UFR d'Informatique
L3 Informatique et Math-Info

Grammaires et analyse syntaxique
Année 2021-2022

TD n°3

Langages algébriques et grammaires

Les exercices avec une étoile sont à faire éventuellement chez vous.

Exercice 1 *Soit G la grammaire donnée par les règles suivantes, où les capitales sont des non-terminaux, les autres caractères sont des terminaux, et S est l'axiome de G.*

$$S \to F \qquad F \to a$$
$$S \to (F + S)$$

1. *Donner une dérivation gauche pour les mots $(a+a)$ et $(a+(a+a))$. Donner une dérivation droite pour le mot $(a+a)$.*
2. *Est-ce que les mots $((a+a)+a)$ et $(a+a))$ sont dans le langage engendré par G ?*
3. *Donner les arbres de dérivation pour les mots $(a+a)$ et $(a+(a+a))$.*

Exercice 2 *Décrire les langages engendrés par les grammaires suivantes (S : axiome, capitales : non terminaux, autres : terminaux) :*

1. $S \longrightarrow \varepsilon \mid aSa \mid bSb \mid a \mid b$
2. $S \longrightarrow \varepsilon \mid [S]S$
3. $S \longrightarrow \varepsilon \mid aS \mid aSbS$

Est-ce que ces grammaires sont ambiguës ou pas ?

Exercice 3 *Montrer que chacun des langages suivants est algébrique en donnant une grammaire qui le reconnaît :*

1. $L_1 = \{a^n b^n \mid n \geq 0\}$;
2. $L_2 = \{a^n b^m \mid m \geq n \geq 0\}$;
3. (*) $L_3 = \{a^n b^* c^n \mid n \geq 0\}$;
4. $L_4 = \{a^n b^m c^k \mid n = m + k\}$;
5. (*) $L_5 = \{a^n b^m c^m d^n \mid n, m \geq 0\}$;
6. $L_6 = \{w \mid w \text{ est un palindrome}\}$.
7. $L_7 = \{w \mid w \text{ n'est pas un palindrome}\}$.

À noter : *il n'est pas vrai en général que le complémentaire d'un langage algébrique soit algébrique.*

L'un de ces langages est-il rationnel ?

Exercice 4 *La grammaire suivante engendre le langage $\{a^n b^n c^m \mid m, n \geq 0\} \cup \{a^m b^n c^n \mid m, n \geq 0\}$.*

$$
\begin{aligned}
S &\to UV \mid XY \\
U &\to \varepsilon \mid aUb \\
V &\to \varepsilon \mid cV \\
X &\to \varepsilon \mid aX \\
Y &\to \varepsilon \mid bYc
\end{aligned}
$$

Montrer que cette grammaire est ambiguë.

Exercice 5 *(*) On considère la grammaire suivante qui engendre des expressions polonaises inverses (S : axiome, E : non terminal, i et v : terminaux) :*

$$
\begin{aligned}
&S \to E \\
&E \to EE+ \mid EE* \mid EE- \mid EE/ \\
&E \to v \mid i
\end{aligned}
$$

Donner une dérivation gauche, une dérivation droite et un arbre de dérivation pour $i\,v+i\,*$.

Exercice 6 *(facultatif)*

Utilisez la forme de Backus-Naur **non étendue** *qui correspond à la déclaration (avec ou sans initialisation) d'un attribut de type objet en Java : On se limitera au cas où l'initialisation se fait avec un* `new` *ou une affectation à partir d'un autre attribut. Les arguments du constructeur seront uniquement des constantes de type* `int` *ou* `String` *ou des variables.*

Par exemple, devront être reconnues les déclarations suivantes :

```
Object obj = new Bidule(12, "abc", n);
private static Bidule bid;
public Truc tr = obj;
```

On a les terminaux id (pour les identifiants), entier pour les valeurs entières, chaine pour les chaînes de caractères (`"abc"` *par exemple).*

Utilisez maintenant la forme de Backus-Naur étendue pour faire la même chose.