Declarative Programming, Handout 8

File mc.lp

% Three missionaries and three cannibals must cross a river using
% a boat which can carry at most two people, under the constraint
% that, for both banks, if there are missionaries present on the
% bank, they cannot be outnumbered by cannibals.

% horizon
#const h=11.

% locations
loc(bank1;bank2).

% number of missionaries/cannibals
num(0..3).

% number of boats
num_boats(0..1).

% Choice rule for actions move(NM,NC,Dest,T): moving NM missionaries
% and NC cannibals to destination Dest between times T and T+1.
{ move(0..2,0..2,Dest,T) } 1 :- loc(Dest), T = 0..h-1.

% boat doesn't move by itself
:- move(0,0,_,_).

% capacity of the boat
:- move(NM,NC,_,_), NM+NC>2.

% Moving missionaries increases their number at the destination by
% the amount moved, and similarly for cannibals and boats
m(Dest,N+NM,T+1) :- m(Dest,N,T), move(NM,_,Dest,T), num(N).
c(Dest,N+NC,T+1) :- c(Dest,N,T), move(_,NC,Dest,T), num(N).
b(Dest,N+1,T+1) :- b(Dest,N,T), move(_,Dest,T), num_boats(N).

% definition of the opposite bank
opposite(B1,B2) :- loc(B1), loc(B2), B1 != B2.

% The number of missionaries at the opposite bank is the total
% number minus the number on this bank, and similarly for cannibals
% and boats
m(B2,3-N,T) :- m(B1,N,T), opposite(B1,B2), num(N).
c(B2,3-N,T) :- c(B1,N,T), opposite(B1,B2), num(N).
b(B2,1-N,T) :- b(B1,N,T), opposite(B1,B2), num_boats(N).

% Impossible to move more missionaries than available at the source, % and similarly for cannibals and boats.
:- m(B,N,T), move(NM,_,Dest,T), opposite(B,Dest), NM>N.
:- c(B,N,T), move(_,NC,Dest,T), opposite(B,Dest), NC>N.
:- b(B,0,T), move(_,_,Dest,T), opposite(B,Dest).

% inertia
m(B,N,T+1) :- m(B,N,T), not -m(B,N,T+1), loc(B), num(N), T=0..h-1.
c(B,N,T+1) :- c(B,N,T), not -c(B,N,T+1), loc(B), num(N), T=0..h-1.
b(B,N,T+1) :- b(B,N,T), not -b(B,N,T+1), loc(B), num(N), T=0..h-1.

% uniqueness rules
-m(B,NN,T) :- m(B,N,T), loc(B), num(NN), NN != N.
-c(B,NN,T) :- c(B,N,T), loc(B), num(NN), NN != N.
-b(B,NN,T) :- b(B,N,T), loc(B), num_boats(NN), NN != N.

% missionaries can't be outnumbered
:- m(B,NM,T), c(B,NC,T), loc(B), num(NM), num(MC), NC>NM, NM>0.

% initial conditions
m(bank1,3,0).
c(bank1,3,0).
b(bank1,1,0).

% goal
:- not m(bank2,3,h).
:- not c(bank2,3,h).

#show move/4.

Output:
move(1,1,bank2,0) move(1,0,bank1,1) move(0,2,bank2,2) move(0,1,bank1,3) move(2,0,bank2,4) move(1,1,bank1,5) move(2,0,bank2,6) move(0,1,bank1,7) move(0,2,bank2,8) move(0,1,bank1,9) move(0,2,bank2,10)