CompSci 275, Constraint Networks

Rina Dechter, Fall 2020

Introduction, the constraint network model
Chapters 1-2

Class information

- Instructor: Rina Dechter
- Lectures: Monday & Wednesday
- Time: 3:30 - 4:50 pm

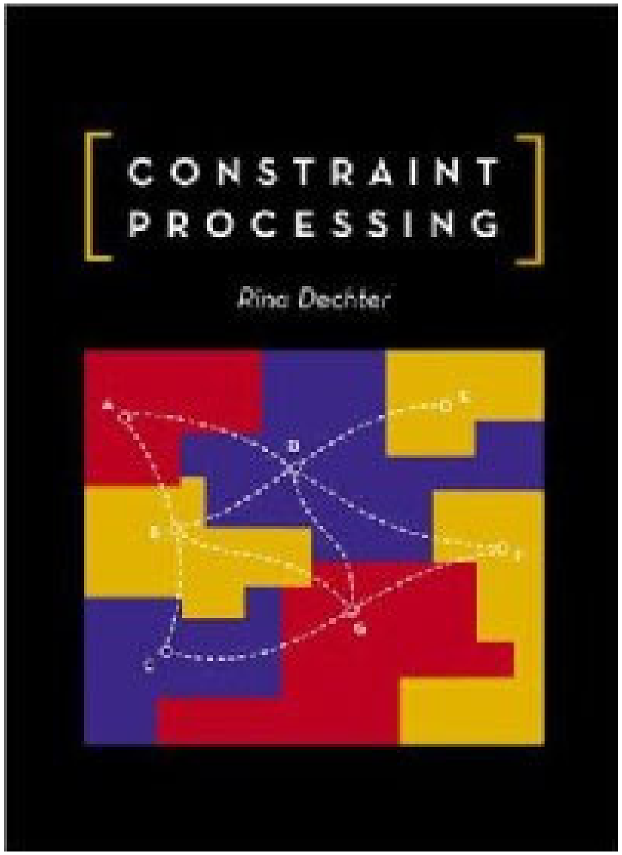
- Class page: https://www.ics.uci.edu/~dechter/courses/ics-275/fall-2020/

Text book (required)

Rina Dechter,

[Constraint Processing](#),

Morgan Kaufmann



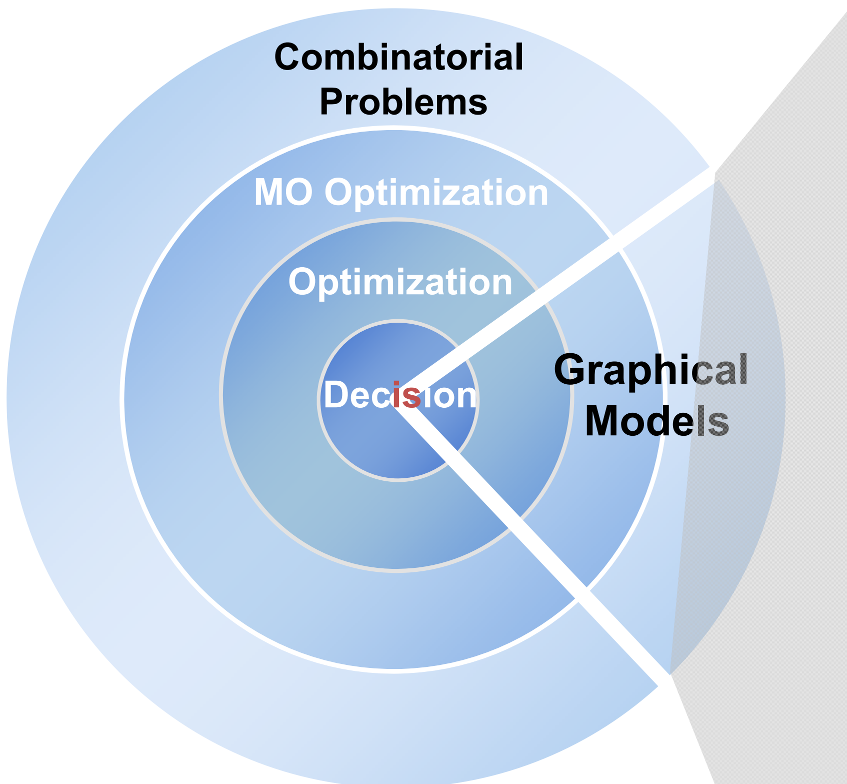
Outline

- ✓ 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



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

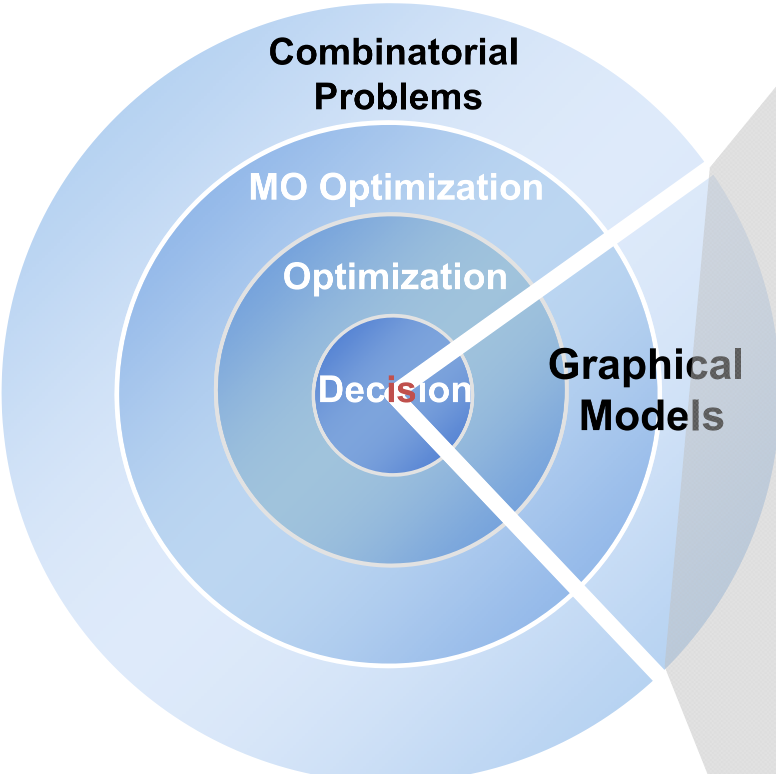
A set of local functions

Main advantage:

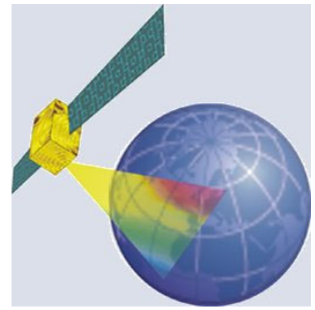
They provide unifying algorithms:

