“Can we do this for our KKR project?” Asking the right questions in open-world planning problems

Taeyeong Choi and **Sailik Sengupta** and **Sarath Sreedharan** and **Aditya Vallabhajosyula**
CSE 579 Knowledge Representation, Arizona State University
{tchoi4,sailiks,ssreedh3,avallab3}@asu.edu

Abstract

Artificial Intelligent (AI) agents planning in open world environment often face uncertainty about the domain physics associated with the environment or objects present in it. Existing literature tries to generate *robust plans* in such scenarios, i.e. plans which are executable in the largest set of possible domain models generated by choosing to make an uncertain precondition or effect certain. With the recent interest in Human Aware AI systems (HAAI), using the presence of a human, who has knowledge about the domain, can help an agent to ask questions and resolve uncertainty, thus reducing the search space over the possible domain models. Now, asking the correct question(s) is not a trivial problem. In this paper, we address this question with the use of LP^{MLN}– a technique for generating answer sets with weighted rules. We show a way of encoding a planning problem in this formalism such that the answer sets generated provide the questions that should be asked. We evaluate our approach on a set of task planning problems for a delivery robot. We end our discussion by talking about the future directions on how this work can be strengthen.

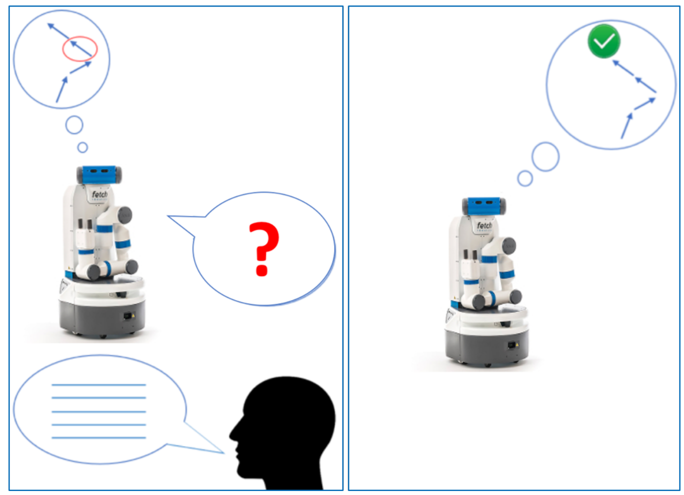

Figure 1: (1) Robot asks minimum questions to the domain expert to resolve uncertainty. (2) Robot successfully executes plan.

Introduction

Planning Agents, in real world scenarios, often have to account of uncertainty in the domain. This uncertainty can arise when an environment is stochastic is nature or if there is uncertainty/incompleteness about the knowledge of the underlying deterministic domain. As an example of the latter case, consider a delivery domain where a robot can deliver mail or coffee to a professor. The robot might not be sure as to whether (i) coffee is present in the kitchen, (ii) the professor is present in his room, (iii) mail for the professor is in the mail box, (iv) common corridor to the mailroom is blocked by people.

The best a planning agent can do in such cases is to generate plans that are robust enough to work in the maximum number of possible domain models induced by the uncertain domain model (Nguyen, Kambhampati, and Do 2013). Unfortunately, this process is inefficient (specifically $\#P$-complete). Human in the Loop planning (HILP) is ubiquitous in many complex planning environments today. In fact, recent studies in mixed initiative planning have shown that planners are more efficient when humans help in the planning process (Kim, Banks, and Shah 2017). With this interest in Human-Aware Artificial Intelligence, we try to see if the present of humans in the planning environment can help us in efficiently generating robust plans.

In our setup, we have an underlying deterministic model with some uncertainty regarding this model. We assume the human in the loop is a domain expert, who has information about these uncertain domain ‘rules’. When asked about whether a rule is true, (s)he can reply with a yes or know answer. Given this setup, when the robot is given a task, if it chooses to execute a generated *robust* plan, it may execute successfully in the domain or land up in catastrophic circumstances. On the other hand, it may choose to ask questions to the domain expert, and based on his answer, generate a plan that will execute with 100% certainty (provided the domain expert is always correct or does not lie).

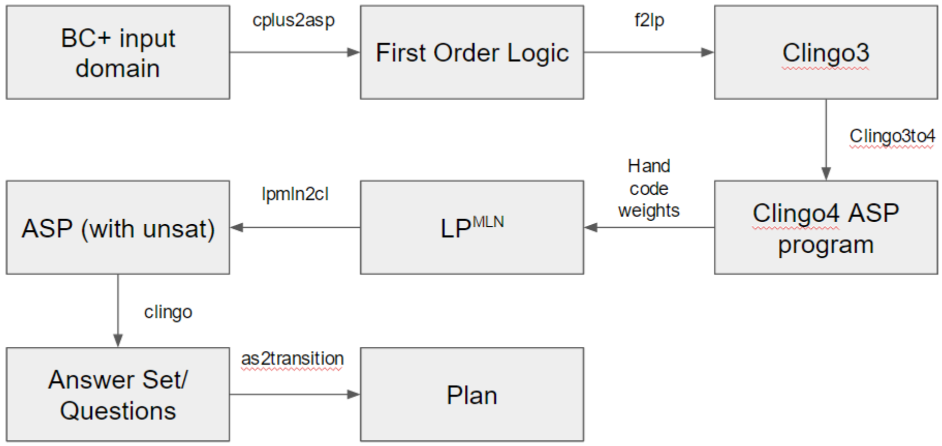

Figure 2: Using BC+ for writing planning domains and LP^{MLN} for minimum question and plan generation.

Now, form the agent's perspective, it may be useful to ask the domain expert questions about all the uncertain rules in the domain, after which it can always produce a successful plan for any task (as long as the domain does not evolve). But this may be highly inefficient and annoying to the human in the loop. Consider our delivery robot was asked to deliver coffee to the professor. If it asks questions about all the four uncertain rules mentioned above, the domain expert is confused (and annoyed) since the question about available mails and blocked common corridor do not make sense at all in the context of delivering coffee.

Finding the least set of questions among the set of all possible questions is the problem we address in the paper. In order to achieve this, we encode all uncertain knowledge in the planning problem as soft constrains. These soft constraints are given negative weights in LP^{MLN}. Thus, any answer set (possible plan) that includes one of this has a lower weight. The answer set with the highest weight uses the smallest number of uncertain rules. We show that these rules used can essentially be transformed into the question that needs to be asked.

In the next section, we talk about an example planning domain for the delivery robot, highlighting rules that we may be uncertain about and formulate this domain with negative weights in LP^{MLN} to generate the right set of questions. We then talk of our system architecture on how this was implemented in a human-robot environment. We show a few experimental scenarios based on this domain. Lastly, we highlight the individual work of our team members and conclude the paper with future directions.

Formulation

In order to describe the formulation, we will use the examples from the domain we designed. Although we have to ideally use an LP^{MLN} domain formulation for the purposes of our project, coding elaborate planning domains in ASP and then adding weights to it is often cumbersome and error prone. Thus, we use the action language BC+ (Babb and Lee 2015) to design our domain (available in Appendix A). But since this does not support the addition of weights to uncertain rules (and apparently it might not be trivial to do so) we will use the generated ASP file for introducing uncertainty. This entire work flow can be seen in Figure 2. To aid others in executing this workflow till the hard coding of weights is needed, we develop a script (available in Appendix B).

To give a concrete example of what we did to generate minimum amount of questions, let us consider an example in our domain. Suppose the robot is asked to execute the task of delivering coffee to the professor. Let us say, in the initial state it is uncertain about the following predicates in the BC+ domain.

```
:- query
maxstep :: 1..20;
0:
  -blocked(c) &
  in(coffee, k) &
  in(mail, m) &
  in(mail, m) &
  ...
maxstep:
  has(coffee, prof).
```

When we convert this to ASP, we associate negative weights with only these predicates to generate the LP^{MLN} file, which looks as follows:

```
-5 :- not c_f_1_blocked(o_c,o_false,0),
   query_label(q_0).
-5 :- not c_f_2_in(o_coffee,o_k,o_true,0),
   query_label(q_0).
-5 :- not c_f_2_in(o_mail,o_m,o_true,0),
   query_label(q_0).
-5 :- not c_f_2_in(o_prof,o_p,o_true,0),
   query_label(q_0).
```

Notice that, if an interpretation (which is going to be a plan in our case) in LP^{MLN}uses all four of these constraints, then the weight of the interpretation will be less ($\frac{1}{Z}e^{y+4\cdot(-5)}$) than say an interpretation with satisfies two soft constrains ($\frac{1}{Z}e^{y+2\cdot(-5)}$) where Z is the normalization constant and y is the weight of all the other formulas satisfied. To generate the plan, we convert this into ASP that generates unsatclauses for all of these (Lee and Yang 2017). We show conversion for only two such rules of which only one is relevant to the context of delivering coffee, for space considerations.

```
unsat(112,"-5.000000") :-  not c_f_2_in(
    o_coffee,o_k,o_true,0),query_label(q_0).
:-  not c_f_2_in(o_coffee,o_k,o_true,0),
    query_label(q_0), not unsat
    (112,"-5.000000").
:~ unsat(112,"-5.000000").[-5@0,112]
unsat(113,"-5.000000") :-  not c_f_2_in(
    o_mail,o_m,o_true,0),query_label(q_0).
:-  not c_f_2_in(o_mail,o_m,o_true,0),
    query_label(q_0), not unsat
    (113,"-5.000000").
:~ unsat(113,"-5.000000").[-5@0,113]
```

