CSCI-564
CONSTRAINT PROCESSING AND HEURISTIC SEARCH

LECTURE 21 – CONSTRAINT PROCESSING (CONT'D)

Dr. Jean-Alexis Delamer

Recap

- Constraint satisfaction problems (CSP) incudes two important components:
  - Variable and domains.
  - Constraints.

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

Mathematical background

- Set of variables $X = \{x_1, \dots, x_k\}$.
- The associated domains $D = \{D_1, \dots, D_k\}$
- A relation R on X is any subset of the Cartesian product of D.
- The set of variables on which a relation is defined is called the scope of the relation denoted $scope(R)$.

Mathematical background

- Example.

$$x_1 = \text{color} \qquad D_1 = \{\text{black}, \text{green}\}$$
$$x_2 = \text{drink} \qquad D_2 = \{\text{coffee}, \text{tea}\}$$

$$R = \{(\text{black, coffee}), (\text{black,tea}), (\text{green,tea})\}$$

The scope of this relation is $\{x_1, x_2\}$

Mathematical background

- How can we represent relations?
  - Explicitly.
  - Implicitly.

$x_1 = \text{color}$ $\quad D_1 = \{\text{black}, \text{green}\}$

$x_2 = \text{drink}$ $\quad D_2 = \{\text{apple juice, coffee, tea}\}$

$R = \{(\text{black, coffee}), (\text{black,tea}), (\text{green,tea})\}$

$\{(x_1, x_2) | x_1 \in D_1, x_2 \in D_2,$
and $x_1$is before $x_2$in dictionary ordring$\}$

=

$\{(x_1, x_2) | x_1 \in D_1, x_2 \in D_2, x_1 \leq x_2\},$
where $\leq$ is lexicographic ordering

Mathematical background

- There are two additional ways to explicitly express a relation.

x_1	x_2
black	coffee
black	tea
green	tea

(a) table

$$
\begin{array}{cc}
 & \underline{x_2} \\
 & \begin{array}{ccc} \text{apple juice} & \text{coffee} & \text{tea} \end{array} \\
\underline{x_1} \quad \begin{array}{c} \text{black} \\ \text{green} \end{array} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}
\end{array}
$$

(b) (0,1)-matrix

Mathematical background

- In constraint satisfaction problem we can have more than one relation.

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s

(a) Relation R

x_1	x_2	x_3
b	b	c
c	b	c
c	n	n

(b) Relation R'

x_2	x_3	x_4
a	a	1
b	c	2
b	c	3

(c) Relation R''

What are the operations available?

Operations on Relations

- The usual operations on sets:
  - Intersection
  - Union
  - Difference

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s

(a) Relation R

x_1	x_2	x_3
b	b	c
c	b	c
c	n	n

(b) Relation R'

x_2	x_3	x_4
a	a	1
b	c	2
b	c	3

(c) Relation R''

$R \cap R'$

x_1	x_2	x_3
b	b	c
c	b	c

$R \cup R'$

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s
c	n	n

x_1	x_2	x_3
a	b	c
c	b	s

$R - R'$

Operations on Relations

- Operations specific to relations:
  - Selection:
    - Select a subset of R with specified values on specified variables.
  - Projection:
    - A new relation with the tuples of R with certain components removed.
  - Join:
    - Join two relations R_S and R_T and combine all their common variables in S and T.

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s

(a) Relation R

x_1	x_2	x_3
b	b	c
c	b	c
c	n	n

(b) Relation R'

x_2	x_3	x_4
a	a	1
b	c	2
b	c	3

(c) Relation R''

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

x_1	x_2	x_3
b	b	c
c	b	c

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

x_2	x_3
b	c
n	n

$R' \bowtie R''$

x_1	x_2	x_3	x_4
b	b	c	2
b	b	c	3
c	b	c	2
c	b	c	3

Can you think of something similar in another field of computer science?

