ST. FRANCIS XAVIER UNIVERSITY

CSCI-564 CONSTRAINT PROCESSING AND HEURISTIC SEARCH

LECTURE 20 – CONSTRAINT PROCESSING

Dr. Jean-Alexis Delamer

Constraint Processing

- What is a constraint?
  - A constraint is a restriction on a space of possibilities.
  - It is a piece of knowledge that narrows the scope of this space.

- Do you have any example of a constraint?
  - The amount of memory in a PC.
  - The number of seats in a car.
  - Hours in a day.
  - Money in the bank.

Constraint Processing

- So, we're always solving constraint satisfaction problem:
  - Saving money but still going in vacation.
  - Etc.

- Try to think about the complexity of the problem when the number of constraints and variables grow.
  - Any example of hard constraint problem (for a human)?
    - Optimal seating arrangement for a dinner party.
  - It can be complicated.

Constraint Processing

- Now imagine the difficulty in scheduling classrooms for the fall term.

- What constraint do we have?
  - A classroom for every course.
  - Only one class per classroom simultaneously.
  - No professor can teach in two different classroom at the same time.
  - No class during the night.
  - The capacity of the classroom.
  - Some class cannot be scheduled at the same time.
  - Etc.

Constraint Processing

- It is a problem very hard for a group of human.

- But is it for a computer?
  - Yes, in general it is NP-Hard.

- You cannot expect to design algorithms that scale efficiently for every problem.

Constraint Processing

- But!
  - We can identify special properties of a problem class.
  - Help to design general algorithm that are efficient for this class.

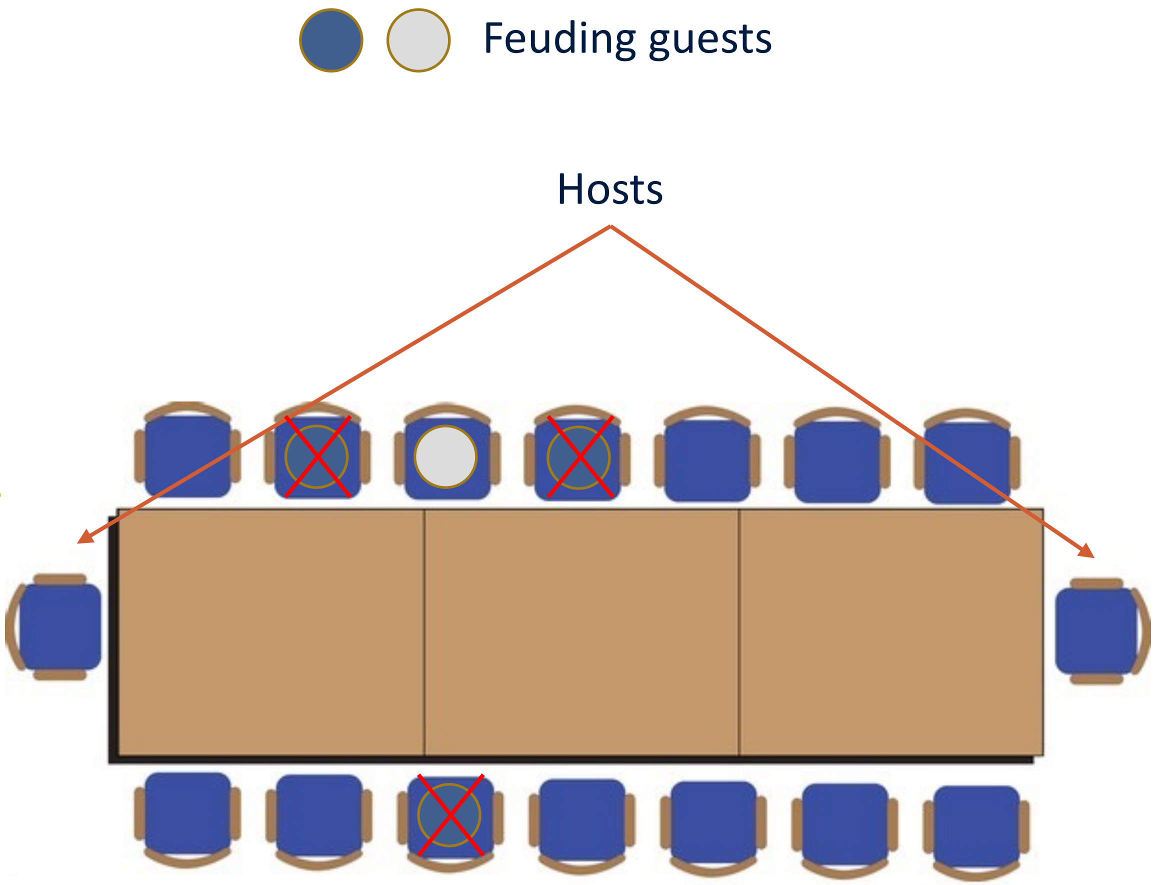
- Algorithms have been developed for restricted subclasses.
  - Solve them efficiently.
  - Or propose a good approximation.
  - They are called tractable classes.

