Artificial Intelligence

Assignment 5

Matthias Horbach

mhorbach@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: 13.06.2020, 23:59

Tutorial on: 14.06.2020

Abhinav Ralhan (221202684), abhinavr8@uni-koblenz.de
Hammad Ahmed (221202832), hammadahmed@uni-koblenz.de
Muhammad Asad Chaudhry (221202659), maac@uni-koblenz.de
Vishal Vidhani (221202681), vvidhani@uni-koblenz.de

1 Prolog (4 Points)

Here are four different ways to extend our family example with childOf/2 to arbitrary descendants:

1. descendantOf(X,Y) :- childOf(X,Y).
   descendantOf(X,Y) :- childOf(X,Z), descendantOf(Z,Y).
2. descendantOf(X,Y) :- descendantOf(Z,Y), childOf(X,Z). descendantOf(X,Y) :- childOf(X,Y).
3. descendantOf(X,Y) :- childOf(X,Y).
   descendantOf(X,Y) :- descendantOf(Z,Y), childOf(X,Z).
4. descendantOf(X,Y) :- childOf(X,Z), descendantOf(Z,Y). descendantOf(X,Y) :- childOf(X,Y).

All those definitions are logically equivalent. Which ones work well in Prolog and which ones do not? Why?

The Prolog command trace. might help you.

Solution

1. Correct.

2. There are cyclic relationships found, which means that the program gets stuck in a recursion loop as shown below.

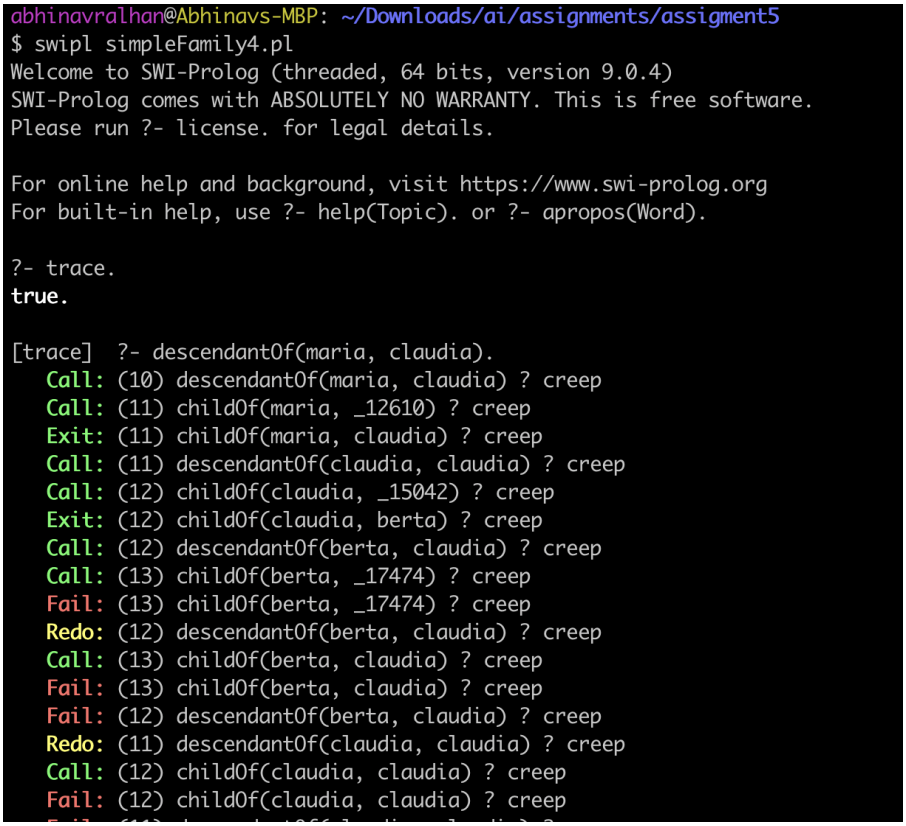
```
abhinavralhan@Abhinavs-MBP: ~/Downloads/ai/assignments/assigment5
$ swipl simpleFamily2.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- trace.
true.

[trace]  ?- descendantOf(maria, claudia).
   Call: (10) descendantOf(maria, claudia) ? creep
   Call: (11) descendantOf(_12616, claudia) ? creep
   Call: (12) descendantOf(_13428, claudia) ? creep
   Call: (13) descendantOf(_14240, claudia) ? creep
   Call: (14) descendantOf(_15052, claudia) ? creep
   Call: (15) descendantOf(_15864, claudia) ? creep
   Call: (16) descendantOf(_16676, claudia) ? creep
   Call: (17) descendantOf(_17488, claudia) ? creep
   Call: (18) descendantOf(_18300, claudia) ? creep
   Call: (19) descendantOf(_19112, claudia) ? creep
   Call: (20) descendantOf(_19924, claudia) ? creep
   Call: (21) descendantOf(_20736, claudia) ? creep
   Call: (22) descendantOf(_21548, claudia) ? creep
   Call: (23) descendantOf(_22360, claudia) ? creep
   Call: (24) descendantOf(_23172, claudia) ? creep
   Call: (25) descendantOf(_23984, claudia) ? creep
```

3. There are cyclic relationships found, which means that the program gets stuck in a recursion loop as shown below.



4. Correct.

2 Default rules (6 Points)

Represent the following sentences as default rules:
1. Spouses of Germans rarely hate soccer.
2. People usually have the same hometown as their spouse.
3. Most of the time, a person's hometown is where his or her employer is located.

