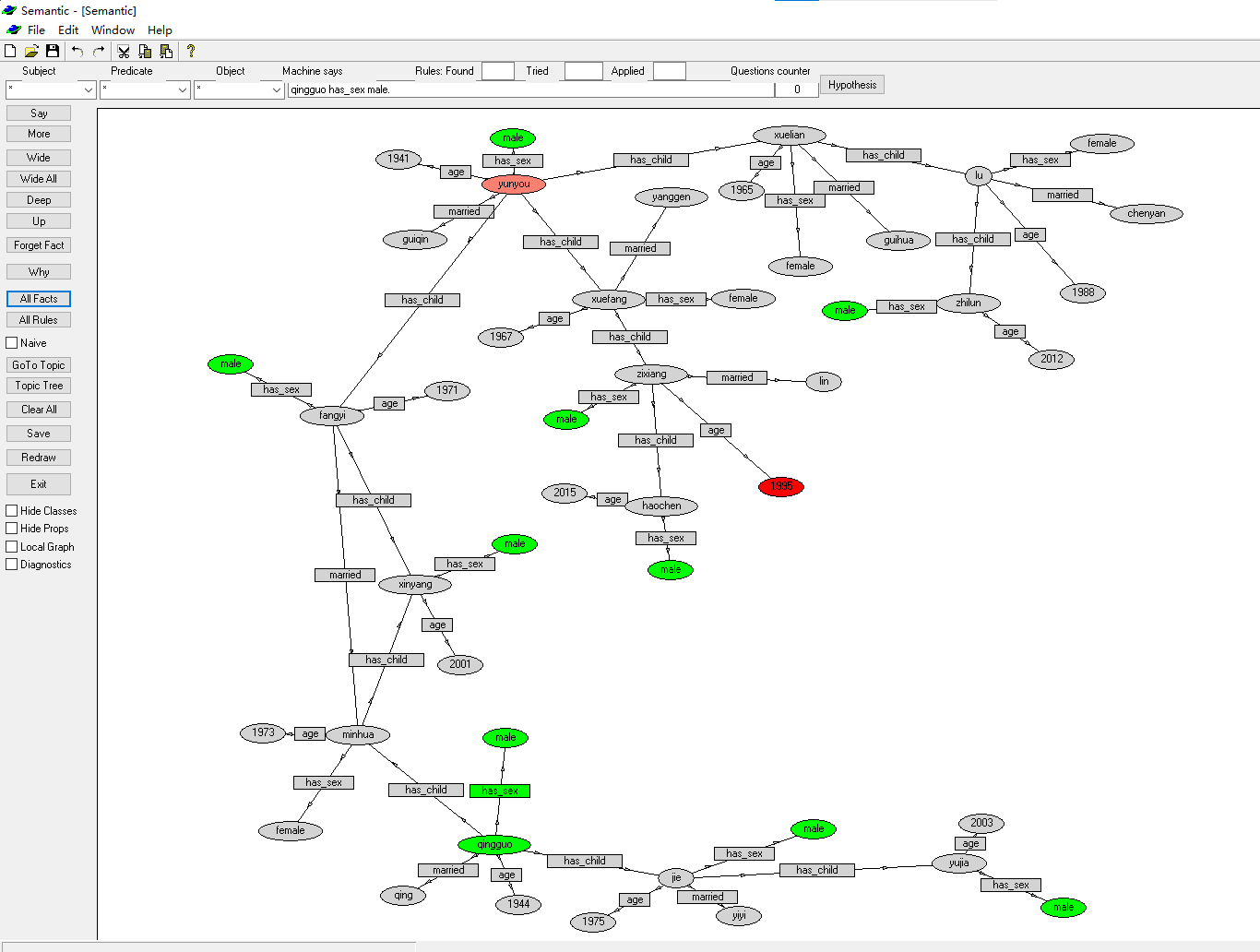
AI Systems Work2

Name: CAO Xinyang

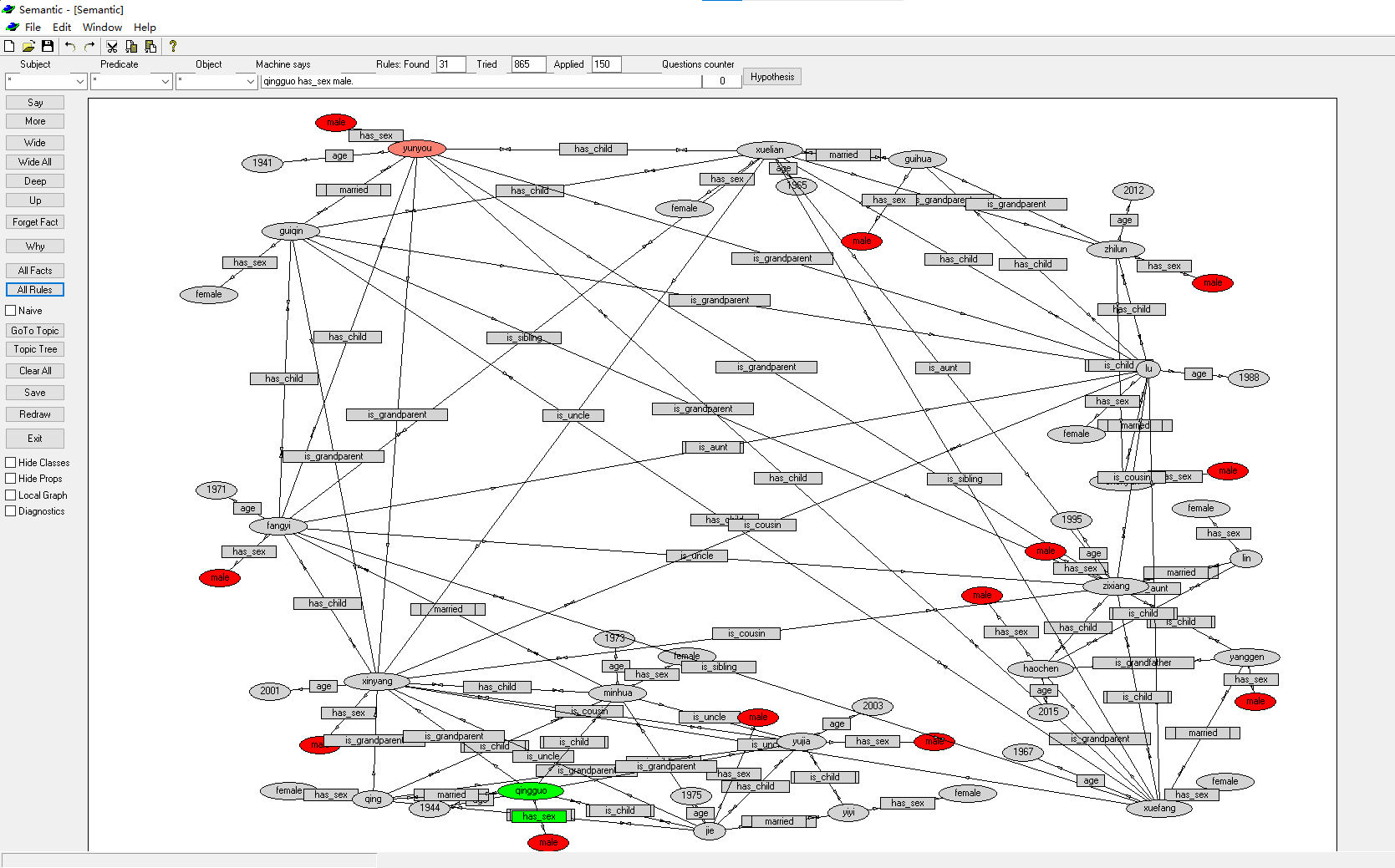
HDU ID: 20321308

1. Screenshots with the goal resolutions.

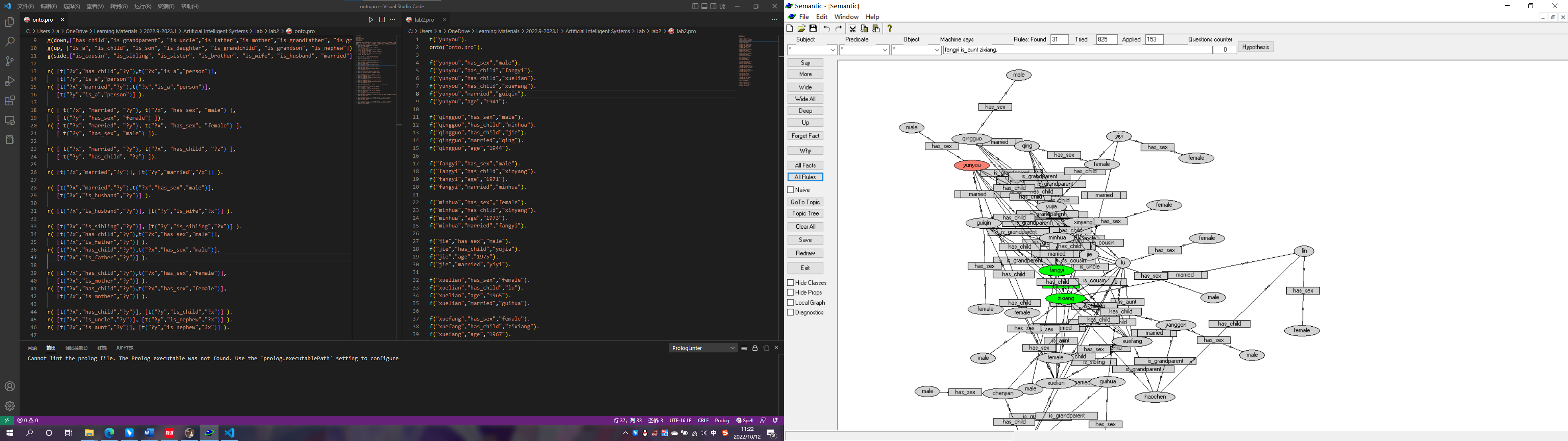
All Facts:



All Rules:



Screenshot on the working time:



2. Codes

lab2.pro：

t("yunyou").

onto("onto.pro").

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

f("yunyou","has\_child","fangyi").

f("yunyou","has\_child","xuelian").

f("yunyou","has\_child","xuefang").

f("yunyou","married","guiqin").

f("yunyou","age","1941").

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

f("qingguo","has\_child","minhua").

f("qingguo","has\_child","jie").

f("qingguo","married","qing").

f("qingguo","age","1944").

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

f("fangyi","has\_child","xinyang").

f("fangyi","age","1971").

f("fangyi","married","minhua").

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

f("minhua","has\_child","xinyang").

f("minhua","age","1973").

f("minhua","married","fangyi").

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

f("jie","has\_child","yujia").

f("jie","age","1975").

f("jie","married","yiyi").

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

f("xuelian","has\_child","lu").

f("xuelian","age","1965").

f("xuelian","married","guihua").

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

f("xuefang","has\_child","zixiang").

f("xuefang","age","1967").

f("xuefang","married","yanggen").

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

f("zixiang","has\_child","haochen").

f("zixiang","age","1995").

f("zixiang","married","lin").

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

f("lu","has\_child","zhilun").

f("lu","age","1988").

f("lu","married","chenyan").

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

f("xinyang","age","2001").

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

f("yujia","age","2003").

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

f("zhilun","age","2012").