Constraint Processing

- Constraint satisfaction problems (CSP) incudes two important components:
  - Variable and domains.
  - Constraints.
- Variables:
  - Objects or items that can take on a variety of values. (Like in math).
  - The set of possible values for a variable is called its domain.
- Constraints:
  - Rules that impose a limitation on the values that a variable may be assigned.

Constraint Processing

- Example: Seating arrangement.
  - What would be the variables?
    - Chairs.
  - What would be the domain?
    - The list of guests.
    - Same domain for each variable.
  - What are the constraints?
    - The hosts must sit at the two ends of the table.
    - Two feuding guests must not be placed next to or directly opposite one another.



Constraint Processing

- There is often more than one way to model a problem.
- In the previous example:
  - Variables could be the guests.
  - And the domain the chairs.
  - In this example it doesn't make any difference.
- However, in other problem (more complex) some model make it easier to obtain a solution.

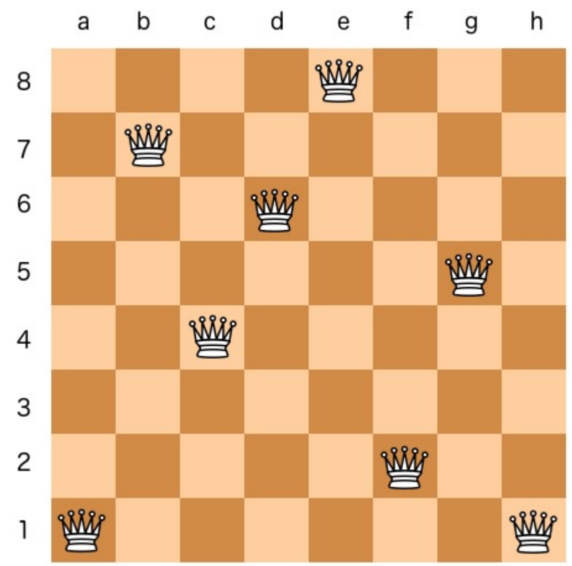
Constraint Processing

- Constraint problems are often referred as constraint network.
  - The constraints dependency structure are captured naturally with a network.

- Do you think that CSP is just about finding a solution?
  - Determine whether a solution exist.
  - Finding one or all solution.
  - Finding the optimal solution.
  - Determine if a partial solution can be extended to a full solution.

N-Queens Problem

- First example (Informal).
- Variables:
  - One for each column.
  - $x_1, \dots, x_n$.
- Domains:
  - Possible row positions.
  - $D_i = \{1, \dots, n\}$
- Constraints:
  - On each pair of columns, two queens must no share
    - A row
    - Or a diagonal