Notice that when the unsat(112,"-5.000000" is a part of the answer-set, the penalty incurred is reduced (due to the -5). Thus, ASP would try to add soft constraints to the answer set to produce optimal models. The detailed output of ASP for the above program can be found in Appendix C. He we only show the section optimal model in the the output:

```
...
Answer: 3
... unsat(108,"-5.000000") unsat
    (113,"-5.000000") ...
Optimization: 10
OPTIMUM FOUND

Models       : 3
  Optimum    : yes
Optimization : 10
```

Notice that for the optimal model found, the 2 unsatpredicates for, one of which (113) is irrelevant to the problem. In the LP^{MLN}this ensures that this constraint does not need to be satisfied since for having a successful plan for delivering coffee does not need to have mailbox in the mail room. Thus, the unsat-s that are absent in the answer-set are the questions that need to be asked. In our case, these can be simply obtained by use a subtracting the set of unsat-s in the answer set from the set of all unsat-s in the clingo code generated from the LP^{MLN}program. These in our case are the unsat(112) and unsat(114). They are about the predicates regarding coffee in the kitchen and professor in the room, both of which are necessary to know for generating a plan that guaranteed success on execution for the task of delivering coffee to the professor.

Although this formulation was developed for generating optimal number of queries, it is general enough to handle cases where uncertain actions or uncertain constraints should be completely avoided to generate *robust* plans. With the capability of handling former case of avoiding uncertain actions, our formulation shows that robust planning is just a special case. With the latter case, we can handle more that what *robust* planning mostly developed for PDDL style planners can generate. To show these cases, we need to make specific changes in the LP^{MLN}file that we describe now.

Avoid uncertain actions for generating *robust* plans Consider an action **unblock** in our domain, that when executed by a robot can clear/unblock a *blocked*(c) where c is a corridor. But say, this may not always be successful many a times. In this case, the robot should ideally come up with a plan that avoids this action altogether and finds a different path to the professor's room from the kitchen. To make this possible, the exogenous, precondition and effect rules related to the uncertain action in the ASP program are made soft constrains by adding LP^{MLN}program.

```
% exogenous action rules
-5 :- not 1{c_a_1_unblock(GVAR_location_2,
    LVAR_boolean_1,t-1):s_boolean(
    LVAR_boolean_1)}1,s_location(
    GVAR_location_2).
-5 {c_a_1_unblock(GVAR_location_1,
    GVAR_boolean_1,t-1)} :- true,s_boolean(
    GVAR_boolean_1),s_location(
    GVAR_location_1).
-5 :- c_a_1_unblock(GVAR_location_1,VAR,t-1)
    ,not s_boolean(VAR),s_location(
    GVAR_location_1).
% effect rule
-5 c_f_1_blocked(V_L1,o_false,t) :-
    c_a_1_unblock(V_L1,o_true,t-1),
    s_location(V_L1).
% precondition rule
-5 :- c_a_1_unblock(V_L1,o_true,t-1),
    c_f_1_blocked(V_L1,o_false,t-1),
    s_location(V_L1).
```

Avoiding uncertain predicates for generating *robust* plans As a simple example, consider that we want the robot to avoid the common corridor altogether (since say the domain expert is not sure as to whether it is blocked or not and does not want to check this every time the robot asks this). In our formalism, all one needs to do is have 2 soft constraints for the both the predicates associated with the uncertain predicate. This example should make it clear:

```
-5 c_f_1_blocked(o_c,o_true,0).
-5 c_f_1_blocked(o_c,o_false,0).
```

In this case, the robot avoids the blocked corridor predicate (and any actions that is associated with it).

Experiments with a Robot

In this brief section, we talk about an experiment we performed with a Fetch, which is a one armed robot. Initially, the robot was assigned the task of delivering

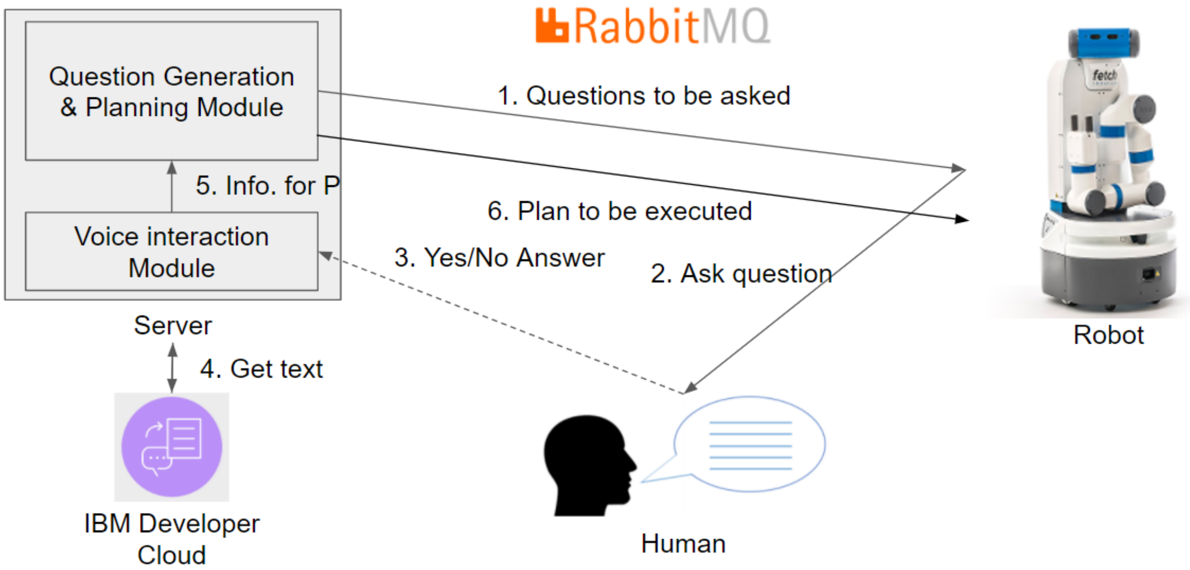

Figure 3: Implementation architecture for the Robot experiment

mail to the professor located at BYENG 5th floor. The locations of the professor's room, the kitchen and the mailroom are hard coded into the 5th floor map that the robot has. We now describe the work flow of what happens (as seen in Figure 3)[1].

The LP^{MLN} module given the goal and uncertainty about the initial predicates generates the set of questions that needs to be asked to the janitor, our domain expert. The questions are mapped to full English sentences hashed by the `unsat` predicates. These are sent to the robot, who then asks the question to the janitor. The janitors reply is sent to the IBM Developer cloud for conversion and the translated texts are sent to the LP^{MLN} module who now makes the soft constrains hard based on the human's answer. It then generates a plan to achieve the goal (if one is feasible) or asks further questions (when not).

Future Work

At present, the negative weights for the soft constraints are given without much formalism, which we plan to define in a context to the robustness score in (Nguyen, Kambhampati, and Do 2013). We hope to incorporate successful plan traces as evidence alongside our LP^{MLN} domains so as to resolve uncertainty about certain rules even without having to ask the janitor, thus reducing load on the human in the loop. We hope to try our developed approach on benchmark domains for robust planning and compare the results with the state-of-the-art solvers.

Contribution of Team Members

Taeyeong Choi Given the scenario design, came up with the BC+ domain that was used for generating the ASP program.

Sailik Sengupta Came up with a scenario that could showcase the main ideas for the project keeping in mind that this was feasible for a robot to execute. Coded the backend module responsible for question and plan generation (this led to uncovering the flow of BC+ to ASP code generation and identifying bug for `lpmln2cl`). Made most of the diagrams and presentations associated with the project. Wrote a backend script for the robot to call. Cinematography.

Sarath Sreedharan Came up with the project idea while we were having a chat on mixed initiative planning in the lab. Helped in suggesting manual fix for the `lpmln2cl` bug. Setup the robot demo code and interaction via RabbitMQ.

Aditya Vallabhajosyula Wrote the code for the audio detection and audio to text and text to audio conversion necessary for the robotic experiment. Modified the domain to create a scenario when human responding with a negative needs re-planning.

[1] Video demo link: https://goo.gl/nZbKz7

References

Babb, J., and Lee, J. 2015. Action language bc+: Preliminary report. In *AAAI*, 1424–1430.

Kim, J.; Banks, C. J.; and Shah, J. A. 2017. Collaborative planning with encoding of users' high-level strategies.

Lee, J., and Yang, Z. 2017. Lpmln, weak constraints, and p-log. In *AAAI*.

Nguyen, T. A.; Kambhampati, S.; and Do, M. 2013. Synthesizing robust plans under incomplete domain models. In *Advances in Neural Information Processing Systems*, 2472–2480.

