% ------------------------------------

% Monkey and Banana Problem (Safe Version)

% ------------------------------------

% State represented as:

% state(MonkeyPosition, BoxPosition, MonkeyStatus, BananaStatus)

% MonkeyPosition = atdoor | middle | window

% BoxPosition = atdoor | middle | window

% MonkeyStatus = onfloor | onbox

% BananaStatus = has | hasnot

% Initial state

initial\_state(state(atdoor, window, onfloor, hasnot)).

% Goal state

goal\_state(state(\_, \_, \_, has)).

% ------------------------------------

% Actions

% ------------------------------------

% Monkey walks

move(state(atdoor, Box, onfloor, hasnot),

walk\_door\_to\_window,

state(window, Box, onfloor, hasnot)).

move(state(window, Box, onfloor, hasnot),

walk\_window\_to\_door,

state(atdoor, Box, onfloor, hasnot)).

move(state(atdoor, Box, onfloor, hasnot),

walk\_door\_to\_middle,

state(middle, Box, onfloor, hasnot)).

move(state(middle, Box, onfloor, hasnot),

walk\_middle\_to\_door,

state(atdoor, Box, onfloor, hasnot)).

move(state(window, Box, onfloor, hasnot),

walk\_window\_to\_middle,

state(middle, Box, onfloor, hasnot)).

move(state(middle, Box, onfloor, hasnot),

walk\_middle\_to\_window,

state(window, Box, onfloor, hasnot)).

% Push box

move(state(Pos, Pos, onfloor, hasnot),

push\_box(Pos, middle),

state(middle, middle, onfloor, hasnot)).

% Climb up / down

move(state(Pos, Pos, onfloor, hasnot),

climb\_box,

state(Pos, Pos, onbox, hasnot)).

move(state(Pos, Pos, onbox, hasnot),

climb\_down,

state(Pos, Pos, onfloor, hasnot)).

% Grab banana

move(state(middle, middle, onbox, hasnot),

grab\_banana,

state(middle, middle, onbox, has)).

% ------------------------------------

% Planner with Visited List

% ------------------------------------

plan(State, Moves) :-

plan(State, [], Moves).

plan(State, \_, []) :-

goal\_state(State).

plan(State, Visited, [Move|Rest]) :-

move(State, Move, NewState),

\+ member(NewState, Visited), % avoid cycles

plan(NewState, [NewState|Visited], Rest).

% ------------------------------------

% Main Program

% ------------------------------------

:- initialization(main).

main :-

nl, write('--- Monkey and Banana Problem ---'), nl,

initial\_state(Start),

plan(Start, Moves),

write('Plan to get banana:'), nl,

forall(member(M, Moves), (write(' - '), write(M), nl)),

nl, write('Banana successfully reached!'), nl,

halt.