3.1 Are the birds Flying South:

p = Birds are flying south

q = Leaves are turning

r = It is Fall

s = It is cold

1) (p^q) => r
   ~(p^q) V r
   ~p V ~q V r
2) r => s
   ~r V s
3) q ^ ~s
4) q
5) ~s

Applying resolution to 2 and 5

6) ~r

Applying resolution to 1 and 6

7) ~p V ~q

Applying resolution to 4 and 7

8) ~p

Knowledge base input:

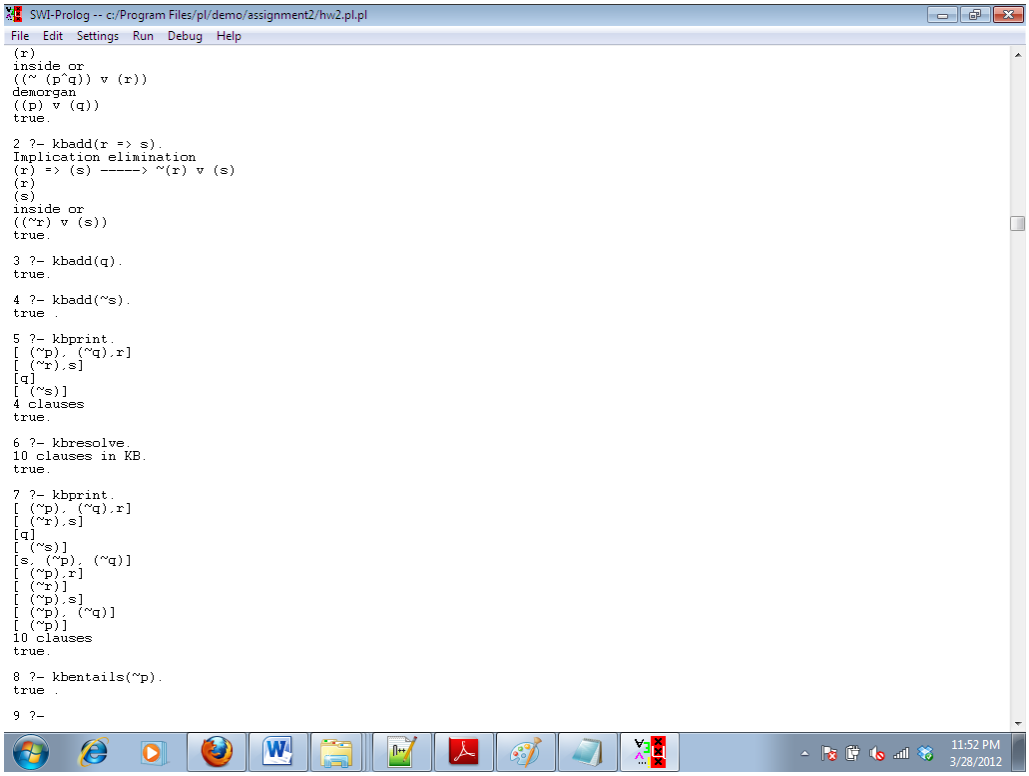
kbadd((p^q) => r)

kbadd(r => s)

kbadd(q)

kbadd(~s)

Prolog Session:



Result: The birds are not flying south

3.2 Solve the murder:

p = Adam speaking truth

q = Bob speaking truth

r = Chuck speaking truth

lc = Chuck likes victim

kb = Bob knows victim

tb = Bob wasn't in town when he died

ka = Adam knows victim

kb = Bob knows victim

tb = Bob wasn't in town when he died

ka = Adam knows victim

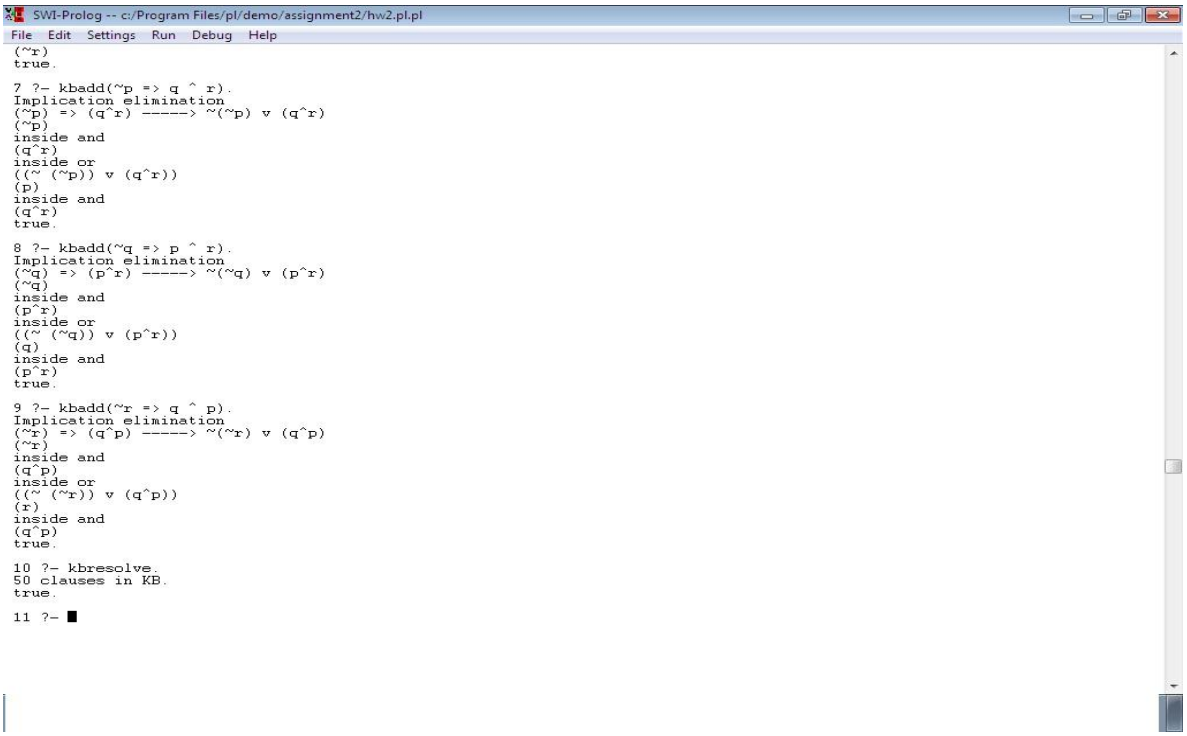
kb = Bob knows victim

Below are the facts that are derived from the given conversation which are given to the inference system. Some of the unwanted terms like out of town ~tb are omitted. Depending on the size of the knowledge base the number of inference will be large. Thus unwanted cases are omitted.

