# CST8503 Assignment 1 Family Tree

## Overview

This assignment introduces Prolog concepts using a family tree example. Your assignment is to implement the Family Tree example described in this document, and complete the exercises below. This is an individual assignment.

When you have finished this assignment, you will have the ability to use the following Prolog concepts in your Prolog programs:

* Clause, fact, rule, question
* The head of a clause, the body of a clause
* Recursive rule, recursive definition
* Procedure
* Atom, variable
* Instantiation of a variable
* Goal
* Goal is satisfiable, goal succeeds
* Goal is unsatisfiable, goal fails
* Backtracking
* Declarative meaning, procedural meaning

## Family Tree Assignment

Implement the family diagram below in a Prolog file so that you can demonstrate it running to your lab instructor. You will define the parent/2 relation where parent(x,y) means x is the parent of y. Each arrow in the diagram represents an instance of the parent relationship. Remember that capital letters represent variables in Prolog, so at this stage, you will have no capital letters in your program.

susan=

bill

sally

joe

chris

jeff

jack

For the following exercises, when asked to write Prolog code, add the additional code to your prolog file and copy/paste the additional code into this document as your answer under the corresponding question. When asked to formulate a Prolog question/query, answer in-place in this document. When asked to draw a diagram/picture, it is acceptable to draw in this document with MS Word or draw on paper and take a picture with your phone or laptop to include the image file in this document.

* 1. What will be Prolog's answers to the following questions? (0.5 mark each)

1. ?- parent(jack, X). false.
2. ?- parent(X, jack). X = jeff.
3. ?- parent(bill, X), parent(X, jeff). X = joe.
4. ?- parent(bill, X), parent(X, Y), parent(Y, jack). X = joe, Y=jeff.
   1. Formulate in Prolog the following questions about the parent relation: (0.5 mark each)
      1. Who is jack's parent?

X = jeff

* + 1. Does chris have a child? No

?- parent(X, chris)

false

* + 1. Who is jack's grandparent? joe  
       ?- parent(X, Y), parent(Y, jack)

X = joe, Y = jeff

* 1. Translate the following statements into Prolog rules (0.5 mark each)
     1. Everybody who has a parent is educated (introduce a one-argument relation educated).

% rule: everybody who is a parent of someone else is also a parent

educated(X) :- parent(\_, X).

* + 1. For all X, if X has a child who has a sibling, then X is poor (introduce a new relation poor, and assume sibling(X,Y) is already defined to mean X and Y are siblings. ).

% sibling(X, Y) is true if X and Y share at least one parent, and X is not the same as Y.

sibling(X, Y) :-

parent(P, X), % P is a parent of X

parent(P, Y), % P is a parent of Y

X \= Y. % X and Y are different individuals

% X is poor if X has a child C who has a sibling S.

poor(X) :- parent(X, C), sibling(C, S).

* 1. Define the relation grandchild using the parent relation. Hint: It will be similar to the grandparent relation below (0.5 mark)

grandparent(X,Z):-parent(X,Y),parent(Y,Z).

* 1. For each person named in our program, use male/1 and female/1 to assert that they are male or female. Then define a sister(X,Y) relation in terms of parent and female (X is the sister of Y if Z is parent of X, and Z is parent of Y, and X is female, and X is not the same as Y. There is a built-in predicate you can use, dif(X,Y) that means X is not the same as Y. Define the relation aunt(X,Y) in terms of the relations parent and sister. (1 mark)

% Define sister/2 relation

sister(X, Y) :- parent(Z, X), parent(Z, Y), female(X), X \= Y.

% Define aunt/2: X is the aunt of Y if X is the sister of one of Y's parents

aunt(X, Y) :- sister(X, Z), parent(Z, Y).

* 1. Consider the following alternative definitions of the ancestor relation:

ancestor(X, Z):-

parent(X,Z).

ancestor(X,Z):-

parent(Y,Z),

ancestor(X,Y).

ancestor(X, Z):-

parent(X,Z).

ancestor(X,Z):-

parent(X,Y),

ancestor(Y,Z).

Do these both seem to be a proper definition of ancestor? Draw two diagrams that show X, Y, and Z, and the parent relationships, with vertical dots to indicate an indefinite number of intermediate individuals. Use arrows to indicate and label the parent and ancestor relationships. The first diagram is started for you. (1 mark)

X

Y

.

.

.

.

.

Z

ancestor(X, Z):- parent(X, Z).

ancestor(X, Z):- parent(X, Y), ancestor(Y, Z).

#### Explanation:

Base Case: ancestor(X, Z) is true if X is the parent of Z. (direct parent relationship)

Recursive Case: ancestor(X, Z) is true if X is the parent of Y and Y is an ancestor of Z (i.e., a recursive relationship that connects X to Z through Y).

This is a proper definition of ancestor, as it correctly captures both the direct parent-child relationship and the indirect ancestor relationship via recursion.

X → Y ⇢Z

| →

+---------+  
The arrow from X to Y →indicates that X is the parent of Y.

The arrow from Y to Z ⇢indicates that Y is the ancestor of Z.

The arrow from Y to Z →indicates that Y might be the parent of Z, just

If the first condition [ancestor(X, Z):- parent(X, Z).]works

The +---------+ from X to Z means X is the ancestor of Z, an additional recursive relationship (ancestor(X, Z)) holds because X is the parent of Y, and Y is an ancestor of Z.

ancestor(X, Z):- parent(X, Z).

ancestor(X, Z):- parent(Y, Z), ancestor(X, Y).

#### Explanation:

Base Case: ancestor(X, Z) is true if X is the parent of Z. (direct parent relationship)

Recursive Case: ancestor(X, Z) is true if Y is the parent of Z and X is an ancestor of Y.

This definition also works because it recursively climbs up the ancestor tree, starting from Z and working its way back to X. While the structure is different from the first, it still defines the ancestor relationship correctly.

X ⇢ Y →Z

| →

+---------+

The arrow⇢ from X to Y indicates that Y is the ancestor of Z.

The arrow from X to Y →indicates that X might be the parent of Y, just

If the first condition [ancestor(X, Z):- parent(X, Z).]works

The arrow from Y to X indicates that Y is the parents of Z.

By recursion+---------+, X is also the ancestor of Z.

* 1. Try to understand how Prolog derives answers to the following questions, using the Prolog code from this assignment. Show your work similar to the example, showing the intermediate goals indented, and draw an X where Prolog will backtrack. (3 marks)

Example

?- grandparent(susan,bill).

parent(susan,Y).

parent(susan,joe).

parent(joe,bill).

X backtrack

parent(suzan,Y).

parent(suzan,sally).

parent(sally,bill).

X backtrack

X no more alternatives, fail.

1. ?- parent(bill, joe).

parent(bill, joe). %succeed

% X backtrack

% X no more alternatives

1. ?- grandparent(bill, chri).

parent(bill, Y), parent(Y, chri).

parent(bill,joe), parent(joe, chri). % succeed

% X backtrack

% X no more alternatives.

1. ?- grandparent(joe, jack).

parent(joe, Z), parent(Z, jack).  
 parent(joe, chri), parent(chri,jack).

% X backtrack

parent(joe, jeff), parent(jeff, jack) % succeed

% X backtrack

% X no more alternatives