Formal Languages and Compilers
Proff. Breveglieri, Crespi Reghizzi, Morzenti
Written exam[1]: laboratory question
06/09/2016

SURNAME: .....
NAME: ..... Student ID: .....
Course: ○ Laurea Specialistica ○ 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 vector assignment statement. The assignment copies a portion of contiguous elements of the source vector into the target vector from the position 0 (of the target vector). The set of the elements copied from the source vector are determined by an interval which is of the form `[s:e]`, where `s` and `e` are two **constants** of value s and e, respectively. The semantics of the statement is defined only if the values s and e are between 0 and the size of the source vector minus 1. There are two cases:

- $s \leq e$: the portion to copy consists of elements at position $s, s+1, \ldots, e$.
- $s > e$: the portion to copy consists of elements from position e to $arrSize(source) - 1$ followed by elements from 0 to s.

Checking the size of the target vector and the number of elements to copy is not required. The following code exemplifies the semantics of the statement and shows the syntax. **Remark:** in the

```
1  int a[10];
2  int b[5];
3
4  ...
5
6  //a=[0,1,2,3,4,5,6,7,8,9]
7
8  b = a[0:4];        //b = [0,1,2,3,4]
9
10 b = a[9:2];        //b = [9,0,1,2,4]
11
12 b = b[3:1];        //b = [2,4,9,0,4]
13
14 b = a[6:6];        //b = [6,4,9,0,4]
```

assignment at line 14, vector `b` is both source and target. The solution must prevent undesired loss of elements.

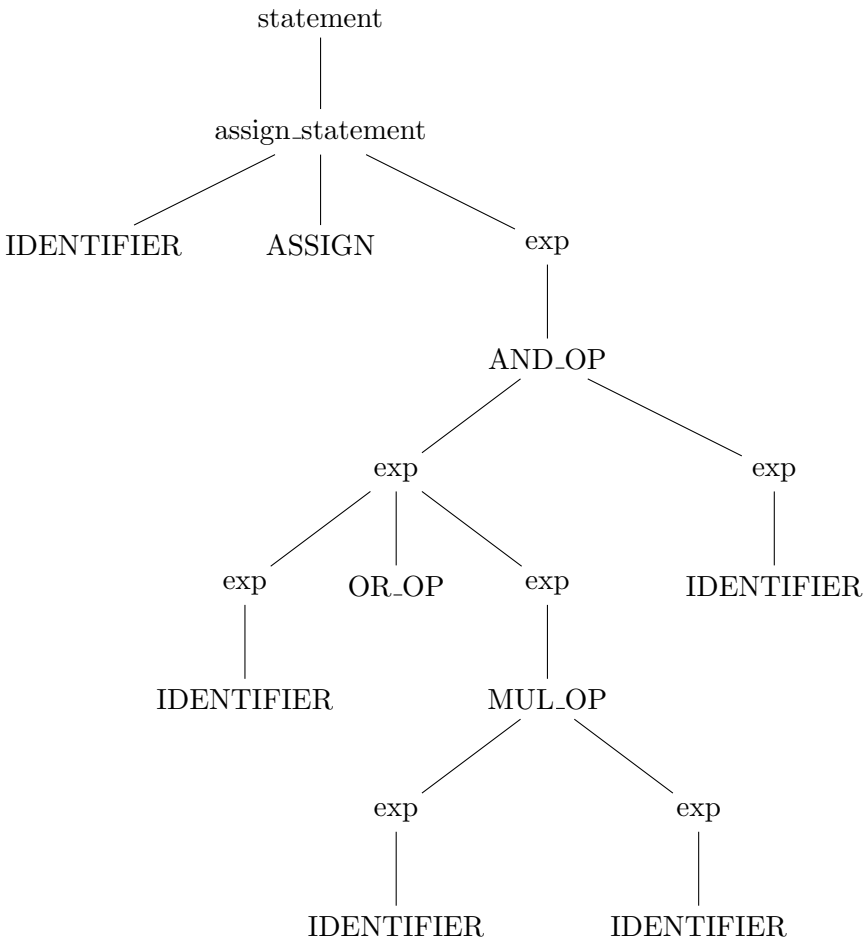
[1] Time 60'. Textbooks and notes can be used.
Pencil writing is allowed. Write your name on any additional sheet.

1. Define the tokens (and the related declarations in **Acse.lex** and **Acse.y**). (3 points)
2. Define the syntactic rules or the modifications required to the existing ones. (4 points)
3. Define the semantic actions needed to implement the required functionality. (18 points)
   The solution is in the attached patch.

4. Given the following `Lance` code snippet:

```
k = x | y*2 & z
```

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


statement
assign_statement
IDENTIFIER
ASSIGN
exp
AND_OP
exp
exp
exp
OR_OP
exp
IDENTIFIER
IDENTIFIER
MUL_OP
exp
exp
IDENTIFIER
IDENTIFIER

5. (**Bonus**) Discuss how to modify your solution to allow the presence of a condition in the assignment statement. The condition is applied to all the element copied from the source vector to the target and determines which element can be copied. For instance, `y = x[e@1:5!e>0]` copies all the strictly positive elements of vactor `x` from position 1 to 5 into vector `y`.