrer que le langage  $L_{\text{copy}} \stackrel{\text{def}}{=} \{ww \mid w \in \{a, b\}^*\}$  n'est pas algébrique.
3. Montrer que le langage  $\{a^n b^n c^p d^p \mid n, p \geq 0\}$  est algébrique mais pas linéaire.
4. Montrer que le langage  $\{a^n b^n c^m \mid n, m \geq 0\} \cup \{a^n b^m c^m \mid n, m \geq 0\}$  est algébrique et inhéremment ambigu. On pourra appliquer le lemme aux mots  $a^{K+K!} b^K c^K$  puis  $a^K b^K c^{K+K!}$  et exhiber deux arbres de dérivations différents pour le mot  $a^{K+K!} b^{K+K!} c^{K+K!}$ .

**Exercice 3 (Langage de programmation [Floyd, *Comm. ACM* 5(9), 1962])**

La première utilisation des grammaires algébriques pour décrire la syntaxe d'un langage de programmation a été faite pour le langage ALGOL 60. Néanmoins, la modélisation par une grammaire algébrique seule n'est pas possible : le programme suivant

```
begin
  real x;
  y := 10
end
```

n'est correct que si les identifiants  $x$  et  $y$  coïncident. Cependant un identifiant ALGOL 60 est une chaîne arbitrairement longue du langage rationnel  $\ell(\ell + c)^*$  où  $\ell$  désigne n'importe quelle lettre majuscule ou minuscule de l'alphabet latin, et  $c$  n'importe quel chiffre de 0 à 9.

Montrer que ce niveau de correction ne peut pas être assuré par une grammaire algébrique seule.

**Exercice 4 (Langage naturel [Shieber, *Lingu. Phil.* 8(3), 1985])**

Un exemple de phénomène syntaxique non exprimable par une grammaire algébrique est issu du Suisse-Allemand :

*Jan säit das mer d'chind em Hans es huus haend wele laa hülfe aastrüche*  
*Jan dit que nous les enfants-ACC Hans-DAT la maison-ACC avons voulu laisser aider peindre*  
 « *Jan dit que nous avons voulu laisser les enfants aider Hans à peindre la maison* »

Dans cette phrase, on peut distinguer

- des syntagmes nominaux à l'accusatif, comme *d'chind*,
- des syntagmes nominaux au datif, comme *em Hans*,
- des verbes recevant exactement un argument accusatif, comme *laa*, et
- des verbes recevant exactement un argument datif, comme *hülfe*.

De plus, cet exemple est productif : on peut continuer à ajouter des subordinées comme *em Hans... hülfe* et *d'chind... laa* de manière arbitraire.

Montrer qu'il n'existe pas de grammaire algébrique pour le Suisse-Allemand.

**Exercice 5 (Lemme d'Ogden)**

Le but de cet exercice est de prouver le lemme d'Ogden :

**Lemme 1** (Lemme d'Ogden). *Soit  $G = \langle \Sigma, V, P, S \rangle$  une grammaire algébrique. Il existe un entier  $N$  tel que pour tout  $x \in V$  et  $w \in \widehat{L}_G(x)$  contenant au moins  $N$  lettres distinguées, il existe  $y \in V$  et  $\alpha, u, \beta, v, \gamma \in (V \cup \Sigma)^*$  tels que :*

- (a)  $w = \alpha u \beta v \gamma$  ;
- (b)  $x \xrightarrow{*} \alpha y \gamma$ ,  $y \xrightarrow{*} u y v$ , et  $y \xrightarrow{*} \beta$  ;
- (c)  $u \beta v$  contient moins de  $N$  lettres distinguées ;
- (d) ( $\alpha$  et  $u$  et  $\beta$ ) ou ( $\beta$  et  $v$  et  $\gamma$ ) contiennent des lettres distinguées.

1. Soit un arbre  $t$  où certaines feuilles sont distinguées. On dit qu'un noeud est *distingué* lorsque le sous-arbre dont il est racine contient des feuilles distinguées, et *spécial* lorsqu'il a au moins deux fils distingués.

Montrer que si  $t$  est de degré au plus  $m$ , a  $k$  feuilles distinguées, et que chaque branche contient au plus  $r$  noeuds spéciaux, alors  $k \leq m^r$ .

2. Montrer le lemme d'Ogden.