- Search
- Complete Inference
- Incomplete Inference

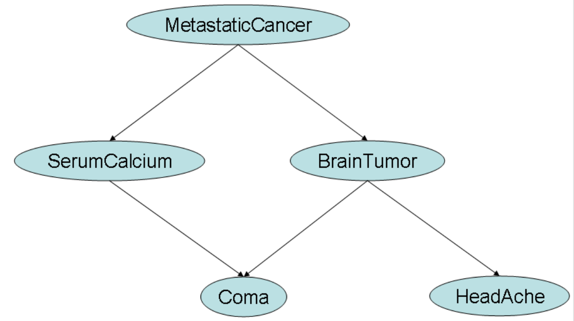
Combinatorial problems



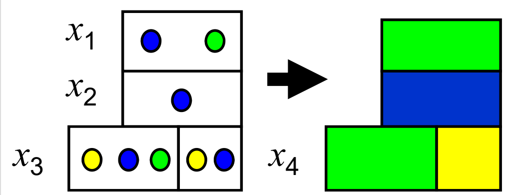
Many Examples



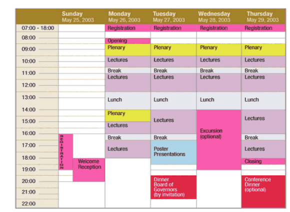
EOS Scheduling



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
  - Courses have prerequisites 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]

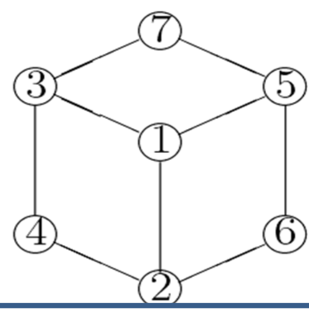
Student course selection (continued)

- **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.

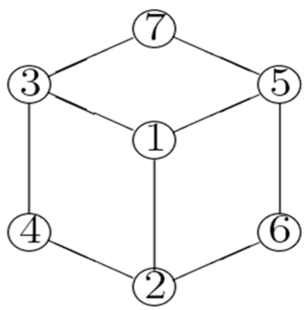
The field of constraint programming

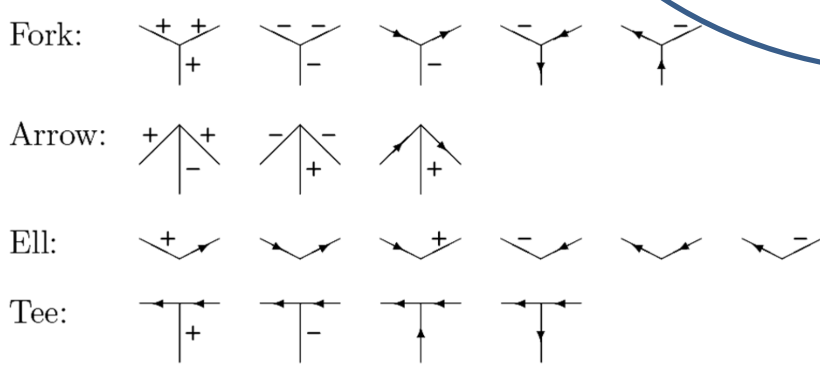
- **How did it start:**
  - Artificial Intelligence (vision)
  - Programming Languages (Logic Programming),
  - Databases (deductive, relational)
  - Logic-based languages (propositional logic)
  - SATisfiability
- **Related areas:**
  - Hardware and software verification
  - Operation Research (Integer Programming)
  - Answer set programming
- **Graphical Models; deterministic**

