



Constraint satisfaction and applications

András Z. Salamon

Computing Laboratory, University of Oxford
(supervisor: Peter Jeavons)

Oxford-Man Institute of Quantitative Finance

7 May 2008

CSP examples



- ▶ allocating frequencies to mobile phone cells
- ▶ checking if a logical formula is satisfiable
- ▶ laying out components on circuit board
- ▶ fitting a protein structure to measurements
- ▶ finding DNA sequence from set of contigs
- ▶ drawing up an examination timetable
- ▶ scheduling set of tasks in a project
- ▶ **finance?**

Constraint satisfaction



assign values to variables to satisfy constraints

Constraint satisfaction



assign **values** to variables to satisfy constraints

Constraint satisfaction



assign values to **variables** to satisfy constraints

Constraint satisfaction



assign values to variables to **satisfy constraints**

What is a constraint?



each constraint has two parts:

- ▶ scope: ordered list of variables
- ▶ relation: allowed combinations of values
- ▶ relation R of arity $\rho(R)$ over a domain D :
 $R \subseteq D^{\rho(R)}$

Nonogram



		2				
		1	1	3	3	3
1						
1						
3						
3						
4						

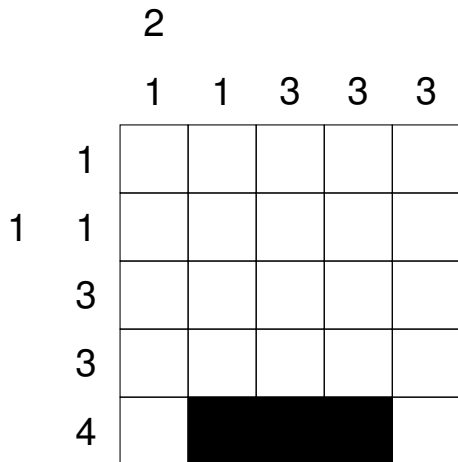
Nonogram



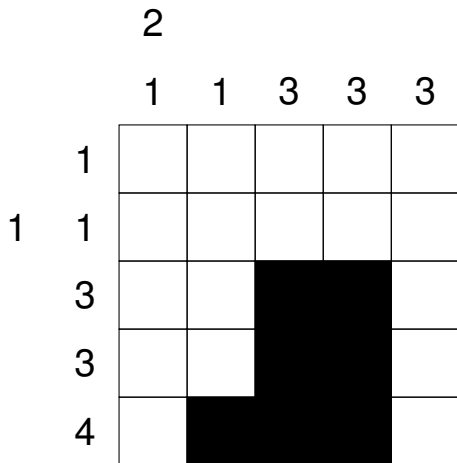
		2				
		1	1	3	3	3
1						
1						
3						
3						
4						