Appendix A

```
:- sorts
  robot;
  human;
  thing;
  location.

:- objects
  robo                                                         :: robot;
  prof                                                         :: human;
  coffee, mail                                                 :: thing;
  % start, kitchen, mailRoom, profRoom, commonCorridor
  s, k, m, p, c    :: location.

:- variables
  TH                                                           :: thing;
  L1, L2                                                       :: location.

:- constants
  % --- fluent --- %
  loc(robot)                                                   :: inertialFluent(location);
  has(thing, robot+prof),
  in(thing+prof, location),
  blocked(location)                                            :: inertialFluent;
  edge(location, location)                                     :: sdFluent;
  % --- actions --- %
  pickUp(thing),
  deliver(thing),
  move(location)                                               :: exogenousAction.

% mail, coffee, and prof only can be present at a specific room.
caused false if in(mail, L1)&L1\=m.
caused false if in(coffee, L1)&L1\=k.
caused false if in(prof, L1)&L1\=p.

% --- Graph --- %
% (S,C), (S,K)
% (K,C), (K,S), (K,P)
% (M,C)
% (P,C), (P,K)
% (C,S), (C,K), (C,M), (C,P)
% --- %
default -edge(L1, L2).
caused edge(L1, L2) if  L1=s & (L2=c ++ L2=k) ++
                        L1=k & (L2=c ++ L2=s ++ L2=p) ++
                        L1=m & L2=c ++
                        L1=p & (L2=c ++ L2=k) ++
                        L1=c & (L2=s ++ L2=k ++ L2=m ++ L2=p).

% --- Move --- %
move(L1) causes loc(robo) = L1.
nonexecutable move(L1) if loc(robo) = L1.
nonexecutable move(L1) if blocked(L1).
nonexecutable move(L1) if -edge(loc(robo),L1).

% --- Pick up --- %
pickUp(TH) causes has(TH, robo).
pickUp(TH) causes -in(TH, loc(robo)).
nonexecutable pickUp(TH) if -in(TH, loc(robo)).
nonexecutable pickUp(TH) if has(TH, robo).

% --- Deliver --- %
deliver(TH) causes has(TH, prof).
deliver(TH) causes -has(TH, robo).
```

```
nonexecutable deliver(TH) if -has(TH, robo).
nonexecutable deliver(TH) if -in(prof, loc(robo)).

caused false if has(mail, robo) & has(coffee, robo).

nonexecutable pickUp(TH) & move(L1).
nonexecutable move(L1) & deliver(TH).
nonexecutable pickUp(TH) & deliver(TH).
```

Appendix B

```
#!/bin/sh

usage() {
    echo "Usage: $0 -f <bc+ file>";
    exit 1;
}

while getopts "f:" o; do
    case "${o}" in
        f)
            f=${OPTARG}
            ;;
        *)
            usage
            ;;
    esac
done

if [ -z "${f}" ]; then
    usage
fi

# Convert bc+ file to asp
cplus2asp3.bin "${f}" --symtab-out /tmp/9d05-386f-5ab1-b1a9 --language=bc+ > "/tmp/${f}_fol"
f2lp -i "/tmp/${f}_fol" > "/tmp/${f}_clingo3"
clingo3to4 -o -i -f "/tmp/${f}_clingo3" > "/tmp/${f}_asp"

printf "\nSuccessfully generated the asp problem...\n"

# Convert asp to a lp program with hard constraints
cp "/tmp/${f}_asp" "/tmp/${f}.lpmln"
python -c 'print "\n#include <incmode>."' >> "/tmp/${f}.lpmln"

printf "\nSuccessfully generated the lp problem with hard constrains...\n"

# Check the generated file works
lpmln2cl /tmp/${f}.lpmln 1 -c queryl=q_0 -c maxstep=20 -c minstep=1 -c imin=2 -c imax=21 --
    warn=no-atom-undefined > "/tmp/${f}_output"
printf "\nSuccessfully ran the lpmln2cl on the generated lp problem.\n"
printf "\nOutput file: /tmp/${f}_output\n"
printf "\nDone.\n"

# --- One has to manually add soft contraints here --- #

# --- Viusalize plans from the output file --- #
#as2transition -i /tmp/kr_project_output
```