Scene labeling constraint network

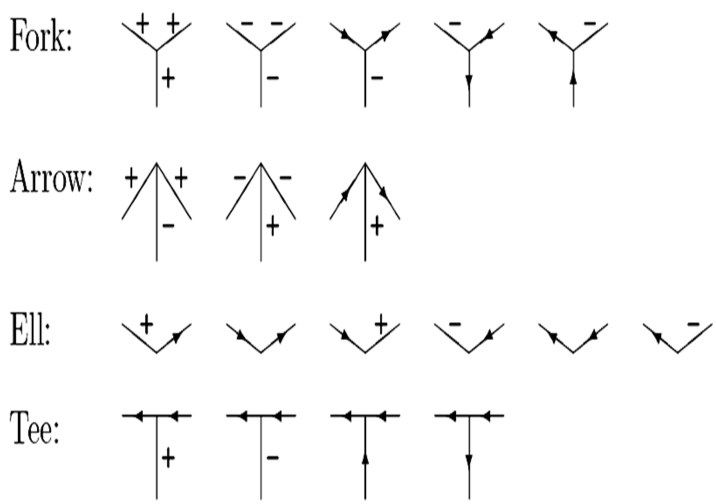


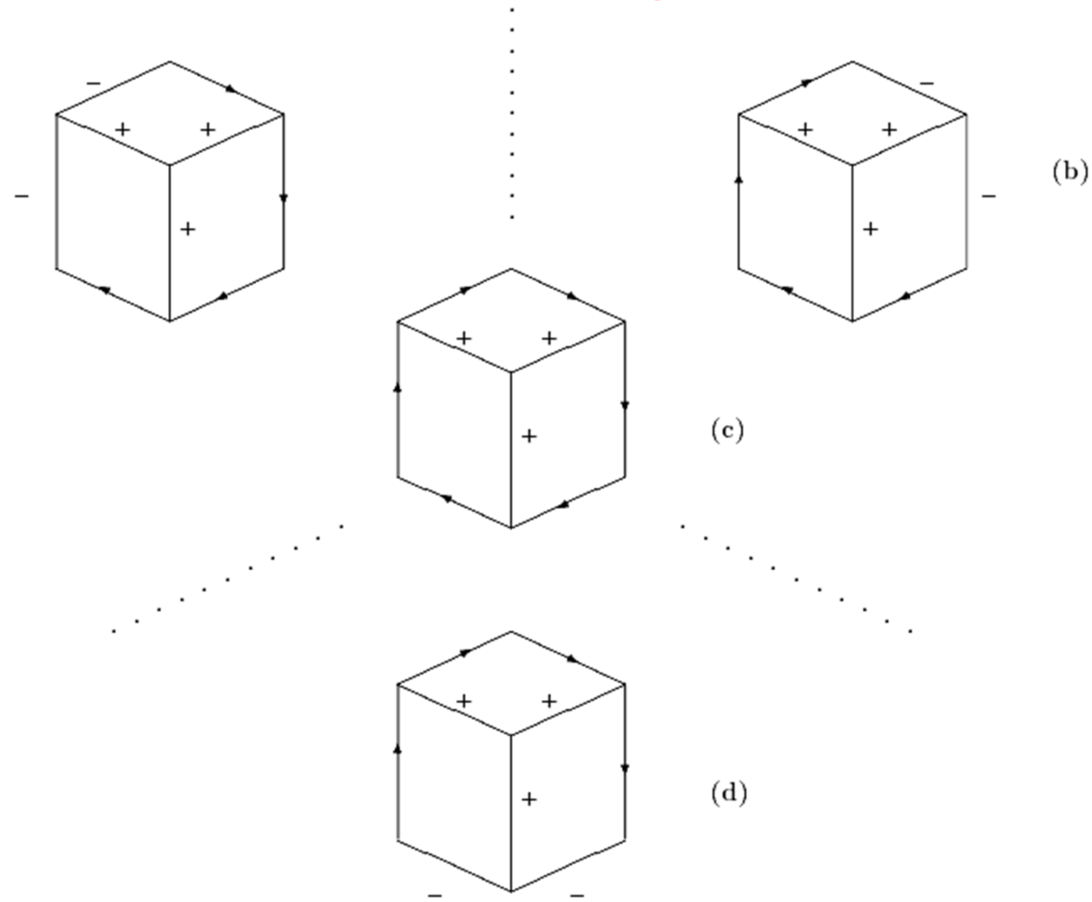
Scene labeling constraint network





3-dimentional interpretation of 2-dimentional drawings



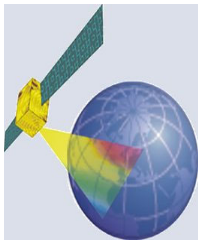


The field of constraint programming

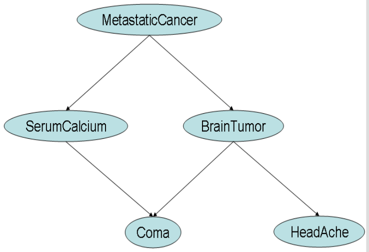
- **How did it start:**
  - Artificial Intelligence (vision)
  - Programming Languages (Logic Programming),
  - Databases (deductive, relational)
  - Logic-based languages (propositional logic)
  - SATisfiability
- **Related areas:**
  - Hardware and software verification
  - Operations Research (Integer Programming)
  - Answer set programming
- **Graphical Models; deterministic**

Applications

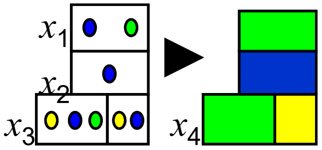
- Radio resource management (RRM)
- Databases (computing joins, view updates)
- Temporal and spatial reasoning
- Planning, scheduling, resource allocation
- Design and configuration
- Graphics, visualization, interfaces
- Hardware verification and software engineering
- HC Interaction and decision support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



EOS Scheduling



Bayesian Networks



Graph Coloring

Timetabling

… and many others.

Outline

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

Constraint networks

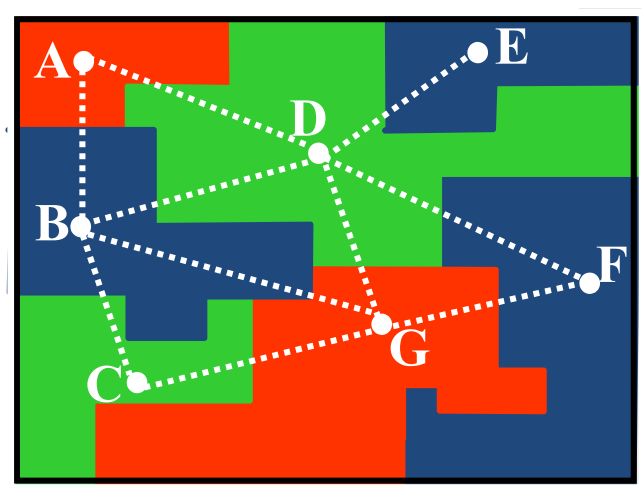
Example: map coloring

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

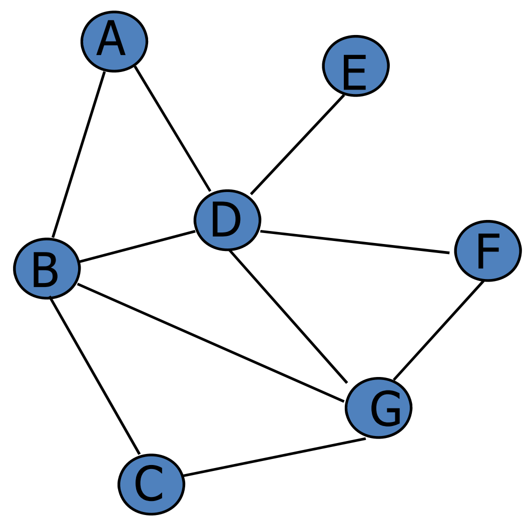
Values - colors (red, green, blue)

Constraints: **A $\neq$ B, A $\neq$ D, D $\neq$ E,** *etc.*

A	B
red	green
red	yellow
green	red
green	yellow
yellow	green
yellow	red



Constraint graph



Constraint satisfaction tasks

Example: map coloring

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

Values - colors (e.g., red, green, yellow)

Constraints:

$$A \neq B,\ A \neq D,\ D \neq E,\ etc.$$

A	B	C	D	E…
red	green	red	green	blue
red	blue	green	green	blue
…	…	…	…	green
…	…	…	…	red
red	blue	red	green	red

Are the constraints consistent?

Find a solution, find all solutions

Count all solutions

Find a good solution

Information as constraints

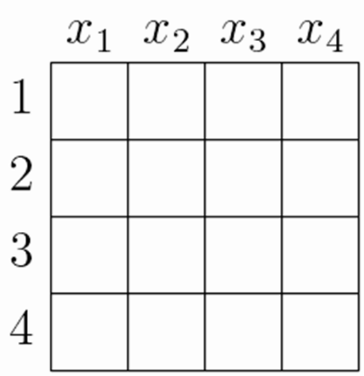
- I have to finish my class in 50 minutes
- 180 degrees in a triangle
- Memory in our computer is limited
- The four nucleotides that makes up a DNA only combine in a particular sequence
- Sentences in English must obey the rules of syntax
- Susan cannot be married to both John and Bill
- Alexander the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

Constraint network; definition

- A constraint network is: **R=*(X,D,C)***
  - **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** $$C = \{C_1,\ldots C_t\},,, C_i = (S_i, R_i)$$
  - ***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.



