Exercises on ProLog

Try to answer the following questions first “by hand” and then verify your answers using a Prolog interpreter.

a. Which of the following are valid Prolog atoms?

   f, loves(john,mary), Mary, _c1, ’Hello’, this_is_it

b. Which of the following are valid names for Prolog variables?

   a, A, Paul, ’Hello’, a_123, _, _abc, x2

d. c.Would the following query succeed?

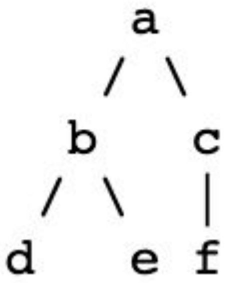
   ?- loves(mary, john) = loves(John, Mary). Why?

Exercises on Logic Programming and Prolog

Assume given a set of facts of the form father(name1,name2) (name1 is the father of name2).

1. Define a predicate brother(X,Y) which holds iff X and Y are brothers.
2. Define a predicate cousin(X,Y) which holds iff X and Y are cousins.
3. Define a predicate grandson(X,Y) which holds iff X is a grandson of Y.
4. Define a predicate descendent(X,Y) which holds iff X is a descendent of Y.
5. Consider the following genealogical tree:

```
father(a,b).
father(a,c).
father(b,d).
 father(b,e).
father(c,f).
```



Say which answers, and in which order, are generated by your definitions for the queries

```
?- brother(X,Y).

  ?- cousin(X,Y).

  ?- grandson(X,Y).

  ?- descendent(X,Y).
```

Solution

1. brother(X,Y) :- father(Z,X), father(Z,Y), not(X=Y).
2. cousin(X,Y) :- father(Z,X), father(W,Y), brother(Z,W).
3. grandson(X,Y) :- father(Z,X), father(Y,Z).
4. descendent(X,Y) :- father(Y,X).
5. descendent(X,Y) :- father(Z,X), descendent(Z,Y).

Draw the family tree corresponding to the following Prolog program:

```
female(mary).
female(sandra).
female(juliet).
female(lisa).
male(peter).
male(paul).
```

```
male(dick).
male(bob).
male(harry).
parent(bob, lisa).
parent(bob, paul).
parent(bob, mary).
parent(juliet, lisa).
```

```
parent(juliet, paul).
parent(juliet, mary).
parent(peter, harry).
parent(lisa, harry).
parent(mary, dick).
parent((mary, sandra)
```

After having copied the given program, define new predicates (in terms of rules using male/1, female/1 and parent/2) for the following family relations:

(a) father

(b) sister

(c) grandmother

(d) cousin

You may want to use the operator \=, which is the opposite of =. A goal like X \= Y succeeds, if the two terms X and Y cannot be matched.

Example: X is the brother of Y, if they have a parent Z in common and if X is male and if X and Y don't represent the same person. In Prolog this can be expressed through the following rule: brother(X, Y) :- parent(Z, X), parent(Z, Y), male(X), X \= Y.

Command to clear the console

tty_clear.