CONSTRAINTS IN BOOLEAN LOGIC (SESSION 2)

ANTHONY W. LIN
YALE-NUS COLLEGE, SINGAPORE

CLASSICAL PROGRAMMING STYLE (PROCEDURAL)

```
// Sorting the array A

for i = 1 to length(A) - 1
    x = A[i]
    j = i
    while j > 0 and A[j-1] > x
        A[j] = A[j-1]
        j = j - 1
    end while
    A[j] = x[3]
 end for
```

Procedure for obtaining sorted(A)
is explicitly given

CONSTRAINT PROGRAMMING WITH BOOLEAN LOGIC

To write a program that output problem specs as formulas in boolean logic and let computers “fill in the details”.

Used to be only a dream, but SAT-solvers (for boolean logic) are fast nowadays!

OBJECTIVES OF THIS LECTURE

- To get started with constraint programming in boolean logic for an “interesting” toy problem:

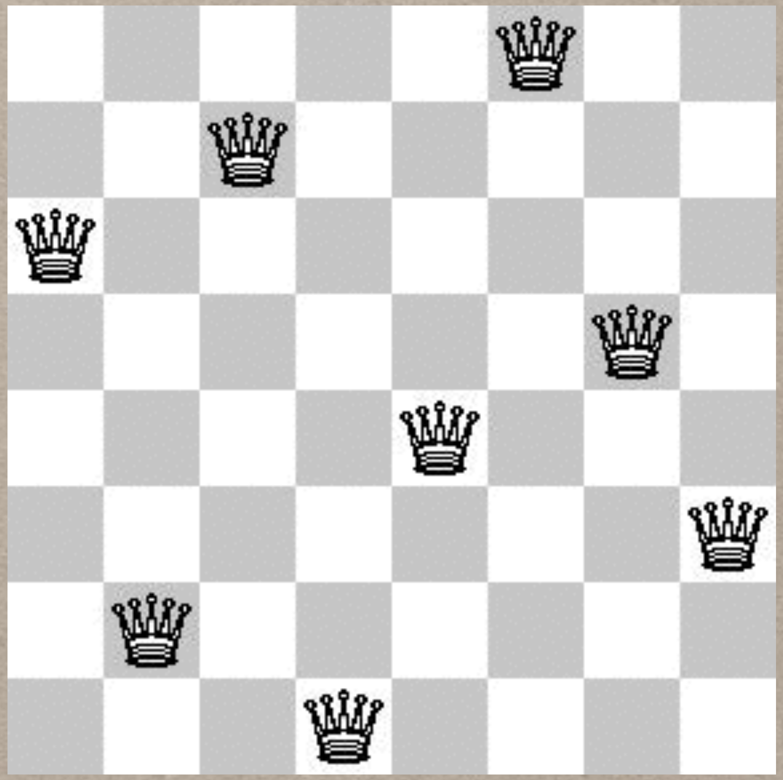
  n-queens (what else?)

- To get started with SAT-solvers

Note: more will be covered by Ric and others

N-QUEENS

A valid 8-queen configuration

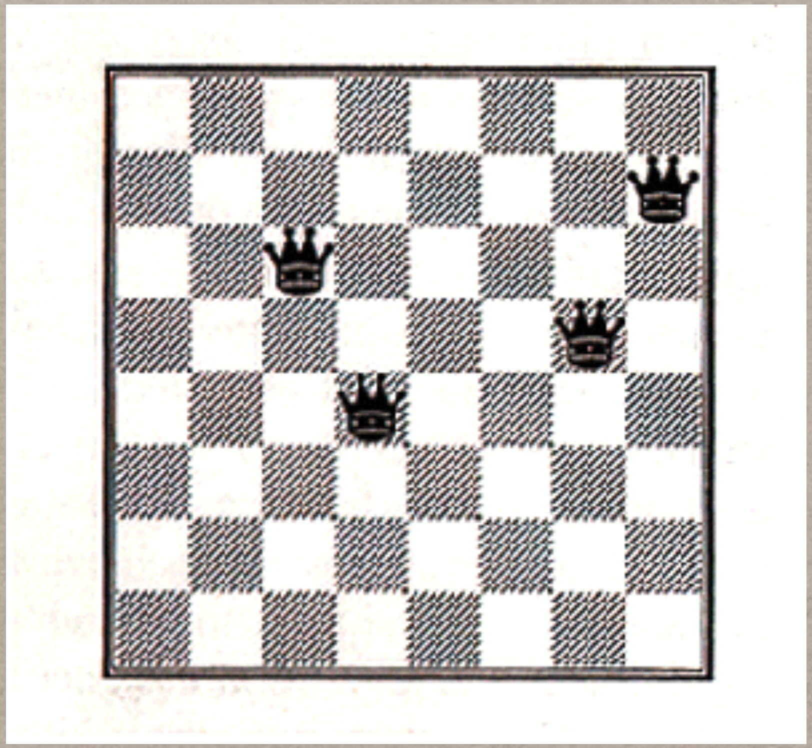


C1: Each row/column has exactly one queen

C2: The queens do not attack each other

Input: a number n
Output: a valid n-queen configuration

N-QUEENS (WITH INITIALISATION)