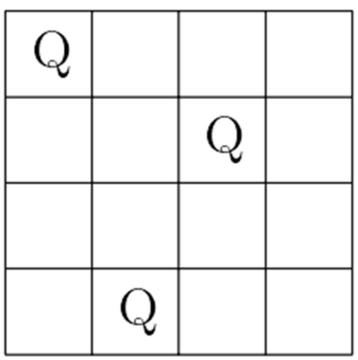
(a)

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

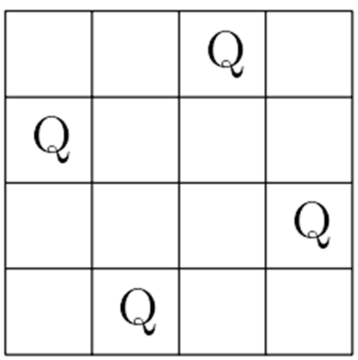
(b)

A solution and a partial consistent tuple (configuration)

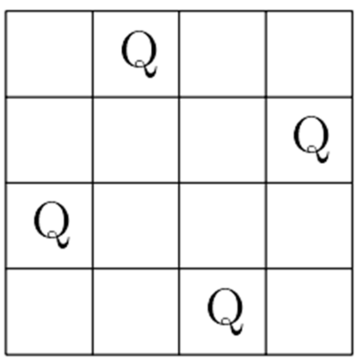
Not all consistent instantiations are part of a solution:
(a) A consistent instantiation that is not part of a solution.
(b) The placement of the queens corresponding to the solution (2, 4, 1,3).
(c) The placement of the queens corresponding to the solution (3, 1, 4, 2).



(a)



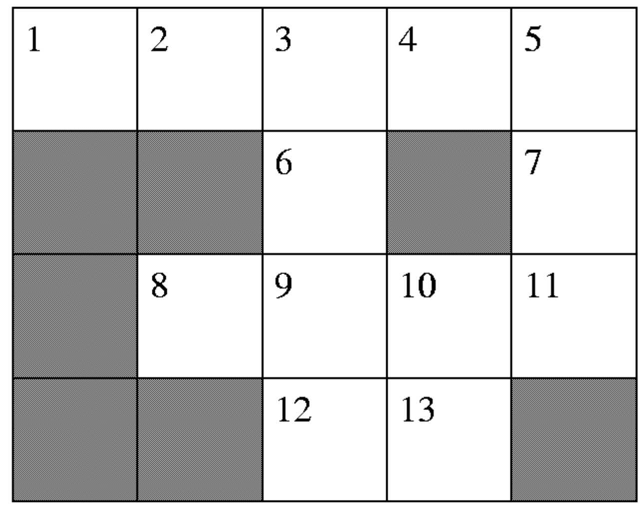
(b)



(c)

Example: crossword puzzle

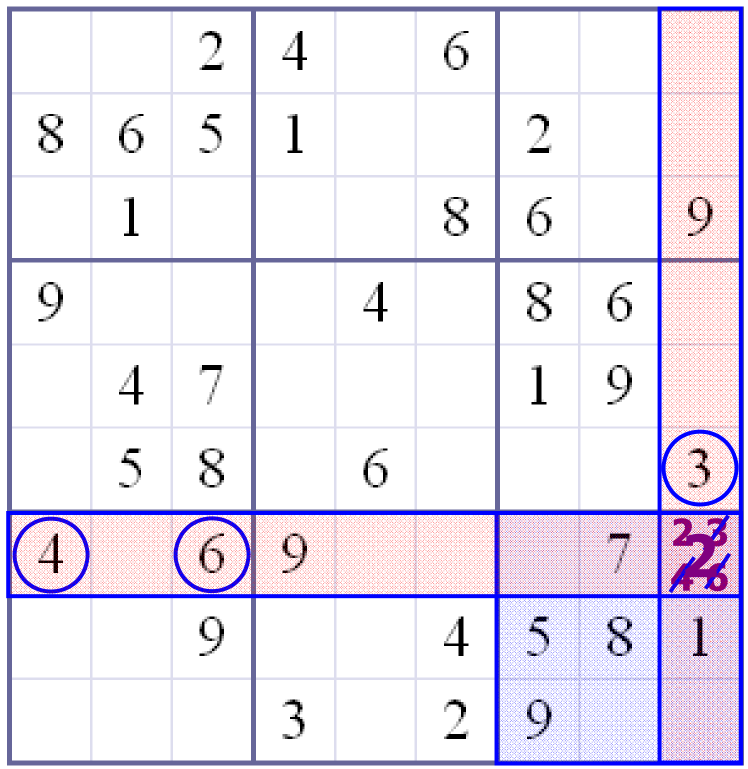
- Variables: x_1, …, x_{13}
- Domains: letters
- Constraints: words from



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

Example: Sudoku (constraint propagation)

Constraint propagation



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

Each row, column and major block must be alldifferent

"Well posed" if it has unique solution: 27 constraints

Sudoku (inference)

		2	1	5				6
			3	6	8		1	
6	1	8			2			4
		5		2				3
	9	3				5	4	
1				3		6		
3			8			4		7
	8		6	4	3			
5				1	7	9		

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

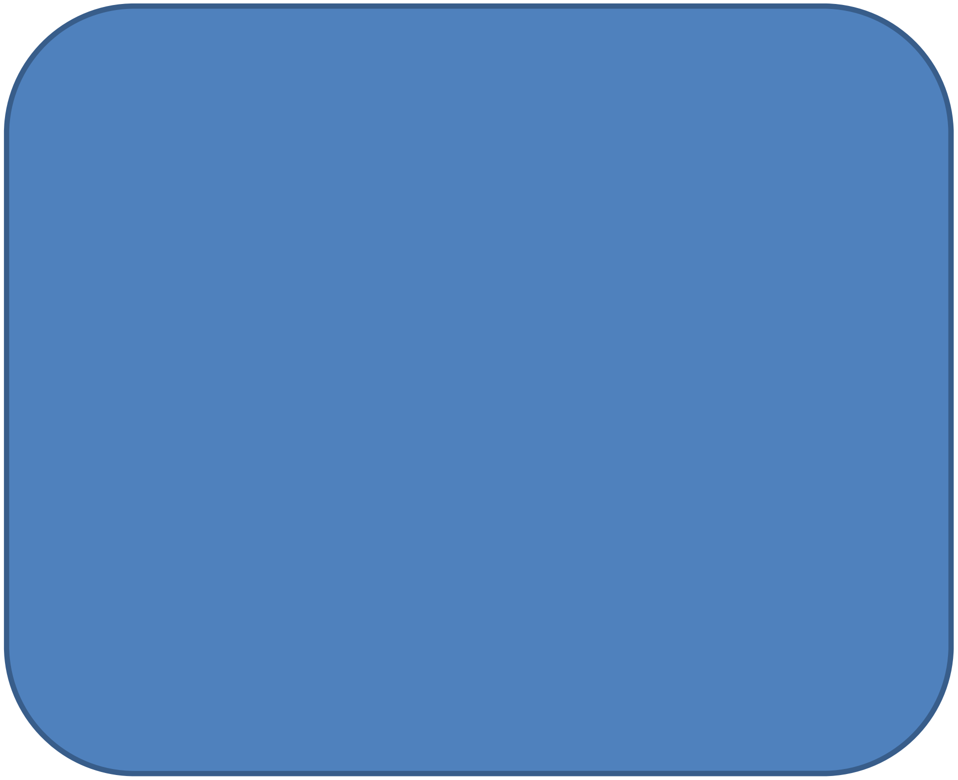
- Sets, domains, tuples
- Relations
- Operations on relations
- Graphs
- Complexity

