CompSci 275, CONSTRAINT Networks

Rina Dechter, Fall 2022

Introduction, the constraint network model
Chapters 1-2
Class information

• Instructor: Rina Dechter
• Lectures: Monday & Wednesday
• Time: 11:00 – 12:20 pm

• Class page: https://www.ics.uci.edu/~dechter/courses/ics-275/fall-2022/
Text book (required)

Rina Dechter,

**Constraint Processing**, Morgan Kaufmann
Outline

- Motivation, applications, history
- CSP: Definition, and simple modeling examples
- Mathematical concepts (relations, graphs)
- Representing constraints
- Constraint graphs
- The binary Constraint Networks properties
Outline

✓ Motivation, applications, history
✓ CSP: Definition, representation and simple modeling examples
✓ Mathematical concepts (relations, graphs)
✓ Representing constraints
✓ Constraint graphs
✓ The binary Constraint Networks properties
Combinatorial problems

Those problems that can be expressed as:

A set of variables

Each variable takes its values from a finite set of domain values

A set of local functions

Main advantage:

They provide unifying algorithms:
- Search
- Complete Inference
- Incomplete Inference
Combinatorial problems

Many Examples

- Metastatic Cancer
- Serum Calcium
- Brain Tumor
- Coma
- Headache

EOS Scheduling
Bayesian Networks

Graph Coloring
Timetabling

... and many others.
Example: student course selection

- **Context:** You are a senior in college
- **Problem:** You need to register in 4 courses for the Spring semester
- **Possibilities:** Many courses offered in Math, CSE, EE, CBA, etc.
- **Constraints:** restrict the choices you can make
  - Courses have **prerequisites** (e.g., take 171 before 175)
  - General course restrictions [https://www.ics.uci.edu/grad/Course_updates.php](https://www.ics.uci.edu/grad/Course_updates.php)
  - you have/don't have Courses/instructors you like/dislike
  - Courses are scheduled at the same time
  - In CE: 4 courses from 5 tracks such that at least 3 tracks are covered

- **You have choices, but are restricted by constraints**
  - Make the right decisions!!
  - [ICS Graduate program](https://www.ics.uci.edu/grad/Course_updates.php)
Student course selection (continued)

• **Given**
  – A set of variables: 4 courses at your college
  – For each variable, a set of choices (values): the available classes.
  – A set of constraints that restrict the combinations of values the variables can take at the same time

• **Questions**
  – Does a solution exist? (classical decision problem)
  – How many solutions exists? (counting)
  – How two or more solutions differ?
  – Which solution is preferable?
  – etc.
The field of constraint programming

• **How did it start:**
  – Artificial Intelligence (vision)
  – Programming Languages (Logic Programming),
  – Databases (deductive, relational)
  – Logic-based languages (propositional logic)
  – SATisfiability

• **Related areas:**
  – Hardware and software verification
  – Operation Research (Integer Programming)
  – Answer set programming

• **Graphical Models; deterministic**
One of the earliest constraint satisfaction is three-dimensional interpretation of a two dimensional drawing (1970). Huffman and Clowes [291] developed a labelling scheme of the arcs in a block world picture graph, where + for a convex, - for concave and arrow for occluding boundaries. Label the junction of a given drawing such that every junction type is labelled according to one of its legal labeling and that edges common to two junctions receive the same label.
Scene labeling constraint network
3-dimentional interpretation of 2-dimentional drawings
The field of constraint programming

• **How did it start:**
  – Artificial Intelligence (vision)
  – Programming Languages (Logic Programming),
  – Databases (deductive, relational)
  – Logic-based languages (propositional logic)
  – SATisfiability

• **Related areas:**
  – Hardware and software verification
  – Operations Research (Integer Programming)
  – Answer set programming

• **Graphical Models; deterministic**
Applications

- Radio resource management (RRM)
- Databases (computing joins, view updates)
- Temporal and spatial reasoning
- Planning, scheduling, resource allocation
- Design and configuration
- Graphics, visualization, interfaces
- Hardware verification and software engineering
- HC Interaction and decision support
- Molecular biology
- Robotics, machine vision and computational linguistics
- Transportation
- Qualitative and diagnostic reasoning

... and many others.
Motivation, applications, history

CSP: Definitions and simple modeling examples

Mathematical concepts (relations, graphs)

Representing constraints

Constraint graphs

The binary Constraint Networks properties


**Example: map coloring**

Variables - countries (A, B, C, etc.)

