**Programming Assignment 2: More Challenging Answer Set Programming**

Q1

|  |  |
| --- | --- |
| Input  Program | Hint: you only need one program with a new term, whose value will be assigned to 3 or 4 in the command line.  %%%%%%%%%%%%%%%%%%%  % File: blocks-scenario.lp  %%%%%%%%%%%%%%%%%%%    block(1..6).    % initial state  :- not on(1,2,0; 2,table,0; 3,4,0; 4,table,0; 5,6,0; 6,table,0).    % goal  :- not on(3,2,m; 2,1,m; 1,table,m; 6,5,m; 5,4,m; 4,table,m).  %%%%%%%%%%%%%%%%%%%  % File: blocks.lp: Blocks World  %%%%%%%%%%%%%%%%%%%      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % sort and object declaration  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    % every block is a location  location(B) :- block(B).    % the table is a location  location(table).      %%%%%%%%%%%%%%%%%%%%%%%%%%  % state description  %%%%%%%%%%%%%%%%%%%%%%%%%%    % two blocks can't be on the same block at the same time  :- 2{on(BB,B,T)}, block(B), T = 0..m.    % No more than n blocks on the table at once  :- n+1{on(B,table,T)}, T = 0..m.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % effect and preconditions of action  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    % effect of moving a block  on(B,L,T+1) :- move(B,L,T).    % concurrent actions are limited by num of grippers  :- not {move(BB,LL,T)} grippers, T = 0..m-1.    % a block can be moved only when it is clear  :- move(B,L,T), on(B1,B,T).    % a block can't be moved onto a block that is being moved also  :- move(B,B1,T), move(B1,L,T).      %%%%%%%%%%%%%%%%%%%%%%%%%%%%  % domain independent axioms  %%%%%%%%%%%%%%%%%%%%%%%%%%%%    % fluents are initially exogenous  1{on(B,LL,0):location(LL)}1 :- block(B).    % uniqueness and existence of value constraints  :- not 1{on(B,LL,T)}1, block(B), T=1..m.    % actions are exogenous  {move(B,L,T)} :- block(B), location(L), T = 0..m-1.    % commonsense law of inertia  {on(B,L,T+1)} :- on(B,L,T), T < m.    #show move/3. |
| Command  Line | You should write multiple command lines below.  Assuming grippers is not constrained:  ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=5 -c n=3  ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=3 -c n=4 |
| Output  of clingo | You should write multiple outputs, one for each command. These outputs serve as the evidences of your answer to the question below.  Reading from blocks.asp ...  Solving...  Answer: 1  move(3,1,0) move(5,4,0) move(3,table,1) move(6,5,1) move(1,table,2) move(3,6,2) move(2,1,3) move(3,table,3) move(3,2,4)  SATISFIABLE    Models : 1+  Calls : 1  Time : 0.005s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.005s  Reading from blocks.asp ...  Solving...  Answer: 1  move(1,table,0) move(3,6,0) move(5,4,0) move(2,1,1) move(3,table,1) move(3,2,2) move(6,5,2)  SATISFIABLE    Models : 1+  Calls : 1  Time : 0.004s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.004s |
| Answer  to Questions | Fill in the following table that lists the number of steps to solve the modified block world problem for different value of n, where n is the maximal number of blocks that can be placed directly on the table.   |  |  | | --- | --- | | n | Number of steps | | 3 | 5 | | 4 | 3 | |