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

Variables: Drink, color

x_1	x_2
black	coffee
black	tea
green	tea

(a) table



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

Variables: Drink, color

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

Examples

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''

Operations with relations

- Intersection
- Union
- Difference
- Selection
- Projection
- Join
- Composition

Relations are local functions

- Relations are special case of a Local function

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

where

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

A: is a set of valuations

- In constraint networks: functions are boolean

x_1	x_2	f
a	a	true
a	b	false
b	a	false
b	b	true

relation →

x_1	x_2
a	a
b	b

Set operations: intersection, union, difference on relations.

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''

x_1	x_2	x_3
b	b	c
c	b	c

(a) $R \cap R'$

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

(b) $R \cup R'$

x_1	x_2	x_3
a	b	c
c	b	s

(b) $R - R'$

Selection, projection, join

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''

x_1	x_2	x_3
b	b	c
c	b	c

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

x_2	x_3
b	c
n	n

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

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

(c) $R' \bowtie R''$

The join and the logical “and”

- Join :

x_1	x_2
a	a
b	b

$\bowtie$

x_2	x_3
a	a
a	b
b	a

$=$

x_1	x_2	x_3
a	a	a
a	a	b
b	b	a

- Logical AND: $f \wedge g$

x_1	x_2	f
a	a	true
a	b	false
b	a	false
b	b	true

$\wedge$

x_2	x_3	g
a	a	true
a	b	true
b	a	true
b	b	false

$=$

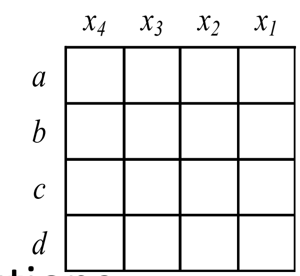
x_1	x_2	x_3	h
a	a	a	true
a	a	b	true
a	b	a	false
a	b	b	false
b	a	a	false
b	a	b	false
b	b	a	true
b	b	b	false

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 problems

- If a CSP M = <X,D,C> 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 $\varphi = \{(A \vee B), (C \vee \neg B)\}$ 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:

A	B
0	1
1	0
1	1

B	C
0	0
0	1
1	1

Constraint's representations

- Relation: allowed tuples

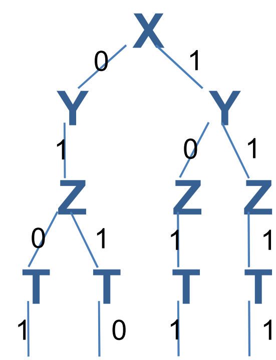
X	Y	Z
1	3	2
2	1	3

- Algebraic expression: $X + Y^2 \leq 10, X \neq Y$
- Propositional formula: $(a \vee b) \rightarrow \neg c$
- A decision tree, a procedure
- Semantics: by a relation

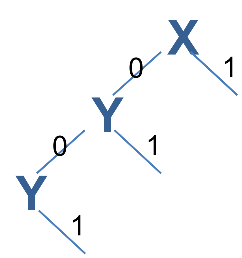
A decision tree or a neural network

Decision tree representations

X	Y	Z	T
0	1	1	0
1	0	1	1
1	1	1	1
0	1	0	1



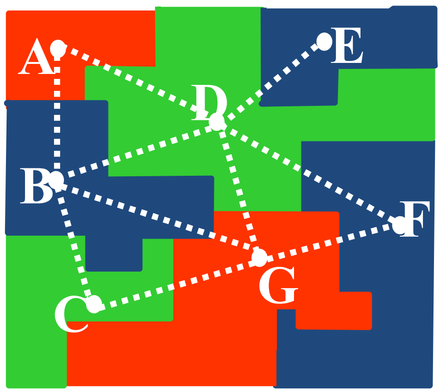
$X \vee Y \vee Z$



X	Y	Z
1	0	0
1	1	0
1	0	1
1	1	1
0	1	0
0	1	1
0	0	1

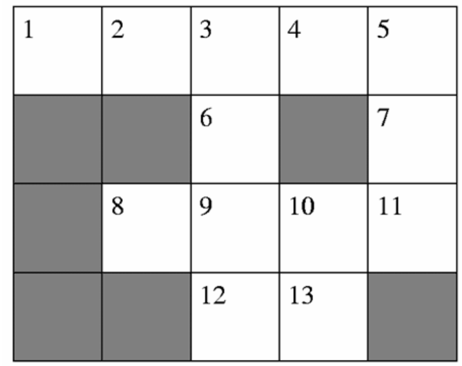
Outline

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

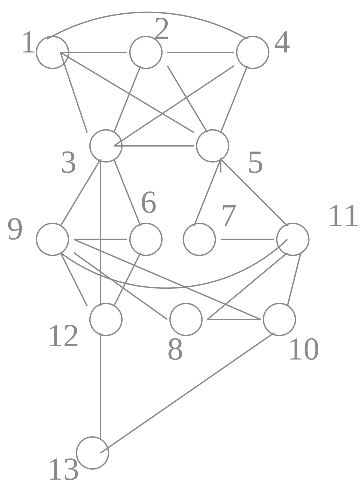


Constraint graphs:

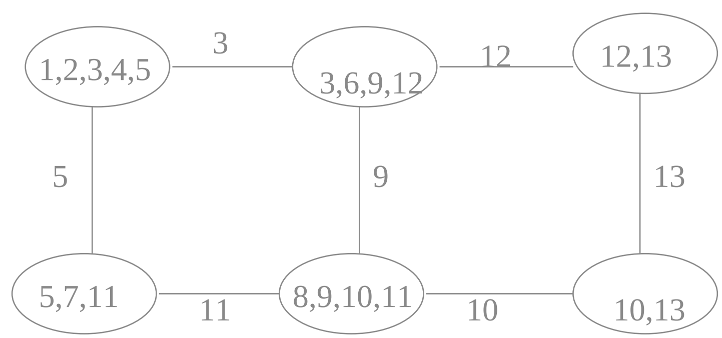
Primal, Dual and Hypergraphs



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



(a)



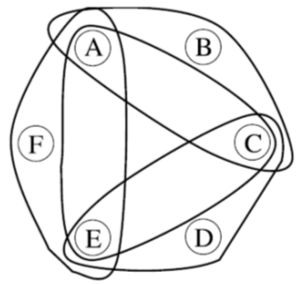
(b)

