

Formal Languages and Compilers

Laboratory 7

1 Exercice (Translator from mini C to Pseudo Assembler - Type Checking)

Modify the translator from **mini C** to **Pseudo Assembler** realising a *type checker* module.

The type checker must recognise and print the following errors and warning:

- **Variable <id> not declared.** (Error): when the programmer tries to use a variable that is not declared.
- **Array index (<used index>) exceed array size (<array size>).** (Error): when the index used to index an element of an array exceed the array dimension.
- **Operation between int and double, int number casted to double.** (Warning): when there is an operation between an integer number and a double number (i.e., 3+2.5).
- **Assignment of a double value to an int variable.** (Warning): when there is an assignment of a double value to a variable declared as integer (i.e., int a; a=3.5;).
- **Assignment of an int value to a double variable.** (Warning): when there is an assignment of an integer value to a variable declared as double (i.e., double a; a=3;).

When a type error is found do not stop the parser, try to find the maximum number of type errors. When type errors are found do not produce the Pseudo Assembler code as output.

2 Exercice (Type Checking)

Using the JFLEX lexer generator and the CUP parser generator, realize a Java program capable of recognizing type declarations and variable definition according to the C language syntax. The program should build and print the type expressions concerning all the variables defined.

Grammar:

```

S ::= /* empty */           ;
    | S Decl ';'           ;
    | S error ';'         ;
;
Decl ::= T Vlist;
T ::= TYPE;
Vlist ::= V
        | Vlist ',' V
V ::= Ptr ID Array;
Ptr ::= /* empty */
       | Ptr '*'
;
Array ::= /* empty */
        | Array '[' NUM ']'
;

```

Input

```

int i;
char **v[9];
int m[3][2],*j;
int k;

```

Output

```

var i : [INT]
var v : array(9, pointer(pointer([CHAR])))
var m : array(2, array(3, [INT]))
var j : pointer([INT])
var k : [INT]

```