Appendix C

```
lpmln2cl
Clingo executed with command:
clingo /home/local/ASUAD/ssengu15/scenario1_sarath/out.txt 0 -c queryl=q_0 -c maxstep=20 -c
    minstep=1 -c imin=2 -c imax=21 --warn=no-atom-undefined
clingo version 4.5.4
Reading from ...SUAD/ssengu15/scenario1_sarath/out.txt
Solving...
Solving...
Solving...
Solving...
Solving...
Answer: 1
true s_robot(o_robo) s_human(o_prof) s_thing(o_coffee) s_thing(o_mail) s_location(o_s)
    s_location(o_k) s_location(o_m) s_location(o_p) s_location(o_c) s_robot__prof(o_robo)
    s_robot__prof(o_prof) s_boolean(o_true) s_boolean(o_false) c_f_1_loc(o_robo,o_s,0)
    s_thing__prof(o_coffee) s_thing__prof(o_mail) s_thing__prof(o_prof)
c_f_2_has(o_coffee,o_robo,o_false,0) c_f_2_has(o_coffee,o_prof,o_false,0) c_f_2_has(o_mail,
    o_robo,o_false,0) c_f_2_has(o_mail,o_prof,o_false,0) c_f_2_in(o_coffee,o_s,o_false,0)
    c_f_2_in(o_coffee,o_k,o_true,0) c_f_2_in(o_coffee,o_m,o_false,0) c_f_2_in(o_coffee,o_p,
    o_false,0) c_f_2_in(o_coffee,o_c,o_false,0) c_f_2_in(o_mail,o_s,o_false,0) c_f_2_in(
    o_mail,o_k,o_false,0) c_f_2_in(o_mail,o_m,o_true,0) c_f_2_in(o_mail,o_p,o_false,0)
    c_f_2_in(o_mail,o_c,o_false,0) c_f_2_in(o_prof,o_s,o_false,0) c_f_2_in(o_prof,o_k,o_false
    ,0) c_f_2_in(o_prof,o_m,o_false,0) c_f_2_in(o_prof,o_p,o_true,0) c_f_2_in(o_prof,o_c,
    o_false,0) c_f_1_blocked(o_s,o_true,0) c_f_1_blocked(o_k,o_false,0) c_f_1_blocked(o_m,
    o_false,0) c_f_1_blocked(o_p,o_false,0) c_f_1_blocked(o_c,o_false,0) c_f_2_edge(o_s,o_s,
    o_false,0) c_f_2_edge(o_s,o_m,o_false,0) c_f_2_edge(o_s,o_p,o_false,0) c_f_2_edge(o_k,o_k
    ,o_false,0) c_f_2_edge(o_k,o_m,o_false,0) c_f_2_edge(o_m,o_m,o_false,0) c_f_2_edge(o_m,
    o_c,o_false,0) c_f_2_edge(o_p,o_s,o_false,0) c_f_2_edge(o_p,o_m,o_false,0) c_f_2_edge(o_p
    ,o_p,o_false,0) c_f_2_edge(o_c,o_c,o_false,0) c_f_2_edge(o_s,o_c,o_true,0) c_f_2_edge(o_s
    ,o_k,o_true,0) c_f_2_edge(o_k,o_c,o_true,0) c_f_2_edge(o_k,o_s,o_true,0) c_f_2_edge(o_k,
    o_p,o_true,0) c_f_2_edge(o_m,o_s,o_true,0) c_f_2_edge(o_m,o_k,o_true,0) c_f_2_edge(o_m,
    o_p,o_true,0) c_f_2_edge(o_p,o_c,o_true,0) c_f_2_edge(o_p,o_k,o_true,0) c_f_2_edge(o_c,
    o_s,o_true,0) c_f_2_edge(o_c,o_k,o_true,0) c_f_2_edge(o_c,o_m,o_true,0) c_f_2_edge(o_c,
    o_p,o_true,0) query_label(q_0) c_f_1_loc(o_robo,o_k,1) c_a_1_move(o_k,o_true,0) c_f_2_has
    (o_coffee,o_robo,o_false,1) c_f_2_has(o_mail,o_robo,o_false,1) c_f_2_in(o_coffee,o_s,
    o_false,1) c_f_2_in(o_coffee,o_m,o_false,1) c_f_2_in(o_coffee,o_p,o_false,1) c_f_2_in(
    o_coffee,o_c,o_false,1) c_f_2_in(o_mail,o_s,o_false,1) c_f_2_in(o_mail,o_k,o_false,1)
    c_f_2_in(o_mail,o_p,o_false,1) c_f_2_in(o_mail,o_c,o_false,1) c_f_2_edge(o_s,o_s,o_false
    ,1) c_f_2_edge(o_s,o_m,o_false,1) c_f_2_edge(o_s,o_p,o_false,1) c_f_2_edge(o_k,o_k,
    o_false,1) c_f_2_edge(o_k,o_m,o_false,1) c_f_2_edge(o_m,o_m,o_false,1) c_f_2_edge(o_m,o_c
    ,o_false,1)
c_f_2_edge(o_p,o_s,o_false,1) c_f_2_edge(o_p,o_m,o_false,1) c_f_2_edge(o_p,o_p,o_false,1)
    c_f_2_edge(o_c,o_c,o_false,1) c_f_2_edge(o_s,o_c,o_true,1) c_f_2_edge(o_s,o_k,o_true,1)
    c_f_2_edge(o_k,o_c,o_true,1) c_f_2_edge(o_k,o_s,o_true,1) c_f_2_edge(o_k,o_p,o_true,1)
    c_f_2_edge(o_m,o_s,o_true,1) c_f_2_edge(o_m,o_k,o_true,1) c_f_2_edge(o_m,o_p,o_true,1)
    c_f_2_edge(o_p,o_c,o_true,1) c_f_2_edge(o_p,o_k,o_true,1) c_f_2_edge(o_c,o_s,o_true,1)
    c_f_2_edge(o_c,o_k,o_true,1) c_f_2_edge(o_c,o_m,o_true,1) c_f_2_edge(o_c,o_p,o_true,1)
    c_f_2_has(o_coffee,o_prof,o_false,1) c_f_2_has(o_mail,o_prof,o_false,1) c_f_2_in(o_coffee
    ,o_k,o_true,1) c_f_2_in(o_mail,o_m,o_true,1) c_f_2_in(o_prof,o_s,o_false,1) c_f_2_in(
    o_prof,o_k,o_false,1) c_f_2_in(o_prof,o_m,o_false,1) c_f_2_in(o_prof,o_p,o_true,1)
    c_f_2_in(o_prof,o_c,o_false,1) c_f_1_blocked(o_s,o_true,1) c_f_1_blocked(o_k,o_false,1)
    c_f_1_blocked(o_m,o_false,1) c_f_1_blocked(o_p,o_false,1) c_f_1_blocked(o_c,o_false,1)
    c_a_1_pickUp(o_coffee,o_false,0) c_a_1_pickUp(o_mail,o_false,0) c_a_1_deliver(o_coffee,
    o_false,0) c_a_1_deliver(o_mail,o_false,0) c_a_1_move(o_s,o_false,0) c_a_1_move(o_m,
    o_false,0) c_a_1_move(o_p,o_false,0) c_a_1_move(o_c,o_false,0) c_f_1_loc(o_robo,o_k,2)
    c_f_2_has(o_coffee,o_robo,o_true,2) c_a_1_pickUp(o_coffee,o_true,1) c_f_2_has(o_mail,
    o_robo,o_false,2) c_f_2_in(o_coffee,o_s,o_false,2) c_f_2_in(o_coffee,o_k,o_false,2)
    c_f_2_in(o_coffee,o_m,o_false,2) c_f_2_in(o_coffee,o_p,o_false,2) c_f_2_in(o_coffee,o_c,
    o_false,2) c_f_2_in(o_mail,o_s,o_false,2) c_f_2_in(o_mail,o_k,o_false,2) c_f_2_in(o_mail,
    o_p,o_false,2) c_f_2_in(o_mail,o_c,o_false,2) c_f_2_edge(o_s,o_s,o_false,2) c_f_2_edge(
    o_s,o_m,o_false,2) c_f_2_edge(o_s,o_p,o_false,2) c_f_2_edge(o_k,o_k,o_false,2) c_f_2_edge
    (o_k,o_m,o_false,2) c_f_2_edge(o_m,o_m,o_false,2) c_f_2_edge(o_m,o_c,o_false,2)
    c_f_2_edge(o_p,o_s,o_false,2) c_f_2_edge(o_p,o_m,o_false,2) c_f_2_edge(o_p,o_p,o_false,2)
```

```
     c_f_2_edge(o_c,o_c,o_false,2) c_f_2_edge(o_s,o_c,o_true,2) c_f_2_edge(o_s,o_k,o_true,2)
    c_f_2_edge(o_k,o_c,o_true,2) c_f_2_edge(o_k,o_s,o_true,2) c_f_2_edge(o_k,o_p,o_true,2)
    c_f_2_edge(o_m,o_s,o_true,2) c_f_2_edge(o_m,o_k,o_true,2) c_f_2_edge(o_m,o_p,o_true,2)
    c_f_2_edge(o_p,o_c,o_true,2) c_f_2_edge(o_p,o_k,o_true,2) c_f_2_edge(o_c,o_s,o_true,2)
    c_f_2_edge(o_c,o_k,o_true,2) c_f_2_edge(o_c,o_m,o_true,2) c_f_2_edge(o_c,o_p,o_true,2)
    c_f_2_has(o_coffee,o_prof,o_false,2) c_f_2_has(o_mail,o_prof,o_false,2) c_f_2_in(o_mail,
    o_m,o_true,2)
c_f_2_in(o_prof,o_s,o_false,2) c_f_2_in(o_prof,o_k,o_false,2) c_f_2_in(o_prof,o_m,o_false,2)
    c_f_2_in(o_prof,o_p,o_true,2) c_f_2_in(o_prof,o_c,o_false,2) c_f_1_blocked(o_s,o_true,2)
    c_f_1_blocked(o_k,o_false,2) c_f_1_blocked(o_m,o_false,2) c_f_1_blocked(o_p,o_false,2)
    c_f_1_blocked(o_c,o_false,2) c_a_1_pickUp(o_mail,o_false,1) c_a_1_deliver(o_coffee,
    o_false,1) c_a_1_deliver(o_mail,o_false,1) c_a_1_move(o_s,o_false,1) c_a_1_move(o_k,
    o_false,1) c_a_1_move(o_m,o_false,1) c_a_1_move(o_p,o_false,1) c_a_1_move(o_c,o_false,1)
    c_f_1_loc(o_robo,o_p,3) c_a_1_move(o_p,o_true,2) c_f_2_has(o_coffee,o_robo,o_true,3)
    c_f_2_has(o_mail,o_robo,o_false,3) c_f_2_in(o_coffee,o_s,o_false,3) c_f_2_in(o_coffee,o_k
    ,o_false,3) c_f_2_in(o_coffee,o_m,o_false,3) c_f_2_in(o_coffee,o_p,o_false,3) c_f_2_in(
    o_coffee,o_c,o_false,3) c_f_2_in(o_mail,o_s,o_false,3) c_f_2_in(o_mail,o_k,o_false,3)
    c_f_2_in(o_mail,o_p,o_false,3) c_f_2_in(o_mail,o_c,o_false,3) c_f_2_edge(o_s,o_s,o_false
    ,3) c_f_2_edge(o_s,o_m,o_false,3) c_f_2_edge(o_s,o_p,o_false,3) c_f_2_edge(o_k,o_k,
    o_false,3) c_f_2_edge(o_k,o_m,o_false,3) c_f_2_edge(o_m,o_m,o_false,3) c_f_2_edge(o_m,o_c
    ,o_false,3) c_f_2_edge(o_p,o_s,o_false,3) c_f_2_edge(o_p,o_m,o_false,3) c_f_2_edge(o_p,
    o_p,o_false,3) c_f_2_edge(o_c,o_c,o_false,3) c_f_2_edge(o_s,o_c,o_true,3) c_f_2_edge(o_s,
    o_k,o_true,3) c_f_2_edge(o_k,o_c,o_true,3) c_f_2_edge(o_k,o_s,o_true,3) c_f_2_edge(o_k,
    o_p,o_true,3) c_f_2_edge(o_m,o_s,o_true,3) c_f_2_edge(o_m,o_k,o_true,3) c_f_2_edge(o_m,
    o_p,o_true,3) c_f_2_edge(o_p,o_c,o_true,3) c_f_2_edge(o_p,o_k,o_true,3) c_f_2_edge(o_c,
    o_s,o_true,3) c_f_2_edge(o_c,o_k,o_true,3) c_f_2_edge(o_c,o_m,o_true,3) c_f_2_edge(o_c,
    o_p,o_true,3) c_f_2_has(o_coffee,o_prof,o_false,3) c_f_2_has(o_mail,o_prof,o_false,3)
    c_f_2_in(o_mail,o_m,o_true,3) c_f_2_in(o_prof,o_s,o_false,3) c_f_2_in(o_prof,o_k,o_false
    ,3) c_f_2_in(o_prof,o_m,o_false,3) c_f_2_in(o_prof,o_p,o_true,3) c_f_2_in(o_prof,o_c,
    o_false,3) c_f_1_blocked(o_s,o_true,3) c_f_1_blocked(o_k,o_false,3) c_f_1_blocked(o_m,
    o_false,3) c_f_1_blocked(o_p,o_false,3) c_f_1_blocked(o_c,o_false,3) c_a_1_pickUp(
    o_coffee,o_false,2) c_a_1_pickUp(o_mail,o_false,2) c_a_1_deliver(o_coffee,o_false,2)
    c_a_1_deliver(o_mail,o_false,2) c_a_1_move(o_s,o_false,2) c_a_1_move(o_k,o_false,2)
    c_a_1_move(o_m,o_false,2) c_a_1_move(o_c,o_false,2) c_f_1_loc(o_robo,o_p,4) c_f_2_has(
    o_coffee,o_prof,o_true,4) c_a_1_deliver(o_coffee,o_true,3) c_f_2_has(o_coffee,o_robo,
    o_false,4)
c_f_2_has(o_mail,o_robo,o_false,4) c_f_2_in(o_coffee,o_s,o_false,4) c_f_2_in(o_coffee,o_k,
    o_false,4) c_f_2_in(o_coffee,o_m,o_false,4) c_f_2_in(o_coffee,o_p,o_false,4) c_f_2_in(
    o_coffee,o_c,o_false,4) c_f_2_in(o_mail,o_s,o_false,4) c_f_2_in(o_mail,o_k,o_false,4)
    c_f_2_in(o_mail,o_p,o_false,4) c_f_2_in(o_mail,o_c,o_false,4) c_f_2_edge(o_s,o_s,o_false
    ,4) c_f_2_edge(o_s,o_m,o_false,4) c_f_2_edge(o_s,o_p,o_false,4) c_f_2_edge(o_k,o_k,
    o_false,4) c_f_2_edge(o_k,o_m,o_false,4) c_f_2_edge(o_m,o_m,o_false,4) c_f_2_edge(o_m,o_c
    ,o_false,4) c_f_2_edge(o_p,o_s,o_false,4) c_f_2_edge(o_p,o_m,o_false,4) c_f_2_edge(o_p,
    o_p,o_false,4) c_f_2_edge(o_c,o_c,o_false,4) c_f_2_edge(o_s,o_c,o_true,4) c_f_2_edge(o_s,
    o_k,o_true,4) c_f_2_edge(o_k,o_c,o_true,4) c_f_2_edge(o_k,o_s,o_true,4) c_f_2_edge(o_k,
    o_p,o_true,4) c_f_2_edge(o_m,o_s,o_true,4) c_f_2_edge(o_m,o_k,o_true,4) c_f_2_edge(o_m,
    o_p,o_true,4) c_f_2_edge(o_p,o_c,o_true,4) c_f_2_edge(o_p,o_k,o_true,4) c_f_2_edge(o_c,
    o_s,o_true,4) c_f_2_edge(o_c,o_k,o_true,4) c_f_2_edge(o_c,o_m,o_true,4) c_f_2_edge(o_c,
    o_p,o_true,4) c_f_2_has(o_mail,o_prof,o_false,4) c_f_2_in(o_mail,o_m,o_true,4) c_f_2_in(
    o_prof,o_s,o_false,4) c_f_2_in(o_prof,o_k,o_false,4) c_f_2_in(o_prof,o_m,o_false,4)
    c_f_2_in(o_prof,o_p,o_true,4) c_f_2_in(o_prof,o_c,o_false,4) c_f_1_blocked(o_s,o_true,4)
    c_f_1_blocked(o_k,o_false,4) c_f_1_blocked(o_m,o_false,4) c_f_1_blocked(o_p,o_false,4)
    c_f_1_blocked(o_c,o_false,4) c_a_1_pickUp(o_coffee,o_false,3) c_a_1_pickUp(o_mail,o_false
    ,3) c_a_1_deliver(o_mail,o_false,3) c_a_1_move(o_s,o_false,3) c_a_1_move(o_k,o_false,3)
    c_a_1_move(o_m,o_false,3) c_a_1_move(o_p,o_false,3) c_a_1_move(o_c,o_false,3) query(4)
Optimization: 20
Answer: 2
true s_robot(o_robo) s_human(o_prof) s_thing(o_coffee) s_thing(o_mail) s_location(o_s)
    s_location(o_k) s_location(o_m) s_location(o_p) s_location(o_c) s_robot__prof(o_robo)
    s_robot__prof(o_prof) s_boolean(o_true) s_boolean(o_false) c_f_1_loc(o_robo,o_s,0)
    s_thing__prof(o_coffee) s_thing__prof(o_mail) s_thing__prof(o_prof) c_f_2_has(o_coffee,
    o_robo,o_false,0) c_f_2_has(o_coffee,o_prof,o_false,0) c_f_2_has(o_mail,o_robo,o_false,0)
     c_f_2_has(o_mail,o_prof,o_false,0) c_f_2_in(o_coffee,o_s,o_false,0) c_f_2_in(o_coffee,
    o_k,o_true,0) c_f_2_in(o_coffee,o_m,o_false,0) c_f_2_in(o_coffee,o_p,o_false,0) c_f_2_in(
    o_coffee,o_c,o_false,0) c_f_2_in(o_mail,o_s,o_false,0) c_f_2_in(o_mail,o_k,o_false,0)
```

```
    c_f_2_in(o_mail,o_m,o_false,0) c_f_2_in(o_mail,o_p,o_false,0) c_f_2_in(o_mail,o_c,o_false
    ,0) c_f_2_in(o_prof,o_s,o_false,0) c_f_2_in(o_prof,o_k,o_false,0) c_f_2_in(o_prof,o_m,
    o_false,0) c_f_2_in(o_prof,o_p,o_true,0) c_f_2_in(o_prof,o_c,o_false,0) c_f_1_blocked(o_s
    ,o_true,0) c_f_1_blocked(o_k,o_false,0) c_f_1_blocked(o_m,o_false,0) c_f_1_blocked(o_p,
    o_false,0) c_f_1_blocked(o_c,o_false,0) c_f_2_edge(o_s,o_s,o_false,0) c_f_2_edge(o_s,o_m,
    o_false,0) c_f_2_edge(o_s,o_p,o_false,0) c_f_2_edge(o_k,o_k,o_false,0) c_f_2_edge(o_k,o_m
    ,o_false,0) c_f_2_edge(o_m,o_m,o_false,0) c_f_2_edge(o_m,o_c,o_false,0) c_f_2_edge(o_p,
    o_s,o_false,0)
c_f_2_edge(o_p,o_m,o_false,0) c_f_2_edge(o_p,o_p,o_false,0) c_f_2_edge(o_c,o_c,o_false,0)
    c_f_2_edge(o_s,o_c,o_true,0) c_f_2_edge(o_s,o_k,o_true,0) c_f_2_edge(o_k,o_c,o_true,0)
    c_f_2_edge(o_k,o_s,o_true,0) c_f_2_edge(o_k,o_p,o_true,0) c_f_2_edge(o_m,o_s,o_true,0)
    c_f_2_edge(o_m,o_k,o_true,0) c_f_2_edge(o_m,o_p,o_true,0) c_f_2_edge(o_p,o_c,o_true,0)
    c_f_2_edge(o_p,o_k,o_true,0) c_f_2_edge(o_c,o_s,o_true,0) c_f_2_edge(o_c,o_k,o_true,0)
    c_f_2_edge(o_c,o_m,o_true,0) c_f_2_edge(o_c,o_p,o_true,0) query_label(q_0) unsat
    (113,"-5.000000") c_f_1_loc(o_robo,o_k,1) c_a_1_move(o_k,o_true,0) c_f_2_has(o_coffee,
    o_robo,o_false,1) c_f_2_has(o_mail,o_robo,o_false,1) c_f_2_in(o_coffee,o_s,o_false,1)
    c_f_2_in(o_coffee,o_m,o_false,1) c_f_2_in(o_coffee,o_p,o_false,1) c_f_2_in(o_coffee,o_c,
    o_false,1) c_f_2_in(o_mail,o_s,o_false,1) c_f_2_in(o_mail,o_k,o_false,1) c_f_2_in(o_mail,
    o_m,o_false,1) c_f_2_in(o_mail,o_p,o_false,1) c_f_2_in(o_mail,o_c,o_false,1) c_f_2_edge(
    o_s,o_s,o_false,1) c_f_2_edge(o_s,o_m,o_false,1) c_f_2_edge(o_s,o_p,o_false,1) c_f_2_edge
    (o_k,o_k,o_false,1) c_f_2_edge(o_k,o_m,o_false,1) c_f_2_edge(o_m,o_m,o_false,1)
    c_f_2_edge(o_m,o_c,o_false,1) c_f_2_edge(o_p,o_s,o_false,1) c_f_2_edge(o_p,o_m,o_false,1)
     c_f_2_edge(o_p,o_p,o_false,1) c_f_2_edge(o_c,o_c,o_false,1) c_f_2_edge(o_s,o_c,o_true,1)
     c_f_2_edge(o_s,o_k,o_true,1) c_f_2_edge(o_k,o_c,o_true,1) c_f_2_edge(o_k,o_s,o_true,1)
    c_f_2_edge(o_k,o_p,o_true,1) c_f_2_edge(o_m,o_s,o_true,1) c_f_2_edge(o_m,o_k,o_true,1)
    c_f_2_edge(o_m,o_p,o_true,1) c_f_2_edge(o_p,o_c,o_true,1) c_f_2_edge(o_p,o_k,o_true,1)
    c_f_2_edge(o_c,o_s,o_true,1) c_f_2_edge(o_c,o_k,o_true,1)
c_f_2_edge(o_c,o_m,o_true,1) c_f_2_edge(o_c,o_p,o_true,1) c_f_2_has(o_coffee,o_prof,o_false
    ,1) c_f_2_has(o_mail,o_prof,o_false,1) c_f_2_in(o_coffee,o_k,o_true,1) c_f_2_in(o_prof,
    o_s,o_false,1) c_f_2_in(o_prof,o_k,o_false,1) c_f_2_in(o_prof,o_m,o_false,1) c_f_2_in(
    o_prof,o_p,o_true,1) c_f_2_in(o_prof,o_c,o_false,1) c_f_1_blocked(o_s,o_true,1)
    c_f_1_blocked(o_k,o_false,1) c_f_1_blocked(o_m,o_false,1) c_f_1_blocked(o_p,o_false,1)
    c_f_1_blocked(o_c,o_false,1) c_a_1_pickUp(o_coffee,o_false,0) c_a_1_pickUp(o_mail,o_false
    ,0) c_a_1_deliver(o_coffee,o_false,0) c_a_1_deliver(o_mail,o_false,0) c_a_1_move(o_s,
    o_false,0) c_a_1_move(o_m,o_false,0) c_a_1_move(o_p,o_false,0) c_a_1_move(o_c,o_false,0)
    c_f_1_loc(o_robo,o_k,2) c_f_2_has(o_coffee,o_robo,o_true,2) c_a_1_pickUp(o_coffee,o_true
    ,1) c_f_2_has(o_mail,o_robo,o_false,2) c_f_2_in(o_coffee,o_s,o_false,2) c_f_2_in(o_coffee
    ,o_k,o_false,2) c_f_2_in(o_coffee,o_m,o_false,2) c_f_2_in(o_coffee,o_p,o_false,2)
    c_f_2_in(o_coffee,o_c,o_false,2) c_f_2_in(o_mail,o_s,o_false,2) c_f_2_in(o_mail,o_k,
    o_false,2) c_f_2_in(o_mail,o_m,o_false,2) c_f_2_in(o_mail,o_p,o_false,2) c_f_2_in(o_mail,
    o_c,o_false,2) c_f_2_edge(o_s,o_s,o_false,2) c_f_2_edge(o_s,o_m,o_false,2) c_f_2_edge(o_s
    ,o_p,o_false,2) c_f_2_edge(o_k,o_k,o_false,2) c_f_2_edge(o_k,o_m,o_false,2) c_f_2_edge(
    o_m,o_m,o_false,2) c_f_2_edge(o_m,o_c,o_false,2) c_f_2_edge(o_p,o_s,o_false,2) c_f_2_edge
    (o_p,o_m,o_false,2) c_f_2_edge(o_p,o_p,o_false,2) c_f_2_edge(o_c,o_c,o_false,2)
    c_f_2_edge(o_s,o_c,o_true,2) c_f_2_edge(o_s,o_k,o_true,2) c_f_2_edge(o_k,o_c,o_true,2)
    c_f_2_edge(o_k,o_s,o_true,2) c_f_2_edge(o_k,o_p,o_true,2)
c_f_2_edge(o_m,o_s,o_true,2) c_f_2_edge(o_m,o_k,o_true,2) c_f_2_edge(o_m,o_p,o_true,2)
    c_f_2_edge(o_p,o_c,o_true,2) c_f_2_edge(o_p,o_k,o_true,2) c_f_2_edge(o_c,o_s,o_true,2)
    c_f_2_edge(o_c,o_k,o_true,2) c_f_2_edge(o_c,o_m,o_true,2) c_f_2_edge(o_c,o_p,o_true,2)
    c_f_2_has(o_coffee,o_prof,o_false,2) c_f_2_has(o_mail,o_prof,o_false,2) c_f_2_in(o_prof,
    o_s,o_false,2) c_f_2_in(o_prof,o_k,o_false,2) c_f_2_in(o_prof,o_m,o_false,2) c_f_2_in(
    o_prof,o_p,o_true,2) c_f_2_in(o_prof,o_c,o_false,2) c_f_1_blocked(o_s,o_true,2)
    c_f_1_blocked(o_k,o_false,2) c_f_1_blocked(o_m,o_false,2) c_f_1_blocked(o_p,o_false,2)
    c_f_1_blocked(o_c,o_false,2) c_a_1_pickUp(o_mail,o_false,1) c_a_1_deliver(o_coffee,
    o_false,1) c_a_1_deliver(o_mail,o_false,1) c_a_1_move(o_s,o_false,1) c_a_1_move(o_k,
    o_false,1) c_a_1_move(o_m,o_false,1) c_a_1_move(o_p,o_false,1) c_a_1_move(o_c,o_false,1)
    c_f_1_loc(o_robo,o_p,3) c_a_1_move(o_p,o_true,2) c_f_2_has(o_coffee,o_robo,o_true,3)
    c_f_2_has(o_mail,o_robo,o_false,3) c_f_2_in(o_coffee,o_s,o_false,3) c_f_2_in(o_coffee,o_k
    ,o_false,3) c_f_2_in(o_coffee,o_m,o_false,3) c_f_2_in(o_coffee,o_p,o_false,3) c_f_2_in(
    o_coffee,o_c,o_false,3) c_f_2_in(o_mail,o_s,o_false,3) c_f_2_in(o_mail,o_k,o_false,3)
    c_f_2_in(o_mail,o_m,o_false,3) c_f_2_in(o_mail,o_p,o_false,3) c_f_2_in(o_mail,o_c,o_false
    ,3) c_f_2_edge(o_s,o_s,o_false,3) c_f_2_edge(o_s,o_m,o_false,3) c_f_2_edge(o_s,o_p,
    o_false,3) c_f_2_edge(o_k,o_k,o_false,3) c_f_2_edge(o_k,o_m,o_false,3) c_f_2_edge(o_m,o_m
    ,o_false,3) c_f_2_edge(o_m,o_c,o_false,3) c_f_2_edge(o_p,o_s,o_false,3) c_f_2_edge(o_p,
    o_m,o_false,3) c_f_2_edge(o_p,o_p,o_false,3) c_f_2_edge(o_c,o_c,o_false,3) c_f_2_edge(o_s
```

```
    ,o_c,o_true,3) c_f_2_edge(o_s,o_k,o_true,3)
c_f_2_edge(o_k,o_c,o_true,3) c_f_2_edge(o_k,o_s,o_true,3) c_f_2_edge(o_k,o_p,o_true,3)
    c_f_2_edge(o_m,o_s,o_true,3) c_f_2_edge(o_m,o_k,o_true,3) c_f_2_edge(o_m,o_p,o_true,3)
    c_f_2_edge(o_p,o_c,o_true,3) c_f_2_edge(o_p,o_k,o_true,3) c_f_2_edge(o_c,o_s,o_true,3)
    c_f_2_edge(o_c,o_k,o_true,3) c_f_2_edge(o_c,o_m,o_true,3) c_f_2_edge(o_c,o_p,o_true,3)
    c_f_2_has(o_coffee,o_prof,o_false,3) c_f_2_has(o_mail,o_prof,o_false,3) c_f_2_in(o_prof,
    o_s,o_false,3) c_f_2_in(o_prof,o_k,o_false,3) c_f_2_in(o_prof,o_m,o_false,3) c_f_2_in(
    o_prof,o_p,o_true,3) c_f_2_in(o_prof,o_c,o_false,3) c_f_1_blocked(o_s,o_true,3)
    c_f_1_blocked(o_k,o_false,3) c_f_1_blocked(o_m,o_false,3) c_f_1_blocked(o_p,o_false,3)
    c_f_1_blocked(o_c,o_false,3) c_a_1_pickUp(o_coffee,o_false,2) c_a_1_pickUp(o_mail,o_false
    ,2) c_a_1_deliver(o_coffee,o_false,2) c_a_1_deliver(o_mail,o_false,2) c_a_1_move(o_s,
    o_false,2) c_a_1_move(o_k,o_false,2) c_a_1_move(o_m,o_false,2) c_a_1_move(o_c,o_false,2)