Values    - colors (red, green, blue)

Constraints: \( A \neq B, \ A \neq D, \ D \neq E, \) etc.
Constraint satisfaction tasks

Example: map coloring

Variables - countries (A, B, C, etc.)
Values - colors (e.g., red, green, yellow)
Constraints: \( A \neq B, A \neq D, D \neq E, \text{ etc.} \)

Are the constraints consistent?
Find a solution, find all solutions
Count all solutions
Find a good solution
Information as constraints

- I have to finish my class in 50 minutes
- 180 degrees in a triangle
- Memory in our computer is limited
- The four nucleotides that make up a DNA only combine in a particular sequence
- Sentences in English must obey the rules of syntax
- Susan cannot be married to both John and Bill
- Alexander the Great died in 333 B.C.
Constraint network; definition

- A constraint network is: \( R=\langle X, D, C \rangle \)
  - \( X \) variables \( X = \{X_1, \ldots, X_n\} \)
  - \( D \) domain \( D = \{D_1, \ldots, D_n\}, D_i = \{v_1, \ldots, v_k\} \)
  - \( C \) constraints \( C = \{C_1, \ldots, C_t\}, \ldots, C_i = (S_i, R_i) \)
  - \( R \) expresses allowed tuples over scopes

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

- Tasks: consistency?, one or all solutions, counting, optimization
% Colouring book-map using nc colours
int: nc = 3;

var 1..nc: A; var 1..nc: B; var 1..nc: C; var 1..nc: D;
var 1..nc: E; var 1..nc: F; var 1..nc: G;

constraint A != D;
constraint A != B;
constraint B != D;
constraint B != G;
constraint B != C;
constraint C != G;
constraint D != G;
constraint D != F;
constraint G != F;
solve satisfy;
The N-queens problem

The network has four variables, all with domains $D_i = \{1, 2, 3, 4\}$. (a) The labeled chess board. (b) The constraints between variables.

\[
\begin{array}{cccc}
  & x_1 & x_2 & x_3 & x_4 \\
1 & & & & \\
2 & & & & \\
3 & & & & \\
4 & & & & \\
\end{array}
\]

\[
R_{12} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\} \\
R_{13} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\} \\
R_{14} = \{(1,2), (1,3), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,2), (4,3)\} \\
R_{23} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\} \\
R_{24} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\} \\
R_{34} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}
\]
A solution and a partial consistent tuple (configuration)

Not all consistent instantiations are part of a solution:
(a) A consistent instantiation that is not part of a solution.
(b) The placement of the queens corresponding to the solution (2, 4, 1,3).
(c) The placement of the queens corresponding to the solution (3, 1, 4, 2).
Example: crossword puzzle

- Variables: $x_1, \ldots, x_{13}$
- Domains: letters
- Constraints: words from

{HOSES, LASER, SHEET, SNAIL, STEER, ALSO, EARN, HIKE, IRON, SAME, EAT, LET, RUN, SUN, TEN, YES, BE, IT, NO, US}
Example: Sudoku (constraint propagation)

Each row, column and major block must be all different

“Well posed” if it has unique solution: 27 constraints
Sudoku (inference)

Each row, column and major block must be alldifferent

“Well posed” if it has unique solution
Outline

✓ Motivation, applications, history
✓ CSP: Definition, representation and simple modeling examples
✓ Mathematical concepts (relations, graphs)
✓ Representing constraints
✓ Constraint graphs
✓ The binary Constraint Networks properties
Mathematical background