← easy

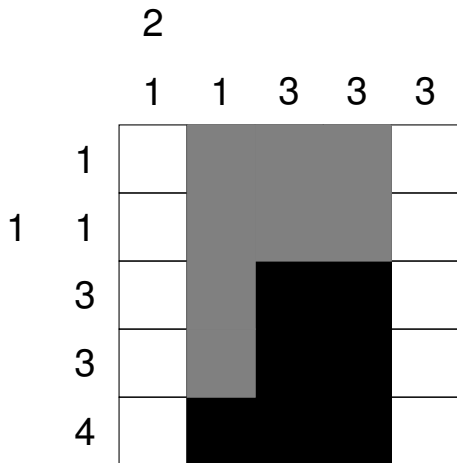
Nonogram



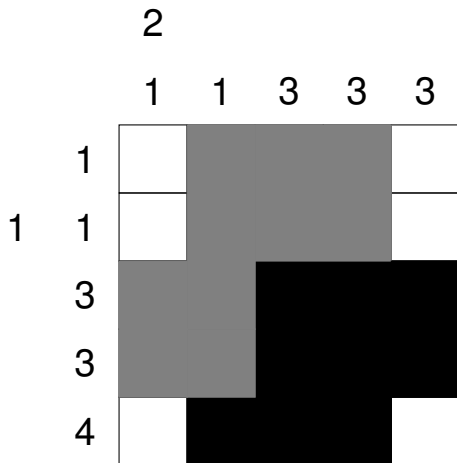
Nonogram



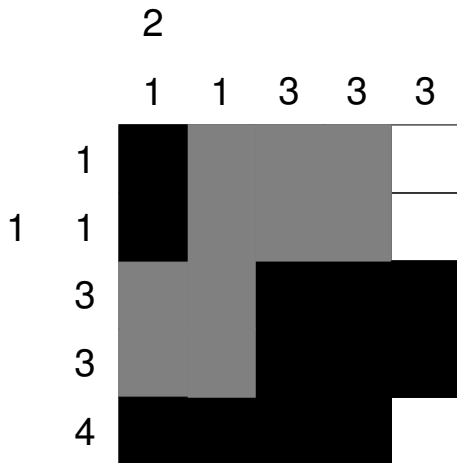
Nonogram



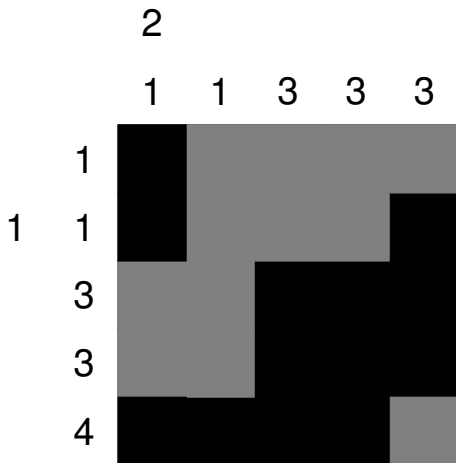
Nonogram



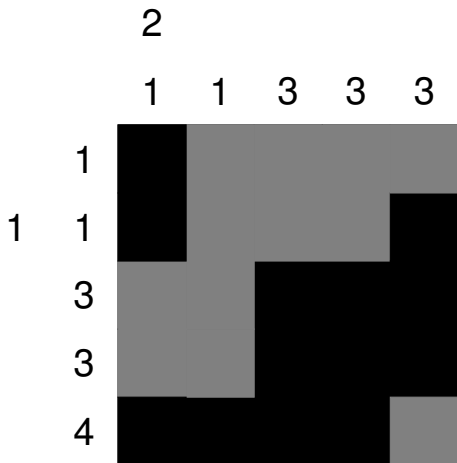
Nonogram



Nonogram



Nonogram (CSP)



Variable-value representation



Constraint satisfaction problem **instance**

- ▶ variables: set V of size $|V| = s$
- ▶ values: set D of size $|D| = t$
- ▶ constraints: set C
- ▶ solution: function $f: V \rightarrow D$ such that if $((v_1, v_2, \dots, v_r), R) \in C$ then $(f(v_1), f(v_2), \dots, f(v_r)) \in R$

Variable-value representation



Constraint satisfaction problem instance

- ▶ **variables**: set V of size $|V| = s$
- ▶ values: set D of size $|D| = t$
- ▶ constraints: set C
- ▶ solution: function $f: V \rightarrow D$ such that if $((v_1, v_2, \dots, v_r), R) \in C$ then $(f(v_1), f(v_2), \dots, f(v_r)) \in R$

Variable-value representation



Constraint satisfaction problem instance

- ▶ variables: set V of size $|V| = s$
- ▶ **values**: set D of size $|D| = t$
- ▶ constraints: set C
- ▶ solution: function $f: V \rightarrow D$ such that if $((v_1, v_2, \dots, v_r), R) \in C$ then $(f(v_1), f(v_2), \dots, f(v_r)) \in R$

Variable-value representation



Constraint satisfaction problem instance

- ▶ variables: set V of size $|V| = s$
- ▶ values: set D of size $|D| = t$
- ▶ **constraints**: set C
- ▶ solution: function $f: V \rightarrow D$ such that if $((v_1, v_2, \dots, v_r), R) \in C$ then $(f(v_1), f(v_2), \dots, f(v_r)) \in R$