Operations on Relations

- How Join works?
  - Join:

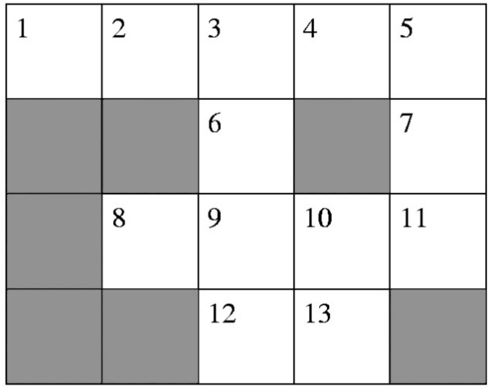
$$\begin{array}{cc} x_1 & x_2 \\ \hline a & a \\ b & b \end{array} \bowtie \begin{array}{cc} x_2 & x_3 \\ \hline a & a \\ a & b \\ b & a \end{array} = \begin{array}{ccc} x_1 & x_2 & x_3 \\ \hline a & a & a \\ a & a & b \\ b & b & a \end{array}$$

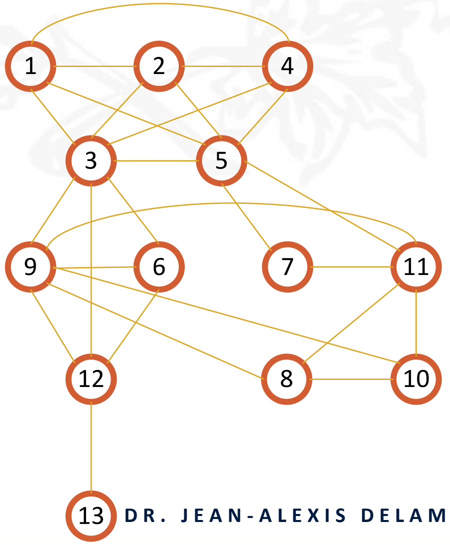
  - Logical AND: $f \wedge g$

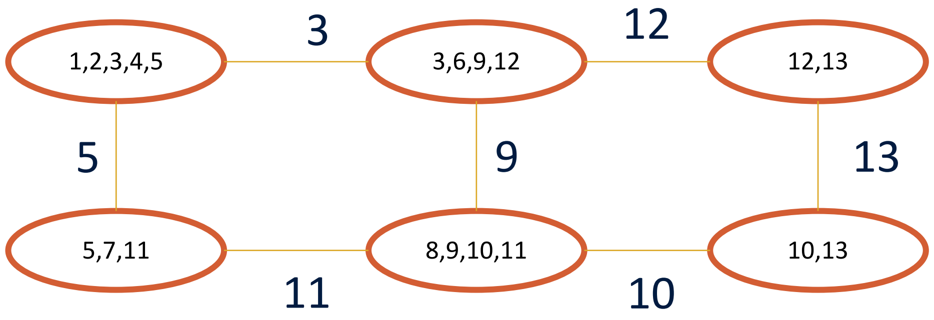
$$\begin{array}{cc|c} x_1 & x_2 & f \\ \hline a & a & \text{true} \\ a & b & \text{false} \\ b & a & \text{false} \\ b & b & \text{true} \end{array} \wedge \begin{array}{cc|c} x_2 & x_3 & g \\ \hline a & a & \text{true} \\ a & b & \text{true} \\ b & a & \text{true} \\ b & b & \text{false} \end{array} = \begin{array}{ccc|c} x_1 & x_2 & x_3 & h \\ \hline a & a & a & \text{true} \\ a & a & b & \text{true} \\ a & b & a & \text{false} \\ a & b & b & \text{false} \\ \mathrm{b} & a & a & \text{false} \\ \mathrm{b} & a & b & \text{false} \\ b & b & a & \text{true} \\ \mathrm{b} & b & b & \text{false} \end{array}$$

Constraint graphs

- A constraint network can be represented by graphs.
  - (Primal) constraint graph:
    - A node per variable.
    - Arcs connect constrained variables.
  - Dual constraint graph (Hypergraph):
    - A node per constraint's scope
    - Arcs connect nodes sharing variables







