

# Formal Languages and Compilers

## Proff. Breveglieri and Morzenti

### Written exam<sup>1</sup>: laboratory question

#### 12/07/2021

SURNAME: .....  
NAME: ..... Student ID: .....  
Course: ☐ Laurea Magistrale ☐ V. O. ☐ Laurea Triennale ☐ Other: ...  
Instructor: ☐ Prof. Breveglieri ☐ Prof Morzenti

The laboratory question must be answered taking into account the implementation of the ACSE compiler given with the exam text.

Modify the specification of the lexical analyser (`flex` input) and the syntactic analyser (`bison` input) and any other source file required to extend the `Lance` language with the **interval assignment**. An interval is syntactically specified by means of *two numerical values* (the upper and the lower bounds) that are separated by the symbol “.” and embraced with squared parenthesis. For instance, `[4..9]` represents the *ordered* set of integers  $\{4, 5, 6, 7, 8\}$  and `[-1..x]` represents the ordered set  $\{-1, \dots, 1\}$  if  $x$  is an ACSE variable of value 2. The interval `[10..1]` represents the ordered set  $\{10, 11, \dots, \text{MAX\_INT}, \text{MIN\_INT}, \dots, -1, 0\}$ , where `MAX\_INT` / `MIN\_INT` are the maximum / minimum (architecture-dependent) integers. Intervals are empty when lower and upper bounds are equivalent (e.g., `[7..7]`). Intervals can be used in assignments. Assigning an empty interval to a variable (scalar or vector) does not produce any change in its value. In all the other cases, if the left-hand side of the assignment is a scalar then the lower bound of the interval is assigned. Otherwise, the left-hand side is a vector, and all the elements of the interval that can fit into it are copied in the same order (i.e.,  $i$ -th element of the interval to the  $i$ -th element of the vector, from position 0). If the number  $n$  of elements of the interval is less than the size  $d$  of the vector, then the positions of the vector between  $n$  and  $d - 1$  remain unchanged.

```
int a[5];
int x,y,i;

a = [0..5];    //a=[0,1,2,3,4]

x=3;
y=10;

a = [x..6];    //a=[3,4,5,3,4]

a = [x+1..y];  //a=[4,5,6,7,8]

x = [1..3];    //x=1
```

---

<sup>1</sup>Time 60'. The only material allowed during the exam is the ACSE reference header.  
Pencil writing is allowed. Write your name on any additional sheet.

1. Define the tokens (and the related declarations in **Acse.lex** and **Acse.y**). (1 points)
2. Define the syntactic rules or the modifications required to the existing ones. (2 points)
3. Define the semantic actions needed to implement the required functionality. (22 points)

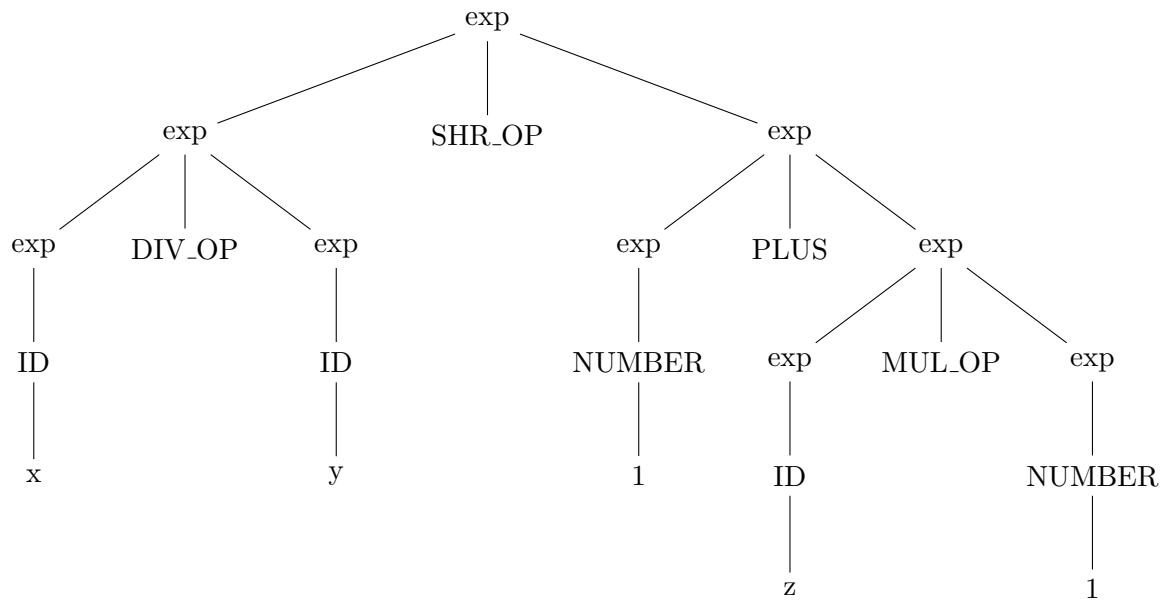
The solution is in the attached patch.



4. Given the following Lance code snippet:

`x/y>>1+z*1`

write down the syntactic tree generated during the parsing with the Bison grammar described in `Acse.y` *starting from the `exp` nonterminal*. (5 points)



5. (**Bonus**) Discuss how you would modify the your solution in order to make interval  $[5..0]$  indicating the set  $\{5, 4, 3, 2, 1\}$ .