Constraint Processing

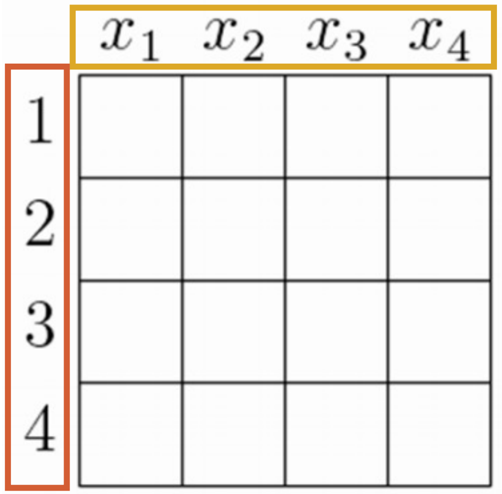
- A constraint network $R = (X, D, C)$
  - A set of finite variables $X = \{\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}\}$
  - A set of domains $D = \{D_1, \ldots, D_n\}$
    - Where $D_i = \{v_1, \ldots, v_k\}$
  - A set of constraints $C = \{C_1, \ldots, C_n\}$
    - Where $C_i = (S_i, R_i)$, R_i expresses allowed tuples called scope.

- A solution is an assignment to all variables that satisfies all constraints (join of all relations).

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

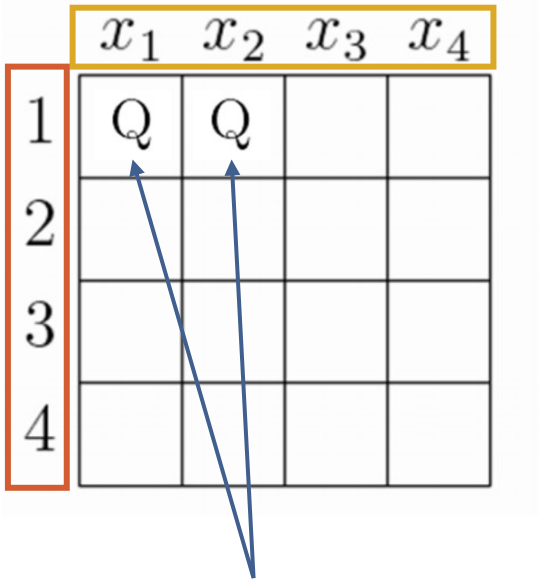
$R_{12} = \{\}$



N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$R_{12} = \{\}$

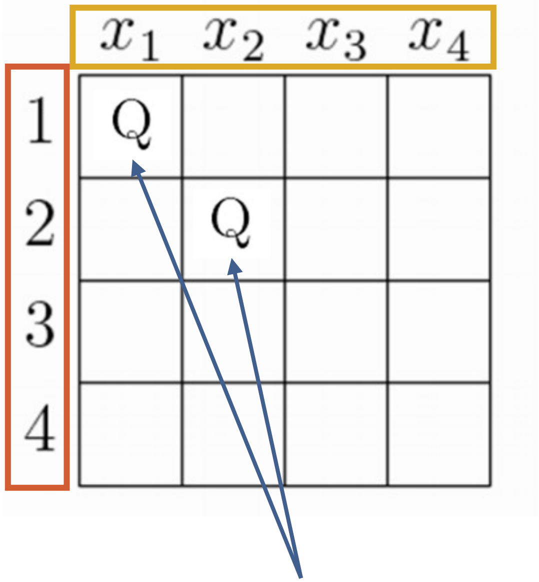


Share a row.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$R_{12} = \{\}$

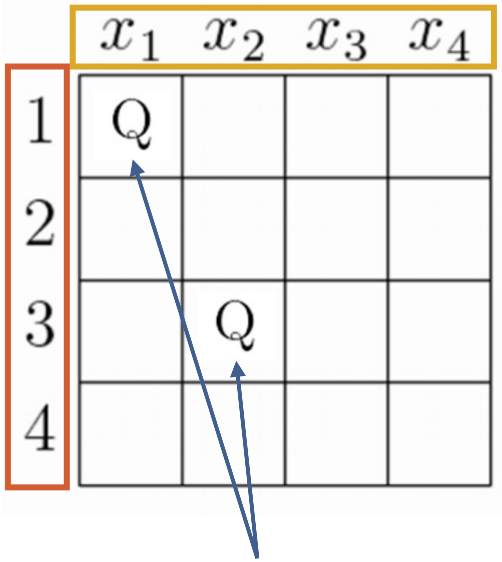


Share a diagonal.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$R_{12} = \{(1,3)\}$