Variable-value representation



Constraint satisfaction problem instance

- ▶ variables: set V of size $|V| = s$
- ▶ values: set D of size $|D| = t$
- ▶ constraints: set C
- ▶ **solution**: function $f: V \rightarrow D$ such that if $((v_1, v_2, \dots, v_r), R) \in C$ then $(f(v_1), f(v_2), \dots, f(v_r)) \in R$

Variable-value representation



Constraint satisfaction problem instance

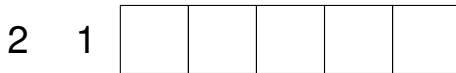
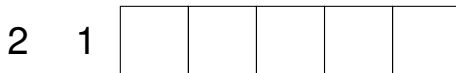
- ▶ variables: set V of size $|V| = s$
- ▶ values: set D of size $|D| = t$
- ▶ constraints: set C
- ▶ solution: function $f: V \rightarrow D$ such that if $((v_1, v_2, \dots, v_r), R) \in C$ then $(f(v_1), f(v_2), \dots, f(v_r)) \in R$

CSP instance: (V, D, C)

Nonogram constraint



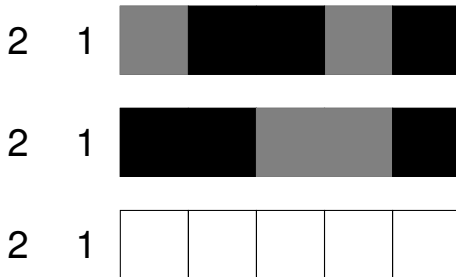
possible patterns



Nonogram constraint



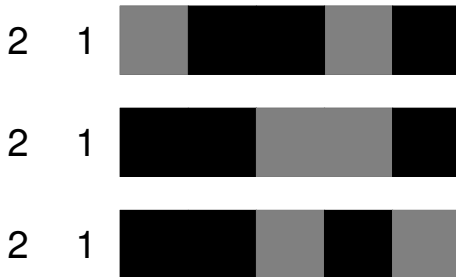
possible patterns



Nonogram constraint



possible patterns



Nonogram constraint



possible assignments

v_{00} v_{01} v_{02} v_{03} v_{04} 0 1 1 0 1

Nonogram constraint



possible assignments

					0	1	1	0	1
v_{00}	v_{01}	v_{02}	v_{03}	v_{04}	1	1	0	0	1

Nonogram constraint



possible assignments

					0	1	1	0	1
					1	1	0	0	1
v_{00}	v_{01}	v_{02}	v_{03}	v_{04}	1	1	0	1	0

Nonogram constraint



scope

relation

$$\left((v_{00}, v_{01}, v_{02}, v_{03}, v_{04}), \left\{ \begin{array}{l} (0, 1, 1, 0, 1), \\ (1, 1, 0, 0, 1), \\ (1, 1, 0, 1, 0) \end{array} \right\} \right)$$

constraint

Sudoku



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

Constraint language



language = set of relations

- ▶ all-different (\neq)
- ▶ anything-goes ($*$)
- ▶ all-equals (\equiv)
- ▶ not-all-equals (\neq)
- ▶ linear order ($<$)
- ▶ black box (compute as required)

Sudoku as CSP



27 all-different constraints (arity 9)
constraints (arity 1) for numbers already in grid

Sudoku as CSP



27 all-different constraints (arity 9)
constraints (arity 1) for numbers already in grid
or play some code golf:

```
$_=$' .$_.$' .<>;split//;${/[@_ [map{$i-($i="@"-")%9+$_,9*$_+$i%9,9*$_%26+$i-$i%27+$i%9-$i%3}0..8]]/o||do$0}for/0/||print..9
```

