

1 Syntax Description

This list is how I initially implemented, but then realized I was supposed to align the syntax given in the assignment page. I modified so for my prolog file, but I will leave this paper as it is because it is too much to fix everything.

parent(PARENT, CHILD)
grandparent(GRANDPARENT, PERSON)
sibling(X, S): S is an array of siblings for X
cousin(X, Cousins): same as sibling but now it is cousins

nth_parent(N, X, Y): X is the Nth ancestor of Y. It is a generalization of the grandparent function. all_children(X, Children): Children is a list of all X's children.
gather(MEMBER, All): ALL is an array of children from all members in the array MEMBER. Used in nth_children.
nth_children(N, X, Children): Children is a list of all X's Nth-generation descendants. It is a generalized form of all_children.
nthcousin(N, X, Cousins): Cousins is a list of X's Nth cousins.
nthcousinkremoved(X, Y, N, K): Y is the Kth removed Nth cousin of X.

age(AGE, PERSON)
sortByAge(Arr, Sorted): Sorted is a list of people ordered by age (ascending).

2 Learning Outcome

I learned a first-order-logic-based declarative language and its difference from an imperative language, although I tried to use my experiences in imperative language here as much as possible. Also, I now remember many ancestors and how they branch out. It is not mine but the one very important as a Japanese person. I hope you can figure out which posterity I referred to because I think it is funny. Additionally, I inserted some creatures in the tree which are shown in the next Evidence of Run section. I hope you can figure out where these names came from, too.

3 Evidence of Run

```
%sibling(kako, S)
%cousin(norihito, C)
%nth_parent(3, PARENT, naruhito)
```

```
 sibling(kako, S) %cousin(norihito, C) %nth_parent(3,
PARENT, naruhito) %nthcousin(2, shoggoth, C)
%nth_children(2, [akihito, masahito], Children)
%nthcousinkremoved(naruhito, Y, 2, 1) %sortByAge([hirohito, mi-go, tsuguko,
norihito], L)
```

S = [mako, hisahito]

```
 %sibling(kako, S) cousin(norihito, C) %nth_parent(3,
PARENT, naruhito) %nthcousin(2, shoggoth, C)
%nth_children(2, [akihito, masahito], Children)
%nthcousinkremoved(naruhito, Y, 2, 1) %sortByAge([hirohito, mi-go, tsuguko,
norihito], L)
```

C = [akihito, masahito]

```
 %sibling(kako, S) %cousin(norihito, C) nth_parent(3,
PARENT, naruhito) %nthcousin(2, shoggoth, C)
%nth_children(2, [akihito, masahito], Children)
%nthcousinkremoved(naruhito, Y, 2, 1) %sortByAge([hirohito, mi-go, tsuguko,
norihito], L)
```


PARENT = yoshihito

Next 10 100 1,000 Stop

```


%nthcousin(2, shoggoth, C)
%nth_children(2, [akihito, masahito], Children)
%nthcousinkremoved(naruhito,Y,2,1)
sortByAge([hirohito, mi-go, tsuguko, norihito], L)

```

 %sibling(kako, S) %cousin(norihito, C) %nth_parent(3, PARENT, naruhito) nthcousin(2, shoggoth, C) %nth_children(2, [akihito, masahito], Children) %nthcousinkremoved(naruhito,Y,2,1) %sortByAge([hirohito, mi-go, tsuguko, norihito], L)


C = [yoko, akiko, tsuguko]

false

 %sibling(kako, S) %cousin(norihito, C) %nth_parent(3, PARENT, naruhito) %nthcousin(2, shoggoth, C) nth_children(2, [akihito, masahito], Children) %nthcousinkremoved(naruhito,Y,2,1) %sortByAge([hirohito, mi-go, tsuguko, norihito], L)


Children = [aiko, kako, mako, hisahito, mi-go]

false

 nthcousinkremoved(naruhito,Y,2,1) %sortByAge([hirohito, mi-go, tsuguko, norihito], L)

Y = [tsathoggua]

false

 sortByAge([hirohito, mi-go, tsuguko, norihito], L)

L = [mi-go, tsuguko, norihito, hirohito]