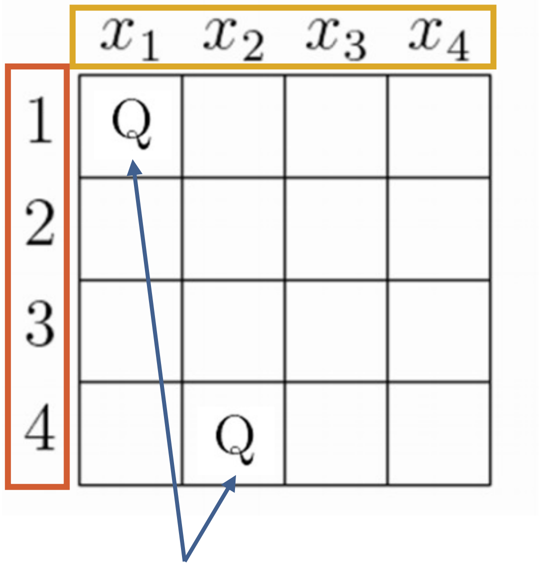


Respect the constraint.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$R_{12} = \{(1,3), (1,4)\}$

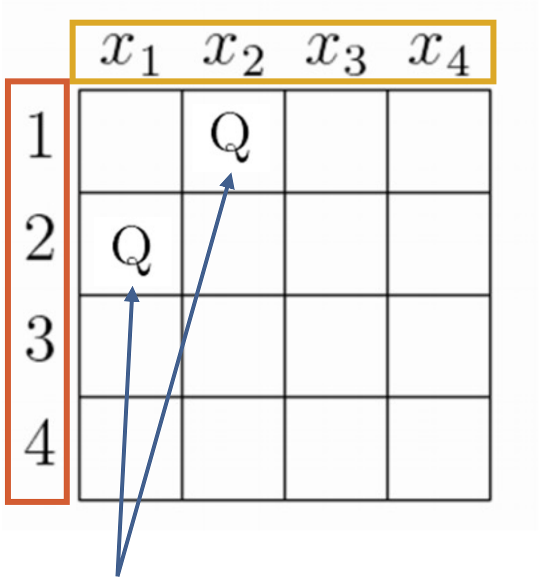


Respect the constraint.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$$R_{12} = \{(1,3), (1,4)\}$$

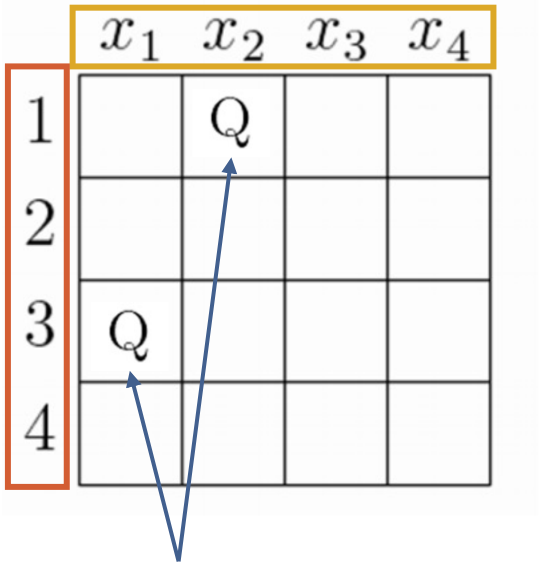


Share a diagonal.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$$R_{12} = \{(1,3), (1,4), (3,1)\}$$