(multiple contributors) **120 bytes of Perl**

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$(v_1, \dots, v_{\rho(R)}) \in Q_i \Rightarrow (f(v_1), \dots, f(v_{\rho(R)})) \in R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ **source**: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$(v_1, \dots, v_{\rho(R)}) \in Q_i \Rightarrow (f(v_1), \dots, f(v_{\rho(R)})) \in R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ **source:** $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ **target:** $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$(v_1, \dots, v_{\rho(R)}) \in Q_i \Rightarrow (f(v_1), \dots, f(v_{\rho(R)})) \in R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T **similar**: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$(v_1, \dots, v_{\rho(R)}) \in Q_i \Rightarrow (f(v_1), \dots, f(v_{\rho(R)})) \in R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ **solution**: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$\begin{aligned} (v_1, \dots, v_{\rho(R)}) \in Q_i &\Rightarrow \\ (f(v_1), \dots, f(v_{\rho(R)})) &\in R_i \end{aligned}$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$(v_1, \dots, v_{\rho(R)}) \in Q_i \Rightarrow (f(v_1), \dots, f(v_{\rho(R)})) \in R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$(v_1, \dots, v_{\rho(R)}) \in Q_i \Rightarrow f(v_1, \dots, v_{\rho(R)}) \in R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: function $f: V \rightarrow D$ s.t. $\forall i \in I$

$$f(Q_i) \subseteq R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: **function** $f: V \rightarrow D$ s.t. $\forall i \in I$

$$f(Q_i) \subseteq R_i$$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: **homomorphism** $f: S \rightarrow T$

Homomorphism representation



Constraint satisfaction problem instance

- ▶ source: $S = (V, (Q_i)_{i \in I})$, $|V| = s$
- ▶ target: $T = (D, (R_i)_{i \in I})$, $|D| = t$
- ▶ S and T similar: arities of Q_i and R_i match
- ▶ solution: homomorphism $f: S \rightarrow T$

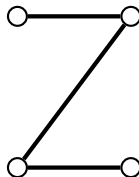
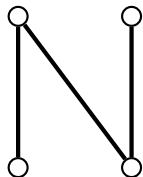
CSP instance: $S \rightarrow T?$

Graph isomorphism



Are graphs $G = (V, Q)$, $H = (D, R)$ isomorphic?

- ▶ $S = (V, (Q, \overline{Q}))$, $T = (D, (R, \overline{R}))$
- ▶ $|V| = |D|$



Graph isomorphism



Are graphs $G = (V, Q)$, $H = (D, R)$ isomorphic?

- ▶ $S = (V, (Q, \overline{Q}))$, $T = (D, (R, \overline{R}))$
- ▶ $|V| = |D|$

G is isomorphic to H if and only if $S \rightarrow T$



Graph isomorphism

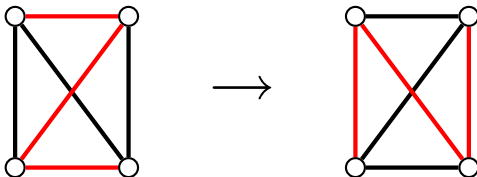


Are graphs $G = (V, Q)$, $H = (D, R)$ isomorphic?

▶ $S = (V, (Q, \overline{Q}))$, $T = (D, (R, \overline{R}))$

▶ $|V| = |D|$

G is isomorphic to H if and only if $S \rightarrow T$

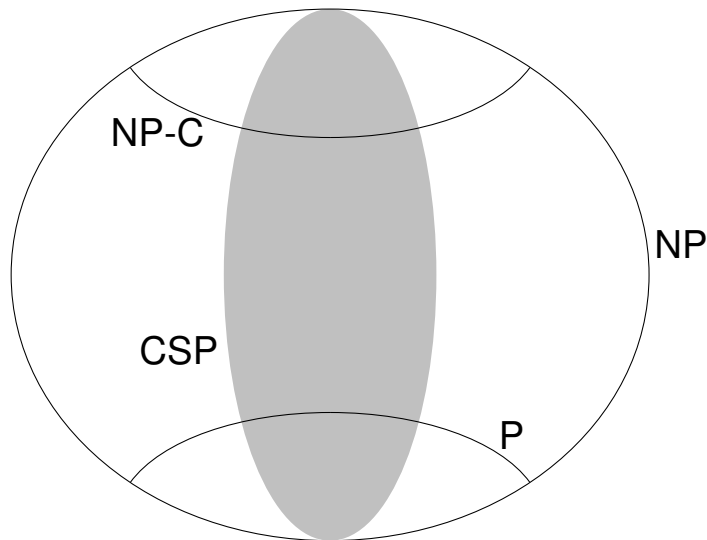


Constraints and logic



- ▶ $NP = SO\exists$ (Fagin 1973)
- ▶ MMSNP: monotone monadic strict NP without inequality (Feder & Vardi 1993)
- ▶ every fragment of MMSNP is computationally equivalent to $CSP(_, \mathcal{T})$ for some finite \mathcal{T} (Kun 2006)

