Using Prolog to Represent Family Relationships

# 1. What is Prolog?

Prolog (PROgramming in LOGic) is a logic programming language ideal for representing relationships and rules. It’s especially good for problems like family trees where logic and rules are used.

# 2. How to Use Prolog Online with SWISH

1. Open your browser and go to: https://swish.swi-prolog.org

2. You will see a code editor on the left and output/results on the right.

3. Start writing Prolog code in the editor.

# 3. Prolog Basics

• Facts: Simple true statements like male(john).

• Rules: Logic connections between facts.

• Queries: Questions asked to the system using ?-.

# 4. Define a Small Family Tree

Here are some example Prolog facts:

% gender  
male(john).  
male(david).  
male(raj).  
female(mary).  
female(sita).  
female(rita).  
  
% parent relationships  
parent(john, david).  
parent(mary, david).  
parent(john, rita).  
parent(mary, rita).  
parent(david, raj).  
parent(sita, raj).

# 5. Define Rules for Relationships

% father(X, Y): X is father of Y  
father(X, Y) :- male(X), parent(X, Y).  
  
% mother(X, Y): X is mother of Y  
mother(X, Y) :- female(X), parent(X, Y).  
  
% sibling(X, Y): X and Y are siblings  
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.  
  
% grandparent(X, Y): X is grandparent of Y  
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

# 6. Ask Questions (Queries)

?- father(john, david). % true  
?- sibling(david, rita). % true  
?- grandparent(john, raj). % true  
?- mother(sita, raj). % true

# 7. How to Save Your Program in SWISH

1. Click the ☰ (menu) icon in the top-left of the page.

2. Click 'Download' → 'As Prolog file (.pl)'.

3. The file is now saved and can be reused anytime.

# 8. Tips for Students

• All names (like john, david) should be lowercase.

• Variables (like X, Y) start with uppercase letters.

• Use :- to define rules. It means 'if'.

# 9. Line-by-Line Explanation of Family Facts

## Gender Facts:

These lines declare the gender of each person. Each line is a Prolog fact and ends with a period (.) to show it is complete.

|  |  |
| --- | --- |
| Line of Code | Meaning in English |
| male(john). | John is a male. |
| male(david). | David is a male. |
| male(raj). | Raj is a male. |
| female(mary). | Mary is a female. |
| female(sita). | Sita is a female. |
| female(rita). | Rita is a female. |

## Parent Relationships:

These lines define parent-child relationships using facts. The first name is the parent, and the second name is the child.

|  |  |
| --- | --- |
| Line of Code | Meaning in English |
| parent(john, david). | John is a parent of David. |
| parent(mary, david). | Mary is a parent of David. |
| parent(john, rita). | John is a parent of Rita. |
| parent(mary, rita). | Mary is a parent of Rita. |
| parent(david, raj). | David is a parent of Raj. |
| parent(sita, raj). | Sita is a parent of Raj. |

# 10. What Does the Period (.) Mean in Prolog?

In Prolog, a period (.) is used to mark the end of a statement, just like a full stop in English. It tells Prolog that the current fact, rule, or query is complete.

Example:

male(john). % This is correct

male(john) % This will cause an error because it's missing the period

Always remember to end every statement in Prolog with a period.

# 11. Explanation of Prolog Rules (father, mother, sibling, grandparent)

## Rule: father(X, Y)

This rule means: X is the father of Y if X is male and X is a parent of Y.

Code:

father(X, Y) :- male(X), parent(X, Y).

Explanation:

• male(X): X is a male.

• parent(X, Y): X is a parent of Y.

So, if both are true, then X is the father of Y.

## Rule: mother(X, Y)

This rule means: X is the mother of Y if X is female and X is a parent of Y.

Code:

mother(X, Y) :- female(X), parent(X, Y).

Explanation:

• female(X): X is a female.

• parent(X, Y): X is a parent of Y.

So, if both are true, then X is the mother of Y.

## Rule: sibling(X, Y)

This rule means: X and Y are siblings if they have the same parent and X is not equal to Y.

Code:

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

Explanation:

• parent(Z, X): Z is a parent of X.

• parent(Z, Y): Z is a parent of Y.

• X \= Y: X and Y are not the same person.

So, if they share a parent and are different people, they are siblings.

## Rule: grandparent(X, Y)

This rule means: X is a grandparent of Y if X is a parent of Z, and Z is a parent of Y.

Code:

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

Explanation:

• parent(X, Z): X is a parent of Z.

• parent(Z, Y): Z is a parent of Y.

So, X is a grandparent of Y.

Sample code:

% Gender facts

male(john).

male(david).

male(raj).

female(mary).

female(sita).

female(rita).

% Parent relationships

parent(john, david). % John is a parent of David

parent(mary, david). % Mary is a parent of David

parent(john, rita). % John is a parent of Rita

parent(mary, rita). % Mary is a parent of Rita

parent(david, raj). % David is a parent of Raj

parent(sita, raj). % Sita is a parent of Raj

% Rule: X is the father of Y if X is male and a parent of Y

father(X, Y) :- male(X), parent(X, Y).

% Rule: X is the mother of Y if X is female and a parent of Y

mother(X, Y) :- female(X), parent(X, Y).

% Rule: X and Y are siblings if they share a parent and are not the same

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

% Rule: X is a grandparent of Y if X is parent of Z and Z is parent of Y

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

## How to Run on SWISH

1. Go to 👉 https://swish.swi-prolog.org
2. Paste the above code in the left panel.
3. Below the code, type the following **queries** to test.

## 🔍 Example Queries to Try

% Who is the father of David?

father(X, david).

% Who is the mother of Rita?

mother(X, rita).

% Are David and Rita siblings?

sibling(david, rita).

% Who are the siblings?

sibling(X, Y).

% Who is the grandparent of Raj?

grandparent(X, raj).

% Who are all fathers?

father(X, Y).

% Who are all grandparents?

grandparent(X, Y).

|  |  |
| --- | --- |
| father(X, david). | X = john |

|  |  |  |
| --- | --- | --- |
| mother(X, rita). |  | X = mary |

|  |  |
| --- | --- |
| sibling(david, rita). | true |

|  |  |
| --- | --- |
| sibling(X, Y). | X = david, Y = rita ; X = rita, Y = david |

|  |  |
| --- | --- |
| grandparent(X, raj). | X = john ; X = mary |

|  |  |
| --- | --- |
| father(X, Y). | X = john, Y = david ; X = john, Y = rita;  X = david, Y = raj |

|  |  |
| --- | --- |
| grandparent(X, Y). | X = john, Y = raj ; X = mary, Y = raj |

For testing

% --- Facts ---

male(john).

parent(john, paul).

% --- Rule ---

father(X, Y) :- male(X), parent(X, Y).

Query

father(john, paul). Ans: true

father(X, paul). Ans: **X** = john