TDDD08 - Exercise 3

Valerio Colitta (950714-T639) valco263@student.liu.se
Théo Foray (980410-T096) thefo365@student.liu.se

October 2018

1 Exercise 3.1 - Parsing

For this lab, we have to create a parser than taken a string from a DCG, transforms it into a term that can be executed by the interpreter (Lab 2) together with an initial binding environment.
We had a specification about it, that we followed and created a DCG.

1.1 Notes

To execute this lab, we had to do some modifications:

- Modify the scanner.pl
- Change the format for the initial environment as [x,1,y,2,...], instead of [x=1,y=2] to make it compliant with our interpreter

Now, with these modifications applied, we can create our grammar.
For example, the clause `pgm(seq(X,Y))` returns the object seq(X,Y) if the token passed matches `cmd(X), ";", pgm(Y)`. This means, that what is before ";" has to match the `cmd(X)`. The other clauses work the same way.
At the end of the execution, we will have a well defined program for our interpreter.

```
%load needed files.
?- load_files([absmach,scanner]).

%construct the DCG according to the specification
pgm(X) --> cmd(X).
pgm(seq(X,Y)) --> cmd(X), [;], pgm(Y).
cmd(skip) --> [skip].
cmd(set(X,Y)) --> id(X), [:=], expr(Y).
cmd(if(X,Y,Z)) --> [if], bool(X), [then], pgm(Y),
                   [else],pgm(Z), [fi].
cmd(while(X,Y)) --> [while], bool(X), [do], pgm(Y), [od].
bool(X > Y) --> expr(X), [>], expr(Y).
bool(X < Y) --> expr(X), [<], expr(Y).
expr(X*Y) --> factor(X), [*], expr(Y).
expr(X) --> factor(X).
factor(X+Y) --> term(X), [+], factor(Y).
factor(X) --> term(X).
term(num(Y)) --> [Y], {number(Y)}.
term(X) --> [X], {nonvar(X)}.
id(X) --> [X], {nonvar(X)}.

%parse(T,O) : T = Tokens, O = Output program object
parse(Tokens, Out) :-pgm(Out, Tokens, []).

%run(I,S,O) : I = Initial Environment, S = String to be parsed,
% O = Output Environment
run(In, String, Out) :-
scan(String, Tokens),
parse(Tokens, AbstStx),
execute(AbstStx, In, Out).

/*
Example query:
-----------------------------------------------------------------
run([x,3],"y:=1; z:=0; while x>z do z:=z+1; y:=y*z od",Res).
Res = [x,3,y,6,z,3]
*/
```

Listing 3: Ex 3.1