Q2

|  |  |
| --- | --- |
| Input  Program | Hint: you don’t need to represent any scenario since you want to find out all possible valid states. Also think about the value of m.  %%%%%%%%%%%%%%%%%%%  % File: blocks-scenario.lp  %%%%%%%%%%%%%%%%%%%    block(1..6).  %%%%%%%%%%%%%%%%%%%  % File: blocks.lp: Blocks World  %%%%%%%%%%%%%%%%%%%      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % sort and object declaration  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    % every block is a location  location(B) :- block(B).    % the table is a location  location(table).      %%%%%%%%%%%%%%%%%%%%%%%%%%  % state description  %%%%%%%%%%%%%%%%%%%%%%%%%%    % two blocks can't be on the same block at the same time  :- 2{on(BB,B,T)}, block(B), T = 0..m.    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % effect and preconditions of action  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    % effect of moving a block  on(B,L,T+1) :- move(B,L,T).    % concurrent actions are limited by num of grippers  :- not {move(BB,LL,T)} grippers, T = 0..m-1.    % a block can be moved only when it is clear  :- move(B,L,T), on(B1,B,T).    % a block can't be moved onto a block that is being moved also  :- move(B,B1,T), move(B1,L,T).    % a block cannot be on itself  :- on(B,B,T).    % No circular stacking of blocks  above(B,table,T) :- on(B,table,T).  above(B1,B,T) :- above(B,table,T), on(B1,B,T), block(B;B1).  above(B2,B1,T) :- above(B1,B,T), on(B2,B1,T), block(B1;B2;B).  :- not {above(B,L,T): location(L)}=1, block(B), T=0..m.    %%%%%%%%%%%%%%%%%%%%%%%%%%%%  % domain independent axioms  %%%%%%%%%%%%%%%%%%%%%%%%%%%%    % fluents are initially exogenous  1{on(B,LL,0):location(LL)}1 :- block(B).    % uniqueness and existence of value constraints  :- not 1{on(B,LL,T)}1, block(B), T=1..m.    % actions are exogenous  {move(B,L,T)} :- block(B), location(L), T = 0..m-1.    % commonsense law of inertia  {on(B,L,T+1)} :- on(B,L,T), T < m.    #show on/3. |
| Command  Line | ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=0 0 |
| Output  of clingo | Models : 4051  Calls : 1  Time : 0.088s (Solving: 0.09s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.088s |
| Answer  to Questions | How many valid states are there when there are 6 blocks? (Note that the limitation of blocks introduced in question 1 is not considered here.)  4051 |

Q3

|  |  |
| --- | --- |
| Input  Program | Hint: the number of grippers is unlimited, meaning that you can have as many movements as you want as far as the movements are serializable.  %%%%%%%%%%%%%%%%%%%  % File: blocks-scenario.lp  %%%%%%%%%%%%%%%%%%%    % a1, b2, c3, d4, e5, f6, g7, h8, i9, j10, k11, l12, m13, n14, o15  block(1..15).    % initial state  :- not on(13,table,0;12,13,0;1,12,0;2,1,0;3,2,0;  15,table,0;14,15,0;4,14,0;5,4,0;10,5,0;  11,10,0;6,table,0;7,6,0;8,7,0;9,8,0).    % goal  :- not on(5,10,m;1,5,m;14,1,m;9,4,m;8,9,m;  13,8,m;15,13,m;11,7,m;3,11,m;2,3,m;12,2,m).  %%%%%%%%%%%%%%%%%%% File: blocks.lp: Blocks World %%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % sort and object declaration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % every block is a location  location(B) :- block(B).  % the table is a location location(table). %%%%%%%%%%%%%%%%%%%%%%%%%% % state description %%%%%%%%%%%%%%%%%%%%%%%%%%  % two blocks can't be on the same block at the same time  :- 2{on(BB,B,T)}, block(B), T = 0..m.    % No more than n blocks on the table at once  % :- n+1{on(B,table,T)}, T = 0..m.  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % effect and preconditions of action  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    % effect of moving a block  on(B,L,T+1) :- move(B,L,T).    % concurrent actions are limited by num of grippers  :- not {move(BB,LL,T)} grippers, T = 0..m-1.    % a block can be moved only when it is clear  :- move(B,L,T), on(B1,B,T).    % a block can't be moved onto a block that is being moved also  :- move(B,B1,T), move(B1,L,T).    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%%%%%%%%%%%%%%%%%%%  % A serializable plan cannot move A onto a block B if anything is  % on B, which was allowed in unserializable plans if you move the  % block on B at the same time stamp.  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  :- move(A, B, T), on(C, B, T), block(B), A != C.    %%%%%%%%%%%%%%%%%%%%%%%%%%%%  % domain independent axioms  %%%%%%%%%%%%%%%%%%%%%%%%%%%%    % fluents are initially exogenous  1{on(B,LL,0):location(LL)}1 :- block(B).    % uniqueness and existence of value constraints  :- not 1{on(B,LL,T)}1, block(B), T=1..m.    % actions are exogenous  {move(B,L,T)} :- block(B), location(L), T = 0..m-1.    % commonsense law of inertia  {on(B,L,T+1)} :- on(B,L,T), T < m.    #show move/3. |
| Command  Line | Please only show the command line that outputs the minimal length plan.  ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=8 |
| Output  of clingo | Reading from blocks.asp ...  Solving...  Answer: 1  move(3,table,0) move(9,table,0) move(11,table,0) move(2,table,1) move(8,table,1) move(10,table,1) move(1,table,2) move(5,table,2) move(7,table,2) move(4,table,3) move(11,7,3) move(12,table,3) move(3,11,4) move(5,10,4) move(9,4,4) move(14,table,4) move(2,3,5) move(8,9,5) move(1,5,6) move(12,2,6) move(13,8,6) move(14,1,7) move(15,13,7)  SATISFIABLE    Models : 1+  Calls : 1  Time : 0.176s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.176s |

