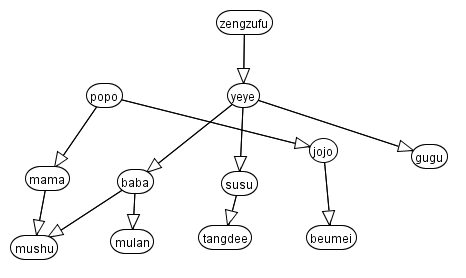
Peter Parianos

Dan Gluth

Project 4

Grail ID: peparian

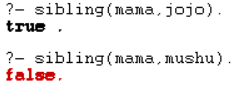
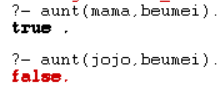
Question 1 Family Tree:



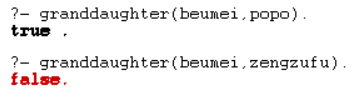
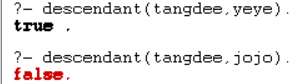
**Problem1:**

I defined a rule called the parent, which could be either a mother or a father. This helped streamline the other rules that followed. For each of the defined rules thereafter from problem1, the logic follows as this, using variables X is <member> of Y. If Z is the parent of Y, and X is the parent of X, and X is not Y, then X is the sibling of Y. Now we can use the sibling rule is subsequent rules. IF X is sibling of Y, and X is male, then X is the brother of Y. If Z is the parent of Y, Z is the sibling of X, and X is female, then X is the aunt of Y. If X is female, Z is the parent of X, Y is the parent of Z, then X, is the granddaughter of Y. If Y is the parent of X, then X is the descendant of Y.

Test cases:





**Problem 2:**

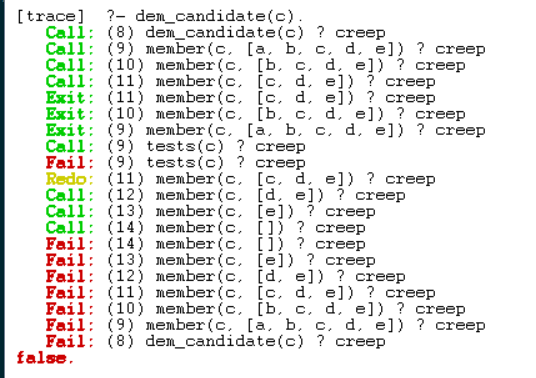
The fact is the base case of the last element algorithm. We know that X is the last element if it is the only element in a list. The rule matches this fact with every element in te list by using the | operator which separates the list into a head (which is an uninstatiated variable), and Z which is the rest of the list.

**Problem 3:**

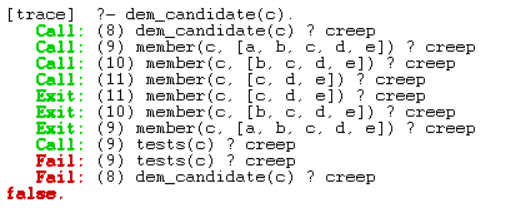
To implement the parser I used a match fact. The match fact also advances to the tail of the list. The language is comma separated and then placed in a list. In order to test this, I had to use a list that the parser accepts. All <letter> is succeeded with a s in order to indicate that is not an id but a non-terminal. We also need to start with the highest level definition s.

**Problem 4:**

The code was found in the Textbook required for this class.



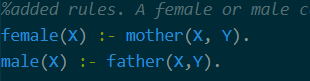
This is with the (!) operator:



**Debugging and Other issues:**

What really helped with me was the fa that I read each rule from right to left. A good example of my thought pattern is when I explained question 1.

I also needed to specify these peripheral rules which helped simplifiy some of the questions:

These rules helped infer that a mother X cis in fact a female as well, and the father X is a male as well.

Mother and father are both parents.

One large problem that I was having during the runtime of prolog was that the method:

**%- prolog family\_tree.pl prog4.pl**

Did not work correctly. I would not recognize facts that are in the famiy\_tree.pl. I solved this problem by looking at the SWI-Prolog Documentation and found a command that could textually include the whole family\_tree.pl:



