

Apendice B

Estos son los algunos de los programas que probamos para realizar nuestro estudio, sin embargo no son todos. Para mayor interés ver <http://www.udlap.mx/~josorio>.

MONKEY AND BANANA

```
% The domain
object(box).
object(monkey).
object(banana).

holds(true,T):-time(T).

%Time
const length = 5.
time(1..length).

const entero = 4.
num(1..entero).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%The fluent

fluent(at(O,L)) :- object(O), num(L).
fluent(onBox).
fluent(hasBanana).
fluent(objectIsSomewhere(O)) :-object(O).

lfluent(F) :- fluent(F).
lfluent(neg(F)) :- fluent(F).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The action

action(walk(L)) :- num(L).
action(pushBox(L)) :- num(L).
action(climbBox).
action(graspBanana).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The executability conditions

executable(walk(L),T) :- L<entero, T<length, not holds(onBox,T),
num(L), time(T).
executable(pushBox(L1),T) :- L<entero, T<length, not holds(onBox,T),
holds(at(monkey,L),T), holds(at(box,L),T),
num(L), L!=L1, num(L1), time(T).
executable(climbBox,T) :- L<entero, T<length, not holds(onBox,T),
holds(at(monkey,L),T), holds(at(box,L),T),
num(L), time(T).
executable(graspBanana,T) :- L<entero, T<length, holds(onBox,T),
holds(at(monkey,L),T), holds(at(banana,L),T),
num(L), time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

%The dynamic causal laws

causes (walk(L), at(monkey,L)) :- num(L).
causes (pushBox(L), at(monkey,L)) :- num(L).
causes (pushBox(L), at(box,L)) :- num(L).
causes (climbBox, onBox).
causes (graspBanana, hasBanana). causes_d(walk(L1),
neg(at(monkey,L),at(monkey,L)) ) :- L!=L1, num(L), num(L1).
causes_d(pushBox(L1), neg(at(monkey,L),at(monkey,L)) ) :- L!=L1, num(L),
num(L1).
causes_d(pushBox(L1), neg(at(box,L),at(box,L)) ) :- L!=L1, num(L),
num(L1).
causes_s(objectIsSomewhere(O), at(O,L)) :- object(O), num(L).
causes_false(onBox, at(monkey,L), neg(at(box,L))) :- object(O), num(L).
causes_false(at(O,L), at(O,L1),true) :- object(O), num(L), num(L1),
L!=L1.
causes_false(neg(objectIsSomewhere(O)), true, true) :- object(O).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The initial state

initially(at(monkey,1)).
initially(at(box,2)).
initially(at(banana,3)).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The goal

finally(hasBanana).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%THE DOMAIN INDEPENDENT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining goal(T)
not_goal(T):- time(T), finally(L), not holds(L,T).
goal(T):- time(T), not not_goal(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Eliminating possible AS
:- not goal(length).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining contrary
contrary(F,neg(F)) :- fluent(F).
contrary(neg(F),F) :- fluent(F).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining executability

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Fluents values in the time point 1
holds(F,1):- initially(F).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Effect axioms
holds(F,T+1) :- T<length, executable(A,T), occurs(A,T), causes(A,F),
action(A), time(T), lfluent(F).

```

```
holds(F,T+1) :- T<length, executable(A,T), occurs(A,T), causes_d(A,F,P),
              action(A), time(T), holds(P,T), lfluent(F).
```

```
holds(A,T) :- T<length, causes_s(A,F),holds(F,T), time(T).
:-causes_false(A,F).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%Inertial
```

```
holds(F,T+1) :- L<entero, lfluent(F), contrary(F,G), T<length,
holds(F,T), not holds(G,T+1), time(T), num(L).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Occurrences of actions
```

```
occurs(A,T) :- action(A), time(T), not goal(T), not not_occurs(A,T).
not_occurs(A,T) :- action(A), action(AA), time(T), occurs(AA,T), A!=AA.
:- action(A), time(T), occurs(A,T), not executable(A,T).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BLOCKS WORDL
```

```
% The domain
```

```
block(a).
block(b).
block(c).
%block(d).
location(table).
location(B) :- block(B).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%The fluent
```

```
%fluents: on(B,L) requires block(B), location(L).
%          occupied(B) requires location(B).
```

```
fluent(on(B,L)) :- block(B), location(L).
fluent(occupied(B)) :- location(B).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%The actions
```

```
%actions: move(B,L) requires block(B), location(L).
```

```
action(move(B,L)) :- block(B), location(L), B !=L.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Time
```

```
const length = 4.
time(1..length).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%The executability conditions
```

```
%executable move(B,L) if not occupied(B), not occupied(L), B <> L
```

```
executable(move(B,L),T) :- T<length, not holds(occupied(B),T), not
holds(occupied(L),T),
```

```

        block(B), location(L), B!=L, time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The dynamic causal laws
%caused occupied(B) if on(B1,B), block(B).
%caused on(B,L) after move(B,L).
%caused -on(B,L1) after move(B,L), on(B,L1), L <> L1.

causes_s(occupied(B), on(B1,B)) :- block(B), block(B1).
causes_d(move(B,L), on(B,L)) :- block(B), location(L).
causes_dp(move(B,L), neg(on(B,L1)), on(B,L1) ) :- block(B), location(L),
        location(L1), L!=L1 .

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The initial state
%initially: on(a,table). on(b,table). on(c,a).

initially(on(a,table)).
initially(on(b,table)).
initially(on(c,a)).
%initially(on(d,b)).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The goal
%goal: on(c,b),on(b,a),on(a,table)? (3)

finally(on(b,a)).
finally(on(c,b)).
finally(on(a,table)).
%finally(on(d,c)).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%THE DOMAIN INDEPENDENT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining goal(T)
not_goal(T):- time(T), finally(B), not holds(B,T).
goal(T):- time(T), not not_goal(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Eliminating possible AS
:- not goal(length).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining contrary
contrary(F,neg(F)) :- fluent(F).
contrary(neg(F),F) :- fluent(F).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining executability

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Fluents values in the time point 1
holds(F,1):- initially(F).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Effect axioms
holds(F,T+1) :- T<length, executable(A,T), occurs(A,T), causes_d(A,F),

```

```

        action(A), time(T).

holds(F,T+1) :- T<length, executable(A,T), occurs(A,T), causes_dp(A,F,P),
               action(A), time(T), holds(P,T).

holds(A,T) :- T<length, causes_s(A,F),holds(F,T), time(T).

%%%%Inertial

%inertial on(B,L).
holds(on(B,L),T+1) :- T<length, holds(on(B,L),T), not
holds(neg(on(B,L)),T+1), block(B), location(L),time(T).
%holds(neg(on(B,L)),T+1) :- T<length, holds(neg(on(B,L)),T), not
holds(on(B,L),T+1), block(B), location(L),time(T).

%%%%Ocurrences of actions
occurs(A,T) :- action(A), time(T), not goal(T), not not_occurs(A,T).
not_occurs(A,T) :- action(A), action(AA), time(T), occurs(AA,T), A!=AA.
:- action(A), time(T), occurs(A,T), not executable(A,T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BOMB IN TOILET
% The domain
package(1).
package(2).
holds(true,T) :- time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%The fluent

fluent(armed(P)) :- package(P).
fluent(unsafe).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%The actions

action(dunk(P)) :- package(P).
action(flush).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Time
const length = 3.
time(1..length).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%The executability conditions

executable(dunk(P),T) :- T<length, package(P), time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%The dynamic causal laws

causes_s(unsafe, armed(P) ) :- package(P). causes_d(dunk(P),
neg(armed(P))) :- package(P).

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The initial state

initially(neg(armed(1))).
initially(unsafe).
initially(armed(2)).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The goal

finally(neg(unsafe)).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%THE DOMAIN INDEPENDENT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining goal(T)
not_goal(T):- time(T), finally(B), not holds(B,T).
goal(T):- time(T), not not_goal(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Eliminating possible AS
%:- not goal(length).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining contrary
contrary(F,neg(F)) :- fluent(F).
contrary(neg(F),F) :- fluent(F).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining executability
%not_executable(A,T):- nexec(A,F), holds(F,T), fluent(F), time(T).
%executable(A,T):- T<length, not not_executable(A,T), action(A), time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Fluents values in the time point 1
holds(F,1):- initially(F).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Effect axioms
holds(F,T+1) :- T<length, executable(A,T), occurs(A,T), causes_d(A,F),
                action(A), time(T).

holds(A,T) :- T<length, causes_s(A,F),holds(F,T), time(T).

%%%TOTAL%%%
holds(armed(P),T) :- T<length, not holds(neg(armed(P)),T), package(P),
time(T).
holds(neg(armed(P)),T) :- T<length, not holds(armed(P),T), package(P),
time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%Inertial

%inertial on(B,L).
holds(armed(P),T+1) :- T<length, holds(armed(P),T), not
holds(neg(armed(P)),T+1),

```

```
package(P), time(T).
%holds(neg(armed(P)),T+1) :- T<length, holds(neg(armed(P)),T), not
holds(armed(P),T+1),
%package(P), time(T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Occurrences of actions
occurs(A,T) :- action(A), time(T), not goal(T), not not_occurs(A,T).
not_occurs(A,T) :- action(A), action(AA), time(T), occurs(AA,T), A!=AA.
%:- action(A), time(T), occurs(A,T), not executable(A,T).
```