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)*}*

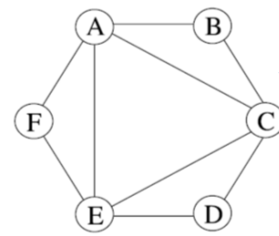
Graph concepts Reviews:

Hyper Graphs and Dual Graphs

A hypergraph

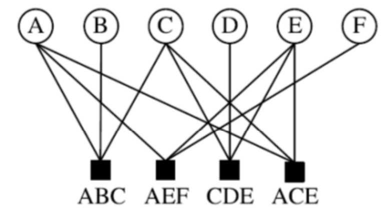


(a)



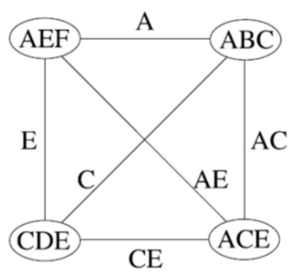
(b)

Primal graphs



(e)

Dual graph



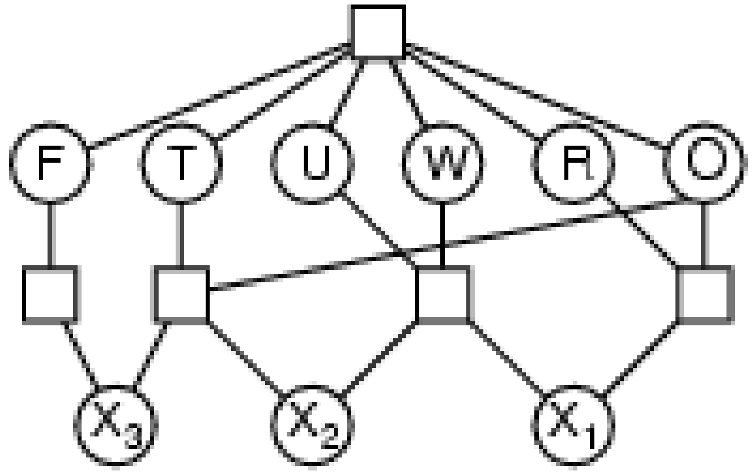
(c)

Factor graphs

Definition 2.1.1 (hypergraph) *A hypergraph is a structure H = (V; S) that consists*
of vertices V = {v_1,…, v_n} and a set of subsets of these vertices S = {S 1,…;S l}

Example: cryptarithmetic

```
  T W O
+ T W O
-------
F O U R
```



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: *Alldiff (F,T,U,W,R,O)*

$O + O = R + 10 \cdot X_1$

$X_1 + W + W = U + 10 \cdot X_2$

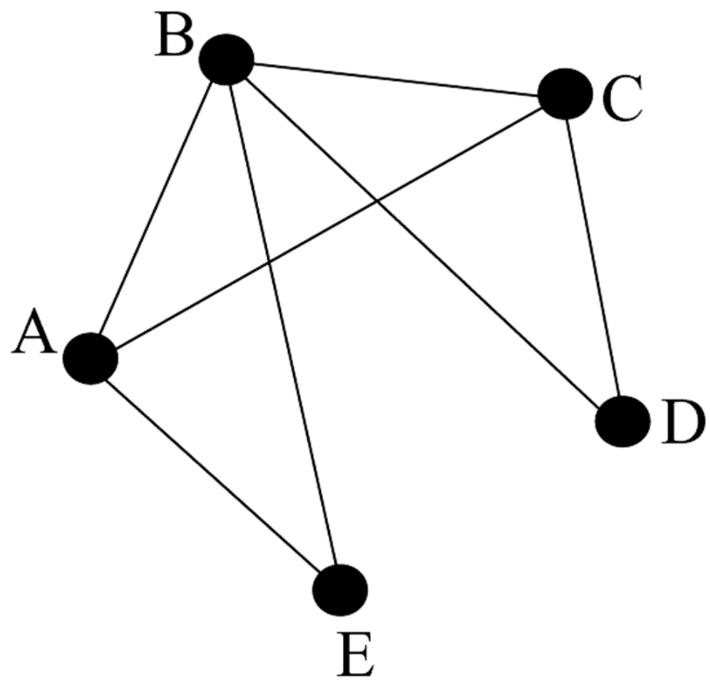
$X_2 + T + T = O + 10 \cdot X_3$

$X_3 = F,\ T \neq 0,\ F \neq 0$

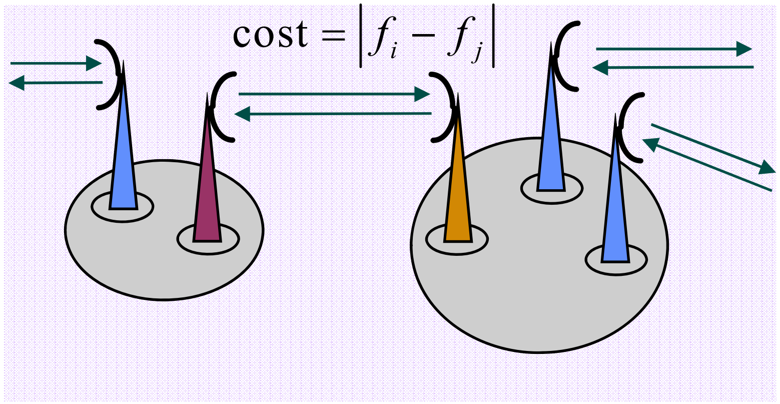
What is the primal graph?
What is the dual graph?

Propositional satisfiability

$\varphi = \{(\neg C), (A \vee B \vee C), (\neg A \vee B \vee E), (\neg B \vee C \vee D)\}.$



Example: radio link assignment



Given a telecommunication network (where each communication link has various antenas) , assign a frequency to each antenna in such a way that all antennas may operate together without noticeable interference.

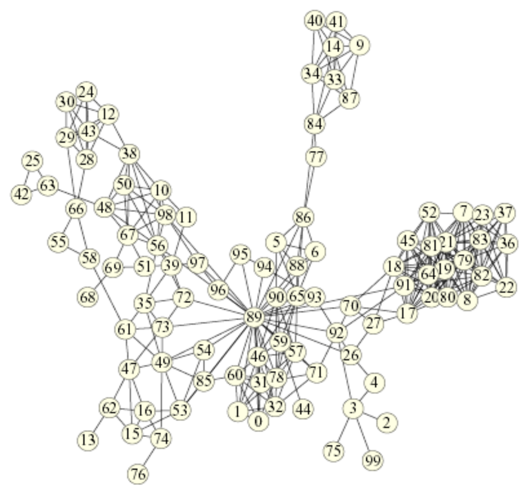
Encoding?