    c_f_1_loc(o_robo,o_p,4) c_f_2_has(o_coffee,o_prof,o_true,4) c_a_1_deliver(o_coffee,o_true
    ,3) c_f_2_has(o_coffee,o_robo,o_false,4) c_f_2_has(o_mail,o_robo,o_false,4) c_f_2_in(
    o_coffee,o_s,o_false,4) c_f_2_in(o_coffee,o_k,o_false,4) c_f_2_in(o_coffee,o_m,o_false,4)
     c_f_2_in(o_coffee,o_p,o_false,4) c_f_2_in(o_coffee,o_c,o_false,4) c_f_2_in(o_mail,o_s,
    o_false,4) c_f_2_in(o_mail,o_k,o_false,4) c_f_2_in(o_mail,o_m,o_false,4) c_f_2_in(o_mail,
    o_p,o_false,4) c_f_2_in(o_mail,o_c,o_false,4) c_f_2_edge(o_s,o_s,o_false,4) c_f_2_edge(
    o_s,o_m,o_false,4) c_f_2_edge(o_s,o_p,o_false,4) c_f_2_edge(o_k,o_k,o_false,4) c_f_2_edge
    (o_k,o_m,o_false,4) c_f_2_edge(o_m,o_m,o_false,4) c_f_2_edge(o_m,o_c,o_false,4)
    c_f_2_edge(o_p,o_s,o_false,4) c_f_2_edge(o_p,o_m,o_false,4) c_f_2_edge(o_p,o_p,o_false,4)
     c_f_2_edge(o_c,o_c,o_false,4) c_f_2_edge(o_s,o_c,o_true,4) c_f_2_edge(o_s,o_k,o_true,4)
    c_f_2_edge(o_k,o_c,o_true,4) c_f_2_edge(o_k,o_s,o_true,4) c_f_2_edge(o_k,o_p,o_true,4)
c_f_2_edge(o_m,o_s,o_true,4) c_f_2_edge(o_m,o_k,o_true,4) c_f_2_edge(o_m,o_p,o_true,4)
    c_f_2_edge(o_p,o_c,o_true,4) c_f_2_edge(o_p,o_k,o_true,4) c_f_2_edge(o_c,o_s,o_true,4)
    c_f_2_edge(o_c,o_k,o_true,4) c_f_2_edge(o_c,o_m,o_true,4) c_f_2_edge(o_c,o_p,o_true,4)
    c_f_2_has(o_mail,o_prof,o_false,4) c_f_2_in(o_prof,o_s,o_false,4) c_f_2_in(o_prof,o_k,
    o_false,4) c_f_2_in(o_prof,o_m,o_false,4) c_f_2_in(o_prof,o_p,o_true,4) c_f_2_in(o_prof,
    o_c,o_false,4) c_f_1_blocked(o_s,o_true,4) c_f_1_blocked(o_k,o_false,4) c_f_1_blocked(o_m
    ,o_false,4) c_f_1_blocked(o_p,o_false,4) c_f_1_blocked(o_c,o_false,4) c_a_1_pickUp(
    o_coffee,o_false,3) c_a_1_pickUp(o_mail,o_false,3) c_a_1_deliver(o_mail,o_false,3)
    c_a_1_move(o_s,o_false,3) c_a_1_move(o_k,o_false,3) c_a_1_move(o_m,o_false,3) c_a_1_move(
    o_p,o_false,3) c_a_1_move(o_c,o_false,3) query(4)
Optimization: 15
Answer: 3
true s_robot(o_robo) s_human(o_prof) s_thing(o_coffee) s_thing(o_mail) s_location(o_s)
    s_location(o_k) s_location(o_m) s_location(o_p) s_location(o_c) s_robot__prof(o_robo)
    s_robot__prof(o_prof) s_boolean(o_true) s_boolean(o_false) c_f_1_loc(o_robo,o_s,0)
    s_thing__prof(o_coffee) s_thing__prof(o_mail) s_thing__prof(o_prof) c_f_2_has(o_coffee,
    o_robo,o_false,0) c_f_2_has(o_coffee,o_prof,o_false,0) c_f_2_has(o_mail,o_robo,o_false,0)
     c_f_2_has(o_mail,o_prof,o_false,0) c_f_2_in(o_coffee,o_s,o_false,0) c_f_2_in(o_coffee,
    o_k,o_true,0) c_f_2_in(o_coffee,o_m,o_false,0) c_f_2_in(o_coffee,o_p,o_false,0) c_f_2_in(
    o_coffee,o_c,o_false,0) c_f_2_in(o_mail,o_s,o_false,0) c_f_2_in(o_mail,o_k,o_false,0)
    c_f_2_in(o_mail,o_m,o_false,0) c_f_2_in(o_mail,o_p,o_false,0) c_f_2_in(o_mail,o_c,o_false
    ,0) c_f_2_in(o_prof,o_s,o_false,0) c_f_2_in(o_prof,o_k,o_false,0) c_f_2_in(o_prof,o_m,
    o_false,0) c_f_2_in(o_prof,o_p,o_true,0) c_f_2_in(o_prof,o_c,o_false,0) c_f_1_blocked(o_s
    ,o_true,0) c_f_1_blocked(o_k,o_false,0) c_f_1_blocked(o_m,o_false,0) c_f_1_blocked(o_p,
    o_false,0) c_f_1_blocked(o_c,o_true,0) c_f_2_edge(o_s,o_s,o_false,0) c_f_2_edge(o_s,o_m,
    o_false,0) c_f_2_edge(o_s,o_p,o_false,0) c_f_2_edge(o_k,o_k,o_false,0) c_f_2_edge(o_k,o_m
    ,o_false,0) c_f_2_edge(o_m,o_m,o_false,0) c_f_2_edge(o_m,o_c,o_false,0) c_f_2_edge(o_p,
    o_s,o_false,0) c_f_2_edge(o_p,o_m,o_false,0) c_f_2_edge(o_p,o_p,o_false,0) c_f_2_edge(o_c
    ,o_c,o_false,0) c_f_2_edge(o_s,o_c,o_true,0) c_f_2_edge(o_s,o_k,o_true,0) c_f_2_edge(o_k,
    o_c,o_true,0) c_f_2_edge(o_k,o_s,o_true,0) c_f_2_edge(o_k,o_p,o_true,0) c_f_2_edge(o_m,
    o_s,o_true,0) c_f_2_edge(o_m,o_k,o_true,0) c_f_2_edge(o_m,o_p,o_true,0) c_f_2_edge(o_p,
    o_c,o_true,0)
c_f_2_edge(o_p,o_k,o_true,0) c_f_2_edge(o_c,o_s,o_true,0) c_f_2_edge(o_c,o_k,o_true,0)
    c_f_2_edge(o_c,o_m,o_true,0) c_f_2_edge(o_c,o_p,o_true,0) query_label(q_0) unsat
    (108,"-5.000000") unsat(113,"-5.000000") c_f_1_loc(o_robo,o_k,1) c_a_1_move(o_k,o_true,0)
     c_f_2_has(o_coffee,o_robo,o_false,1) c_f_2_has(o_mail,o_robo,o_false,1) c_f_2_in(
    o_coffee,o_s,o_false,1) c_f_2_in(o_coffee,o_m,o_false,1) c_f_2_in(o_coffee,o_p,o_false,1)
     c_f_2_in(o_coffee,o_c,o_false,1) c_f_2_in(o_mail,o_s,o_false,1) c_f_2_in(o_mail,o_k,
    o_false,1) c_f_2_in(o_mail,o_m,o_false,1) c_f_2_in(o_mail,o_p,o_false,1) c_f_2_in(o_mail,
    o_c,o_false,1) c_f_2_edge(o_s,o_s,o_false,1) c_f_2_edge(o_s,o_m,o_false,1) c_f_2_edge(o_s
    ,o_p,o_false,1) c_f_2_edge(o_k,o_k,o_false,1) c_f_2_edge(o_k,o_m,o_false,1) c_f_2_edge(
    o_m,o_m,o_false,1) c_f_2_edge(o_m,o_c,o_false,1) c_f_2_edge(o_p,o_s,o_false,1) c_f_2_edge
```

```
    (o_p,o_m,o_false,1) c_f_2_edge(o_p,o_p,o_false,1) c_f_2_edge(o_c,o_c,o_false,1)
    c_f_2_edge(o_s,o_c,o_true,1) c_f_2_edge(o_s,o_k,o_true,1) c_f_2_edge(o_k,o_c,o_true,1)
    c_f_2_edge(o_k,o_s,o_true,1) c_f_2_edge(o_k,o_p,o_true,1) c_f_2_edge(o_m,o_s,o_true,1)
    c_f_2_edge(o_m,o_k,o_true,1) c_f_2_edge(o_m,o_p,o_true,1) c_f_2_edge(o_p,o_c,o_true,1)
    c_f_2_edge(o_p,o_k,o_true,1) c_f_2_edge(o_c,o_s,o_true,1) c_f_2_edge(o_c,o_k,o_true,1)
    c_f_2_edge(o_c,o_m,o_true,1) c_f_2_edge(o_c,o_p,o_true,1) c_f_2_has(o_coffee,o_prof,
    o_false,1) c_f_2_has(o_mail,o_prof,o_false,1) c_f_2_in(o_coffee,o_k,o_true,1) c_f_2_in(
    o_prof,o_s,o_false,1) c_f_2_in(o_prof,o_k,o_false,1) c_f_2_in(o_prof,o_m,o_false,1)
    c_f_2_in(o_prof,o_p,o_true,1) c_f_2_in(o_prof,o_c,o_false,1) c_f_1_blocked(o_s,o_true,1)
    c_f_1_blocked(o_k,o_false,1) c_f_1_blocked(o_m,o_false,1) c_f_1_blocked(o_p,o_false,1)
    c_f_1_blocked(o_c,o_true,1) c_a_1_pickUp(o_coffee,o_false,0) c_a_1_pickUp(o_mail,o_false
    ,0) c_a_1_deliver(o_coffee,o_false,0)
c_a_1_deliver(o_mail,o_false,0) c_a_1_move(o_s,o_false,0) c_a_1_move(o_m,o_false,0)
    c_a_1_move(o_p,o_false,0) c_a_1_move(o_c,o_false,0) c_f_1_loc(o_robo,o_k,2) c_f_2_has(