Q4

|  |  |
| --- | --- |
| Input  Program | %%%%%%%%%%%%%%%%%%%  % File: blocks-scenario.lp  %%%%%%%%%%%%%%%%%%%    % a1, b2, c3, d4, e5, f6, g7, h8, i9, j10, k11, l12, m13, n14, o15  block(1..15).    % initial state  :- not on(13,table,0;12,13,0;1,12,0;2,1,0;3,2,0;  15,table,0;14,15,0;4,14,0;5,4,0;10,5,0;  11,10,0;6,table,0;7,6,0;8,7,0;9,8,0).    % goal  :- not on(5,10,m;1,5,m;14,1,m;9,4,m;8,9,m;  13,8,m;15,13,m;11,7,m;3,11,m;2,3,m;12,2,m).  %%%%%%%%%%%%%%%%%%% File: blocks.lp: Blocks World %%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % sort and object declaration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % every block is a location  location(B) :- block(B).  % the table is a location location(table). %%%%%%%%%%%%%%%%%%%%%%%%%% % state description %%%%%%%%%%%%%%%%%%%%%%%%%%  % two blocks can't be on the same block at the same time  :- 2{on(BB,B,T)}, block(B), T = 0..m.    % No more than n blocks on the table at once  % :- n+1{on(B,table,T)}, T = 0..m.  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % effect and preconditions of action  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    % effect of moving a block  on(B,L,T+1) :- move(B,L,T).    % concurrent actions are limited by num of grippers  :- not {move(BB,LL,T)} grippers, T = 0..m-1.    % a block can be moved only when it is clear  :- move(B,L,T), on(B1,B,T).    % a block can't be moved onto a block that is being moved also  :- move(B,B1,T), move(B1,L,T).    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%%%%%%%%%%%%%%%%%%%  % A serializable plan cannot move A onto a block B if anything is  % on B, which was allowed in unserializable plans if you move the  % block on B at the same time stamp.  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  :- move(A, B, T), on(C, B, T), block(B), A != C.    %%%%%%%%%%%%%%%%%%%%%%%%%%%%  % domain independent axioms  %%%%%%%%%%%%%%%%%%%%%%%%%%%%    % fluents are initially exogenous  1{on(B,LL,0):location(LL)}1 :- block(B).    % uniqueness and existence of value constraints  :- not 1{on(B,LL,T)}1, block(B), T=1..m.    % actions are exogenous  {move(B,L,T)} :- block(B), location(L), T = 0..m-1.    % commonsense law of inertia  {on(B,L,T+1)} :- on(B,L,T), T < m.    % Minimize the number of moves  #minimize{1, B, L, T: move(B, L, T)}.  #show move/3. |
| Command  Line | You should write multiple command lines below.  ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=8  ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=9  ~/clingo-5.4.0-linux-x86\_64/clingo blocks.asp blocks-scenario.asp -c m=10 |
| Output  of clingo | You should write multiple outputs, one for each command. These outputs serve as the evidences of your answer to the question below.  Models : 6  Optimum : yes  Optimization : 18  Calls : 1  Time : 11.887s (Solving: 10.82s 1st Model: 0.01s Unsat: 0.10s)  CPU Time : 11.887s  Models : 8  Optimum : yes  Optimization : 16  Calls : 1  Time : 87.796s (Solving: 87.35s 1st Model: 0.00s Unsat: 6.48s)  CPU Time : 87.796s  Models : 9  Optimum : yes  Optimization : 15  Calls : 1  Time : 1665.337s (Solving: 1664.97s 1st Model: 0.00s Unsat: 1.16s)  CPU Time : 1665.287s |
| Answer  to Questions | What is the least number of actions when maxstep m is 8, 9, and 10?   |  |  | | --- | --- | | m | least number of actions | | 8 | 18 | | 9 | 16 | | 10 | 15 | |