CONSTRAINT Networks
Chapters 1-2

Compisci-275
Fall 2010
Class Information

• Instructor: Rina Dechter
• Days: Tuesday & Thursday
• Time: 11:00 - 12:20 pm
• Class page: http://www.ics.uci.edu/~dechter/ics-275a/fall-2010/
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
Graphical Models

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
Many Examples

Combinatorial Problems

MO Optimization

Optimization

Decision

Graphical Models

Many Examples

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
  – **Unary**: Courses have prerequisites you have/don't have
    Courses/instructors you like/dislike
  – **Binary**: Courses are scheduled at the same time
  – **n-ary**: In CE: 4 courses from 5 tracks such as at least 3 tracks are covered

• **You have choices, but are restricted by constraints**
  – Make the right decisions!!
  – [ICS Graduate program](#)
Student course selection (continued)

• **Given**
  – A set of variables: 4 courses at your college
  – For each variable, a set of choices (values)
  – 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?
  – How two or more solutions differ?
  – Which solution is preferrable?
  – etc.
The field of Constraint Programming

• **How did it started:**
  – 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**
Scene labeling constraint network
Scene labeling constraint network
3-dimensional interpretation of 2-dimensional drawings

Fork: + + - - - - - -

Arrow: + + + + + + + +

Ell: + + + + - - - -

Tee: + + - - - - - -
The field of Constraint Programming

• How did it started:
  – 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
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
Motivation, applications, history

CSP: Definitions and simple modeling examples

Mathematical concepts (relations, graphs)

Representing constraints

Constraint graphs

The binary Constraint Networks properties
Constraint Networks

Example: map coloring

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>green</td>
</tr>
<tr>
<td>red</td>
<td>yellow</td>
</tr>
<tr>
<td>green</td>
<td>red</td>
</tr>
<tr>
<td>green</td>
<td>yellow</td>
</tr>
<tr>
<td>yellow</td>
<td>green</td>
</tr>
<tr>
<td>yellow</td>
<td>red</td>
</tr>
</tbody>
</table>

Constraint graph
Example: map coloring

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

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.
A constraint network is: \( R = (X, D, C) \)

- **X variables**
  \[ X = \{ X_1, \ldots, X_n \} \]

- **D domain**
  \[ D = \{ D_1, \ldots, D_n \}, \quad D_i = \{ v_1, \ldots, v_k \} \]

- **C constraints**
  \[ C = \{ C_1, \ldots, C_t \}, \quad C_i = (S_i, R_i) \]

- **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
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

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\}
Configuration and design

- Want to build: recreation area, apartments, houses, cemetery, dump
  - Recreation area near lake
  - Steep slopes avoided except for recreation area
  - Poor soil avoided for developments
  - Highway far from apartments, houses and recreation
  - Dump not visible from apartments, houses and lake
  - Lots 3 and 4 have poor soil
  - Lots 3, 4, 7, 8 are on steep slop
  - Lots 2, 3, 4 are near lake
  - Lots 1, 2 are near highway
Example: Sudoku

Each row, column and major block must be all different

“Well posed” if it has unique solution: 27 constraints

• Variables: 81 slots
• Domains = \{1,2,3,4,5,6,7,8,9\}
• Constraints: • 27 not-equal

Constraint propagation
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 graphical representation and views of a relation: 
\( R = \{(\text{black, coffee}), (\text{black, tea}), (\text{green, tea})\} \).

<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

\[
\begin{array}{c}
\text{apple juice} \\
\text{coffee} \\
\text{tea}
\end{array}
\]

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

\[
\begin{bmatrix}
\text{black} & 0 & 1 & 1 \\
\text{green} & 0 & 0 & 1
\end{bmatrix}
\]
Operations with relations

- Intersection
- Union
- Difference
- Selection
- Projection
- Join
- Composition
• Local function

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

where

\[ \text{var}(f) = Y \subseteq X: \text{ scope of function } f \]

A: is a set of valuations

• 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>
Example of set operations intersection, union, and difference applied to 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>b</td>
<td>s</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>b</td>
<td>b</td>
<td>c</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>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>
<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>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<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>

(c) $R - R'$

<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>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>a</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>
Selection, Projection, and Join operations 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) $\sigma_{x_3=c}(R')$  

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

(c) $R' \bowtie R''$
- **Join:** \( f \Join g \)