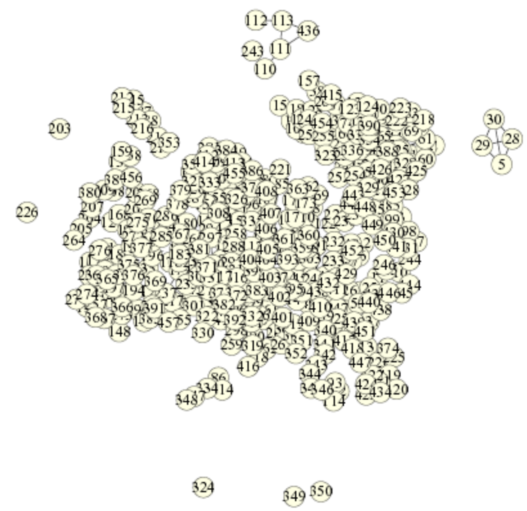
Variables: one for each antenna

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

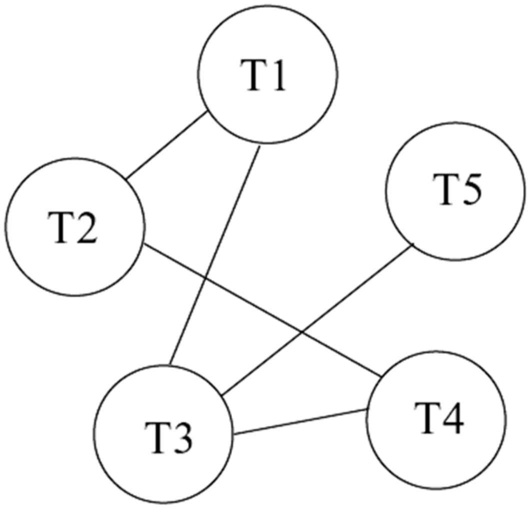




Example: scheduling problem

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

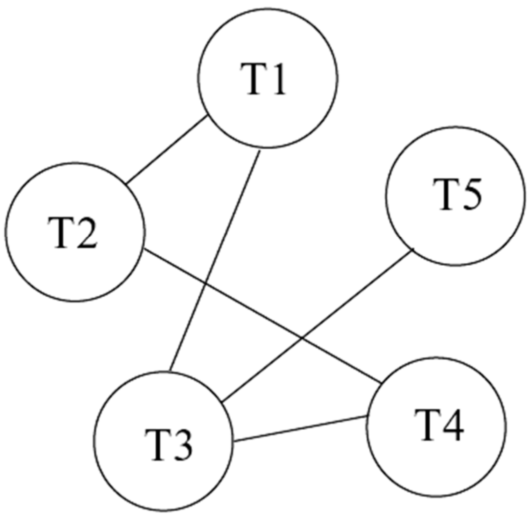


Variables: one for each task

Domains: $D_{T1} = D_{T2} = D_{T3} = D_{T3} = \{1{:}00, 2{:}00, 3{:}00\}$

T4
1:00
3:00

The constraint graph and relations of scheduling problem



Unary constraint

$D_{T4} = \{1{:}00,\ 3{:}00\}$

Binary constraints

$R_{\{T1,T2\}}$: $\{(1{:}00,2{:}00), (1{:}00,3{:}00), (2{:}00,1{:}00), (2{:}00,3{:}00), (3{:}00,1{:}00), (3{:}00,2{:}00)\}$

$R_{\{T1,T3\}}$: $\{(2{:}00,1{:}00), (3{:}00,1{:}00), (3{:}00,2{:}00)\}$

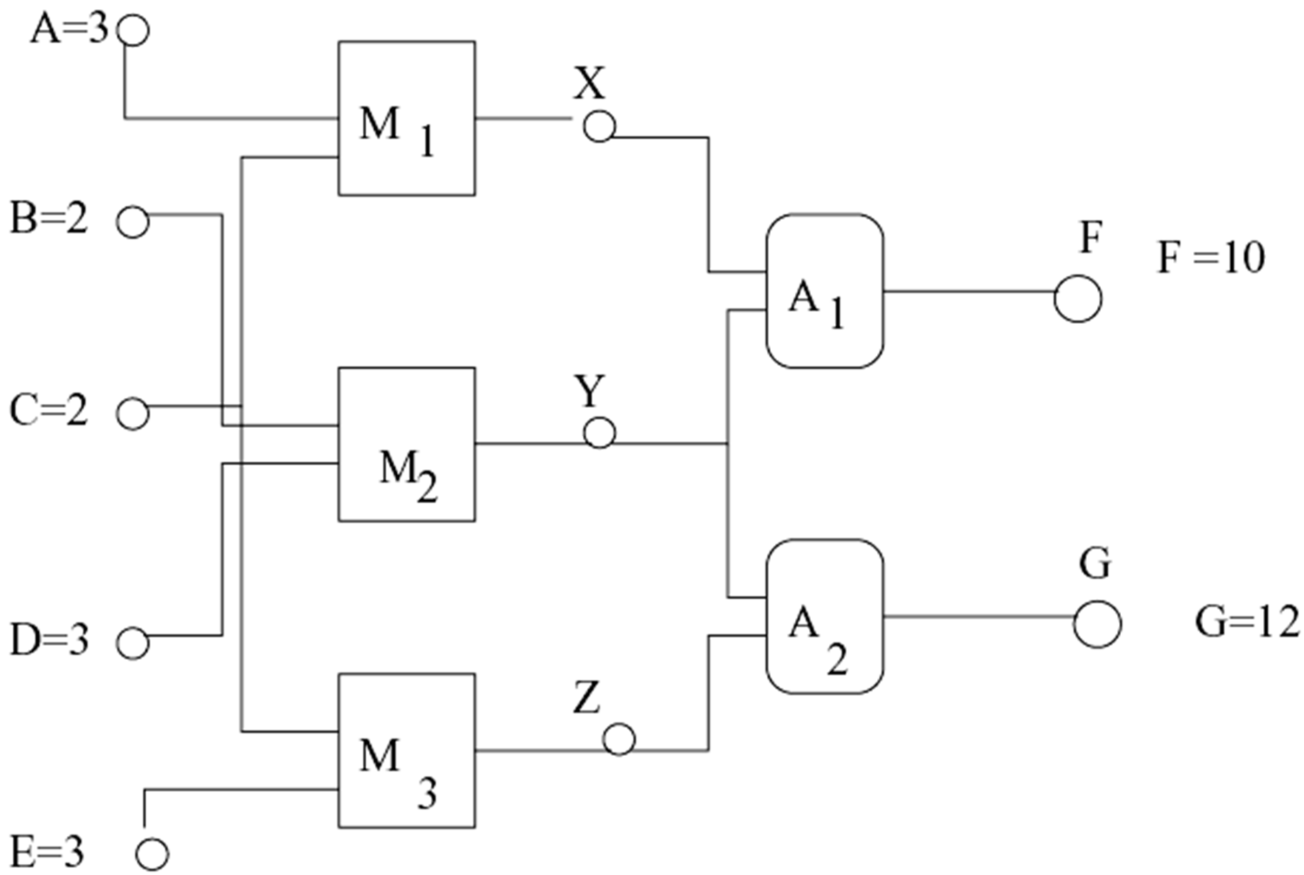
$R_{\{T2,T4\}}$: $\{(1{:}00,2{:}00), (1{:}00,3{:}00), (2{:}00,1{:}00), (2{:}00,3{:}00), (3{:}00,1{:}00), (3{:}00,2{:}00)\}$