    o_coffee,o_robo,o_true,2) c_a_1_pickUp(o_coffee,o_true,1) c_f_2_has(o_mail,o_robo,o_false
    ,2) c_f_2_in(o_coffee,o_s,o_false,2) c_f_2_in(o_coffee,o_k,o_false,2) c_f_2_in(o_coffee,
    o_m,o_false,2) c_f_2_in(o_coffee,o_p,o_false,2) c_f_2_in(o_coffee,o_c,o_false,2) c_f_2_in
    (o_mail,o_s,o_false,2) c_f_2_in(o_mail,o_k,o_false,2) c_f_2_in(o_mail,o_m,o_false,2)
    c_f_2_in(o_mail,o_p,o_false,2) c_f_2_in(o_mail,o_c,o_false,2) c_f_2_edge(o_s,o_s,o_false
    ,2) c_f_2_edge(o_s,o_m,o_false,2) c_f_2_edge(o_s,o_p,o_false,2) c_f_2_edge(o_k,o_k,
    o_false,2) c_f_2_edge(o_k,o_m,o_false,2) c_f_2_edge(o_m,o_m,o_false,2) c_f_2_edge(o_m,o_c
    ,o_false,2) c_f_2_edge(o_p,o_s,o_false,2) c_f_2_edge(o_p,o_m,o_false,2) c_f_2_edge(o_p,
    o_p,o_false,2) c_f_2_edge(o_c,o_c,o_false,2) c_f_2_edge(o_s,o_c,o_true,2) c_f_2_edge(o_s,
    o_k,o_true,2) c_f_2_edge(o_k,o_c,o_true,2) c_f_2_edge(o_k,o_s,o_true,2) c_f_2_edge(o_k,
    o_p,o_true,2) c_f_2_edge(o_m,o_s,o_true,2) c_f_2_edge(o_m,o_k,o_true,2) c_f_2_edge(o_m,
    o_p,o_true,2) c_f_2_edge(o_p,o_c,o_true,2) c_f_2_edge(o_p,o_k,o_true,2) c_f_2_edge(o_c,
    o_s,o_true,2) c_f_2_edge(o_c,o_k,o_true,2) c_f_2_edge(o_c,o_m,o_true,2) c_f_2_edge(o_c,
    o_p,o_true,2) c_f_2_has(o_coffee,o_prof,o_false,2) c_f_2_has(o_mail,o_prof,o_false,2)
    c_f_2_in(o_prof,o_s,o_false,2) c_f_2_in(o_prof,o_k,o_false,2) c_f_2_in(o_prof,o_m,o_false
    ,2) c_f_2_in(o_prof,o_p,o_true,2) c_f_2_in(o_prof,o_c,o_false,2) c_f_1_blocked(o_s,o_true
    ,2) c_f_1_blocked(o_k,o_false,2) c_f_1_blocked(o_m,o_false,2) c_f_1_blocked(o_p,o_false
    ,2) c_f_1_blocked(o_c,o_true,2) c_a_1_pickUp(o_mail,o_false,1)
c_a_1_deliver(o_coffee,o_false,1) c_a_1_deliver(o_mail,o_false,1) c_a_1_move(o_s,o_false,1)
    c_a_1_move(o_k,o_false,1) c_a_1_move(o_m,o_false,1) c_a_1_move(o_p,o_false,1) c_a_1_move(
    o_c,o_false,1) c_f_1_loc(o_robo,o_p,3) c_a_1_move(o_p,o_true,2) c_f_2_has(o_coffee,o_robo
    ,o_true,3) c_f_2_has(o_mail,o_robo,o_false,3) c_f_2_in(o_coffee,o_s,o_false,3) c_f_2_in(
    o_coffee,o_k,o_false,3) c_f_2_in(o_coffee,o_m,o_false,3) c_f_2_in(o_coffee,o_p,o_false,3)
     c_f_2_in(o_coffee,o_c,o_false,3) c_f_2_in(o_mail,o_s,o_false,3) c_f_2_in(o_mail,o_k,
    o_false,3) c_f_2_in(o_mail,o_m,o_false,3) c_f_2_in(o_mail,o_p,o_false,3) c_f_2_in(o_mail,
    o_c,o_false,3) c_f_2_edge(o_s,o_s,o_false,3) c_f_2_edge(o_s,o_m,o_false,3) c_f_2_edge(o_s
    ,o_p,o_false,3) c_f_2_edge(o_k,o_k,o_false,3) c_f_2_edge(o_k,o_m,o_false,3) c_f_2_edge(
    o_m,o_m,o_false,3) c_f_2_edge(o_m,o_c,o_false,3) c_f_2_edge(o_p,o_s,o_false,3) c_f_2_edge
    (o_p,o_m,o_false,3) c_f_2_edge(o_p,o_p,o_false,3) c_f_2_edge(o_c,o_c,o_false,3)
    c_f_2_edge(o_s,o_c,o_true,3) c_f_2_edge(o_s,o_k,o_true,3) c_f_2_edge(o_k,o_c,o_true,3)
    c_f_2_edge(o_k,o_s,o_true,3) c_f_2_edge(o_k,o_p,o_true,3) c_f_2_edge(o_m,o_s,o_true,3)
    c_f_2_edge(o_m,o_k,o_true,3) c_f_2_edge(o_m,o_p,o_true,3) c_f_2_edge(o_p,o_c,o_true,3)
    c_f_2_edge(o_p,o_k,o_true,3) c_f_2_edge(o_c,o_s,o_true,3) c_f_2_edge(o_c,o_k,o_true,3)
    c_f_2_edge(o_c,o_m,o_true,3) c_f_2_edge(o_c,o_p,o_true,3) c_f_2_has(o_coffee,o_prof,
    o_false,3) c_f_2_has(o_mail,o_prof,o_false,3) c_f_2_in(o_prof,o_s,o_false,3) c_f_2_in(
    o_prof,o_k,o_false,3) c_f_2_in(o_prof,o_m,o_false,3) c_f_2_in(o_prof,o_p,o_true,3)
    c_f_2_in(o_prof,o_c,o_false,3) c_f_1_blocked(o_s,o_true,3) c_f_1_blocked(o_k,o_false,3)
    c_f_1_blocked(o_m,o_false,3) c_f_1_blocked(o_p,o_false,3) c_f_1_blocked(o_c,o_true,3)
    c_a_1_pickUp(o_coffee,o_false,2) c_a_1_pickUp(o_mail,o_false,2) c_a_1_deliver(o_coffee,
    o_false,2) c_a_1_deliver(o_mail,o_false,2)
c_a_1_move(o_s,o_false,2) c_a_1_move(o_k,o_false,2) c_a_1_move(o_m,o_false,2) c_a_1_move(o_c,
    o_false,2) c_f_1_loc(o_robo,o_p,4) c_f_2_has(o_coffee,o_prof,o_true,4) c_a_1_deliver(
    o_coffee,o_true,3) c_f_2_has(o_coffee,o_robo,o_false,4) c_f_2_has(o_mail,o_robo,o_false
    ,4) c_f_2_in(o_coffee,o_s,o_false,4) c_f_2_in(o_coffee,o_k,o_false,4) c_f_2_in(o_coffee,
    o_m,o_false,4) c_f_2_in(o_coffee,o_p,o_false,4) c_f_2_in(o_coffee,o_c,o_false,4) c_f_2_in
    (o_mail,o_s,o_false,4) c_f_2_in(o_mail,o_k,o_false,4) c_f_2_in(o_mail,o_m,o_false,4)
    c_f_2_in(o_mail,o_p,o_false,4) c_f_2_in(o_mail,o_c,o_false,4) c_f_2_edge(o_s,o_s,o_false
    ,4) c_f_2_edge(o_s,o_m,o_false,4) c_f_2_edge(o_s,o_p,o_false,4) c_f_2_edge(o_k,o_k,
    o_false,4) c_f_2_edge(o_k,o_m,o_false,4) c_f_2_edge(o_m,o_m,o_false,4) c_f_2_edge(o_m,o_c
    ,o_false,4) c_f_2_edge(o_p,o_s,o_false,4) c_f_2_edge(o_p,o_m,o_false,4) c_f_2_edge(o_p,
    o_p,o_false,4) c_f_2_edge(o_c,o_c,o_false,4) c_f_2_edge(o_s,o_c,o_true,4) c_f_2_edge(o_s,
```

```
    o_k,o_true,4) c_f_2_edge(o_k,o_c,o_true,4) c_f_2_edge(o_k,o_s,o_true,4) c_f_2_edge(o_k,
    o_p,o_true,4) c_f_2_edge(o_m,o_s,o_true,4) c_f_2_edge(o_m,o_k,o_true,4) c_f_2_edge(o_m,
    o_p,o_true,4) c_f_2_edge(o_p,o_c,o_true,4) c_f_2_edge(o_p,o_k,o_true,4) c_f_2_edge(o_c,
    o_s,o_true,4) c_f_2_edge(o_c,o_k,o_true,4) c_f_2_edge(o_c,o_m,o_true,4) c_f_2_edge(o_c,
    o_p,o_true,4) c_f_2_has(o_mail,o_prof,o_false,4) c_f_2_in(o_prof,o_s,o_false,4) c_f_2_in(
    o_prof,o_k,o_false,4) c_f_2_in(o_prof,o_m,o_false,4) c_f_2_in(o_prof,o_p,o_true,4)
    c_f_2_in(o_prof,o_c,o_false,4) c_f_1_blocked(o_s,o_true,4) c_f_1_blocked(o_k,o_false,4)
    c_f_1_blocked(o_m,o_false,4) c_f_1_blocked(o_p,o_false,4) c_f_1_blocked(o_c,o_true,4)
    c_a_1_pickUp(o_coffee,o_false,3) c_a_1_pickUp(o_mail,o_false,3) c_a_1_deliver(o_mail,
    o_false,3) c_a_1_move(o_s,o_false,3) c_a_1_move(o_k,o_false,3) c_a_1_move(o_m,o_false,3)
    c_a_1_move(o_p,o_false,3) c_a_1_move(o_c,o_false,3) query(4)
Optimization: 10
OPTIMUM FOUND

Models       : 3
  Optimum    : yes
Optimization : 10
Calls        : 5
Time         : 0.065s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.060s
```