$R\{1,2,3,4,5\}$= {(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)}

Exercise

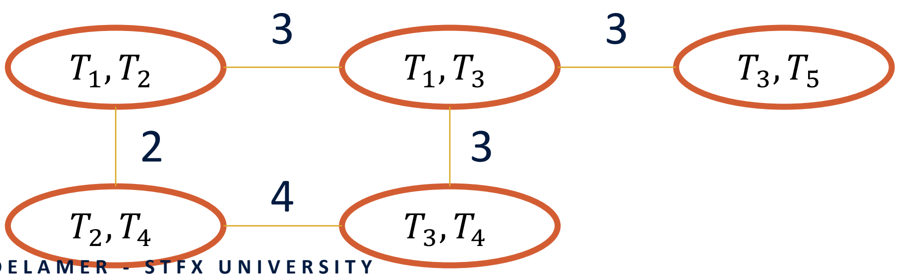
- Scheduling problem.
  - 5 tasks to complete: $T_1, \dots, T_5$.
  - Each task requires 1 hour to complete.
  - Tasks can start at 1:00, 2:00 or 3:00
  - Tasks can be executed simultaneously, but
    - T_1 must start after T_3.
    - T_3 must start before T_4 and after T_5.
    - T_2 cannot execute at the same time as T_1 or T_4.
    - T_4 cannot start at 2:00.
- Write the constraint relations and draw the constraint graphs (Primal and dual).
  - $X = \{T_1, \dots, T_5\}$
  - $D_i = \{1{:}00, 2{:}00, 3{:}00\}$

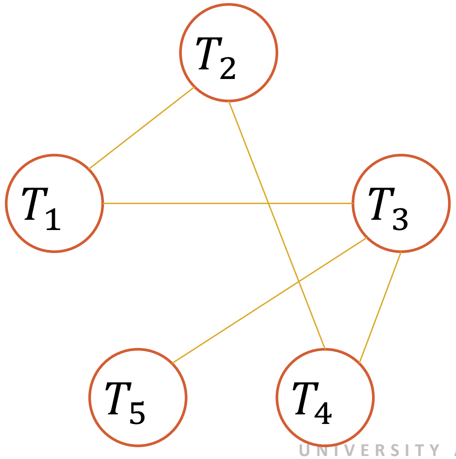
Exercise

- Unary constraint:
  - $D_{T_4} = \{1:00, 3:00\}$
- Binary constraints:
  - $R_{\{T_1,T_2\}}$: {(1:00, 2:00), (1:00, 3:00), (2:00, 1:00), (2:00, 3:00), (3:00,1:00), (3:00,2:00)}
  - $R_{\{T_1,T_3\}}$: {(2:00, 1:00), (3:00,1:00), (3:00,2:00)}
  - $R_{\{T_2,T_4\}}$: {(1:00, 2:00), (1:00, 3:00), (2:00, 1:00), (2:00, 3:00), (3:00,1:00), (3:00,2:00)}
  - $R_{\{T_3,T_4\}}$: {(1:00, 2:00), (1:00, 3:00), (2:00, 3:00)}
  - $R_{\{T_3,T_5\}}$: {(2:00, 1:00), (3:00,1:00), (3:00,2:00)}

Summary:

- $T_1 > T_3$
- $T_3 < T_4, T_5$
- $T_2 \neq T_1, T_4$
- $T_4 \neq 2:00$





Binary Constraint Networks

- Many processing concepts were initially introduced for binary networks.
  - Minimal networks.
  - Decomposability.
  - Etc.

- These concepts can be extended to the other constraint problems.

