**CSE 579**

**Programming Assignment 1**

**Template for clingo Work**

Problem 1

|  |  |
| --- | --- |
| Input  Program | % 10\*10 board size queens problem.  #const n = 10.  % 1. Create a n \* n board  {queen(R, 1..n)}=1:-R=1..n.  % 2. Queens not in the same row.  :- queen(R1, C), queen(R2, C), R1 != R2.  % 3. Queens not in the same column.  :- queen(R, C1), queen(R, C2), C1 != C2.  % 4. No 2 queens in same diagonal  :- queen(R1, C1), queen(R2, C2), R1 != R2, |R1 - R2| == |C1 - C2|.  % 5. No queens in the 4 \* 4 = 16 squares in the 4 to 7 rows and columns of the board  :- queen(R, C), R >= 4, R <= 7, C >= 4, C <= 7. |
| Command  Line | clingo.exe .\P1\_NQueensMod\_Q1.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_NQueensMod\_Q1.lp  Solving...  Answer: 1  queen(2,1) queen(9,10) queen(7,9) queen(5,8) queen(3,7) queen(4,2) queen(1,6) queen(6,3) queen(8,4) queen(10,5)  Answer: 2  queen(7,8) queen(3,5) queen(5,1) queen(6,10) queen(2,7) queen(4,3) queen(1,9) queen(10,2) queen(8,6) queen(9,4)  Answer: 3  queen(3,4) queen(6,8) queen(1,5) queen(4,9) queen(5,3) queen(7,2) queen(2,10) queen(9,1) queen(8,7) queen(10,6)  Answer: 4  queen(1,2) queen(2,4) queen(3,6) queen(4,8) queen(5,10) queen(6,1) queen(7,3) queen(10,9) queen(9,7) queen(8,5)  Answer: 5  queen(9,10) queen(5,8) queen(3,7) queen(4,2) queen(1,6) queen(7,1) queen(2,9) queen(6,3) queen(8,4) queen(10,5)  Answer: 6  queen(1,2) queen(3,6) queen(4,8) queen(5,10) queen(6,1) queen(2,9) queen(7,3) queen(9,7) queen(8,5) queen(10,4)  Answer: 7  <Truncated>  Answer: 33  queen(10,10) queen(3,4) queen(7,9) queen(4,1) queen(5,8) queen(1,5) queen(2,7) queen(6,2) queen(9,3) queen(8,6)  Answer: 34  queen(3,4) queen(7,9) queen(4,1) queen(5,8) queen(2,7) queen(6,2) queen(1,10) queen(9,3) queen(8,6) queen(10,5)  Answer: 35  queen(3,5) queen(1,4) queen(5,1) queen(6,10) queen(4,9) queen(2,8) queen(7,2) queen(9,3) queen(8,6) queen(10,7)  Answer: 36  queen(2,2) queen(7,8) queen(3,5) queen(1,4) queen(5,1) queen(6,10) queen(4,9) queen(9,3) queen(8,6) queen(10,7)  Answer: 37  queen(2,3) queen(7,9) queen(3,6) queen(4,2) queen(5,10) queen(6,1) queen(1,7) queen(9,8) queen(8,5) queen(10,4)  Answer: 38  queen(3,6) queen(4,2) queen(5,10) queen(6,1) queen(1,7) queen(2,9) queen(7,3) queen(9,8) queen(8,5) queen(10,4)  Answer: 39  queen(2,4) queen(6,9) queen(3,7) queen(1,6) queen(4,10) queen(5,3) queen(7,2) queen(10,1) queen(9,8) queen(8,5)  Answer: 40  queen(1,1) queen(2,4) queen(6,9) queen(3,7) queen(4,10) queen(5,3) queen(7,2) queen(9,8) queen(8,5) queen(10,6)  SATISFIABLE  Models : 40  Calls : 1  Time : 0.149s (Solving: 0.08s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.016s |

Problem 2

|  |  |
| --- | --- |
| Input  Program | % 1. Create a n \* n board  {queen(R, 1..n)}=1 :- R=1..n.  % 2. Queens not in the same row.  :- queen(R1, C), queen(R2, C), R1 != R2.  % 3. Queens not in the same column.  :- queen(R, C1), queen(R, C2), C1 != C2.  % 4. No 2 queens in same diagonal  :- queen(R1, C1), queen(R2, C2), R1 != R2, |R1-R2|=|C1-C2|. |
| Command  Line | You should write multiple command lines below.   1. clingo -c n=3 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 2. clingo -c n=4 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 3. clingo -c n=5 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 4. clingo -c n=6 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 5. clingo -c n=7 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 6. clingo -c n=8 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 7. clingo -c n=9 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 8. clingo -c n=10 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 9. clingo -c n=11 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 10. clingo -c n=12 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 |
| Output  of clingo | Since the output is large, do not copy them into the submission. |
| Answer  to Questions | Draw a table that lists the number of solutions and the times to compute all solutions. Use CPU time that clingo returns.   |  |  |  | | --- | --- | --- | | Value n | Number of solutions | time | | 3 | 0 | 0.005s | | 4 | 2 | 0.000s | | 5 | 10 | 0.011s | | 6 | 4 | 0.007s | | 7 | 40 | 0.022s | | 8 | 92 | 0.049s | | 9 | 352 | 0.191s | | 10 | 724 | 0.540s | | 11 | 2680 | 3.310s | | 12 | 14200 | 5.281s | |

Problem 3

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line |  |
| Output  of clingo |  |

Problem 4

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line |  |
| Output  of clingo |  |

Problem 5

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line |  |
| Output  of clingo |  |

Problem 6

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line |  |
| Output  of clingo |  |

Problem 7

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line |  |
| Output  of clingo |  |

Problem 8

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line | You should write multiple command lines below. |
| Output  of clingo |  |
| Answer  to Questions | Draw a table that lists the maximum value of bishops when the chessboard is n by n, where n is 3, 4, 5, 6, 7, 8. Infer the general function f(n) that returns the maximum value of bishops.   |  |  | | --- | --- | | Value n | f(n) | | 3 |  | | 4 |  | | 5 |  | | 6 |  | | 7 |  | | 8 |  |   f(n) = |

Problem 9

|  |  |
| --- | --- |
| Input  Program |  |
| Command  Line | You should write multiple command lines below. |
| Output  of clingo |  |
| Answer  to Questions | Fill in the values accordingly.   |  |  | | --- | --- | | Exact value of A(1) |  | | Exact value of A(2) |  | | Exact value of A(3) |  | | Largest lower bound for A(4)  Note: it would take longer time when you increase the value of n. Thus, you may stop increasing the value of n when your program does not terminate within 10 minutes and submit the last trial of n. |  | |