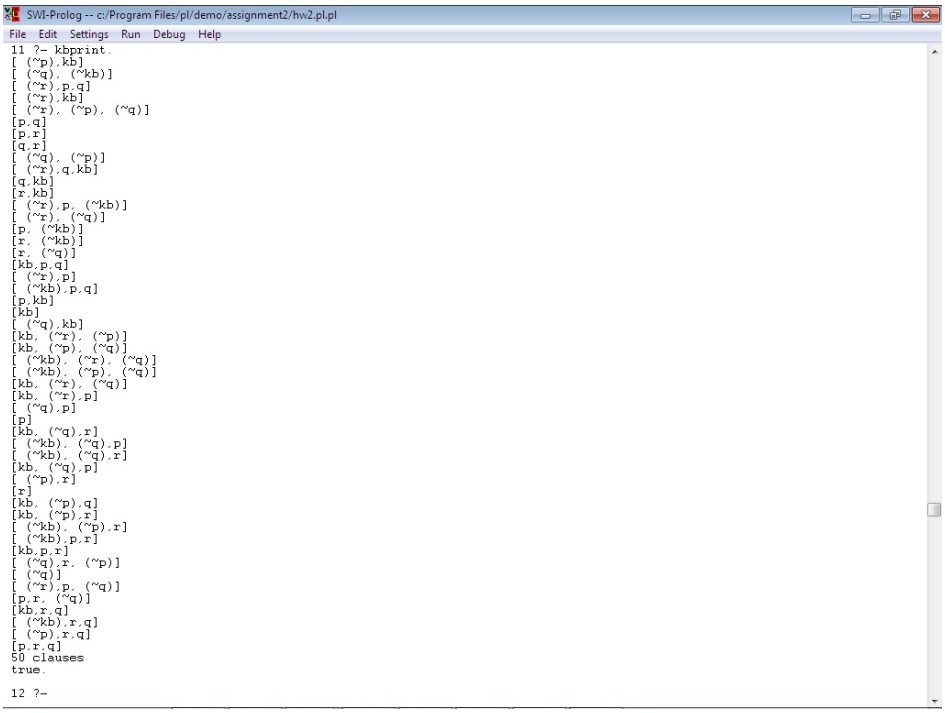
1) p => kb
2) q => ~kb
3) r => p v q
4) r => kb
5) r => ~p v ~q
6) ~p v ~q v ~r
7) ~p => q ^ r
8) ~q => p ^ r
9) ~r => q ^ p

Prolog Sessions:

