CIS 671: Programming Assignment No.4

Sameer Shende
Matthew Giger
Junying Wang
Saikiran Sheth

November 29, 1995

1. Discussion

In this programming assignment, we are trying to create a "learning agent" that will learn some knowledge based upon the background knowledge which was defined by the family tree shown in Figure 21.5 on page 638 of the text. The background knowledge was defined in the file named "background.scm". The frustrated the thing is that we did not get a working version of the learning agent even though we tried our best to implemente it.

2. Implementation

- The hypo function

  One experimental algorithm for hypo would be defined as follows:

```
--------------------------------------------------------------------
function hypo(relationship, background, predicates)
         returns a set of Horn clauses
  inputs: relationship, the given relationship, e.g. (uncle ?x ?y)
          background,   the given background knowledge
          predicates,   the related predicates, e.g. parent, male, female
begin
  binary-subset = get all the binary relationship from the background
  unary-subset  = get all the unary relationship from the background
  relation-set  = get all the relation pairs from the background
  allpath = get all path which was related with relationship
  maximumset = get all the relationship set
  generalizableset = substitute the variable to the maximumset eliminate
                     the duplicates
  return generalizableset
end
--------------------------------------------------------------------
```

  The **hypo** can propose a most specific list of (conjunction of) conditions for a given *relationship*, based upon the content of a given list of *background* descriptions in terms of the given list of *predicates*. The algorithm should do this by finding a maximal set of common, generalizable set of conditions in the background that is common to all individual terms seen to share the given relationship.

- The generalize condition function
  The generalize condition function will substitute the variables to the maximumset which eliminate the duplicates. If the input is as follows:

  ```
  ((grandfather philip william)
    (parent charles william)
    (parent philip charles)
    (male charles)
    (male william)
    (male philip))
  ```

  The output will be as follows:

  ```
  (parent ?v1 ?y) (parent ?x ?v1) (male ?v1) (male ?y) (male ?x))
  ```

- Results
  What is the result of
  (hypo '(uncle ?x ?y) background '(parent male female))?
  It would be nice if it were something like
  ((parent ?v1 ?y) (parent ?v2 ?v1) (parent ?v2 ?x) (male ?x))
- Kinship relationships.
  The background is a complete set of kinship relationships for a particular set of people that is sufficient to illustrate the following relationships between people: parent, son, daughter, sister, brother, uncle, aunt, cousin, grandparent, and grandchild; information regarding the unary predicates male and female is included.
- Extra Credit Part
  Theorically, it is possible to combine a set of rule conditions to simplify the background, using one rule conclusion as a rule condition for another. For example, if we have the background:

  $$father(a,b) \wedge father(b,c) \wedge father(c,d)$$

  then by combine a set of rule conditions:

  $$father(a,b) \wedge father(b,c) \Rightarrow grandfather(a,c)$$

  we can simplify the background to as follows:

  $$grandfather(a,c) \wedge father(c,d)$$

3. Software

The software for the project is located in directory:

`/home/grads/sameer/cis671/p4`

The files are as follows :

- satisfy_test.sch - the satisfy routine which takes propositions and facts.
- match.sch - the match routine which matches a pattern with an expression.
- background.scm - the background knowledge that determine the family relationships.
- hypo.scm - the main funtion and subfunction for the hypo algorithm.

The script files for the main program on kinship relationships and other test cases are stored as follows :

- result.script - the expected test result only.

4. Conclusions

5. Appendix

The appendix contains the source codes and the script files for the assignment.