The Expressive Power of Valued Constraints: Hierarchies and Collapses

Stanislav Živný, University of Oxford

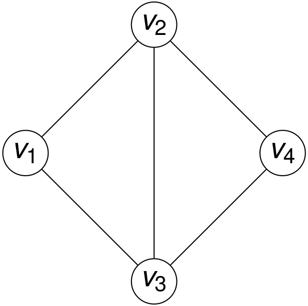
St Andrews, 6 December, 2007

Outline

My research:

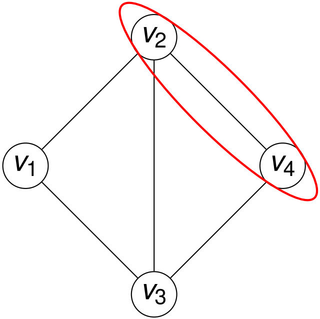
- valued constraints and their expressive power
- complexity and algebraic properties of valued constraints
- submodular function minimisation

3-Colouring Revisited



3-Colouring Revisited

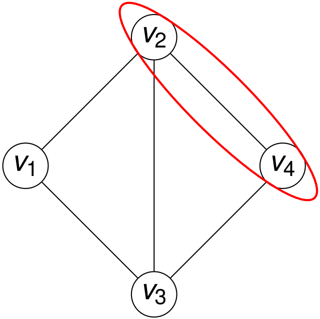
$c = \langle \langle v_2, v_4 \rangle, \phi \rangle$



3-Colouring Revisited

$c = \langle \langle v_2, v_4 \rangle, \phi \rangle$

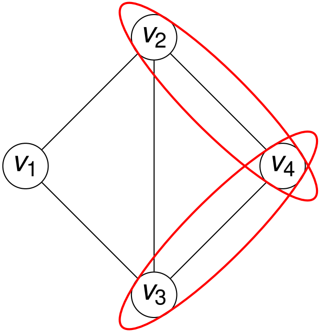
$$\phi(x, y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$



3-Colouring Revisited

$c = \langle \langle v_2, v_4 \rangle, \phi \rangle$

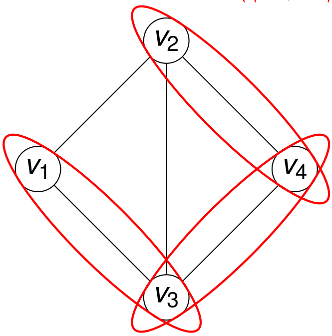
$$\phi(x, y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$



3-Colouring Revisited

$c = \langle \langle v_2, v_4 \rangle, \phi \rangle$

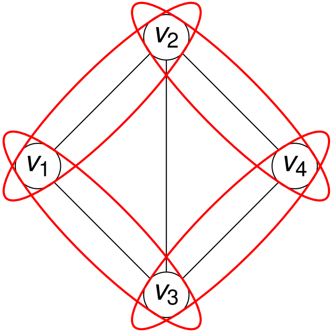
$$\phi(x, y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$



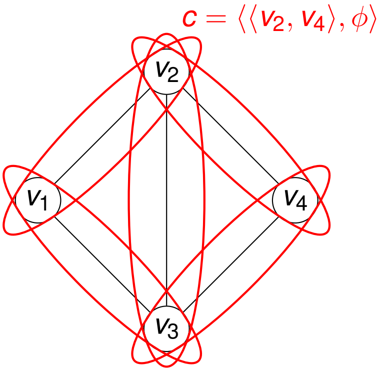
3-Colouring Revisited

$c = \langle\langle v_2, v_4\rangle, \phi\rangle$

$$\phi(x,y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$



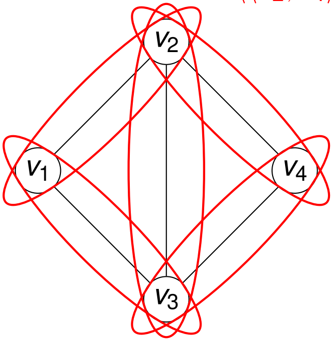
3-Colouring Revisited



$$\phi(x, y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$

3-Colouring Revisited

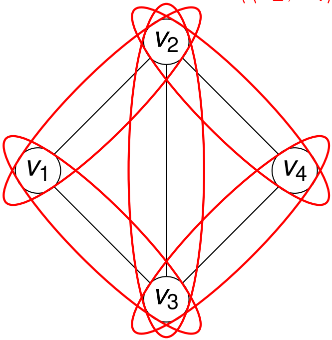
$c = \langle\langle v_2, v_4\rangle, \phi\rangle$



