

# Compilation et théorie des langages

## TP7 - Flex et Bison

Licence 3 - 2022/2023

### Abstract

Le but de ce TP est de construire un traducteur de séquences d'expressions arithmétiques vers la machine RAM du cours d'algorithmique de 2ème année. Pour cela nous utiliserons conjointement **Flex** et **Bison** ainsi qu'une table de symbole élémentaire pour la gestion des variables.

## 1 Utilisation conjointe de Flex et Bison

Il est possible d'écrire de puissant analyseurs syntaxiques avec **Bison** tout en sous-traitant la construction de l'analyseur lexical à **Flex**. Pour cela on commence par écrire l'analyseur syntaxique puis l'analyseur lexical chargé de reconnaître les unités lexicales (tokens) définies par **Bison**.

Les fichiers `parser.y` et `lexer.lex` fournissent un exemple élémentaire d'utilisation conjointe de **Flex** et **Bison**. Le programme **Flex** récupère la définition des unités lexicales grâce au fichier `.h` qui sera produit par **Bison** lors de la compilation du fichier `parser.y` avec l'option `-d`. La compilation de la calculatrice se fait alors avec la séquence de commandes suivantes :

```
$ bison -o parser.c -d parser.y
$ flex -o lexer.c lexer.lex
$ gcc -Wall -o parser parser.c lexer.c -lfl
```

## 2 Traduction d'expressions arithmétiques

Le compilateur doit être capable de traduire des expressions arithmétiques utilisant les opérateurs habituels (+, -, \*, /), les parenthèses et le moins unaire. Un programme à traduire devra commencer par le mot clé **DEBUT** et se terminer par le mot clé **FIN**:

```
DEBUT
10 + (21 -(12+3)*3)/2;
FIN
```

## 3 Gestion de la mémoire

Le compilateur doit être capable de gérer l'affectation et l'utilisation de variables à l'aide d'identificateurs ainsi que affichage:

```
DEBUT
toto <- 10 + (21 -(12+3)*3)/2;
titi <- toto * 8;
AFFICHER titi;
FIN
```

La gestion des variables et l'adresse mémoire qu'elles représentent sera géré par une table de symbole sous forme de liste chaînée basée sur la structure décrite dans les fichiers `ts.[ch]`.