Logic Programming– Laboratory 2

Syntax and Data Structures

Isabela Drămnesc

1 Questions!

- What is Prolog? How does it work?
- What are the facts in Prolog? Give an example!
- What are the rules in Prolog? Give an example!

2 Short theoretical part

- Terms
- Constants
- Variables
- Structures
- Arithmetic operators
- Functors
- Lists
- Structures as Trees
- Representation as a Tree

3 Exercises

3.1 Constants

Constants are simple terms and integers.

Example of **atoms**:

```
a, likes, ion_albulescu, =, -->, 'snow', 'Today';
```

not atoms:

```
A, George, _likes, 234white2, ion-albulescu.
```

3.2 Anonymous variables, Variables

We use the anonymous variables when we do not want to know all the possible instantiations.

1)

```
logic(_).

?-logic(a).
?-logic(b).
?-logic(c).
?-logic(X).
```

2)

```
likes(ana,book).
likes(john,beer).
likes(marian,beer).

?-likes(_,beer). /* Does anyone like beer? */
?-likes(ana,_). /* Does ana like someone? */
?-likes(_,_).   /* Does anyone like anyone? */
```

3)

```
r(a).
r(b).
s(X):-r(X).

?-s(a).
?-s(b).
?-s(c).
?-s(X).
?-r(X).
```

3.3 Structures

Examples:

4)

```
has(john, book(eminescu,poems)).
book(poems,author(mihai,eminescu)).

?- has(john,book(X,author(Y,eminescu))).
    /* does john have a book X which has the author (Y,eminescu)?
*/
```

5)

```
point(1,4).
segment(point(1,2),point(5,3)).
```

```
triangle(point(1,1),point(1,3),point(2,3)).
```

6)

```
+(1,*(2,3)).
```

3.4 Arithmetic Operators

Operators do not cause evaluation in Prolog.

7)

```
?- 5>2.
?- 2<3.
```

Test for:

```
<, >,=, =<, >=, =\=
```

8)

```
?- b == b.
?- p(a) == p(a).
?- p(X) == p(b).
?- 10 == 10.
?- 2 + 8 == 1 + 9.
```

9)

```
?- b =:= b.
?- p(a) =:= p(a).
?- p(X) =:= p(b).
?- 10 =:= 10.
?- 2 + 8 =:= 1 + 9.
```

is forces the evaluation of the expression.
Test for:

10)

```
E1 + E2
E1 - E2
E1 * E2
E1 / E2
E1 // E2 -- integer division
E1 rem E2
E1 ** E2
E1 /\ E2
E1 \/ E2
E1 ^ E2
E1 << E2
E1 >> E2
E1 =\= E2
E1 =:= E2
```

+ in front of the expression means that it has to be instantiated;

- means that it has not be instantiated

Examples:

11)
```
?- X is 3+4.
?- X is +(1,*(2,3)).
```

12)
```
 ?- X is 32 mod 12.
 ?- X is 35 mod 10.
```

13)
```
 ?- X is abs(14.3).
 ?- X is abs(-3.4).
```

14)
```
 ?- X is max(56,12).
 ?- X is max(-56,-23.5).
```

15)
```
 ?- X is round(23.45).
 ?- X is round(-23.45).
 ?- X is round(-23).
  ?- X is round(-29.8).
```

16)
```
 ?- integer(-23.5).
  ?- integer(34).

  ?- X is integer(-32.5).
  ?- X is integer(32.5).
```

17)
```
?- X is rationalize(0.7).
?- X is rationalize(-0.1).

?- X is rationalize(12).

?- X is rdiv(5,10).
?- X is rdiv(5,15).
?- X is rdiv(7,9).
```

18)
```
    ?- X is log(32).
    ?- X is log10(1000).
```

19)
```
?- X is popcount(15).
?- X is popcount(13).
```

20)
```
 ?- between(12,17,X).
 ?- succ(X,Y).
 ?- succ(11,12).
 ?- plus(5,7,12).
```

Test for: inc(Expr), dec(Expr), sign(Expr), floor(Expr), ceiling(Expr), sqrt(Expr), exp(Expr), cos(Expr), sin(Expr), ...

3.5 Unification

The predicate for unification is =.

In general the unification procedure of two terms T1 and T2 takes place when:

1. T1,T2 are constants;
2. T1, T2 are uninstantiated variables;
3. T1 is an uninstantiated variable, T2 is a constant or a structure;
4. T1, T2 are uninstantiated variables;
5. T1, T2 are structures: $T1 = f(A_1, A_2, ..., A_n)$, $T2 = f(B_1, B_2, ..., B_n)$.

Exercises:

22)

```
?- X=12.
?- yesterday='yesterday'.
?- likes(X,Y)=likes(Z,T).
?- likes(X,Y)=likes(z,t).
?- likes(x,y)=likes(Z,t).

?- occurs(a,B,c(D,e,f,g(H)))=occurs(A,b,c(d,E,F,g(h))).

?- eats(paul,apricots)=X.
```

3.6 Representing as Trees

`father(popescu,john).`

Is represented as:

father

popescu john

`registration(popescu,john,date-of-birth(Day,Month,Year)).`

Is represented by:

registration

popescu john date-of-birth

Day Month Year

18 October 1989

3.7 Lists

Any data structure can be represented by lists. Inductive domain:

- []-the empty list;
- .(h,t) -generic list, where h=head, t=tail of the list. The tail of a list has to be a list.

23)

```
?- [H|T] = [a, b, c].

?- A = a, B = [b, c], C = [A|B].
```

```
?- [a] = [H|T].
?- [ ] = [H|T].

?- "abcd"=X.
?- "abcd" = [97, 98, 99, 100].
?- "abcd" = [H|T].
?- "X*(Y+Z)" = [H|T].

?- abcd = X.
?- abcd = [H|T]. /* It is not a list */
?- a+b = X.
?- 'a+b' = X.
?- "a+b" = X.
?- 'a+b' = [H|T].
?- "a+b" = [H|T].
```

24) Which one are the unifications? What is instantiated on the next cases?

```
?- [X,Y,Z]=[book,library,internet].

?- [cat]=[H|T].

?- [X, Y | Z] = [i, love, sea].

?- [open|X]=[open,mind].

?- [[mary,Y]|Z]=[[X,lives],[in,timisoara]].

?- [white|X]=[Y|horse].

?- X="i love Prolog".
```

25) Introduce the next fact:

```
test(.(a,.(b,.(c,[])))).
```

Which one would be the instantiations of H and of T in the next situations?

```
?- test(.(H,T)).
?- test([H|T]).
```

26) Write a predicate is-list to define that a term is a list.
Example:

```
?- is_list([]).
true.

?- is_list([a,b,c]).
true.

?- is_list(green(snow)).
false.
```

3.8 Homework:

Homework2.

Deadline: Next laboratory.