$R_{\{T3,T4\}}$: $\{(1{:}00,2{:}00), (1{:}00,3{:}00), (2{:}00,3{:}00)\}$

$R_{\{T3,T5\}}$: $\{(2{:}00,1{:}00), (3{:}00,1{:}00), (3{:}00,2{:}00)\}$

A combinatorial circuit

M a multiplier, *A* is an adder

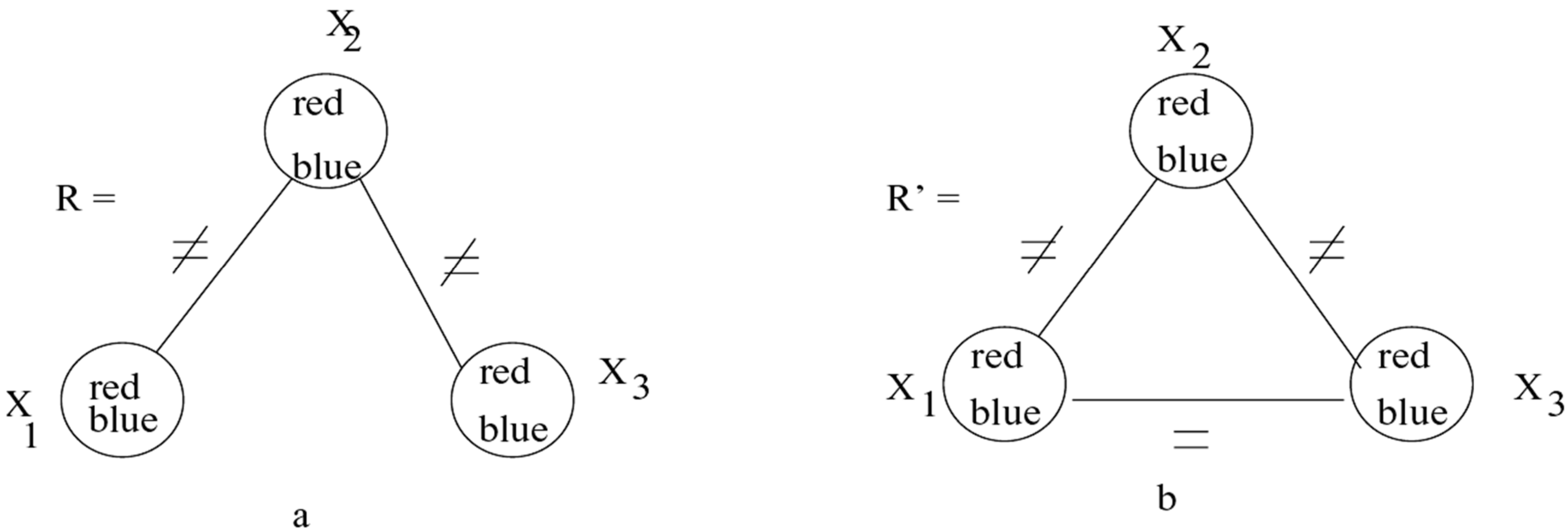


Outline

- ✓ 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 $\mathfrak{R}$ to be colored by two colors,
an equivalent representation $\mathfrak{R}'$ having a newly inferred constraint between $x1$ and $x3$.



Equivalence and deduction with constraints (composition)

Composition of relations (*Montanari'74*)

Constraint deduction can be accomplished through the *composition* operation.

Input: two binary relations $\boldsymbol{R}_{ab}$ and $\boldsymbol{R}_{bc}$ with one variable in common.

Output: a new induced relation $\boldsymbol{R}_{ac}$.

Bit-matrix operation: matrix multiplication

$$R_{ac} = R_{ab} \cdot R_{bc}$$

$$R_{ab} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad R_{bc} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad R_{ac} = ?$$

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:*

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

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

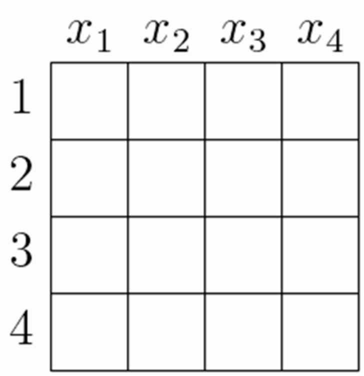
$$R_{xz} = \pi_{xz}(R_{xy} \bowtie R_{yz})$$

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^{k^n}
- Number of networks: $2^{n^2 k^2}$
- Most relations cannot be represented by binary constraint networks

The N-queens constraint network is there a tighter network?

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



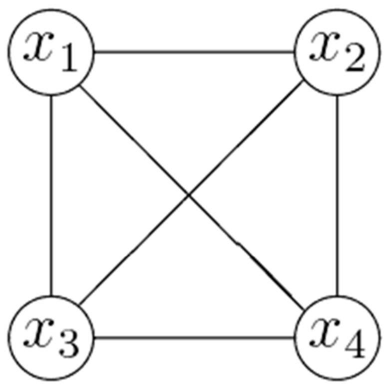
(a)

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

(b)

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

The 4-queens constraint network:
(a) The constraint graph. (b) The minimal binary constraints.
(c) The minimal unary constraints (the domains).



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

$$
\begin{aligned}
D_1 &= \{2,3\}\\
D_2 &= \{1,4\}\\
D_3 &= \{1,4\}\\
D_4 &= \{2,3\}
\end{aligned}
$$

(a) (b) (c)

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).
- *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?

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

Example: Sudoku (constraint propagation)

What is the minimal network?
The projection network?

Constraint propagation

		2	4		6			
8	6	5	1			2		
	1				8	6		9
9				4		8	6	
	4	7				1	9	
	5	8		6				3
4		6	9				7	2 ~~3~~ ~~4~~ ~~6~~ 2
		9			4	5	8	1
			3		2	9		

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

Each row, column and major block must be alldifferent

"Well posed" if it has unique solution: 27 constraints

Outline

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