Constraint Inference

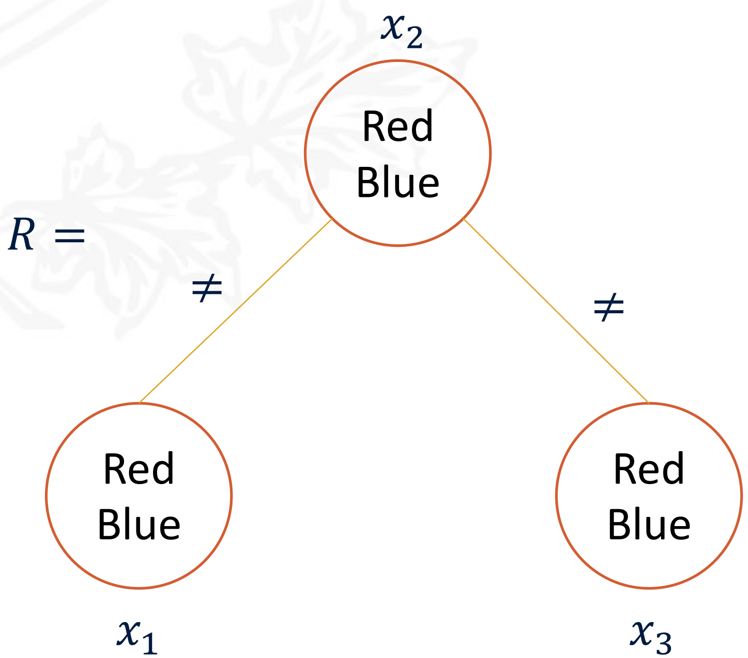
- Constraint deduction or constraint inference is a central concept.
  - New constraints can be inferred from an initial set of constraints.

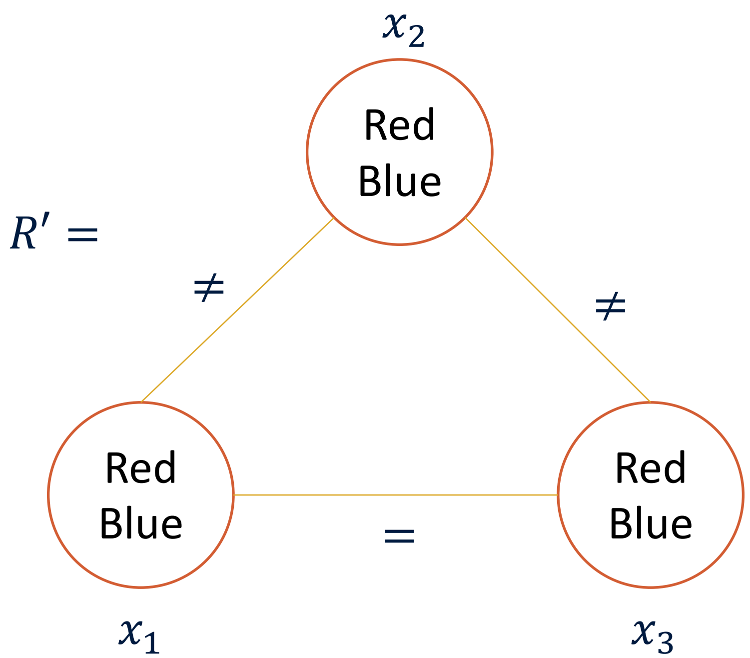
- The new constraints might:
  - Create constraints between variables that were not initially constrained.
  - Tighten existing constraints.