CSP vs. NP



Optimization



- ▶ n variables
- ▶ objective function: $f: \mathbb{R}^n \rightarrow \mathbb{R}$
- ▶ feasible region: $S \subseteq \mathbb{R}^n$
- ▶ problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in S \end{array}$$

Optimization



- ▶ often $S = \{x \mid x \models \bigwedge_{i=1}^L (g_i(x) \geq 0)\}$
- ▶ global method: conditions hold for each solution
- ▶ local information usually just added together
- ▶ if S convex: solutions can often be computed in polynomial time, typically $O(n^3 L)$
- ▶ S not convex: NP-hard

Linear programming



$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0 \end{array}$$

Linear programming



- ▶ Polytope: multi-dimensional polygon
- ▶ Convex polytope: intersection of half-spaces
- ▶ Convex polytope: feasible region of linear program
- ▶ Linear programming: find minimum or maximum of linear function on boundary of convex polytope

Linear programming



- ▶ simplex method: consider sequence of vertices s.t. f increases
- ▶ ... but number of vertices can be exponential in L, n
- ▶ convex: ellipsoid or interior point methods
- ▶ polynomial in L, n

Integer programming



$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0, \\ & x \in \mathbb{Z}^n \end{array}$$

Some things really are discrete



$$\begin{aligned} &\text{maximize} && 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ &\text{subject to} && \{x_1, x_2, x_3, x_4\} \subseteq \{0, 1\}, \\ &&& 6.7x_1 + 10x_2 + 5.5x_3 + 3.4x_4 \leq 19 \end{aligned}$$

- ▶ LP: $(1, 0.89, 0, 1)$
- ▶ IP: $(0, 1, 1, 1)$
- ▶ rounding doesn't work

Integer programming



- ▶ NP-hard
- ▶ sometimes LP relaxation works
- ▶ otherwise start searching. . .

Quadratic programming



$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T Qx + c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0 \end{array}$$

Quadratic programming



- ▶ not known to be in NP
- ▶ if Q positive semidefinite then f convex, so polynomial time
- ▶ if Q has any negative eigenvalues then NP-hard

Constraint satisfaction



- ▶ global properties from local information
- ▶ objective function arbitrary (valued CSP)
- ▶ feasible region seldom convex
- ▶ is feasible region non-empty?
- ▶ is there a solution at all?

Search



- ▶ LP: branch-and-bound
- ▶ bounds on f for each state
- ▶ prune B when there is A s.t. $\text{lb}(B) > \text{ub}(A)$
- ▶ CSP: backtrack search
- ▶ consistency enforcement
- ▶ prune domain of each variable

Constraint satisfaction problems



- ▶ problem: class of instances (**restricted**)
- ▶ (**structure**) $\text{CSP}(\mathcal{S}, _) = \{S \rightarrow _ \mid S \in \mathcal{S}\}$
CLIQUE, PLANAR GRAPH, ...
- ▶ (**language**) $\text{CSP}(_, \mathcal{T}) = \{_ \rightarrow T \mid T \in \mathcal{T}\}$
SAT, SUDOKU, COLOURING, 2-SAT, ...
- ▶ $\text{CSP}(\mathcal{S}, \mathcal{T}) = \{S \rightarrow T \mid S \in \mathcal{S}, T \in \mathcal{T}\}$
SUBGRAPH ISOMORPHISM, SUBSET
SUM, GRAPH ISOMORPHISM, LP, ...

Constraint satisfaction problems



- ▶ problem: class of instances (**restricted**)
- ▶ (structure) $\text{CSP}(\mathcal{S}, _) = \{S \rightarrow _ \mid S \in \mathcal{S}\}$
CLIQUE, PLANAR GRAPH, ...
- ▶ (language) $\text{CSP}(_, \mathcal{T}) = \{_ \rightarrow T \mid T \in \mathcal{T}\}$
SAT, SUDOKU, COLOURING, 2-SAT, ...