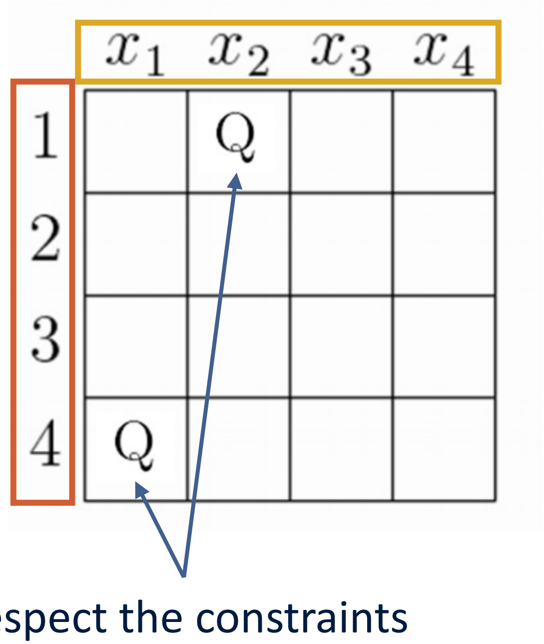


Respect the constraints

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

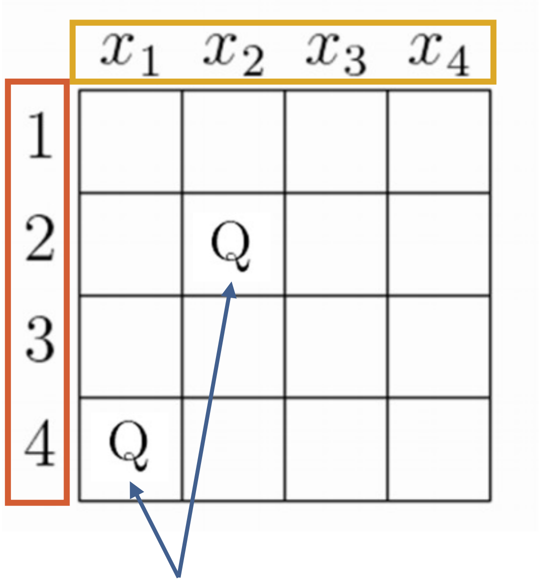
$$R_{12} = \{(1,3), (1,4), (3,1), (4,2)\}$$



N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$$R_{12} = \{(1,3), (1,4), (3,1), (4,2), (4,3)\}$$