$$x \geq y, \quad y \geq z \quad \longrightarrow \quad x \geq z$$

Constraint Inference

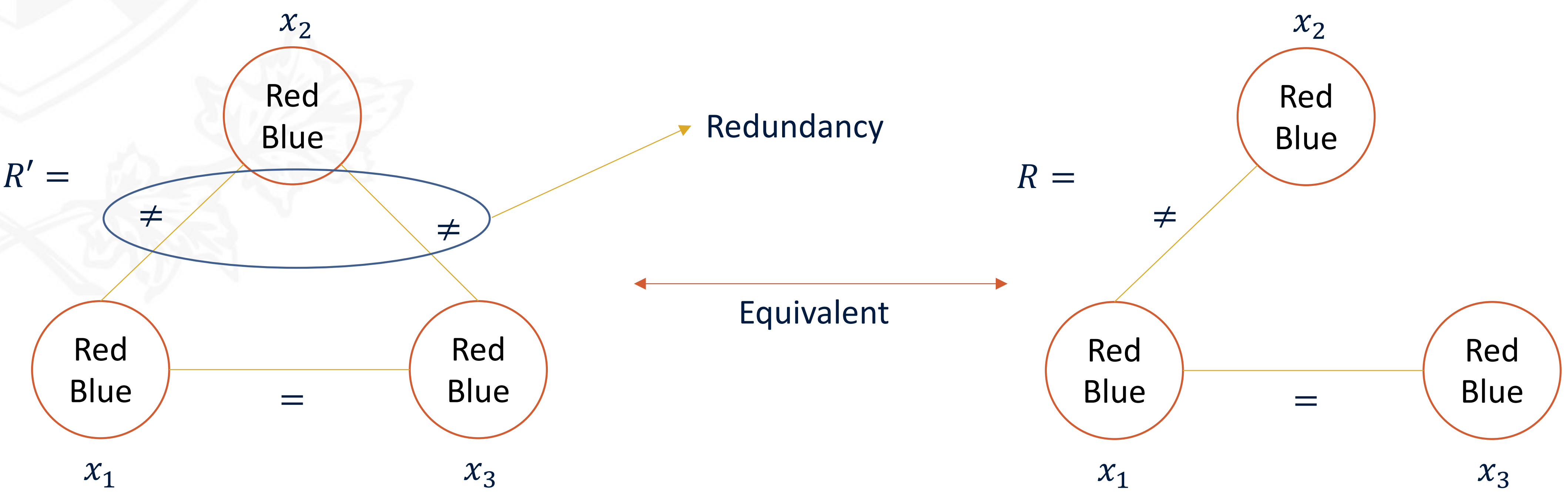
- A graph R to be colored by two colors.
- What can you deduce?
  - A newly inferred constraint between x_1 and x_3.





Constraint Inference

- A constraint R_{ij} is redundant relative to R' iff R' is equivalent to R when removed.



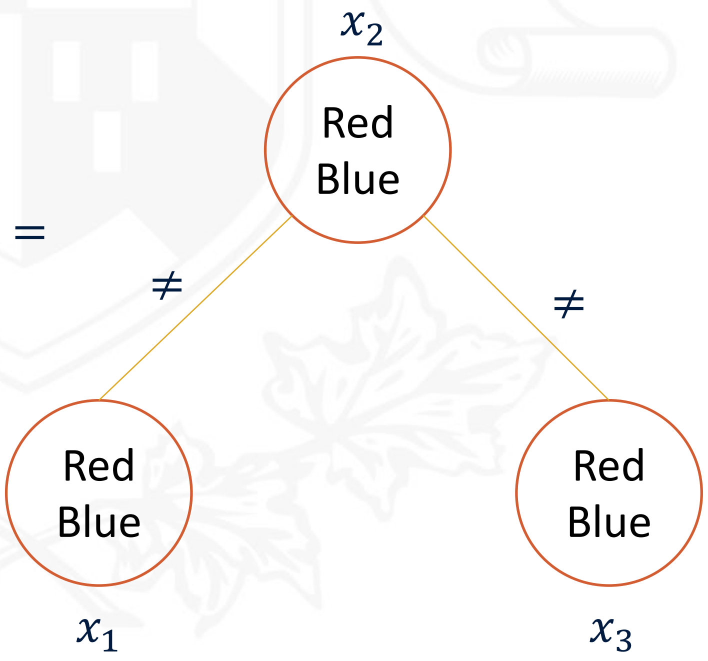
Constraint Inference

- Constraint deduction can be accomplished through the composition operation.

- 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
  - $R_{xz} = \{(a, b) | a \in D_x, b \in D_z, \exists c \in D_y$ such that $(a, c) \in R_{xy}$ and $(c, b) \in R_{yz}\}$

