

Travaux Dirigés d'analyse syntaxique n°3

Licence d'informatique

Grammaires algébriques

L'objectif de ce TD est

- de réaliser un script shell de déploiement de tests,
 - d'interfacer **flex** et **bison** dans le cas où on utilise des terminaux symboliques,
 - de tracer les numéros de lignes depuis **flex** jusqu'à **bison** et de les inclure dans les messages d'erreur,
 - de supprimer la récursivité à gauche d'une grammaire algébrique,
 - de détecter l'ambiguïté.
-

► Exercice 1. *Déploiement de tests*

1. Modifiez le programme **bison** du TD 2, exercice 2, pour que la valeur renvoyée par le **main** soit 1 quand le mot est reconnu par la grammaire, et 0 sinon. Testez votre analyseur sur plusieurs fichiers.
2. Créez un script **bash** qui teste l'analyseur sur un dossier de fichiers de tests, et écrit les valeurs de retour dans un rapport de tests.

► Exercice 2. *Terminaux symboliques*

Faites un analyseur lexical et un analyseur syntaxique pour la grammaire d'expressions booléennes suivante :

$$\left\{ \begin{array}{l} E_b \rightarrow E_b \text{ ou } T_b \mid T_b \\ T_b \rightarrow T_b \text{ et } F_b \mid F_b \\ F_b \rightarrow \text{non } F_b \mid (E_b) \mid \text{vrai} \mid \text{faux} \end{array} \right.$$

Dans l'analyseur lexical, les lexèmes sont **ou** , **et** , **non** , **vrai** , **faux** et les parenthèses. Il doit pouvoir y avoir autant d'espace blanc qu'on veut entre deux lexèmes, y compris des sauts de lignes. Adaptez votre **makefile** et testez sur quelques fichiers contenant chacun une expression booléenne.

► Exercice 3. *Traçage des numéros de lignes*

Complétez les programmes **flex** et **bison** de l'exercice 2 pour que les messages d'erreur de syntaxe donnent le numéro de ligne de l'erreur dans le fichier d'entrée. Testez avec un fichier qui occupe plusieurs lignes, puis avec un fichier contenant une erreur.

► **Exercice 4. Récursivité à gauche**

On donne la grammaire suivante :

$$B \rightarrow (B) \mid B \text{ et } B \mid B \text{ ou } B \mid \text{non } B \mid \text{vrai} \mid \text{faux}$$

1. Quelles règles de cette grammaire créent une récursivité à gauche ?
2. Donnez une grammaire équivalente qui ne soit pas récursive à gauche.

► **Exercice 5. Ambiguïté et Bison**

1. Adaptez l'analyseur de l'exercice 1 à la grammaire suivante :

$$S \rightarrow aS \mid Sa \mid aaS \mid aSa \mid b$$

La commande `bison` fait-elle des avertissements ? la grammaire est-elle ambiguë ?
trouvez-vous des mots du langage que l'analyseur produit par `bison` rejette ?

2. Proposez une grammaire pour le langage $\{a^n b^m \mid n \leq m \leq 2n\}$
3. Adaptez l'analyseur à cette nouvelle grammaire. Mêmes questions.

► **Exercice 6. Grammaires ambiguës**

Prouvez que chacune des grammaires suivantes est ambiguë :

$$1) \begin{cases} P \rightarrow SbS \\ S \rightarrow \varepsilon \mid Sa \mid Sb \end{cases} \quad 2) \begin{cases} S \rightarrow \text{if } C \text{ then } S \text{ else } S \mid \text{if } C \text{ then } S \mid \text{instructions} \\ C \rightarrow \text{conditions} \end{cases}$$

$$3) B \rightarrow (B) \mid B \text{ et } B \mid B \text{ ou } B \mid \text{non } B \mid \text{vrai} \mid \text{faux}$$