- ▶ **$\text{CSP}(\mathcal{S}, \mathcal{T}) = \{S \rightarrow T \mid S \in \mathcal{S}, T \in \mathcal{T}\}$**
SUBGRAPH ISOMORPHISM, SUBSET
SUM, GRAPH ISOMORPHISM, LP, ...

hybrid

A red curved arrow originates from the word 'hybrid' and points upwards and to the left towards the definition of CSP(S, T) in the list above.

Finance?



... in progress

- ▶ combinatorial auctions?
- ▶ constructing an index fund?
- ▶ portfolio optimization with thresholds?
- ▶ optimization of price spreads?

ideal: “is feasible region empty?”

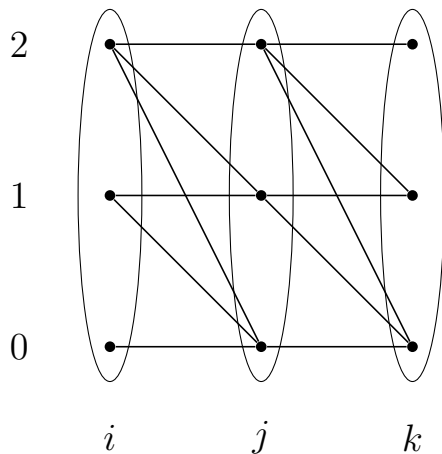
Microstructure complement



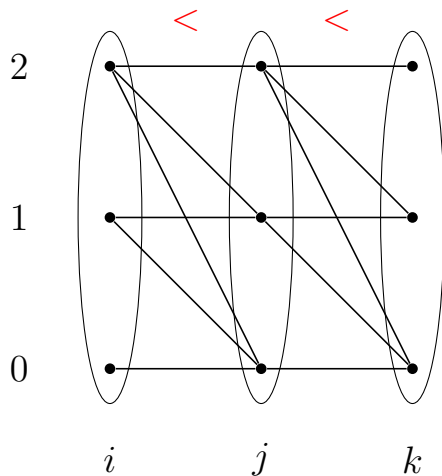
how to deal with hybrid restrictions?

- ▶ direct product: $((u, v), (x, y)) \in Q \times R$ iff $(u, x) \in Q$ and $(v, y) \in R$
- ▶ microstructure complement:
 $MSC(S, T) = (S \times \overline{T}) \cup (= \times \neq)$

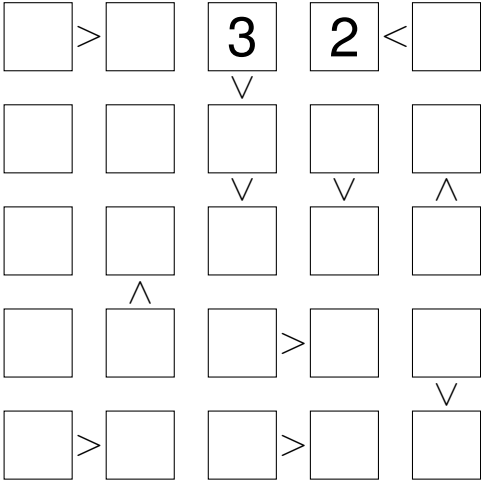
MSC illustrated



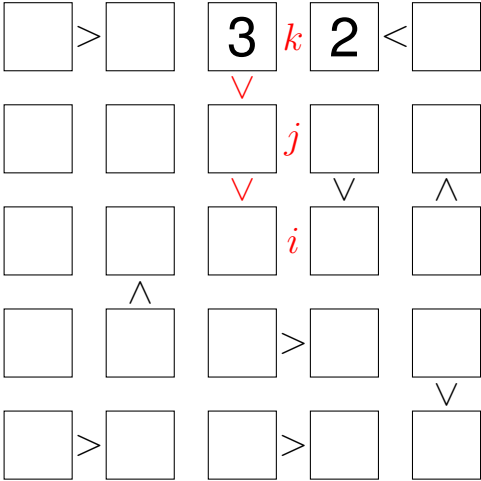
MSC illustrated



Futoshiki



Futoshiki



MSC applied



$S \rightarrow T$ iff $MSC(S, T)$ contains a size $|V(S)|$ independent set (Jégou 1993, Cohen 2003)