• Sets, domains, tuples
• Relations
• Operations on relations
• Graphs
• Complexity
Two Representations of a relation: 
\[ R = \{(\text{black}, \text{coffee}), (\text{black}, \text{tea}), (\text{green}, \text{tea})\}. \]

Variables: Drink, color

<table>
<thead>
<tr>
<th>(x_1)</th>
<th>(x_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>coffee</td>
</tr>
<tr>
<td>black</td>
<td>tea</td>
</tr>
<tr>
<td>green</td>
<td>tea</td>
</tr>
</tbody>
</table>

(a) table
Two Representations of a relation:
\( R = \{(\text{black, coffee}), (\text{black, tea}), (\text{green, tea})\} \).

Variables: Drink, color

(a) Table

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>coffee</td>
</tr>
<tr>
<td>black</td>
<td>tea</td>
</tr>
<tr>
<td>green</td>
<td>tea</td>
</tr>
</tbody>
</table>

(b) \((0,1)\)-matrix

\[
\begin{bmatrix}
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\]
Examples

\[
\begin{array}{ccc}
  x_1 & x_2 & x_3 \\
  a & b & c \\
  b & b & c \\
  c & b & c \\
  c & b & s \\
\end{array}
\quad
\begin{array}{ccc}
  x_1 & x_2 & x_3 \\
  b & b & c \\
  c & b & c \\
  c & n & n \\
\end{array}
\quad
\begin{array}{ccc}
  x_2 & x_3 & x_4 \\
  a & a & 1 \\
  b & c & 2 \\
  b & c & 3 \\
\end{array}
\]

(a) Relation \( R \)  \quad (b) Relation \( R'' \)  \quad (c) Relation \( R''' \)
Operations with relations

• Intersection
• Union
• Difference
• Selection
• Projection
• Join
• Composition
Relations are local functions

- Relations specify local functions.

- A general function

$$f : \prod_{x_i \in Y} D_i \rightarrow A$$

where

$$\text{scope}(f) = Y \subseteq X: \text{scope of function } f$$

$$A: \text{is a set of valuations}$$

In a relation \( A = \{\text{false, true}\} \), telling if the tuple is in or out of the relation.

- In constraint networks: functions are boolean

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>true</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>true</td>
</tr>
</tbody>
</table>

relation

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
</tr>
</tbody>
</table>
Set operations: intersection, union, difference on relations.

(a) Relation $R$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>s</td>
</tr>
</tbody>
</table>

(b) Relation $R'$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

(c) Relation $R''$

<table>
<thead>
<tr>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>3</td>
</tr>
</tbody>
</table>

(a) $R \cap R'$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>s</td>
</tr>
<tr>
<td>c</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

(b) $R \cup R'$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
</tr>
</tbody>
</table>

(b) $R - R'$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>s</td>
</tr>
</tbody>
</table>
Selection, projection, join

(a) Relation $R$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>s</td>
</tr>
</tbody>
</table>

(b) Relation $R'$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

(c) Relation $R''$

<table>
<thead>
<tr>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>3</td>
</tr>
</tbody>
</table>

(a) $\sigma_{x_3=c}(R')$

(b) $\pi_{x_2,x_3}(R')$

(c) $R' \bowtie R''$
The join and the logical “and”

- Join:

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
</tr>
</tbody>
</table>

- Logical AND:

$f \land g$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>true</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>false</td>
</tr>
</tbody>
</table>

Fall 2022

38
Outline

- Motivation, applications, history
- CSP: Definition, representation and simple modeling examples
- Mathematical concepts (relations, graphs)
- Representing constraints/ Languages
- Constraint graphs
- The binary Constraint Networks properties
Modeling; Representing a problem

- If a CSP $M = \langle X, D, C \rangle$ represents a real problem $P$, then every solution of $M$ corresponds to a solution of $P$ and every solution of $P$ can be derived from at least one solution of $M$.
- The variables and values of $M$ represent entities in $P$.
- The constraints of $M$ ensure the correspondence between solutions.
- The aim is to find a model $M$ that can be solved as quickly as possible.
- **Goal of modelling**: choose a set of variables and values that allows the constraints to be expressed easily and concisely.
Example: satisfiability

Given a proposition theory

\[ \varphi = \{(A \lor B), (C \lor \neg B)\} \]

does it have a model?