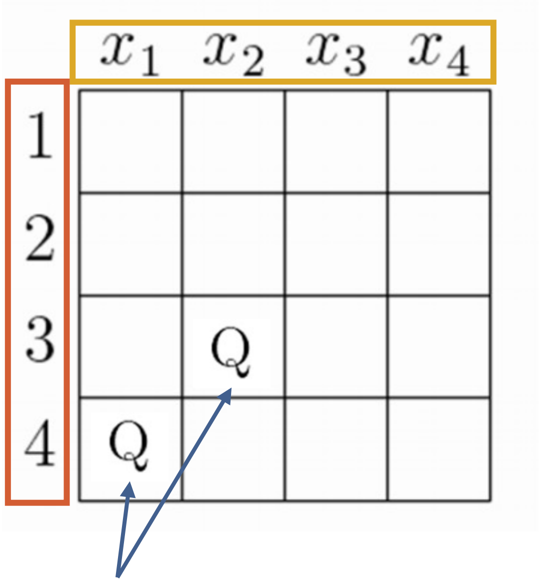


Respect the constraints

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

$$R_{12} = \{(1,3), (1,4), (3,1), (4,2), (4,3)\}$$

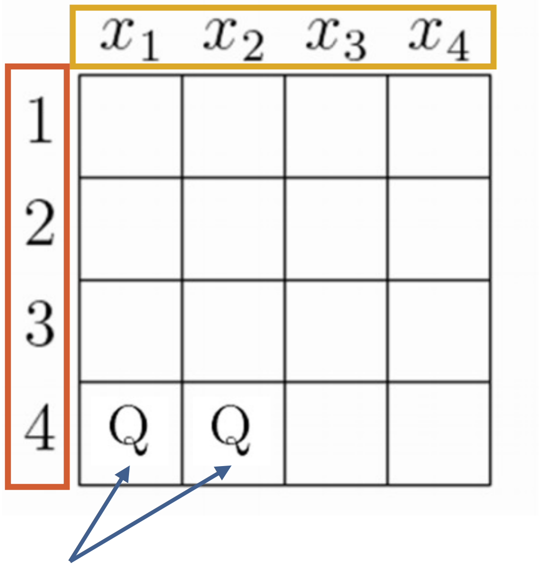


Share a diagonal.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal

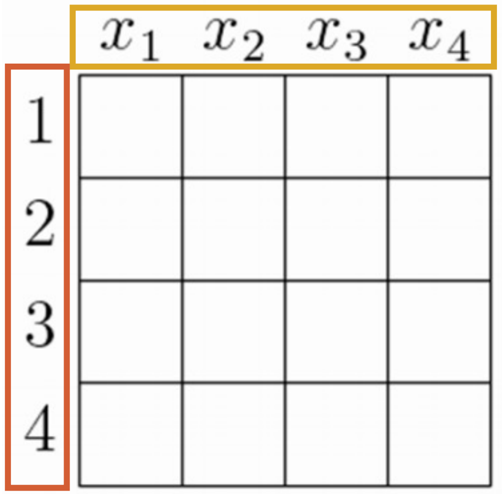
$$R_{12} = \{(1,3), (1,4), (3,1), (4,2), (4,3)\}$$



Share a row.

N-Queens Problem

- We define our constraint network $R = (X, D, C)$.
  - $X = \{x_1, x_2, x_3, x_4\}$
  - $D = \{1,2,3,4\}$
  - $C = \{R_{12}, R_{13}, R_{14}, R_{23}, R_{24}, R_{34}\}$
    - On each pair of columns, two queens must no share a row or a diagonal



$$R_{12} = \{(1,3), (1,4), (3,1), (4,2), (4,3)\}$$
$$R_{13} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}$$
$$R_{14} = \{(1,2), (1,3), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,2), (4,3)\}$$
$$R_{23} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$$
$$R_{24} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}$$
$$R_{34} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$$

Solutions of a Constraint Network

- When a variable is assigned a value, we say that the variable has been instantiated.

- An instantiation of a set of variables $\{x_{i_1}, \dots, x_{i_k}\}$ is a tuple of ordered pairs $\left((x_{i_1}, a_{i_1}), \dots, (x_{i_k}, a_{i_k})\right)$
  - Where each pair (x, a) represents the assignment of value a to x.
  - We can abbreviate $\bar{a} = \{a_1, \dots, a_i\}$.

Solutions of a Constraint Network

- Satisfying constraint:
  - An instantiation satisfies a constraint (S, R) iff
    - It is defined over all the variables in S
    - And the components of the tuple $\bar{a}$ are present in the relation R
- Example:
  - Let $R_{xyz} = \{(1,1,1), (1,0,1), (0,0,0)\}$
  - An instantiation $\bar{a}$ with a scope $\{x, y, z, t\}$, such $\bar{a} = \{(x, 1), (y, 1), (z, 1), (t, 0)\}$.
    - It satisfies the constraint, because $(1,1,1)$ is in R_{xyz}.
  - An instantiation $\bar{a}$ with a scope $\{x, y, z, t\}$, such $\bar{a} = \{(x, 1), (y, 0), (z, 0), (t, 0)\}$.
    - It does not satisfy the constraint, because $(1,0,0)$ is not in R_{xyz}.

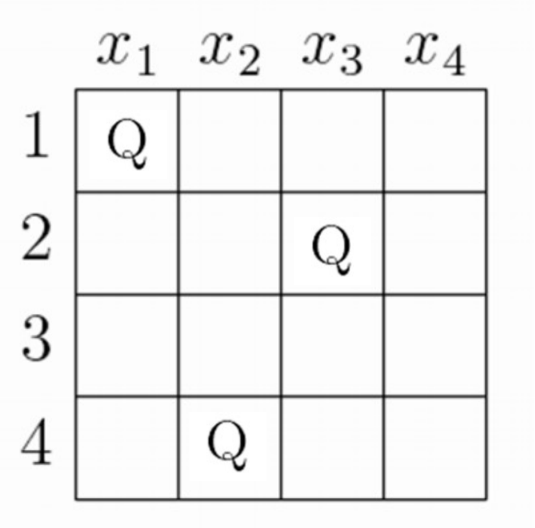
Solutions of a Constraint Network

- A partial instantiation is consistent if:
  - It satisfies all the constraints whose scopes have no uninstantiated variables.