- ▶ now: graphs (later: hypergraphs)
- ▶ www.teo.informatik.uni-rostock.de/isgci/
- ▶ 1031 classes (2008-04-08), many tractable for INDEPENDENT SET
- ▶ e.g. perfect, claw-free, “greedy”

Perfect microstructures



- ▶ chordal-MSC \subseteq (hole, antihole)-free
- ▶ MSC(TREE) \subseteq (hole, odd-antihole)-free
- ▶ MSC(alldiff) \subseteq (odd-hole, odd-antihole)-free
- ▶ (odd-hole, odd-antihole)-free = PERFECT
(Chudnovsky et al. 2004)

(S., Jeavons 2008)

Broken-triangle property



if $i < j < k$ and

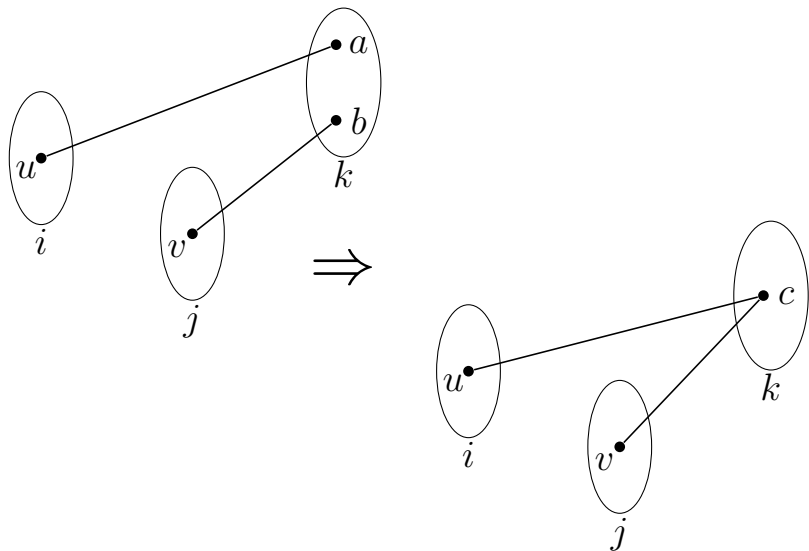
- ▶ (i, u) compatible with (j, v)
- ▶ (i, u) compatible with (k, a)
- ▶ (j, v) compatible with (k, b)

then (i, u) compatible with (k, b)

OR (j, v) compatible with (k, a)

kind of “disjunctive transitivity”

Broken-triangle property



Broken-triangle property



- ▶ if there exists a variable order such that BTP holds everywhere, then solution can be found efficiently
- ▶ can find such order (if it exists) efficiently

(Cooper, Jeavons, S. 2008)

Summary



- ▶ **CSP**
- ▶ code golf
- ▶ broken triangles

Summary



- ▶ CSP (vs. optimization)
- ▶ code golf
- ▶ broken triangles

Summary



- ▶ CSP (vs. optimization)
- ▶ code golf
- ▶ broken triangles

Summary



- ▶ CSP (vs. optimization)
- ▶ Sudoku is CSP
- ▶ broken triangles

Summary



- ▶ CSP (vs. optimization)
- ▶ Sudoku is CSP
- ▶ **broken triangles**

Summary



- ▶ CSP (vs. optimization)
- ▶ Sudoku is CSP
- ▶ microstructure: hybrid tractability

What next?



