Logic and Functional Programming
Laboratory 13 Prolog
Revision

Isabela Drămnesc

1. Read all from an input file until end of file

```
?- real_all_from_file('C:\\Users\\....\\pb.txt',L).
```

L will contain all the elements **read** from the input **file**

2. Generate random lists

```
?-generate_random_list(5,101,L), write_to_a_file('C:\\Users\\...\\out.txt',L).

% write_to_a_file writes each of the elements of a list on a line
%followed by '.'
% e.g. if L=[1,4,5,7,8], then the file out.txt is of the form
1.
4.
5.
7.
8.
```

3. Quick-sort

```
% splitting (pivoting) % consider H be the pivot
% L=[4,5,6,1,2,3,4] L1=[1,2,3,4,4], L2=[5,6]
% append

?-read_all_from_file('C:\\Users\\....\\pb.txt',L), quick_sort(L,S),
write_to_a_file('C:\\Users\\...\\qsorted.txt',L).
```

4. Merge-sort

```
?-read_all_from_file('C:\\Users\\....\\pb.txt',L), selection_sort(L,S),
write_to_a_file('C:\\Users\\...\\msorted.txt',L).
```

5. Delete duplicates (keeps the first occurrence)

```
?-read_all_from_file('C:\\Users\\....\\pb.txt',L), delete_duplicates(L,S),
write_to_a_file('C:\\Users\\...\\without_doubles.txt',L).

; delete_duplicates (without accummulators)
; delete_duplicates (with accummulators)
```

6.

```
animal(duck).
animal(cat).
animal(dog).
animal(horse).

swim(duck).
swim(dog).

r(X):-animal(X),swim(X),!,fail.
r(X):-animal(X).

% explain the definition of r(X).
% what are the answers for ?-r(X).
% What happens if we remove ! from the first %rule?
% What kind of ! uses the definition of r?
%red or green? Explain.
```

7. Cut-fail

```
%Consider the following friendship relation:
friend(radu, elena).
friend(elena, mihai).
friend(zoe, viorel).

% define, using the cut fail combination the
%relation unfriend which takes place if it
%is unknown whether X and Y are friends
```

8. Unification

```
% What kind of structures can be unified with
 _, [_], [_,_], [_|_], [X,X|_]
```

9. A solution to Hanoi's tower

```
% Move all the disks from axe_1 to axe_2
% one disk at a time, do not put a greater disk
%on a smaller disk

% one possible answer:
hanoi(N):-move(N,axe_1,axe_2,axe_3).

move(0,_,_,_):-!.
move(N,A,B,C):- M is N-1, move(M,A,C,B),
say(A,B), move(M,C,B,A).
say(X,Y):-write([move,one,disk,from,X,to,Y]),nl.
```

10. Define a database for the following:

```
% A day of a week can be Monday, Tuesday,
% Wednesday, Thursday, Friday, Saturday, Sunday.

% Define relations for:
% next(X,Y) Y is the next day of X
% twodaysafter(X,Y) Y is the second day after X
```

11. Consider the following database:

```
grade(popescu,'LCS',9.0).
grade(popescu,'FP',7.0).
grade(popescu,'LP',10.0).
grade(albu,'Algebra',9.0).
grade(albu,'Management',8.5).

% define a program general_average(Pers,Ga) which computes
% the general average of the person X. Example:

?-general_average(popescu,Ga).
?- Ga=8.66666

% Sugestion 1: define a predicate similar to %findall

% Sugestion 2: findall(Term,+G,-L)
% computes the list L of all possible instances of Term produced by
% answering to G.
% findall(N,X,grade(popescu,_,N),L).
```