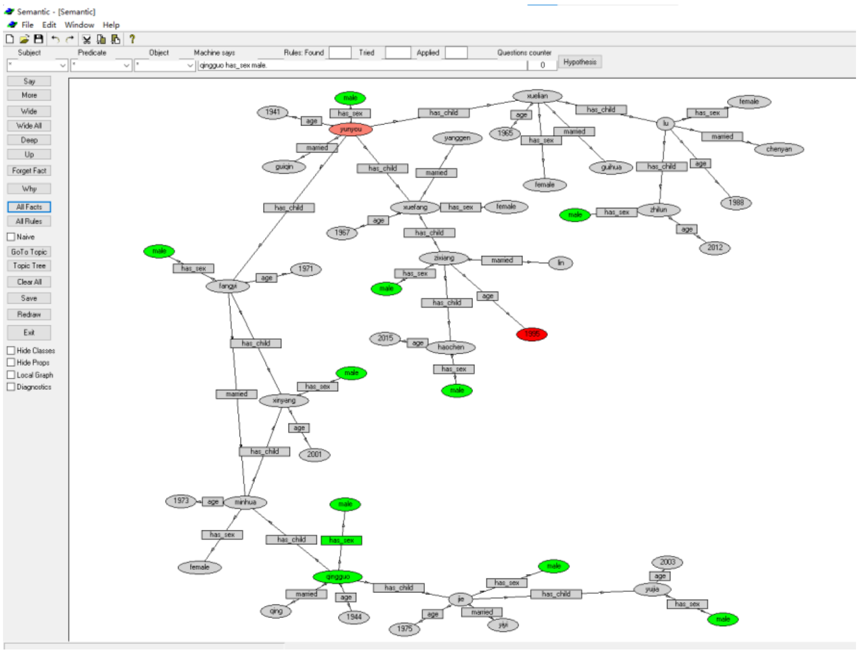
AI Systems Work2

Name: CAO Xinyang
HDU ID: 20321308

1. Screenshots with the goal resolutions.

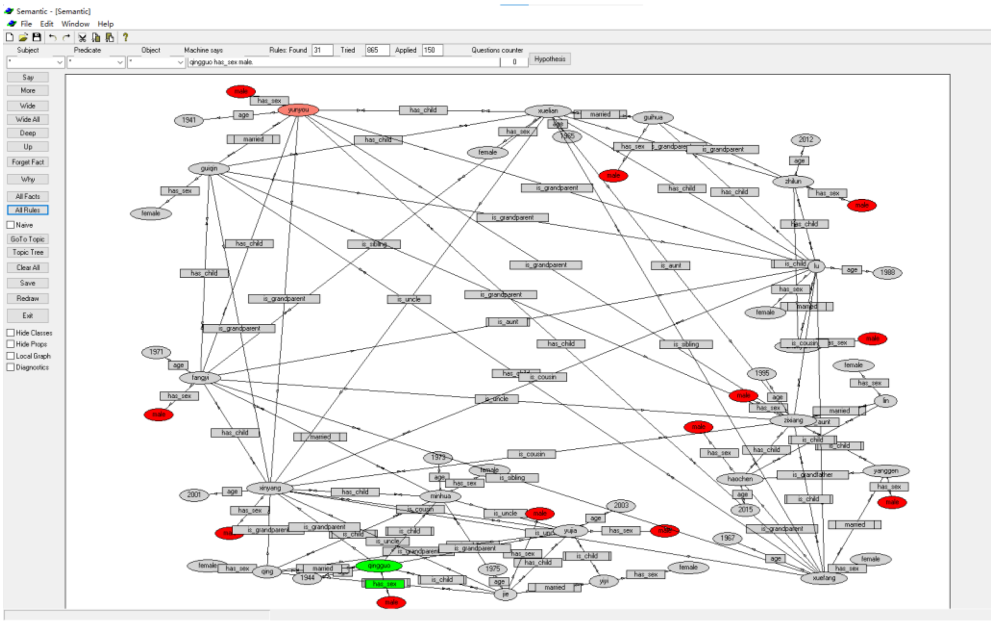
All Facts:



Semantic - [Semantic]

Subject: *
Predicate: *
Object: *
Machine says: qingguo has_sex male.
Rules: Found
Tried
Applied
Questions counter: 0

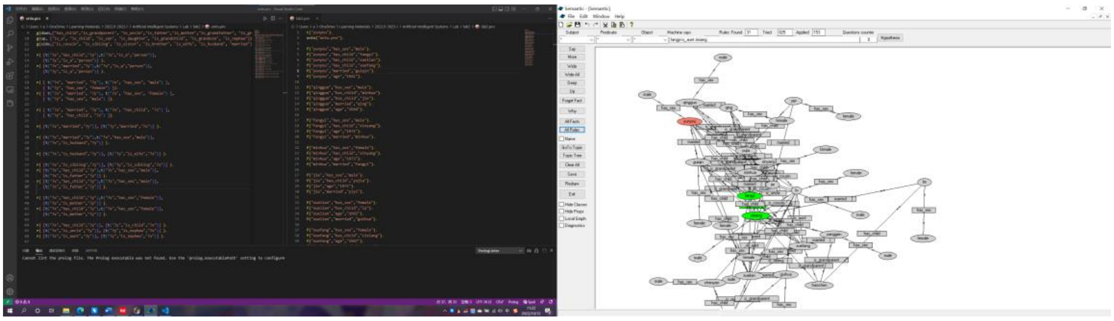
All Rules:



Semantic - [Semantic]

Subject: *
Predicate: *
Object: *
Machine says: qingguo has_sex male.
Rules: Found 31
Tried 865
Applied 150
Questions counter: 0

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.