# Lab2

## Name: Zeng Yuhang

## ID: 222320008



Problem statement

The main purpose of this lab work is to learn how to use ‘Semantic’ program and to understand the differences between prolog and semantic. We should first transfer the knowledge base in prolog to the ‘Semantic’ environment, and then add attributes with some rules to objects. The first step aims to let us know the syntax differences between them, and the second step shows some new capabilities of semantic network.

Knowledge base text

########################## FamilyTree.pl ##############################

t("family\_tree").

onto("Rules.pl").

f("yuhang","has\_sex","male").

f("youlin","has\_sex","male").

f("youquan","has\_sex","male").

f("tao","has\_sex","male").

f("qingming","has\_sex","male").

f("wei","has\_sex","male").

f("du","has\_sex","male").

f("heng","has\_sex","male").

f("shengqiang","has\_sex","male").

f("hang","has\_sex","male").

f("yueyang","has\_sex","male").

f("anran","has\_sex","male").

f("jiayan","has\_sex","male").

f("tinggao","has\_sex","male").

f("dayou","has\_sex","male").

f("yuxi","has\_sex","female").

f("li","has\_sex","female").

f("yesha","has\_sex","female").

f("jun","has\_sex","female").

f("xiujuan","has\_sex","female").

f("jiafeng","has\_sex","female").

f("yuehong","has\_sex","female").

f("youping","has\_sex","female").

f("lu","has\_sex","female").

f("yi","has\_sex","female").

f("yue","has\_sex","female").

f("xiaoou","has\_sex","female").

f("shuang","has\_sex","female").

f("daobi","has\_sex","female").

f("longhui","has\_sex","female").

f("tinggao","is\_father\_of","youquan").

f("tinggao","is\_father\_of","youping").

f("tinggao","is\_father\_of","youlin").

f("youquan","is\_father\_of","yue").

f("tao","is\_father\_of","du").

f("youlin","is\_father\_of","yuhang").

f("youlin","is\_father\_of","yuxi").

f("dayou","is\_father\_of","jun").

f("dayou","is\_father\_of","li").

f("dayou","is\_father\_of","wei").

f("qingming","is\_father\_of","heng").

f("qingming","is\_father\_of","lu").

f("wei","is\_father\_of","yi").

f("wei","is\_father\_of","yueyang").

f("shengqiang","is\_father\_of","anran").

f("hang","is\_father\_of","jiayan").

f("longhui","is\_mother\_of","youquan").

f("longhui","is\_mother\_of","youping").

f("longhui","is\_mother\_of","youlin").

f("yuehong","is\_mother\_of","yue").

f("youping","is\_mother\_of","du").

f("youping","is\_mother\_of","shuang").

f("li","is\_mother\_of","yuhang").

f("li","is\_mother\_of","yuxi").

f("daobi","is\_mother\_of","jun").

f("daobi","is\_mother\_of","li").

f("daobi","is\_mother\_of","wei").

f("jun","is\_mother\_of","heng").

f("jun","is\_mother\_of","lu").

f("xiujuan","is\_mother\_of","yi").

f("xiujuan","is\_mother\_of","yueyang").

f("lu","is\_mother\_of","anran").

f("yi","is\_mother\_of","jiayan").

f("jiafeng","is\_spouse\_of","wei").

f("xiaoou","is\_spouse\_of","du").

f("yesha","is\_lover\_of","yuhang").

f("yuhang","is\_a","person").

f("yesha","is\_a","person").

f("yuxi","is\_a","person").

f("yuhang:hobby","is","sports").

f("yesha:hobby","is","sports").

f("yuxi:hobby","is","painting").

############################## Rules.pl ##############################

r([t("?x","is\_father\_of","?y")],[t("?x","is\_parent\_of","?y")]).

r([t("?x","is\_mother\_of","?y")],[t("?x","is\_parent\_of","?y")]).

r([t("?x","is\_father\_of","?y")],[t("?y","is\_child\_of","?x")]).

r([t("?x","is\_mother\_of","?y")],[t("?y","is\_child\_of","?x")]).

r([t("?x","is\_child\_of","?y"),t("?x","has\_sex","male")],[t("?x","is\_son\_of","?y")]).

r([t("?x","is\_child\_of","?y"),t("?x","has\_sex","female")],[t("?x","is\_daughter\_of","?y")]).

r([t("?x","is\_father\_of","?y"),t("?y","is\_parent\_of","?z")],[t("?x","is\_grandfather\_of","?z")]).

r([t("?x","is\_mother\_of","?y"),t("?y","is\_parent\_of","?z")],[t("?x","is\_grandmother\_of","?z")]).

r([t("?x","is\_child\_of","?y"),t("?z","is\_child\_of","?y"),t("?x","differs","?z")],[t("?x","is\_sibling\_of","?z")]).

r([t("?x","is\_parent\_of","?y"),t("?z","is\_sibling\_of","?x"),t("?z","has\_sex","male")],[t("?z","is\_uncle\_of","?y")]).

r([t("?x","is\_parent\_of","?y"),t("?z","is\_sibling\_of","?x"),t("?z","has\_sex","female")],[t("?z","is\_aunt\_of","?y")]).

r([t("?x","is\_parent\_of","?y"),t("?z","is\_parent\_of","?w"),t("?x","is\_sibling\_of","?z")],[t("?y","is\_cousin\_of","?w")]).

r([t("?x","is\_parent\_of","?y"),t("?z","is\_parent\_of","?y"),t("?x","differs","?z")],[t("?x","is\_spouse\_of","?z")]).

c("person").

o("person","attribute","hobby").

r([t("?a","is\_a","?b"), t("?b","attribute","?c")], [t("?a","attribute","?a:?c")]).