Constraint Inference

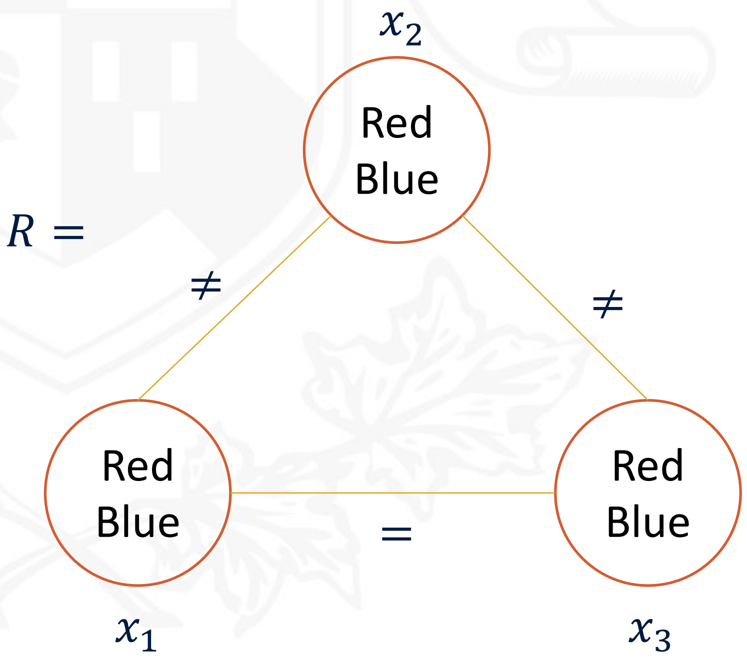
$R =$



$$R_{12} = \begin{array}{cc} \text{red} & \text{blue} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{array}{l} \text{red} \\ \text{blue} \end{array} \end{array} \qquad R_{23} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$R_{13} = R_{12} \cdot R_{23} = ?$$

Constraint Inference



$$R_{12} = \begin{array}{cc} \text{red} & \text{blue} \\ \multicolumn{2}{c}{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}} \end{array} \begin{array}{l} \text{red} \\ \text{blue} \end{array} \qquad R_{23} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$R_{13} = R_{12} \cdot R_{23} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Constraint Inference

- Another method is to use the join operator.

- 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:
  - $R(x_1, x_2, x_3) = \{(0,0,0)(0,1,1)(1,0,1)(1,1,0)\}$
  - $R(x_1, x_2, x_3, x_4) = \{(1,0,0,0)(0,1,0,0)(0,0,1,0)(0,0,0,1)\}$

- Given n variables each having a domain of size k.
  - Maximum number of relations: 2^{k^n}.
  - Maximum number of networks: $2^{n^2k^2}$.

- Most relations cannot be represented by binary constraint networks.

Projection Network

- A relation that cannot be expressed by a binary network may still be approximated by one.
- An approximation is the binary projection network.
  - The projection of a relation R is obtained by projecting ρ onto each pair of its variables.
  - Formally, if R is a relation over $X = \{x_1, \dots, x_n\}$, its projection network, $P(R)$ is defined by the network $\rho = (X, D, P)$, where $D = \{D_i\}, D_i = \pi_i(R), P = \{P_{ij}\}$, and $P_{ij} = \pi_{x_i,x_j}(R)$.

Projection Network

- What is the projection network of R_{123}?

$$R_{123} = \begin{array}{c|c|c} x_1 & x_2 & x_3 \\ \hline 1 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 2 & 1 \end{array}$$

$$R_{12} = \begin{array}{c|c} x_1 & x_2 \\ \hline 1 & 1 \\ 1 & 2 \end{array} \qquad R_{13} = \begin{array}{c|c} x_1 & x_3 \\ \hline 1 & 2 \\ 1 & 1 \end{array} \qquad R_{23} = \begin{array}{c|c} x_2 & x_3 \\ \hline 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{array}$$

$P(R_{123})$

=

Generating all the solution yields $sol(P(R_{123})) = \boxed{\{(1,1,2),(1,2,2),(1,2,1)\}}$

Projection Network

- What is the projection network of R_{123}?

$$R_{123} = \begin{array}{c|c|c} x_1 & x_2 & x_3 \\ \hline 1 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 2 \end{array}$$

$$R_{12} = \begin{array}{c|c} x_1 & x_2 \\ \hline 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{array} \qquad R_{13} = \begin{array}{c|c} x_1 & x_3 \\ \hline 1 & 2 \\ 2 & 3 \\ 2 & 2 \end{array} \qquad R_{23} = \begin{array}{c|c} x_2 & x_3 \\ \hline 1 & 2 \\ 2 & 2 \\ 1 & 3 \end{array}$$

$P(R_{123})$

$\neq$

Generating all the solution yields $sol\big(P(R_{123})\big) = \{(1,1,2), (1,2,2), \boxed{(2,1,2)}, (2,1,3), (2,2,2)\}$

Projection Network

- Theorem:
  - Every relation is included in the set of solutions of its projection network.
  - For every relation $R, R \subseteq sol(P(R))$.

