Koushik Sahu
118CS0597
Artificial Intelligence Lab – 7

Air Cargo Transportation Problem
Code:

```
:- dynamic(at/2).
:- dynamic(in/2).


at('Plane2','SFO').
at('Plane1','JFK').
at('Cargo2','SFO').
at('Cargo1','JFK').
plane('Plane1').
plane('Plane2').
cargo('Cargo1').
cargo('Cargo2').


load_(C,P,R) :-
\+ (in(C,P)),
at(P,R),
at(C,R),


retract(at(C,R)),
assert(in(C,P)),
assert(loade(C,P,R)).


unload_(C,P,R) :-
\+ (at(C,R)),
at(P,R),
in(C,P),


retract(in(C,P)),
assert(at(C,R)),
assert(unload(C,P,R)).


fly_(P,D,A) :-
D \== A,
at(P,D),


retract(at(P,D)),
assert(at(P,A)),
assert(fly(P,D,A)).
```

```
do(Glist) :-
valid(Glist),
do_all(Glist,Glist).


valid(_).


do_all([G|R],Allgoals) :-
call(G),
do_all(R,Allgoals).


do_all([G|_],Allgoals) :-
achieve(G),
call(G),
do_all(Allgoals,Allgoals).


do_all([],Allgoals).


achieve(at(P,R)) :-
\+ (at(P,R)),
plane(P),
at(P,X),
at(C,X),
cargo(C),


load_(C,P,X),
fly_(P,X,R),
unload_(C,P,R).
```

Result:

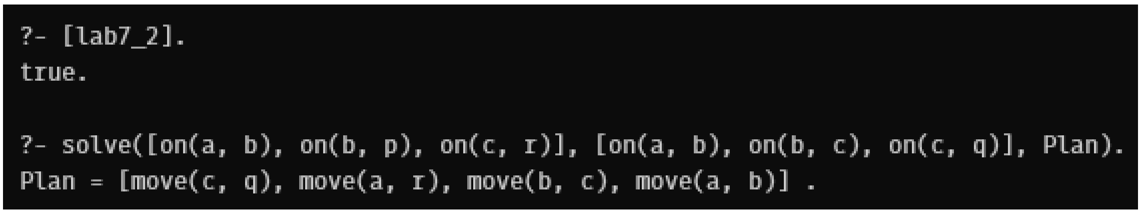
```
?- [lab7_1].
true.
?- do([at('Plane1','SFO'),at('Plane2','JFK'),at('Cargo1','SFO'),at('Cargo2','JFK')]).
true .

?- listing(loade),listing(fly),listing(unload). :-dynamic loade/3.
:- dynamic loade/3.

loade('Cargo1', 'Plane1', 'JFK').
loade('Cargo2', 'Plane2', 'SFO').

:- dynamic fly/3.

fly('Plane1', 'JFK', 'SFO').
fly('Plane2', 'SFO', 'JFK').

:- dynamic unload/3.

unload('Cargo1', 'Plane1', 'SFO').
unload('Cargo2', 'Plane2', 'JFK').

true.
```

Block Problem

Code:

```
block(a).
block(b).
block(c).

place(p).
place(q).
place(r).

solve(Initial, Final, Plan) :- strips(Initial, Final, Plan).

strips(Initial, Final, Plan) :- strips(Initial, Final, [Initial], Plan).

strips(Initial, Final, Visited, Plan) :-
   deepening_strips(1, Initial, Final, Visited, Plan).

deepening_strips(Bound, Initial, Final, Visited, Plan) :-
   bounded_strips(Bound, Initial, Final, Visited, Plan).
deepening_strips(Bound, Initial, Final, Visited, Plan) :-
          succ(Bound, Successor),
   deepening_strips(Successor, Initial, Final, Visited, Plan).

bounded_strips(_, Final, Final, _, []).
bounded_strips(Bound, Initial, Final, Visited, [Action|Actions]) :-
   succ(Predecessor, Bound),
   action(Initial, Action),
   perform(Initial, Action, Intermediate),
   \+ member(Intermediate, Visited),
   bounded_strips(Predecessor, Intermediate, Final, [Intermediate|Visited], Actions).

action(State, move(Block, Destination)) :-
   block(Block),
   \+ Block == Destination,
   free(State, Block),
   free(State, Destination).

free(State, Thing) :-
   thing(Thing),
   \+ member(on(_, Thing), State).

thing(Block) :- block(Block).
thing(Place) :- place(Place).

perform(Source, move(Block, Destination), Target) :-
   substitute(on(Block, _), Source, on(Block, Destination), Target).

substitute(_, [], _, []).
substitute(A, [A|As], B, [B|Bs]) :-
   substitute(A, As, B, Bs), !.
substitute(A, [X|As], B, [X|Bs]) :-
   substitute(A, As, B, Bs).
```

Result:



```
?- [lab7_2].
true.

?- solve([on(a, b), on(b, p), on(c, r)], [on(a, b), on(b, c), on(c, q)], Plan).
Plan = [move(c, q), move(a, r), move(b, c), move(a, b)] .
```