Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

PLIA

Laboratory work # 2

Acquiring ideas about specific mechanisms of Prolog

Author:
Petru Negrei

Supervisor:
L. Luchianov

December 2014

1 Introduction

1.1 Topic

Acquiring ideas about specific mechanisms of Prolog

1.2 Tasks

1. Launch the execution work program established in laboratory work 1 and investigated changes:

- By changing the order of sentences facts;
- By changing the order of sentences rules; (two versions)
- By changing the order of sentences facts; (two versions)
- Make conclusions.

2. Solve the following problems proposed and will follow their proper implementation.

- Develop and test a program to determine a minimum value of two numbers (X and Y), without using predicate cut.
- Develop and test a program to determine a minimum value of two numbers (X and Y), using red and green cut predicates.
- What will be the program answers to given task, do a comparative analysis between predicates using the knowledge base cut in space and space purposes.
- Two children play a game in a tennis tournament if they have the same age. Whether subsequent children and their ages:
  child (peter, 9). child (paul, 10). child (chris, 9). child (sesame, 9).
  Define a predicate which shows all pairs of children who can play a game in a tennis tournament.

3. Enter the appropriate changes in the program (2.1), using green cut at least two rules in the knowledge base.

4. Enter the appropriate changes in the program (2.1), using red cut rules in the knowledge base. Write conclusions.

1.2.1 Report

Report will containt a short description of work done, and will present necessary information about tools, algorithms used or studied.

2 Implementation

2.1

2.1.1

After changing the order of rules I observed that no changes occur in the program, but when I changed the order of facts, I noticed that it changes the order of the showed results.
Thus when changing the rules we change the order in which the results will appear to the user.

2.2

2.2.1

Develop and test a program to determine a minimum value of two numbers (X and Y), without using predicate cut.

```
min(X,Y,X):- X =< Y.
min(X,Y,Y):- X > Y.
```

2.2.2

Develop and test a program to determine a minimum value of two numbers (X and Y), using cut cut predicate red and green.

```
% green cut
min(X, Y, X) :- X =< Y, !.
min(X, Y, Y) :- X > Y.
```

```
% red cut
min(X, Y, X) :- X =< Y, !.
min(X, Y, Y).
```

2.2.3

What will be the program answers to given task, do a comparative analysis between predicates using the knowledge base cut in space and space purposes

In the first case of **p(X)**. we have the following answers: $\mathbf{X = 1; X = 2}$. We have the answers because it enters in backtraking and check all the time until it reaches the clause p(2): - !. and stops because the cut predicate was used, which ignores the responses that follow.

For the purposes of **p(X), p(Y).** the answers will be:

X = Y, Y = 1;
X = 1, Y = 2;
X = 2, Y = 1;

X = Y, Y = 2.

For the purposes of **p(X),!,p(Y).** because of cut the answers will be:

X = Y, Y = 1;
X = 1, Y = 2,

2.2.4

Two children play a game in a tennis tournament if they have the same age. Whether subsequent children and their ages:

```
child(peter, 9).
child(paul, 10).
child(chris, 9).
child(sesame, 9).
```

Define a predicate which shows all pairs of children who can play a game in a tennis tournament.

```
perechi(A,B,C):-copil(A,C),copil(B,C),A @<B,!.
```

2.3 List

Write a predicate called *descomp(N, Lista)* witch will receive an integer number N and return a list of prime facotors of this number.

```
% Determine the prime factors of a given positive integer.

% descomp(N, L) :- L is the list of prime factors of N.
%   (integer,list) (+,?)

descomp(N,L) :- N > 0, descomp(N,L,2).

% descomp(N,L,K) :- L is the list of prime factors of N. It is
% known that N does not have any prime factors less than K.

descomp(1,[],_) :- !.
descomp(N,[F|L],F) :- % N is multiple of F
  R is N // F, N =:= R * F, !, descomp(R,L,F).
descomp(N,L,F) :-
  next_factor(N,F,NF), descomp(N,L,NF). % N is not multiple of F

% next_factor(N,F,NF) :- when calculating the prime factors of N
%   and if F does not divide N then NF is the next larger candidate to
%   be a factor of N.

next_factor(_,2,3) :- !.
next_factor(N,F,NF) :- F * F < N, !, NF is F + 2.
next_factor(N,_,N).                    % F > sqrt(N)
```

3 Conclusion

As a result of this laboratory work I analyzed theoretical material and became acquainted with some features of Prolog. In carrying out the work there have been solved some problems with using predicate cut red and green. After solving can say that the use of the green cut increase the efficiency of the program, also the use of the clauses use the order does not matter, the result will not depend on it. When using red cut it changes procedural significance of the program, in case the order is changed clauses will have an incorrect result.