- Proof:
  - Let $t \in R$. We must show only that $t \in sol(P(R))$, namely that it satisfies every binary constraint in $P(R)$.
  - This is clearly true, since, by its definition, every pair of values of t was included, by projection, in the corresponding constraint of $P(R)$.

Projection Network

- Theorem:
  - The projection network $P(R)$ is the tightest upper bound binary network representation of R; there is no binary network ρ', such that $R \subseteq sol(\rho') \subset sol(P(\rho))$

- Therefore, if a network cannot be represented by its projection network it has no binary network representation.

Minimal Network

- Minimal network:
  - Let $\{\rho_1, \ldots, \rho_l\}$ be the set of all networks equivalent to ρ_0 and let $R = sol(\rho_0)$.
  - Then the minimal network M of ρ_0 or of R is defined by $M(\rho_0) = M(R) = \bigcap_{i=1}^{l} \rho_i$

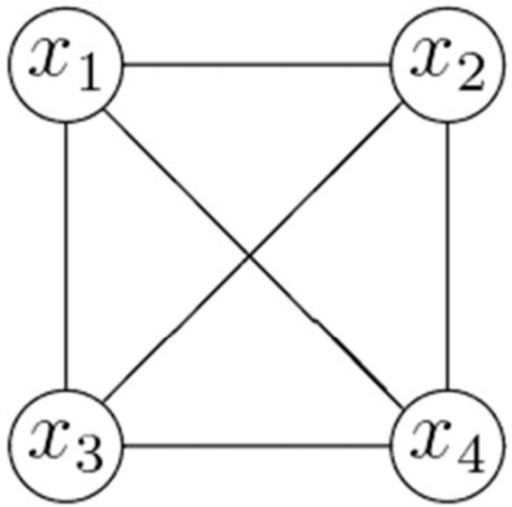
	x_1	x_2	x_3	x_4
1				
2				
3				
4				

$$
\begin{aligned}
R_{12} &= \{(1,3),\ (1,4),\ (2,4),\ (3,1),\ (4,1),\ (4,2)\} \\
R_{13} &= \{(1,2),\ (1,4),\ (2,1),\ (2,3),\ (3,2),\ (3,4),\ (4,1),\ (4,3)\} \\
R_{14} &= \{(1,2),\ (1,3),\ (2,1),\ (2,3),\ (2,4),\ (3,1),\ (3,2),\ (3,4) \\
&\qquad (4,2),\ (4,3)\} \\
R_{23} &= \{(1,3),\ (1,4),\ (2,4),\ (3,1),\ (4,1),\ (4,2)\} \\
R_{24} &= \{(1,2),\ (1,4),\ (2,1),\ (2,3),\ (3,2),\ (3,4),\ (4,1),\ (4,3)\} \\
R_{34} &= \{(1,3),\ (1,4),\ (2,4),\ (3,1),\ (4,1),\ (4,2)\}
\end{aligned}
$$

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

Minimal Network

$D_1 = \{2,3\}$
$D_2 = \{1,4\}$
$D_3 = \{1,4\}$
$D_4 = \{2,3\}$



$$
\begin{aligned}
M_{12} &= \{(2,4),\ (3,1)\} \\
M_{13} &= \{(2,1),\ (3,4)\} \\
M_{14} &= \{(2,3),\ (3,2)\} \\
M_{23} &= \{(1,4),\ (4,1)\} \\
M_{24} &= \{(1,2),\ (4,3)\} \\
M_{34} &= \{(1,3),\ (4,2)\}
\end{aligned}
$$

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