Use predicate symbols spouse/2, employer/2, german/1 and hate/2, as well as function symbols hometown/1, location/1 and soccer/0.

Solution

1. Spouses of Germans rarely hate soccer.

   Spouse(X,Y) ∧ German(Y) : ¬ Hate(X, soccer) / ¬ Hate(X, soccer)

2. People usually have the same hometown as their spouse.

   Spouse(X,Y) ∧ hometown(Y) = Z : hometown(X) = Z / hometown(X) = Z

3. Most of the time, a person's hometown is where his or her employer is located.

   Employer(X,Y) ∧ location(Y) = Z : hometown(X) = Z / hometown(X) = Z

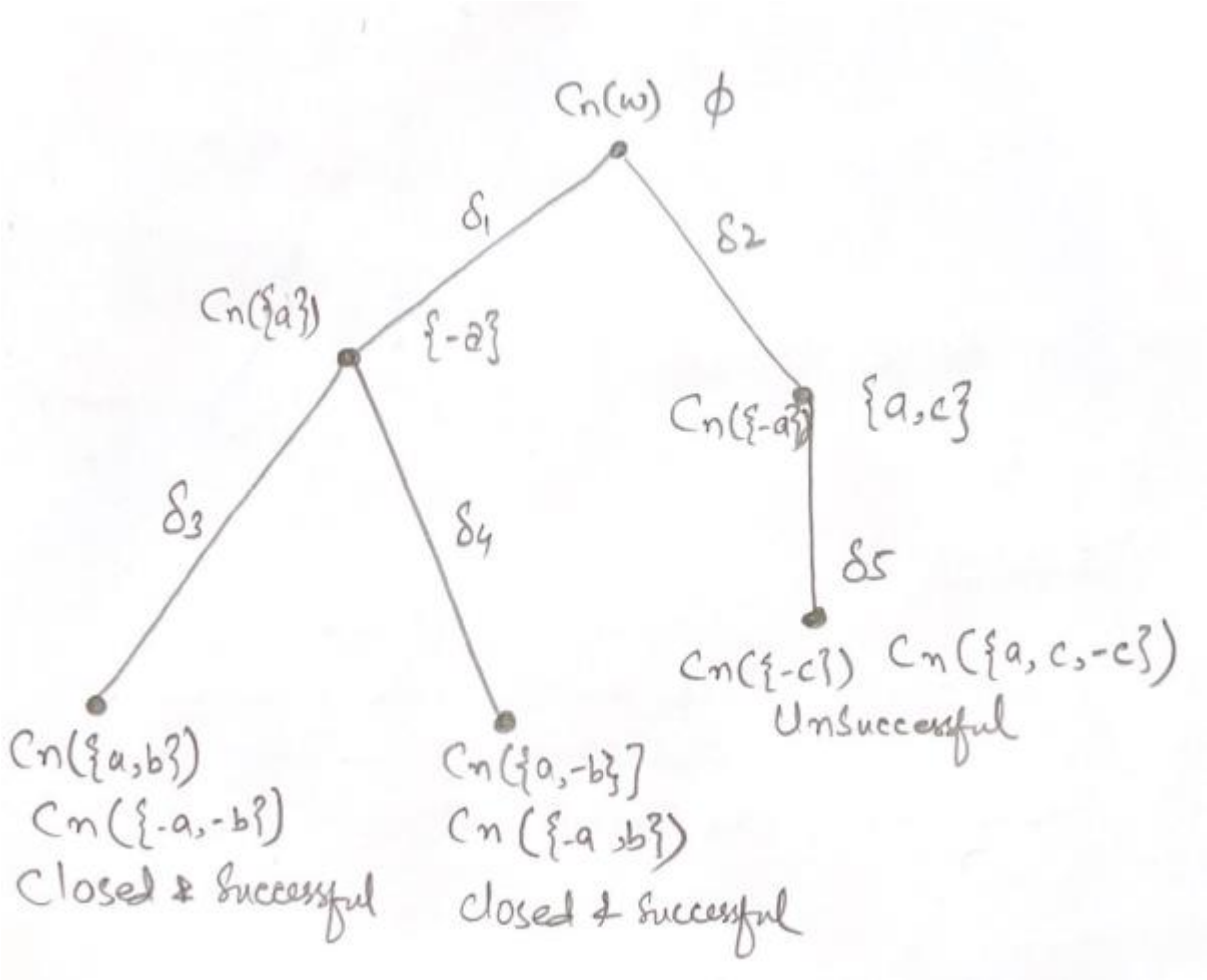
3 Default theory (10 Points)

Let T = (W,Δ) be a default theory with W = ∅ and Δ = {δ_1,δ_2,δ_3,δ_4,δ_5} with

$$\delta_1 = \frac{\top : a}{a}, \quad \delta_2 = \frac{\top : \neg a, \neg c}{\neg a}, \quad \delta_3 = \frac{a : b}{b}, \quad \delta_4 = \frac{a : \neg b}{\neg b}, \quad \delta_5 = \frac{\neg a : c}{c}$$

Calculate all extensions of T using a process tree. Make sure that you both draw the tree and state the resulting extensions.

Solution



The extensions are {a,b} and {a, -b}