Can it be encoded as a constraint network?

Variables: \{A, B, C\}

Domains: \[ D_A = D_B = D_C = \{0, 1\} \]

Relations:

<table>
<thead>
<tr>
<th>(A)</th>
<th>(B)</th>
<th>(B)</th>
<th>(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Constraint’s representations

- Relation: allowed tuples
  
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

- Algebraic expression:
  \[ X + Y^2 \leq 10, X \neq Y \]

- Propositional formula:
  \((a \lor b) \rightarrow \neg c\)

- A decision tree, a procedure

- Semantics: by a relation

A decision tree or a neural network
Decision tree representations

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Outline

✓ Motivation, applications, history
✓ CSP: Definition, representation and simple modeling examples
✓ Mathematical concepts (relations, graphs)
✓ Representing constraints
✓ Constraint graphs
✓ The binary Constraint Networks properties
Constraint graphs:
Primal, Dual and Hypergraphs

- A (primal) constraint graph: a node per variable, arcs connect constrained variables.
- A dual constraint graph: a node per constraint's scope, an arc connects nodes sharing variables = hypergraph.

\[ R\{1,2,3,4,5\} = \{(H,O,S,E,S), (L,A,S,E,R), (S,H,E,E,T), (S,N,A,I,L), (S,T,E,E,R)\} \]
\[ R\{3,6,9,12\} = \{(A,L,S,O), (E,A,R,N), (H,I,K,E), (I,R,O,N), (S,A,M,E)\} \]
\[ R\{5,7,11\} = \{(E,A,T), (L,E,T), (R,U,N), (S,U,N), (T,E,N), (Y,E,S)\} \]
A hypergraph

Primal graphs

Dual graph

Factor graphs

Definition 2.1.1 (hypergraph) A hypergraph is a structure $H = (V, S)$ that consists of vertices $V = \{v_1, \ldots, v_n\}$ and a set of subsets of these vertices $S = \{S_1, \ldots; S_l\}$
Example: cryptarithmetic

Variables: \( F \ T \ U \ W \ R \ O \ X_1 \ X_2 \ X_3 \)

Domains: \( \{0,1,2,3,4,5,6,7,8,9\} \)

Constraints: \textit{Alldiff} \( (F,T,U,W,R,O) \)

\begin{align*}
O + O &= R + 10 \cdot X_1 \\
X_1 + W + W &= U + 10 \cdot X_2 \\
X_2 + T + T &= O + 10 \cdot X_3 \\
X_3 &= F, \ T \neq 0, \ F \neq 0
\end{align*}

What is the primal graph? What is the dual graph?
Propositional satisfiability

\[ \varphi = \{ (\neg C), (A \lor B \lor C), (\neg A \lor B \lor E), (\neg B \lor C \lor D) \}. \]
Given a telecommunication network (where each communication link has various antennas), assign a frequency to each antenna in such a way that all antennas may operate together without noticeable interference.