- **Logical AND:** \( f \land g \)

**Local Functions**

**Combination**

Join:

\[
\begin{array}{c|c|c|}
  x_1 & x_2 & \text{Join} \\
  \hline
  a & a & a \\
  a & a & a \\
  b & b & b \\
  b & b & b \\
\end{array}
\]
\[
\begin{array}{c|c|c|}
  x_2 & x_3 & \text{Join} \\
  \hline
  a & a & a \\
  a & a & a \\
  a & b & b \\
  b & a & a \\
\end{array}
\]

**Logical AND:**

\[
\begin{array}{c|c|c|c|}
  x_1 & x_2 & x_3 & f \land g \\
  \hline
  a & a & a & \text{true} \\
  a & a & b & \text{false} \\
  a & b & a & \text{false} \\
  a & b & b & \text{false} \\
  b & a & a & \text{false} \\
  b & a & b & \text{false} \\
  b & b & a & \text{true} \\
  b & b & b & \text{true} \\
\end{array}
\]

Fall 2010
Global View of the Problem

The problem has a solution if the global view is not empty.

Does the problem a solution?

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

The problem has a solution if there is some true tuple in the global view, the universal relation.

Global View = universal relation

Fall 2010
**Global View of the Problem**

![Table and Diagram]

What about counting?

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

Number of true tuples

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>b</td>
<td>0</td>
</tr>
</tbody>
</table>

Sum over all the tuples

**Logical AND?**
- true is 1
- false is 0

Fall 2010
Outline

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

• If a CSP $M = \langle X, D, C \rangle$ represents a 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 modeling: choose a set of variables and values that allows the constraints to be expressed easily and concisely
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></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

If this constraint network has a solution, then the propositional theory has a model.
Constraint’s representations

- **Relation:** allowed tuples
  
<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<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:**
  
  \( (a \lor b) \rightarrow \neg c \)

- **Semantics:** by a relation
A (primal) constraint graph: a node per variable, arcs connect constrained variables.

A dual constraint graph: a node per constraint’s scope, an arc connect nodes sharing variables = hypergraph.
Graph Concepts Reviews:
Hyper Graphs and Dual Graphs

- A hypergraph
- Dual graphs
- Primal graphs
- Factor graphs
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 of 3 instances of the Radio frequency assignment problem in CELAR’s benchmark
Scene labeling constraint network

\[ R_{21} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad R_{31} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad R_{51} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ R_{24} = R_{37} = R_{56} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \]

\[ R_{26} = R_{34} = R_{57} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \]

Fork: \[ \begin{array}{cccc} + & + & - & - \\ + & - & + & - \end{array} \]

Arrow: \[ \begin{array}{cccc} + & + & - & - & + \\ - & + & + & + \end{array} \]

Ell: \[ \begin{array}{cccc} + & - & + & - \\ + & - & + & - \end{array} \]

Tee: \[ \begin{array}{cccc} + & - & + & - \end{array} \]
Figure 1.5: Solutions: (a) stuck on left wall, (b) stuck on right wall, (c) suspended in mid-air, (d) resting on floor.
Examples

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

Encoding?

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

\[
\begin{align*}
T4 & \\
1:00 & \\
3:00 & 
\end{align*}
\]
The constraint graph and relations of scheduling problem

Unary constraint
\[ D_{T4} = \{1:00, 3:00\} \]

Binary constraints
\[ R_{\{T1,T2\}}: \{(1:00,2:00), (1:00,3:00), (2:00,1:00), (2:00,3:00), (3:00,1:00), (3:00,2:00)\} \]
\[ R_{\{T1,T3\}}: \{(2:00,1:00), (3:00,1:00), (3:00,2:00)\} \]
\[ R_{\{T2,T4\}}: \{(1:00,2:00), (1:00,3:00), (2:00,1:00), (2:00,3:00), (3:00,1:00), (3:00,2:00)\} \]
\[ R_{\{T3,T4\}}: \{(1:00,2:00), (1:00,3:00), (2:00,3:00)\} \]
\[ R_{\{T3,T5\}}: \{(2:00,1:00), (3:00,1:00), (3:00,2:00)\} \]
Can we specify numeric constraints as relations?
More examples