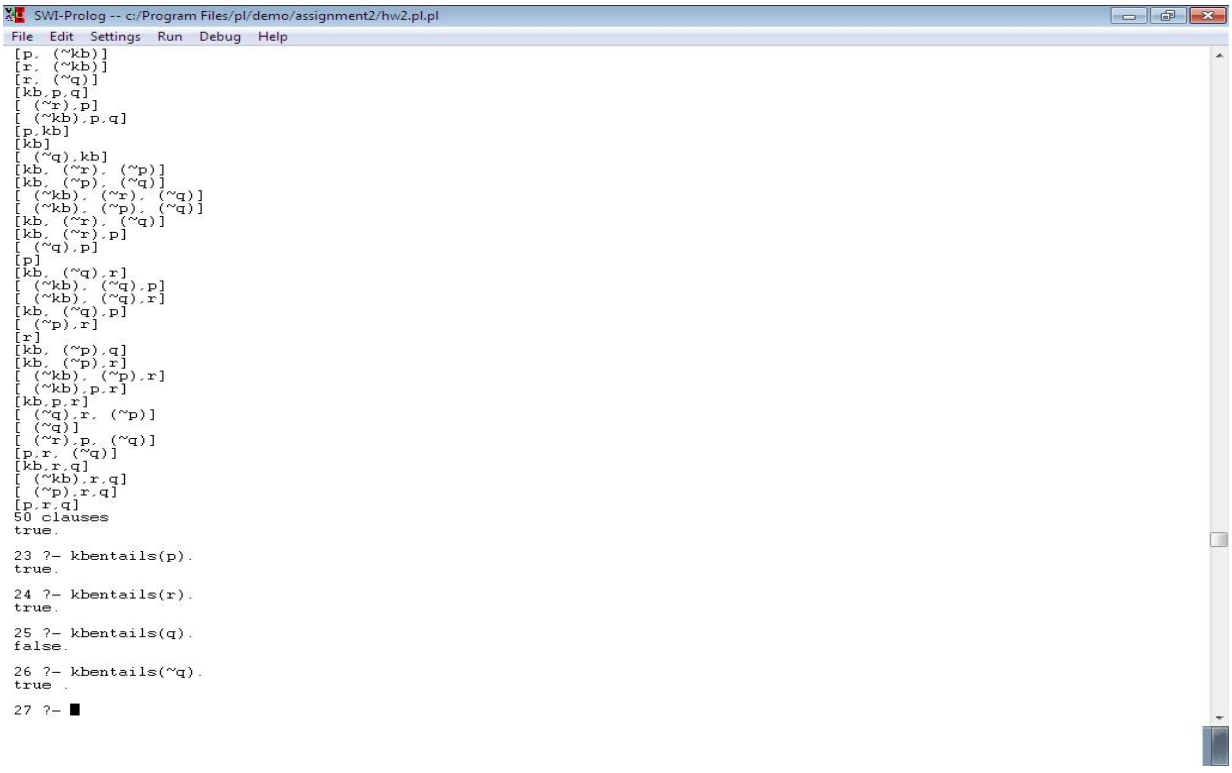
***kbadd, kbresolve**



***Content of knowledge base after applying resolution:**



***Solution: The given model entails p – Adam is saying truth, r – Chuck is saying truth, ~ q – Bob is saying truth**



Solution :- Bob is the murderer.

3.3 Restaurant:

3.3.1 First Order Logic:

1) Manager(X) => ~(Server(X) V Chef(X))

2) Server(X) => ~(Manager(X) V Chef(X))

3) Chef(X) => ~(Manager(X) V Server(X))

4) Friend(Adam, BB)

5) Younger(Adam, Bob)

6) Eyes(Adam, Brown)

7) Server(X) => Eyes(X, Blue)

8) Brother(Bob, BB)

8) Ax[Manager(X) => Ay ~ Younger(X,Y)]

9) Ax[Manager(X) => Ay ~Brother(X,Y)]

Applying CNF for clause 1 :

11) ~Manager(X) V ~ (Server(X) V Chef(X))

(~Manager(X) V ~Server(X)) ^ (~Manager(X) V ~Chef(X))

Similarly applying CNF for clauses 2 and 3, we get

12) (~Server(X) V ~Manager(X)) ^ (~Server(X) V ~Chef(X))

3) (~Chef(X) V ~Manager(X)) ^ (Chef(X) V ~Server(X))

14) ~Server(X) V Eyes(X, Blue)

15) Ax[Manager(X) => Ay ~Younger(X,Y)]

Applying CNF

Ax[~Manager(X) V Ay ~Younger(X,Y)]

~Manager(X) V Younger(X,Y)

16) Ax[Manager(X) => Ay ~Brother(X,Y)]

Applying CNF

Ax[~Manager(X) V Ay ~Brother(X,Y)]

~Manager(X) V Brother(X,Y)

Below are the inferences:

17) Manager(Adam) V Manager(Bob) V Manager(Chuck)

18) Server(Adam) V Server(Bob) V Server(Chuck)

19) Chef(Adam) V Chef(Bob) V Chef(Chuck)

20) Manager(X) V Chef(X) V Server(X)

21) Eyes(X, Blues) => Eyes(X, Brown)

Applying CNF

~Eyes(X, Blue) V ~Eyes(X, Brown)

Applying resolution to 21 and 14, we get

22) ~Server(X) V ~Eyes(X, Brown)

Applying unification to 22 and 6

~Server (Adam) V ~Eyes (Adam, Brown)

Applying resolution to 6 and 22

23) ~Server (Adam)

Applying unification to 20 and 21

24) Manager (Adam) V Chef (Adam) V Server (Adam)

Applying resolution to 5 and 15

25) ~Manager (Adam)

Applying resolution to 23 and 24

26) Manager (Adam) V Chef (Adam)

Applying resolution to 25 and 26

27) Chef (Adam)

Applying converse to 16

28) Ax[Manager(X) => Ay ~Brother(X,Y)]

Ay[Ax ~(~Brother(X,Y)) => ~Manager(X)]

Brother(X,Y) => ~Manager(X)

~Brother(X,Y) V ~Manager(X)

Applying unification and resolution to 8 and 28

29) Brother (Bob, BB)

~ Brother(X,Y) V ~Manager(X)

~Brother (Bob, BB) V ~ Manager(Bob)

Brother(Bob, BB)

~Manager(Bob)

30) We know that,

Manager(Bob) V Chef(Bob) V Server(Bob)

