Analyse Syntaxique et Compilation
TD10 de la semaine du 8 au 14 Décembre 2000

Travaux pratiques en Yacc

Exercice 1.

En utilisant les pages :
`http://dept-info.labri.u-bordeaux.fr/~ cori/Compil/etf1.y,`
`etf2.y, etfcode.y, etf.lex`
réaliser les fonctions suivantes :

1. Utliser `yacc` avec l'option `-v` pour construire l'automate d'analyse syntaxique ascendante de la grammaire E, T, F. Examiner le fichier `y.output` et comparer avec ce qui a été calculé en cours.

2. Ecrire un programme en `Yacc` qui compte le nombre de règles utilisées pour engendrer un mot dans la grammaire E, T, F.

3. Dans `etf2.y`, ajouter des opérations de soustraction et de division à la grammaire et obtenir le résultat en flottant d'une expression qui ne comporte que des nombres et les 4 opérations.

4. On se propose d'écrire différemment le programme `etf.code` : au lieu d'utiliser un modificateur de numéro de registre on s'arrange pour garder en mémoire des informations qui permettent de générer (avec des appels à `printf`) les instructions au fur et à mesure de la reconnaissance des règles. Pour cela on utilisera une variable `numregistre` qui est initialisée à 1, qui est incrémentée quand on reconnaît la règle $F \to a$ et décrémentée lorsque l'on reconnaît $E \to E + T$ ou $T \to T * F$. La génération d'une ligne de code s'effectuera chaque fois que l'on reconnaît l'une de ces trois règles, le numéro du registre correspondant sera déterminé grace à `numregistre` et aux valeurs sémantiques associées aux non terminaux (*i. e.* les variables $$ et $1, $2, $3).

Exercice 2

On utilise ici la grammaire suivante qui engendre des systèmes de parenthèses.

$$S \to aSbS \qquad S \to aSb$$

$$S \to abS \qquad S \to ab$$

Pour toute lettre x, on note $|f|_x$ le nombre de x qui apparaissent dans le mot f.

On remarque que le mot f est un système de paranthèses si $|f|_a = |f|_b$ et si pour tout g tel que $f = gh$ on a $|g|_a \geq |g|_b$. On appelle hauteur d'un mot g la quantité $|g|_a - |g|_b$.

Question 1. On demande d'écrire un programme `Yacc` qui traduit chaque mot f engendré par cette grammaire en une suite d'entiers qui représentent les hauteurs de tous les facteurs gauches de f.

On considère le langage L_1 des systèmes de parenthèses imbriquées sur l'alphabet $A = \{a, b, c, d\}$ où chaque a est fermé par un b et chaque c est fermé par un d. Ainsi a et b sont assimilés à des parenthèses ouvrantes et fermantes et c et d à des crochets ouvrants et fermants. Le mot $acdabbcd$ appartient à L_1 mais $acbd$ n'y appartient pas. Le mot vide n'appartient pas à L_1.

Question 2. On demande de donner une grammaire de L_1.

A tout mot f de $\{a, b, c, d\}^*$ on associe un couple d'entiers $H_2(f) = (x, y)$ où $x = |f|_a - |f|_b$ et $y = |f|_c - |f|_d$.

Question 3. Modifier effectuer sur le programme Yacc que vous avez donné plus haut pour qu'il admette en entrée un mot de L_1 et qu'il affiche les valeurs de $H_2(f)$ pour tous les facteurs gauches de f. Pour le mot $f = aabccddb$ l'affichage doit être :

$$(1, 0), (2, 0)(1, 0), (1, 1)(1, 2), (1, 1), (1, 0), (0, 0)$$