<u>Input</u>: a number n, partial n-queen configuration

<u>Output</u>: a valid n-queen configuration

CONSTRAINTS FOR N-QUEENS

1. Each row has precisely one queen
2. Each column has precisely one queen
3. Each diagonal has at most one queen

Let us write them down as a propositional formula

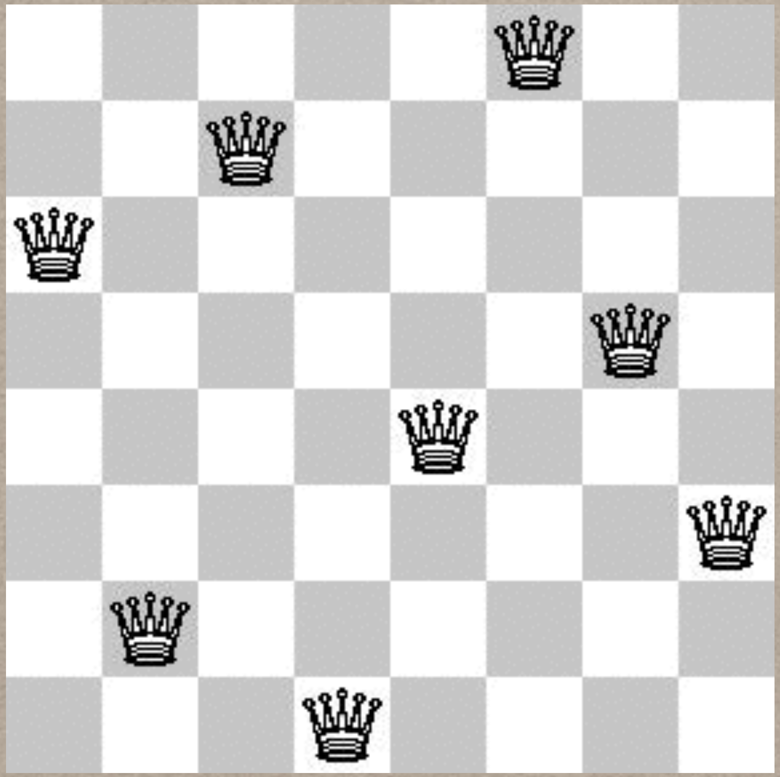
STEP 1: PICK YOUR VARIABLES

Tips: Think how to represent n-queens solutions as satisfying interpretations (i.e. assignments)

One approach: Have a variable $p_{i,j}$ for each row i and column j (1 <= i,j <= n)

$p_{i,j}$ *is 1 iff a queen is placed on row i and column j*

EXAMPLE



$$p_{1,6} = 1$$

$$p_{2,3} = 1$$

$$p_{3,1} = 1$$

$$p_{4,7} = 1$$

$$p_{5,5} = 1$$

$$p_{6,8} = 1$$

$$p_{7,2} = 1$$

$$p_{8,4} = 1$$

STEP 2: EXPRESS CONSTRAINTS

Constraint 1: Each row has precisely one queen

1.a: Each row has at least one queen

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} p_{i,j}$$

1.b: Each row has at most one queen

$$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{\substack{1 \leq k \leq n \\ k \neq j}} (p_{i,j} \to \neg p_{i,k})$$

