**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.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).  %%%%%%%%%%%%%%%%%%%%%%%%%%%% % 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.  % space on table is limited  :- not {on(B,table,T)}s, T=0..m.  #show move/3. |
| Command  Line | You should write multiple command lines below.  clingo blocks-q1 blocks-scenario -c grippers=100 -c s=3 -c m=5  clingo blocks-q1 blocks-scenario -c grippers=100 -c s=4 -c m=3 |
| 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. |
| 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.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).  %%%%%%%%%%%%%%%%%%%%%%%%%%%% % 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.  % two same blocks cannot be top of one another  below(X,Y,T) :- on(Y,X,T).  below(X,Z,T) :- below(X,Y,T), below(Y,Z,T).  :- on(X,X,T).  :- on(X,Y,T), below(X,Y,T).  block(1..6).  #show move/3. |
| Command  Line | clingo blocks-q2 -c grippers=100 -c m=0 0 |
| Output  of clingo |  |
| 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.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).  %%%%%%%%%%%%%%%%%%%%%%%%%%%% % 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.  % serializability constraint  :- move(BB,LL,T), on(X,LL,T), move(X,B,T), block(X), T=1..m-1.  #show move/3. |
| Command  Line | Please only show the command line that outputs the minimal length plan.  clingo blocks-q3 blocks-instances -c grippers=4 -c m=8 |
| Output  of clingo |  |

Q4

|  |  |
| --- | --- |
| Input  Program | %%%%%%%%%%%%%%%%%%% % 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).  %%%%%%%%%%%%%%%%%%%%%%%%%%%% % 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.  % serializability constraint  :- move(BB,LL,T), on(X,LL,T), move(X,B,T), block(X), T=1..m-1.  #minimize{1,B,L,T:move(B,L,T)}.  #show move/3. |
| Command  Line | You should write multiple command lines below.  clingo blocks-q4 blocks-instances -c grippers=4 -c m=8 0  clingo blocks-q4 blocks-instances -c grippers=2 -c m=9 -t4  clingo blocks-q4 blocks-instances -c grippers=4 -c m=10 -t4 |
| 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. |
| 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 | |