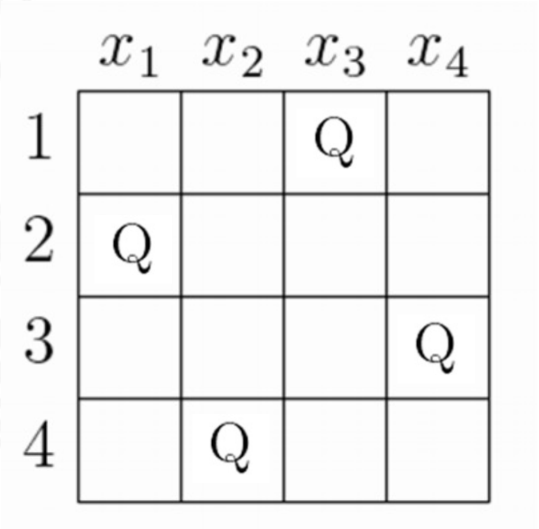
- A solution of a constraint network:
  - Is an instantiation of all its variables that satisfies all the constraints.

N-Queens Problem

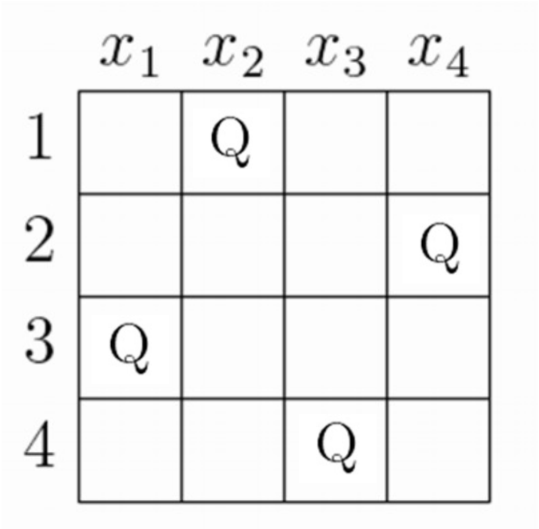
- Let us consider $\bar{a} = (1,4,2)$.
  - It is consistent instantiation.
  - It is not a part of a solution.



N-Queens Problem

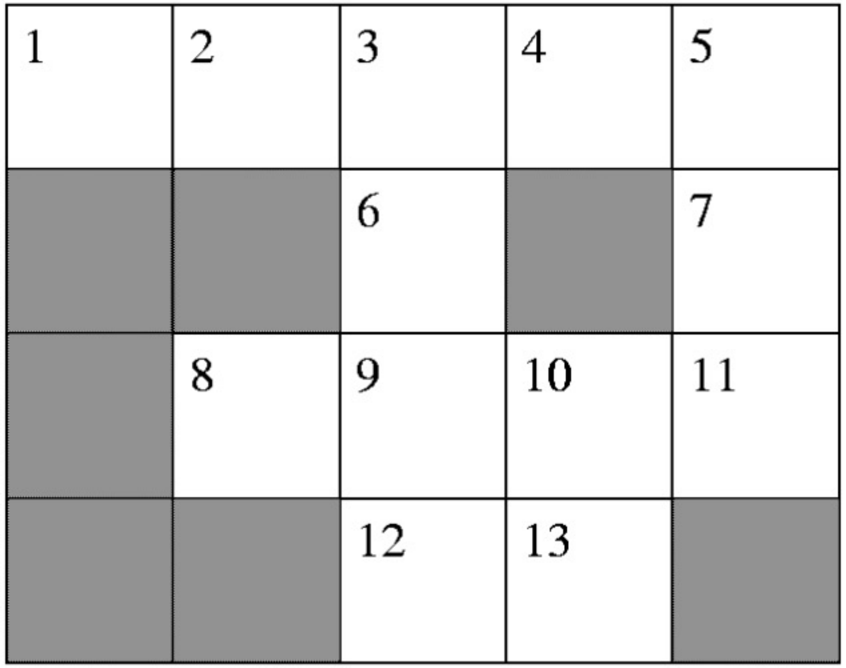


A solution: $\bar{a} = (2,4,1,3)$



A solution: $\bar{a} = (3,1,4,2)$

Exercise: Crossword puzzle



{HOSES, LASER, SHEET, SNAIL, STEER, ALSO, EARN, HIKE, IRON, SAME, EAT, LET, RUN, SUN, TEN, YES, BE, IT, NO, US}