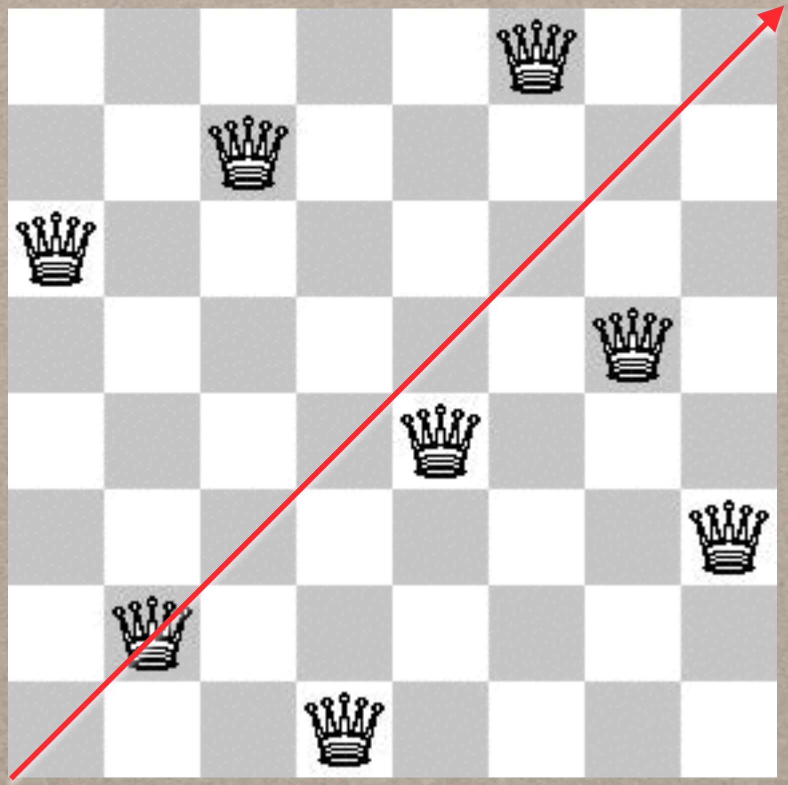
PONDERABLE

Question: How do you represent “Each column has exactly one queen”?

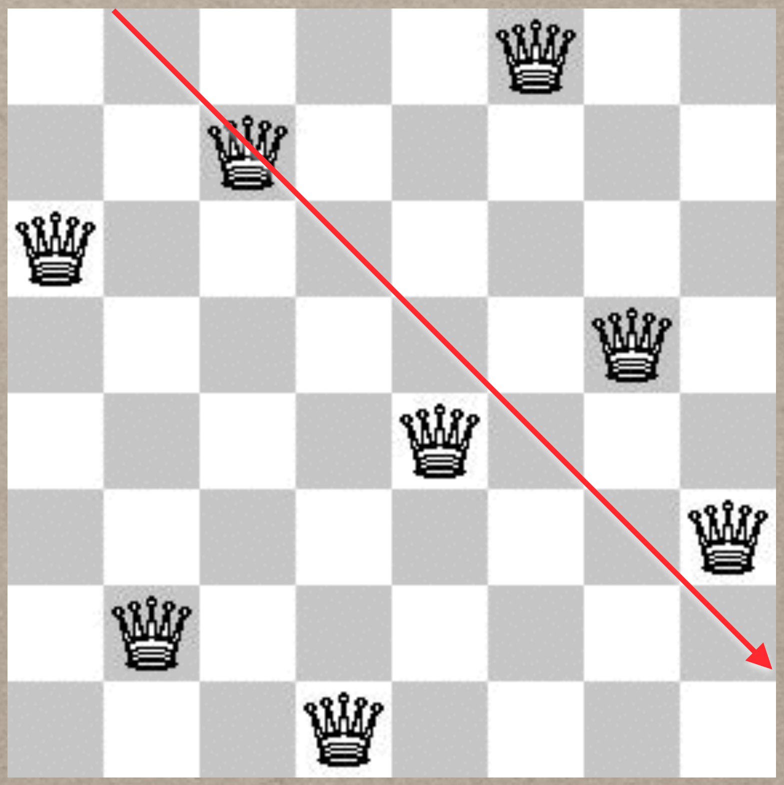
STEP 2.3: DIAGONAL CONSTRAINTS

Constraint 3: Each diagonal has at most one queen

Two types of diagonals!



45 degrees



-45 degrees

STEP 2.3.1: -45 DEGREES DIAGONAL CONSTRAINTS

Observation: other squares that are in the same -45-deg diagonal as $p_{i,j}$ must be of the form:

$$p_{i+k,j+k} \quad \textit{with} \quad 1 \leq i+k, j+k \leq n$$

or

$$p_{i-k,j-k} \quad \textit{with} \quad 1 \leq i-k, j-k \leq n$$

$$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{1 \leq i+k, j+k \leq n} (p_{i,j} \rightarrow \neg p_{i+k,j+k})$$

$$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{1 \leq i-k, j-k \leq n} (p_{i,j} \rightarrow \neg p_{i-k,j-k})$$

STEP 2.3: 45 DEGREES DIAGONAL CONSTRAINTS

Exercise!

TIME TO EXPERIMENT

***Instruction**: using http://logictools.org/index.html*
to solve a 3-queen problem and a 8-queen problem

SUMMARY

- Constraint programming in boolean logic for the n-queen problem
- Homework: can you solve the 4x4 sudoku problem via SAT-solver?