**Encoding?**

- **Variables:** one for each antenna
- **Domains:** the set of available frequencies
- **Constraints:** the ones referring to the antennas in the same communication link
Constraint graphs, 3 instances of radio frequency assignment in CELAR’s benchmark
Example: a scheduling problem

Five tasks: T1, T2, T3, T4, T5
Each one takes one hour to complete
The tasks may start at 1:00, 2:00 or 3:00
Requirements:
  T1 must start after T3
  T3 must start before T4 and after T5
  T2 cannot execute at the same time as T1 or T4
  T4 cannot start at 2:00

Variables: one for each task
Domains: \( D_{T1} = D_{T2} = D_{T3} = D_{T3} = \{1:00, 2:00, 3:00\} \)

\[ \begin{array}{c}
\text{T4} \\
1:00 \\
3:00
\end{array} \]
The constraint graph and relations of scheduling problem

Unary constraint
\( D_{T4} = \{1:00, 3:00\} \)

Binary constraints
\[ R_{\{T1,T2\}}: \{(1:00,2:00), (1:00,3:00), (2:00,1:00), (2:00,3:00), (3:00,1:00), (3:00,2:00)\} \]
\[ R_{\{T1,T3\}}: \{(2:00,1:00), (3:00,1:00), (3:00,2:00)\} \]
\[ R_{\{T2,T4\}}: \{(1:00,2:00), (1:00,3:00), (2:00,1:00), (2:00,3:00), (3:00,1:00), (3:00,2:00)\} \]
\[ R_{\{T3,T4\}}: \{(1:00,2:00), (1:00,3:00), (2:00,3:00)) \}
\[ R_{\{T3,T5\}}: \{(2:00,1:00), (3:00,1:00), (3:00,2:00)\} \]
A combinatorial circuit

$M$ a multiplier, $A$ is an adder

- $A = 3$
- $B = 2$
- $C = 2$
- $D = 3$
- $E = 3$
- $M_1$
- $M_2$
- $M_3$
- $X$
- $Y$
- $Z$
- $A_1$
- $A_2$
- $F$
- $G$
- $F = 10$
- $G = 12$
Outline

✓ Motivation, applications, history
✓ CSP: Definition, and simple modeling examples
✓ Mathematical concepts (relations, graphs)
✓ Representing constraints
✓ Constraint graphs
✓ The binary Constraint Networks properties
Outline

- Motivation, applications, history
- CSP: Definition, representation and simple modeling examples
- Mathematical concepts (relations, graphs)
- Representing constraints
- Constraint graphs
- The binary Constraint Networks properties
Binary constraint networks

A graph \( \mathcal{R} \) to be colored by two colors, an equivalent representation \( \mathcal{R}' \) having a newly inferred constraint between \( x_1 \) and \( x_3 \).

Equivalence and deduction with constraints (composition)
Composition of relations (Montanari'74)

**Definition 2.3.2 (composition)** Given two binary or unary constraints $R_{xy}$ and $R_{yz}$, the composition $R_{xy} \cdot R_{yz}$ generates the binary relation $R_{xz}$ defined by:

$$R_{xz} = \{(a, b) | a \in D_x, \ b \in D_z, \exists c \in D_y \text{ s.t. } (a, c) \in R_{xy} \text{ and } (c, b) \in R_{yz}\}$$

An operational definition of composition by Join-project

$$R_{xz} = R_{xy} \cdot R_{yz} = \pi_{\{x,z\}}(R_{xy} \Join R_{yz}).$$

In Figure 2.12a, we deduced that $R'_{13} = \pi_{\{x_1,x_3\}}(R_{12} \Join R_{23}) = \{(red, red), (blue, blue)\}$,
Composition of relations \((\text{Montanari'74})\)

**Bit-matrix operation**: matrix multiplication

**Example 2.3.3** Continuing with our simple graph-coloring example, the two inequality constraints can be expressed as \(2 \times 2\) matrices having zeros along the main diagonal.

\[
R_{12} = \begin{pmatrix} \text{red} & \text{blue} \\
\text{red} & 0 \\
\text{blue} & 1 \end{pmatrix} \quad R_{23} = \begin{pmatrix} \text{red} & \text{blue} \\
\text{red} & 0 \\
\text{blue} & 1 \end{pmatrix}
\]

Multiplying two such matrices yields the following two-dimensional identity matrix:

\[
R_{12} \cdot R_{23} = R_{13} = \begin{pmatrix} 0 & 1 \\
1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\
1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\
0 & 1 \end{pmatrix} = \begin{pmatrix} \text{red} & \text{blue} \\
\text{red} & 1 \\
\text{blue} & 0 \end{pmatrix}
\]

![Graph coloring example](image)
Equivalence, redundancy, composition

• Equivalence: Two constraint networks are equivalent if they have the same set of solutions.

