CSE 579
Module 5 Graded Assignment
Template for clingo Work

Problem 1

Input Program	Hint: you only need one program with a new term, whose value will be assigned to 4 or 5 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). % Limit on the number of blocks that can be on the table at the same time :- not {on(B,table,T)} table_capacity, 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 move/3. %%%%%%%%%%%%%%%%%% % 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).
Command Lines	You should write multiple command lines below. clingo blocks.lp blocks-scenario.lp -c m=4 -c grippers=2 -c table_capacity=4 clingo blocks.lp blocks-scenario.lp -c m=3 -c grippers=2 -c table_capacity=5
Outputs of clingo	You should write multiple outputs, one for each command. These outputs serve as the evidences of your answer to the following question. Hint 1: Let n be the maximal number of blocks that can be placed directly on the table. There should be 2 command lines and outputs for n=4, where • the 1st command line and output show k steps are not enough and • the 2nd command line and output show k+1 steps are enough. Similarly, there should be another 2 command lines and outputs for n=5. Hint 2: Use only 2 grippers. **When capacity is 4** clingo version 5.6.2 Reading from blocks.lp ... Solving... Answer: 1 move(1,table,0) move(2,4,1) move(3,table,1) move(2,1,2) move(5,4,2) move(3,2,3) move(6,5,3)

	SATISFIABLE Models : 1+ Calls : 1 Time : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.013s **When capacity is 5** clingo version 5.6.2 Reading from blocks.lp ... Solving... Answer: 1 move(1,table,0) move(3,table,0) move(5,1,1) move(2,1,2) move(5,4,2) move(3,2,3) move(6,5,3) SATISFIABLE Models : 1+ Calls : 1 Time : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.013s
Answer to Questions	Fill in the following table that lists the minimum number of steps to solve the modified block world problem for different values of n, where n is the maximal number of blocks that can be placed directly on the table. n Number of steps Number of Moves 4 4 7 5 3 6

Problem 2

Input Program	Hint 1: You don't need to represent any scenario since you want to find out all possible valid states with 6 blocks. Think about the value of m. Hint 2: You don't need to consider the limitation of the maximum number of blocks on the table. That's only required in Problem 1. %%%%%%%%%%%%%%%%%% % 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. % additional constraints for valid initial state :- on(B1, B2, 0), block(B1), block(B2), B1 != B2. :- on(B, B, 0), block(B). #show on/3. %%%%%%%%%%%%%%%%%% % File: blocks-scenario.lp %%%%%%%%%%%%%%%%%% block(1..6). % initial state :- not on(1,table,0; 2,table,0; 3,table,0; 4,table,0; 5,table,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).
Command Line	clingo blocks.lp blocks-scenario.lp -c m=2 -c grippers=2 0
Output of clingo	clingo block.lp block-scenario.lp -c m=2 -c gripper=2 0 clingo version 5.6.2 Reading from block.lp ... Solving... Answer: 1 on(1,table,1) on(2,1,1) on(3,table,1) on(4,table,1) on(5,4,1) on(6,table,1) on(1,table,2) on(2,1,2) on(3,2,2) on(4,table,2) on(5,4,2) on(6,5,2) on(6,table,0) on(5,table,0) on(4,table,0) on(3,table,0) on(2,table,0) on(1,table,0) Answer: 2 on(1,table,1) on(2,1,1) on(3,table,1) on(4,table,1) on(5,4,1) on(6,table,1) on(1,table,2) on(2,1,2) on(3,2,2) on(4,table,2) on(5,4,2) on(6,5,2) on(6,table,0) on(5,table,0) on(4,table,0) on(3,table,0) on(2,table,0) on(1,table,0) Answer: 3 on(1,table,1) on(2,1,1) on(3,table,1) on(4,table,1) on(5,4,1) on(6,table,1) on(1,table,2) on(2,1,2) on(3,2,2) on(4,table,2) on(5,4,2) on(6,5,2) on(6,table,0) on(5,table,0) on(4,table,0) on(3,table,0) on(2,table,0) on(1,table,0) Answer: 4 on(1,table,1) on(2,1,1) on(3,table,1) on(4,table,1) on(5,4,1) on(6,table,1) on(1,table,2) on(2,1,2) on(3,2,2) on(4,table,2) on(5,4,2) on(6,5,2) on(6,table,0) on(5,table,0) on(4,table,0) on(3,table,0) on(2,table,0) on(1,table,0)

	SATISFIABLE Models : 4 Calls : 1 Time : 0.008s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.008s
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.) 18

Problem 3

Reading: A plan may allow multiple actions happening at the same time, e.g., when we have multiple robots working together to increase efficiency. However, if there is a little bit delay on one action, then we may get unexpected results. For example, when 2 robots are moving 2 adjacent blocks to the left at the same time, if there is a delay for the robot on the left-hand side, then these 2 robots may hit with each other. To make sure that our plan will get the expected result, we introduce the restriction "serializable" on the actions happening at the same time. This restriction simply says that, even if some actions in the same time stamp happen in serial with arbitrary ordering, the final result would be the same.