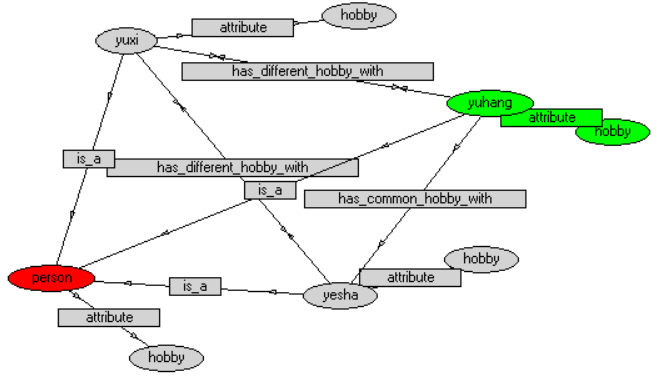
r([t("?a:hobby","is","?b"),t("?c:hobby","is","?b"),t("?a","differs","?c")],[t("?a","has\_common\_hobby\_with","?c")]).

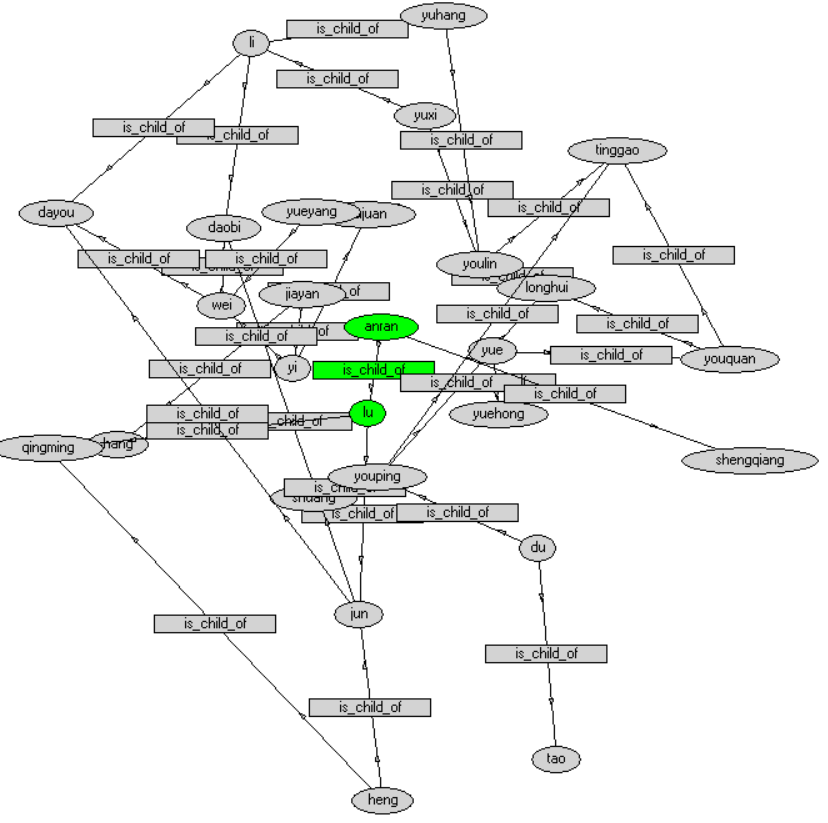
r([t("?a:hobby","is","?b"),t("?c:hobby","is","?d"),t("?b","differs","?d")],[t("?a","has\_different\_hobby\_with","?c")]).

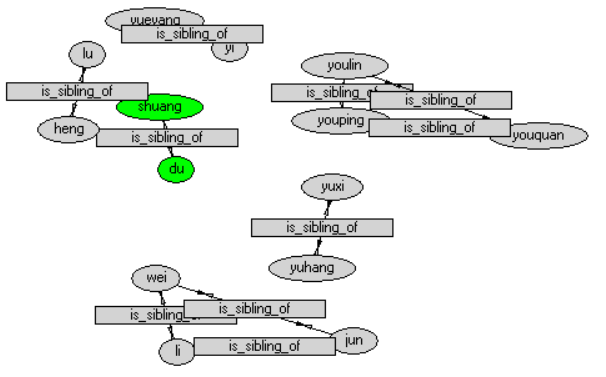
Rules definition

1. If x is y’s father or mother, x is y’s parent.
2. If y is x’s father or mother, x is y’s child.
3. If x is y’s child, and x is male, x is y’s son.
4. If x is y’s child, and x is female, x is y’s daughter.
5. If x is the father of y’s parent, x is y’s grandfather.
6. If x is the mother of y’s parent, x is y’s grandmother.
7. If x and y have a same parent, x and y are siblings.
8. If x is a sibling of y’s parent, and x is male, x is y’s uncle.
9. If x is a sibling of y’s parent, and x is female, x is y’s aunt.
10. If x is a child of a sibling of y’s parent, x is y’s cousin.
11. If x and y have a same child, x and y are spouses.
12. If a is a b, and b has attribute c, a has attribute a: c.
13. If a’s hobby is the same as b’s hobby, a has common hobby with b.
14. If a’s hobby is different from b’s hobby, a has different hobby with b.

Screenshots







Other relations are not suitable for showing, they are too complex in semantic network.

Summary

I have translated prolog into semantic environment, and add one attribute to one ontology with three rules. And try to figure out the differences between them.

The main difference between prolog and semantic is ontology. Prolog can only describe the relations between two entities. However, semantic can define the attributes of entity using ontology. What’s more, semantic can also use attributes to define the relations between entities, which can’t be done by prolog.