Formal Languages and Compilers
Proff. Breveglieri, Morzenti
Written exam[1]: laboratory question
07/02/2018

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 new **array reduction** construct.

The array reduction construct is constituted of the `red` keyword, followed by a variable name enclosed in round braces. The implementation must check that the said variable is an array. The behaviour of the construct is the computation of the sum of all the elements in the array.
An example is provided in the following.

```
int v[2], a;
a = 0;
v[0] = 3;
v[1] = 2;

a=red(v);
while( red(v) ){
   v[0] = v[0]-1;
   v[1] = v[0];
}
```

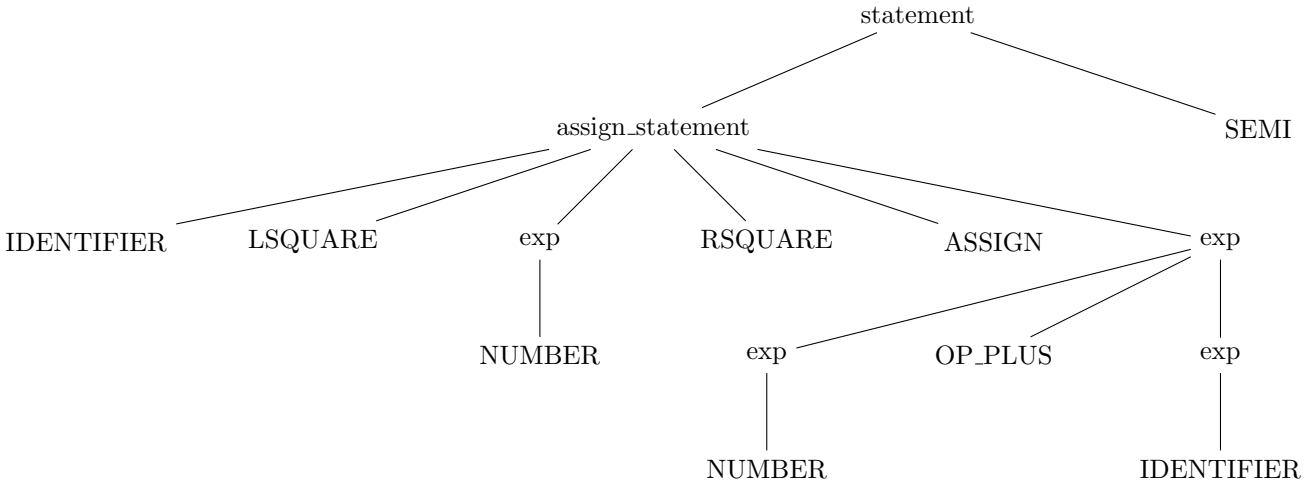
[1]Time 60'. Textbooks and notes can be used.
Pencil writing is allowed. Write your name on any additional sheet.

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:

```
a[5] = 3 + b;
```

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



5. (**Bonus**) Describe how to modify the construct so that the average of the array is computed, instead of its sum.