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

f("haochen","age","2015").

onto.pro：

c("person").

pr(["name"]).

pr(["age", "old"]).

pr(["has\_sex"]).

pr(["male"]).

pr(["female"]).

g(down,["has\_child","is\_grandparent", "is\_uncle","is\_father","is\_mother","is\_grandfather", "is\_grandmother"]).

g(up, ["is\_a", "is\_child", "is\_son", "is\_daughter", "is\_grandchild", "is\_grandson", "is\_nephew"]).

g(side,["is\_cousin", "is\_sibling", "is\_sister", "is\_brother", "is\_wife", "is\_husband", "married"]).

r( [t("?x","has\_child","?y"),t("?x","is\_a","person")],

[t("?y","is\_a","person")] ).

r( [t("?x","married","?y"),t("?x","is\_a","person")],

[t("?y","is\_a","person")] ).

r( [ t("?x", "married", "?y"), t("?x", "has\_sex", "male") ],

[ t("?y", "has\_sex", "female") ]).

r( [ t("?x", "married", "?y"), t("?x", "has\_sex", "female") ],

[ t("?y", "has\_sex", "male") ]).

r( [ t("?x", "married", "?y"), t("?x", "has\_child", "?z") ],

[ t("?y", "has\_child", "?z") ]).

r( [t("?x","married","?y")], [t("?y","married","?x")] ).

r( [t("?x","married","?y"),t("?x","has\_sex","male")],

[t("?x","is\_husband","?y")] ).

r( [t("?x","is\_husband","?y")], [t("?y","is\_wife","?x")] ).

r( [t("?x","is\_sibling","?y")], [t("?y","is\_sibling","?x")] ).

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

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

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

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

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

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

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

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

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

r( [t("?x","is\_uncle","?y")], [t("?y","is\_nephew","?x")] ).

r( [t("?x","is\_aunt","?y")], [t("?y","is\_nephew","?x")] ).

r( [t("?x","has\_child","?y"),t("?y","has\_child","?z")],

[t("?x","is\_grandparent","?z")] ).

r( [t("?x","is\_grandparent","?y"),t("?x","has\_sex","male")],

[t("?x","is\_grandfather","?y")] ).

r( [t("?x","is\_grandparent","?y"),t("?x","has\_sex","male")],

[t("?x","is\_grandfather","?y")] ).

r( [t("?x","is\_grandparent","?y"),t("?x","has\_sex","female")],

[t("?x","is\_grandmother","?y")] ).

r( [t("?x","is\_grandparent","?y"),t("?x","has\_sex","female")],

[t("?x","is\_grandmother","?y")] ).

r( [t("?x","has\_child","?y"),t("?x","has\_child","?z"),t("?y","differs","?z")],

[t("?y","is\_sibling","?z")] ).

r( [t("?x","is\_sibling","?y"),t("?x","has\_sex","male")],

[t("?x","is\_brother","?y")] ).

r( [t("?x","is\_sibling","?y"),t("?x","has\_sex","male")],

[t("?x","is\_brother","?y")] ).

r( [t("?x","is\_sibling","?y"),t("?x","has\_sex","female")],

[t("?x","is\_sister","?y")] ).

r( [t("?x","is\_sibling","?y"),t("?x","has\_sex","female")],

[t("?x","is\_sister","?y")] ).

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

[t("?y","is\_cousin","?a")] ).

r( [t("?x","has\_child","?y"),t("?x","is\_sibling","?z"), t("?z","has\_sex", "male")],

[t("?z","is\_uncle", "?y")] ).

r( [t("?x","has\_child","?y"),t("?x","is\_sibling","?z"), t("?x","has\_sex","male")],

[t("?z","is\_uncle", "?y")] ).

r( [t("?x","has\_child","?y"),t("?x","is\_sibling","?z"), t("?z","has\_sex", "female")],

[t("?z","is\_aunt", "?y")] ).

r( [t("?x","has\_child","?y"),t("?x","is\_sibling","?z"), t("?x","has\_sex","female")],

[t("?z","is\_aunt", "?y")] ).

3. Summary

With software Semantic we can quickly build a diagram to clarify the relationship between each object. This is a very useful software. We should master this skill expertly.

That’s all for lab work2, thank you for checking.