- ▶ feasible region problems
 - ▶ more tractable MSCs
 - ▶ other hybrid approaches
-

Relational structure



tuple $(D, (R_i)_{i \in I})$

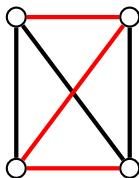
- ▶ domain D
- ▶ tuple $(R_i)_{i \in I}$ of relations
- ▶ ordered index set I
- ▶ relations R_i over D : $R_i \subseteq D^{\rho(R_i)}$

Graph as relational structure



complemented representation: $(V, (E_i)_{i \in \{1,2\}})$

- ▶ E_1 symmetric irreflexive relation over V (arity 2)
- ▶ E_2 complement of E_1
- ▶ colour E_1 edges black, E_2 red

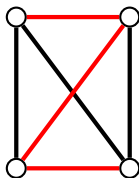


Graph as relational structure



complemented representation: $(V, (E_i)_{i \in \{1,2\}})$

- ▶ E_1 symmetric irreflexive relation over V (arity 2)
- ▶ E_2 complement of E_1
- ▶ colour E_1 edges black, E_2 red

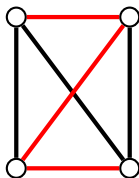


Graph as relational structure



complemented representation: $(V, (E, \bar{E}))$

- ▶ E symmetric irreflexive relation over V (arity 2)
- ▶ \bar{E} **complement** of E
- ▶ colour E edges black, \bar{E} red



Languages



- ▶ $\Gamma(P)$: set of all relations in target structures
- ▶ expressibility: $\langle \Gamma \rangle$ characterises tractability (Jeavons 1996)
- ▶ local language: tuples of relations in each instance

Local-global conjecture



- ▶ due to Bulatov (2006)
- ▶ if $* \rightarrow T$ NP-complete then there is finite $C \subseteq T$ such that $* \rightarrow C$ is NP-complete
- ▶ *is there always a finite reason for hardness?*
- ▶ need infinite chain C of increasing hardness, such that $\bigvee C$ is NP-complete
- ▶ infinite ascending chain in clone lattice

Local languages



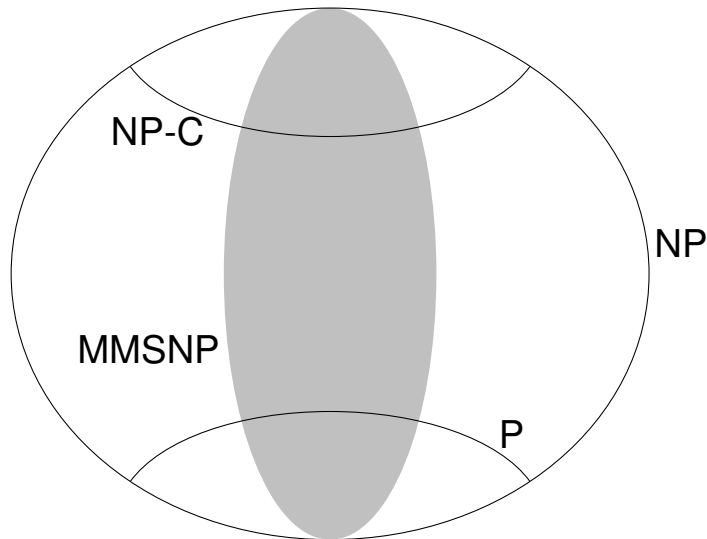
$CS(2, s) \rightarrow T$	$\{R_i\}$	r, s, t
LOG-CLIQUE	E	$t = 2^s$
s -CLIQUE	E	
t -COLOURING	$*, \neq$	
GRAPH ISOMORPHISM	E, \overline{E}	$s = t$
SUBGRAPH ISOMORPHISM	E, \overline{E}	
<hr/>		
$CS(r, s) \rightarrow T$		
HYPERGRAPH t -COLOURING	$*, \neq$	
COMPLETE $(r, _, _)$ -CSP	R_1, \dots, R_q	
<i>E is a symmetric irreflexive relation</i>		

Local languages



$CS(2, s) \rightarrow T$	$\{R_i\}$	r, s, t
<hr/>		
s -CLIQUE	E	
t -COLOURING	$*, \neq$	
GRAPH ISOMORPHISM	E, \overline{E}	$s = t$
SUBGRAPH ISOMORPHISM	E, \overline{E}	
$CS(r, s) \rightarrow T$		
<hr/>		
COMPLETE $(r, _)$ -SUBSTRUCTURE	R	
HYPERGRAPH t -COLOURING	$*, \neq$	

MMSNP vs. NP



MMSNP vs. NP