Applying resolution to 30 and 31, we get

Chef(Bob) V Server(Bob)

From 27, we can infer that

31) ~Chef(Bob)

Applying resolution to 31 and 27

32) Server(Bob)

Applying resolution to 17 and 25

33) Manager(Adam) V Manager(Bob) V Manager(Chuck)

Manager (Bob) V Manager(Chuck)

Applying resolution with 29

Manager(Chuck)

3.3.2 First Order Logic to Propositional Logic:

Applying Universal instantiation to all the first order logic predicates

1) FriendAdamBB

2) YoungerAdamBob

3) EyesAdamBrown

4) BrotherBobBB

5) ~Manager(X) V ~Server(X)

~ManagerAdam V ~ServerAdam

~ManagerBob V ~ServerBob

~ManagerChuck V ~ServerChuck

6) ~Manager(X) V ~Chef(X)

~ManagerAdam V ~ChefAdam

~ManagerBob V ~ChefBob

~ManagerChuck V ~ChefChuck

7) ~Server(X) V ~Manager(X)

~ServerAdam V ~ManagerAdam

~ServerBob V ~ManagerBob

~ServerChuck V ~ManagerChuck

8) ~Server(X) V ~Chef(X)

~ServerAdam V ~ChefAdam

~ServerBob V ~ChefBob

9) ~Chef(X) V ~Manager(X)

~ChefAdam V ~ManagerAdam

~ChefBob V ~ManagerBob

~ChefChuck V ~ManagerChuck

10) Chef(X) V ~Server(X)

ChefAdam V ~ServerAdam

ChefBob V ~ServerBob

ChefChuck V ~ServerChuck

11) ~Server(X) V Eyes(X, Blue)

~ServerAdam V EyesAdamBlue

~ServerBob V EyesBobBlue

~ServerChuck V EyesChuckBlue

12) ~Manager(X) V Younger(X,Y)

~ManagerAdam V YoungerAdamBob

~ManagerAdam V YoungerAdamChuck

~ManagerBob V YoungerBobAdam

~ManagerBob V YoungerBobChuck

~ManagerChuck V YoungerChuckAdam

~ManagerChuck V YoungerChuckBob

13) ~Manager(X) V Brother(X,Y)

~ManagerAdam V BrotherAdamBob

~ManagerAdam V BrotherAdamChuck

~ManagerBob V BrotherBobAdam

~ManagerBob V BrotherBobAdam

~ManagerChuck V BrotherChuckAdam

~ManagerChuck V BrotherChuckBob

14) ManagerAdam V ManagerBob V ManagerChuck

15) ServerAdam V ServerBob V ServerChuck

16) ChefAdam V ChefBob V ChefChuck

17) Manager(X) V Chef(X) V Server(X)

Manager(Adam) V Chef(Adam) V Server(Adam)

Manager(Bob) V Chef(Bob) V Server(Bob)

22) ~Eyes(X, Blue) V ~Eyes(X, Brown)

~EyesAdamBlue V ~EyesAdamBrown

~EyesBobBlue V ~EyesBobBrown

~EyesChuckBlue V ~EyesChuckBrown

23) ~Server(X) V ~Eyes(X, Brown)

~ServerAdam V ~EyesAdamBrown

~ServerBob V ~EyesBobBrown

~ServerChuck V ~EyesBobChuck

24) ~Server (Adam)

25) Manager (Adam) V Chef (Adam) V Server (Adam)

26) ~Manager (Adam)

27) Manager (Adam) V Chef (Adam)

28) Chef (Adam)

29) Brother(X,Y) V ~Manager(X)

30) ~Manager(Bob)

31) Chef(Bob) V Server(Bob)

32) ~Chef(Bob)

33) Server(B)

34)Manager(Chuck)

3.3.3 Entering Propotional Logic in Kb of our inference system:

When all the propositional clauses are entered into the knowledge base of the inference system, it infers exponential amount of further clauses. This halts the working of the inference system. Inference system hangs.