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).

	% Limit to one action per time step :- 2{move(BB,LL,T)}, T = 0..m-1. %%%%%%%%%%%%%%%%%%%%%%%%%%% % 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. %%%%%%%%%%%%%%%%%% % File: blocks-scenario.lp %%%%%%%%%%%%%%%%%% % Declare blocks block(m; l; a; b; c; o; n; d; e; j; k; f; g; h; i). % initial state :- not on(m,table,0; l,m,0; a,l,0; b,a,0; c,b,0; o,table,0; n,o,0; d,n,0; e,d,0; j,e,0; k,j,0; f,table,0; g,f,0; h,g,0; i,h,0). % goal state :- not on(e,j,m; a,e,m; n,a,m; i,d,m; h,i,m; m,h,m; o,m,m; k,g,m; c,k,m; b,c,m; l,b,m).
Command Line	Please only show the command line that outputs the minimal length plan. clingo blocks.lp blocks-scenario.lp -c m=14 -c grippers=2 --opt-mode=optN
Output of clingo	clingo version 5.6.2 Reading from blocks.lp ... Solving... Answer: 1 move(k,c,0) move(j,table,1) move(e,j,2) move(d,table,3) move(i,d,4) move(h,i,5) move(k,g,6) move(c,k,7) move(b,c,8) move(a,e,9) move(l,b,10) move(n,a,11) move(14,h,12) move(o,14,13)

	SATISFIABLE Models : 1+ Calls : 1 Time : 0.243s (Solving: 0.07s 1st Model: 0.07s Unsat: 0.00s) CPU Time : 0.243s

Problem 4

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. #minimize {1, move(B,L,T) : move(B,L,T)}. #show move/3.

	%%%%%%%%%%%%%%%%%% % File: blocks-scenario.lp %%%%%%%%%%%%%%%%%% % Declare blocks block(m; l; a; b; c; o; n; d; e; j; k; f; g; h; i). % initial state :- not on(m,table,0; l,m,0; a,l,0; b,a,0; c,b,0; o,table,0; n,o,0; d,n,0; e,d,0; j,e,0; k,j,0; f,table,0; g,f,0; h,g,0; i,h,0). % goal state :- not on(e,j,m; a,e,m; n,a,m; i,d,m; h,i,m; m,h,m; o,m,m; k,g,m; c,k,m; b,c,m; l,b,m).
Command Line	You should write multiple command lines below. clingo block.lp block-scenario.lp -c m=8 -c gripper=2 --opt-mode=optN clingo block.lp block-scenario.lp -c m=9 -c gripper=2 --opt-mode=optN clingo block.lp block-scenario.lp -c m=10 -c gripper=2 --opt-mode=optN clingo block.lp block-scenario.lp -c m=11 -c gripper=2 --opt-mode=optN
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. **When m=8** clingo version 5.6.2 Reading from block.lp ... Solving... Answer: 4 move(k,table,0) move(i,table,0) move(j,table,1) move(k,g,1) move(h,table,1) move(c,k,2) move(e,j,2) move(b,c,3) move(d,table,3) move(a,e,4) move(i,d,4) move(l,b,5) move(h,i,5) move(8,h,6) move(n,a,6) move(o,8,7) Optimization: 16 OPTIMUM FOUND **When m=9** clingo version 5.6.2 Reading from block.lp ... Solving... Answer: 3

	move(c,table,0) move(k,table,0) move(i,table,0) move(b,table,1) move(j,table,1) move(h,table,1) move(a,table,2) move(e,j,2) move(l,table,3) move(d,table,3) move(n,table,4) move(k,g,4) move(i,d,4) move(c,k,5) move(h,i,5) move(9,h,6) move(a,e,6) move(b,c,6) move(l,b,7) move(o,9,8) move(n,a,8) Optimization: 21 **When m = 10** clingo version 5.6.2 Reading from block.lp ... Solving... Answer: 4 move(c,table,0) move(k,table,0) move(b,table,1) move(j,table,1) move(a,table,2) move(e,table,2) move(l,table,3) move(d,table,3) move(n,table,4) move(i,d,5) move(e,j,6) move(k,g,6) move(h,i,6) move(a,e,7) move(c,k,7) move(10,h,8) move(b,c,8) move(n,a,8) move(l,b,9) move(o,10,9) Optimization: 20 **When m = 11** clingo version 5.6.2 Reading from block.lp ... Solving... Answer: 4 move(c,table,0) move(k,table,0) move(b,table,1) move(j,table,1) move(a,table,2) move(e,table,2) move(l,table,3) move(d,table,3) move(n,table,4) move(i,d,6) move(e,j,7) move(k,g,7) move(h,i,7) move(11,h,8) move(a,e,8) move(c,k,8) move(b,c,9) move(o,11,9) move(n,a,9) move(l,b,10) Optimization: 20
Answer to Questions	What is the least number of actions when maxstep m is 8, 9,10 and 11? m least number of actions 8 16 9 21 10 20 11 20