Logic Programming – Laboratory 7

I/O Operations with files

Isabela Drămnesc

November 24, 2010

Notions

- I/O in Prolog

Exercises

1. Examples of writing

```
a) ?- write('today'),write(' is sunshine').

b) ?- writeq("abc").
[97, 98, 99]
true.

c) ?- write('abc').
abc
true.

d) ?- writeq('abc').
abc
true.

e) ?- write('la la la').
la la la
true.

f) ?- writeq('la la la').
'la la la'
true.

g) ?- writeq('today'),writeq(' is sunshine').

/* for today does not put quotes */

h) ?- put(98).
b
true.
```

```
/* put writes one character (the representation in the ASCII code) */

i) ?- put("99").
ERROR: put/1: Type error: 'character' expected, found '[57,57]'

j) ?- put('99').
ERROR: put/1: Type error: 'character' expected, found '99'

k) ?- get(L).
|: force

L = 102

l) ?- get(L).
|: f

L = 102

/* get reads one single character */

m) ?- write(today is monday),put(8),write(bye).
today is mondabye
true.

n) ?- put(8).
true.

o) ?- write('hello '), put(8), write(lala),put(98), put(99).
hellolalabc
true.

p) ?- write(hello),put(32),write(man).
hello man
true.
```

2. Examples of reading

```
a) ?- read(X).
|    hello.   /* this is what introduces the user */

X = hello.
     /* read unifies X with what the user introduces */

b) ?-read(lala).
|    'today is raining'.
false.

c) ?- read(X).
|    today is sunshine.
```

```
X = (today is sunshine).

d) ?- read(X).
|: 'today is sunshine'.

X = 'today is sunshine'.

e) ?- read(X).
|: "today is sunshine".

X = [116, 111, 100, 97, 121, 32, 105, 115, 32|...].
```

3. Read from files

Create a file pb.txt, then ask in Prolog:

a) the case when the file pb.txt is empty

```
?- see('C:\\Documents and Settings\\ISABELA\\Desktop\\pb.txt'),
read(X),seen.

X = end_of_file.

?- see('C:\\Documents and Settings\\ISABELA\\Desktop\\pb.txt'),
read(X), read(I), seen.

X = end_of_file.
I = end_of_file.
```

b) the case when the file pb.txt contains

```
4.
5.
6.
7.
10.

?- see('C:\\Desktop\\pb.txt'), read(X),read(Y),
read(Z),read(W),read(O),seen.

X = 4,
Y = 5,
Z = 6,
W = 7,
O = 10.
```

4. Write into a file

```
a) ?- tell('C:\\Desktop\\pb.txt'),
write('the first exercise'),nl,write('is solved'),
write('the second problem is to write into another file'),
told.
```

```
true.

b) ?- tell('C:\\Desktop\\pb.txt'),
write('hello'),tab(5),write('again'),told.

true.

c) ?- tell('C:\\Desktop\\pb2.txt'),
write('the second problem '),nl,
write('is to write a predicate which calculates
the sum of all the numbers from a file'),told.

true.

d) ?- tell('C:\\Desktop\\An2.pdf'),
write('writes in pdf but open with notepad'),told.

true.

/* so we can write in any kind of files */
```

5. Read all the elements from the file pb.txt and print the sum of the elements

a) printing the sum on the SWI-Prolog window

```
example:-see('C:\\Desktop\\pb.txt'),
read(X),read(Y),read(Z),read(V),sum([X,Y,Z,V],W),
write('the sum of the elements from the file pb.txt is '),
write(W),seen.

sum([],0).
sum([X|T],W):-sum(T,S),W is X+S.
```

b) print the sum of all the elements from the file pb.txt into another file called sum.txt

6.

The file pb.txt contains one single number on each line followed by dot. Create a predicate in Prolog which sorts the numbers from the file pb.txt and print the result into a file called sorted.txt.

7.

The file pb.txt contains one single number on each line followed by dot. Create a predicate in Prolog which does shift-left (if we have in the file 4.2.3.1. after we apply shift-left we will obtain 2.3.1.4.) and print the result into the file changed1.txt.

8.

The file pb.txt contains one single number on each line followed by dot. Create a predicate in Prolog which does right-shift (if we have in the file 4.2.3.1. after we apply right-shift we will obtain 1.4.2.3.) and print the result in the file changed2.txt.

9.

The file pb.txt contains one single number on each line followed by dot. Create a predicate in Prolog which separates the numbers from the file pb.txt into even numbers and odd numbers. Even numbers are printed into the file even.txt, and odd numbers are printed into the file odd.txt.

Homework:

From homework 3 the exercises 1,7,8,9,13,14,15,16,17,19,24,25 using files. I mean reading from a file, apply the predicate, the result is printed into another file.

As input data you have a file containing 100 lines. On each line we have a number followed by dot. Print into another file the sorted numbers (increasing way)!