• Composition in matrix notation

\[ R_{xz} = R_{xy} \cdot R_{yz} \]

• Composition in relational operation

\[ R_{xz} = \pi_{xz}(R_{xy} \bowtie R_{yz}) \]
Relations vs networks

- Can we represent by binary constraint networks the relations
  - \( R(x_1, x_2, x_3) = \{(0,0,0)(0,1,1)(1,0,1)(1,1,0)\} \)
  - \( R(x_1, x_2, x_3, x_4) = \{(1,0,0,0)(0,1,0,0)(0,0,1,0)(0,0,0,1)\} \)

- Number of relations: \( 2^{kn} \)
- Number of networks: \( 2^{n^2k^2} \)

- Most relations cannot be represented by binary constraint networks
- Most relations cannot be expressed by binary networks but they can be approximated by them.
The N-queens constraint network

is there a tighter network?

The network has four variables, all with domains \( D_i = \{1, 2, 3, 4\} \).

(a) The labeled chess board. (b) The constraints between variables.

<table>
<thead>
<tr>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[
R_{13} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}
\]

\[
R_{14} = \{(1,2), (1,3), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,2), (4,3)\}
\]

\[
R_{23} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}
\]

\[
R_{24} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}
\]

\[
R_{34} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}
\]

Solutions are: (2,4,1,3) (3,1,4,2)
The projection networks

- The projection network of a relation is obtained by projecting it onto each pair of its variables (yielding a binary network).

- \( \text{Relation} = \{(1,1,2)(1,2,2)(1,2,1)\} \)
  - What is the projection network?

- What is the relationship between a relation and its projection network?

- \( R = \{(1,1,2)(1,2,2)(2,1,3)(2,2,2)\} \)
  - What are the solutions of its projection network?
Theorem: Every relation is included in the set of solutions of its projection network.

Theorem: The projection network is the tightest upper bound binary networks representation of the relation.

Therefore, if a network cannot be represented by its projection network it has no binary network representation.
Partial order between networks, The minimal network

Definition 2.3.10 Given two binary networks, \( R' \) and \( R \), on the same set of variables \( x_1, \ldots, x_n \), \( R' \) is at least as tight as \( R \) iff for every \( i \) and \( j \), \( R'_{ij} \subseteq R_{ij} \).

• An intersection of two networks is tighter (as tight) than both
• An intersection of two equivalent networks is equivalent to both

Definition: The minimal network is the intersection of all equivalent networks

Theorem: The projection network is identical to the minimal network
The N-queens constraint network

The network has four variables, all with domains $D_i = \{1, 2, 3, 4\}$.
(a) The labeled chess board. (b) The constraints between variables.

- $R_{12} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$
- $R_{13} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}$
- $R_{14} = \{(1,2), (1,3), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,2), (4,3)\}$
- $R_{23} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$
- $R_{24} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}$
- $R_{34} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$
The 4-queens constraint network:
(a) The constraint graph. (b) The minimal binary constraints. (c) The minimal unary constraints (the domains).

Solutions are: (2,4,1,3) (3,1,4,2)
Outline

✓ Motivation, applications, history
✓ CSP: Definition, and simple modeling examples
✓ Mathematical concepts (relations, graphs)
✓ Representing constraints
✓ Constraint graphs
✓ The binary Constraint Networks properties
MiniZinc Tutorial

CS 275: Constraint Networks

Annie Raichev Fall 2022
MiniZinc References & More Examples

• The minizinc handbook: https://www.minizinc.org/doc-2.5.5/en/index.html

• Lecture 1: https://www.ics.uci.edu/~dechter/courses/ics-275/fall-2022/