- Given \( P = (V, D, C) \), where

\[
V = \{ V_1, V_2, \ldots, V_n \}
\]

\[
D = \{ D_{V_1}, D_{V_2}, \ldots, D_{V_n} \}
\]

\[
C = \{ C_1, C_2, \ldots, C_l \}
\]

Example I:

- Define \( C \) ?

\[
V_1 \rightarrow V_2: v_1 < v_2
\]

\[
V_1 + v_3 < 9
\]

\[
V_2 < v_3
\]

\[
v_2 > v_4
\]

\[
V_3 \rightarrow V_4: v_2 < v_3
\]

\[
V_3\rightarrow V_4: v_1 < v_2
\]
Example: temporal reasoning

• Give one solution: .......
• Satisfaction, yes/no: decision problem
Outline

✓ Motivation, applications, history
✓ CSP: Definition, representation and simple modeling examples
✓ Mathematical concepts (relations, graphs)
✓ Representing constraints
✓ Constraint graphs
✓ The binary Constraint Networks properties
Properties of 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 \((\text{Montanari'74})\)

**Input:** two binary relations \(R_{ab}\) and \(R_{bc}\) with 1 variable in common.

**Output:** a new induced relation \(R_{ac}\) (to be combined by intersection to a pre-existing relation between them, if any).

**Bit-matrix operation:** matrix multiplication

\[
R_{ac} = R_{ab} \cdot R_{bc}
\]

\[
R_{ab} = \begin{pmatrix} 1 & 0 & 1 \\
1 & 0 & 0 \end{pmatrix}, \quad R_{bc} = \begin{pmatrix} 1 & 0 \\
1 & 1 \end{pmatrix}, \quad R_{ac} = ?
\]
Equivalence, Redundancy, Composition

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

• Composition in matrix notation

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

• Composition in relational operation

\[ R_{xz} = \pi_{xz} (R_{xy} \otimes R_{yz}) \]
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 \( 2^{(k^n)} \)
• Number of networks: \( 2^{((k^2)(n^2))} \)
• Most relations cannot be represented by binary networks
The minimal and 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?
- \( \{(1,1,2)(1,2,2)(2,1,3)(2,2,2)\} \), solve its projection network?
Projection network (continued)

• **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, $\mathcal{R}'$ and $\mathcal{R}$, on the same set of variables $x_1, ..., x_n$, $\mathcal{R}'$ is at least as tight as $\mathcal{R}$ 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 2.3.14 Let $\{\mathcal{R}_1, ..., \mathcal{R}_l\}$ be the set of all networks equivalent to $\mathcal{R}_0$ and let $\rho = \text{sol}(\mathcal{R}_0)$. Then the minimal network $M$ of $\mathcal{R}_0$ is defined by $M(\mathcal{R}_0) = \bigcap_{i=1}^l \mathcal{R}_i$.

Theorem 2.3.15 For every binary network $\mathcal{R}$ s.t. $\rho = \text{sol}(\mathcal{R})$, $M(\rho) = P(\rho)$.  

Fall 2010
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.

\begin{align*}
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)\}
\end{align*}
The 4-queens constraint network:
(a) The constraint graph. (b) The minimal binary constraints. 
(c) The minimal unary constraints (the domains).

\[ M_{12} = \{(2,4), (3,1)\} \]
\[ M_{13} = \{(2,1), (3,4)\} \]
\[ M_{14} = \{(2,3), (3,2)\} \]
\[ M_{23} = \{(1,4), (4,1)\} \]
\[ M_{24} = \{(1,2), (4,3)\} \]
\[ M_{34} = \{(1,3), (4,2)\} \]

\[ D_1 = \{1,3\} \]
\[ D_2 = \{1,4\} \]
\[ D_3 = \{1,4\} \]
\[ D_4 = \{1,3\} \]

Solutions are: \((2,4,1,3)\) \((3,1,4,2)\)
The Minimal vs Binary decomposable networks

• The minimal network is perfectly explicit for binary and unary constraints:
  – Every pair of values permitted by the minimal constraint is in a solution.

• Binary-decomposable networks:
  – A network whose all projections are binary decomposable
  – Ex: \((x,y,x,t) = \{(a,a,a,a)(a,b,b,b)(b,b,a,c)\}\):
    is binary representeble? and what about its projection on \(x,y,z\)?

  – Proposition: The minimal network represents fully binary-decomposable networks.