TP n° 2
FLex et Bison

1 Introduction

L'outil yacc, un générateur automatique d'analyseurs syntaxiques basés sur la méthode LALR(1), est mis à la disposition du développeur qui n'a désormais pour charge que l'écriture de la grammaire.
De plus, il accepte même des schémas de traduction en entrée et permet alors de générer automatiquement l'implémentation en langage C du traducteur spécifié.

2 Exercice

L'objectif de cet exercice est d'utiliser, dans un premier temps, l'outil yacc pour générer l'implémentation en langage C d'un analyseur syntaxique pour les déclarations du langage Pascal décrites par la grammaire ci-dessous. Par la suite, on étendra la grammaire en schéma de traduction dont l'implémentation en langage C sera également confiée à yacc.

$$\begin{aligned}
&\textit{declarations} \rightarrow \textit{declarations declarations} \mid \textit{liste : type ;}\\
&\textit{liste} \rightarrow \textit{liste , liste} \mid \textit{id}\\
&\textit{type} \rightarrow \textit{byte} \mid \textit{word} \mid \text{int}\,\textit{eger} \mid \textit{array}\,[\,\text{int}\,\textit{ervalles}\,]\ \textit{of type} \mid \wedge\,\textit{type}\\
&\text{int}\,\textit{ervalles} \rightarrow \text{int}\,\textit{ervalles}\ ,\ \text{int}\,\textit{ervalles} \mid \textit{num . . num}
\end{aligned}$$

Le tableau suivant donne une énumération de l'ensemble des tokens ainsi que leurs modèles respectifs.

Token	FIN	DP	PV	VIR	ID	BY TE	WORD	INTEGER
Modèle	EOF	:	;	,	(l|_)(l|c|_)*	byte	word	integer

Token	ARRAY	CRG	CRD	OF	PTR	NUM	INT	
Modèle	array	[	]	of	^	(+|-)? c+	..	

l = [A-Za-z], c = [0-9].

De plus, on conviendra que :

- toute chaîne comprise entre { et } est un commentaire ;
- les mots réservés peuvent être écrits indifféremment en lettres majuscules et minuscules ;
- NUM est le seul token disposant d'un attribut (la valeur entière).

1. Écrire une spécification Lex décrivant l'analyseur lexical pour le langage décrit ci-dessus et effectuer les compilations nécessaires pour générer l'exécutable.
2. Écrire une spécification yacc pour le langage décrit ci-dessus. Les messages d'erreurs syntaxiques seront de la forme :

Ligne 5: erreur syntaxique autour du symbole 'tab'.

3. Générer le fichier y.output et étudier les conflits d'analyse.
4. Résoudre les conflits de manière satisfaisante et réaliser différentes exécutions sur des déclarations avec et sans erreurs.
5. On se propose maintenant d'étendre la grammaire en schéma de traduction permettant de calculer et d'afficher la taille de l'espace mémoire occupé par une suite de déclarations donnée en entrée.
   En admettant qu'une variable de type *byte*, *word* ou *integer* occupe, respectivement, *un*, *deux* ou *quatre* octets, et qu'une variable de type pointeur en occupe quatre,

   (a) attacher des attributs aux symboles de la grammaire en utilisant les directives *%union* et *%type* ;

   (b) associer aux règles de production des actions sémantiques permettant de réaliser le traitement souhaité ;

   (c) réaliser différentes exécutions et vérifier les valeurs affichées.