$$\phi(x,y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$

minimum total cost

3-Colouring Revisited

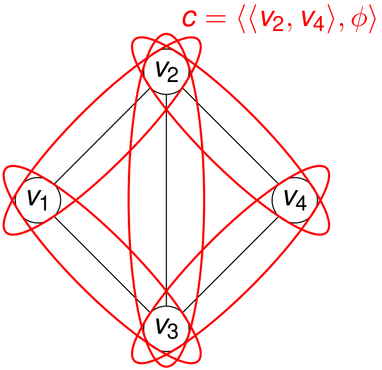


$c = \langle\langle v_2, v_4\rangle, \phi\rangle$

$$\phi(x, y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$

minimum total cost = 0 $\Leftrightarrow$ there is a 3-colouring

3-Colouring Revisited

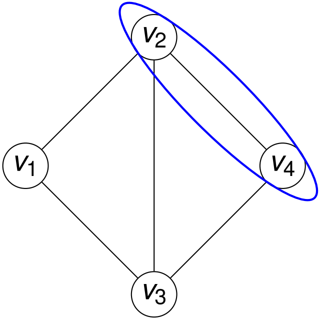


$$\phi(x,y) = \begin{cases} \infty & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$

minimum total cost $= 0 \Leftrightarrow$ there is a 3-colouring
(note: ϕ is a relation)

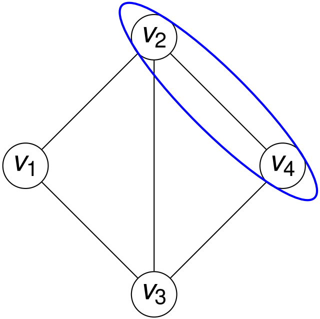
3-Colouring Revisited

$c' = \langle \langle v_2, v_4 \rangle, \phi' \rangle$



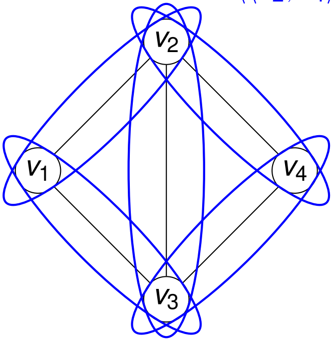
3-Colouring Revisited

$c' = \langle\langle v_2, v_4\rangle, \phi'\rangle \quad \phi'(x,y) = \begin{cases} 1 & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$

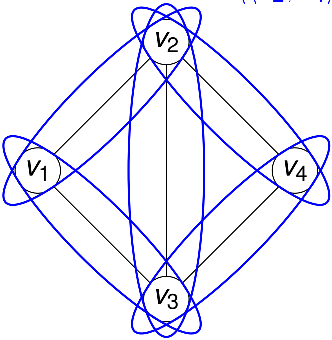


3-Colouring Revisited

$c' = \langle\langle v_2, v_4\rangle, \phi'\rangle$ $\quad \phi'(x,y) = \begin{cases} 1 & \text{if } (x=y) \\ 0 & \text{otherwise} \end{cases}$



3-Colouring Revisited



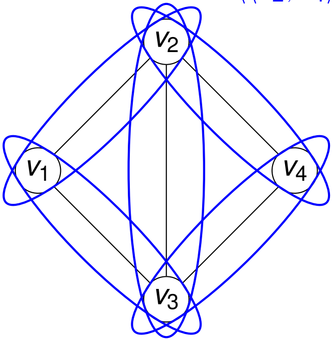
$c' = \langle\langle v_2, v_4\rangle, \phi'\rangle$

$$\phi'(x,y) = \begin{cases} 1 & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$

minimum total cost

3-Colouring Revisited

$c' = \langle\langle v_2, v_4\rangle, \phi'\rangle$

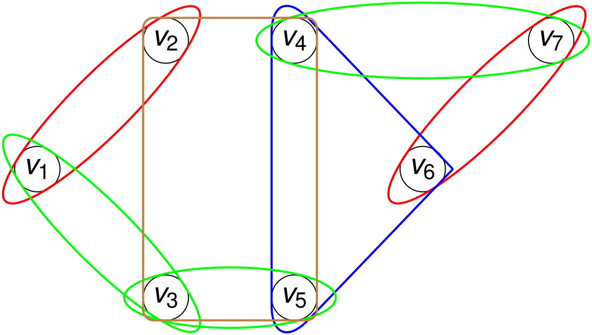


$$\phi'(x, y) = \begin{cases} 1 & \text{if } (x = y) \\ 0 & \text{otherwise} \end{cases}$$

minimum total cost $= k \Leftrightarrow$ there is a "3-colouring" with k mistakes and cannot be improved

"Hypergraph Min-Cost Colouring"

variables = vertices, coloured (typed) edges = constraints
goal: minimum total cost



Valued Constraint Satisfaction Problem

- VCSP $= \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints

Valued Constraint Satisfaction Problem

- $\text{VCSP} = \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables

Valued Constraint Satisfaction Problem

- $\text{VCSP} = \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables
- domain = finite set of values

Valued Constraint Satisfaction Problem

- VCSP $= \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables
- domain = finite set of values
- valued constraint = scope + cost function

Valued Constraint Satisfaction Problem

- VCSP $= \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables
- domain = finite set of values
- valued constraint = scope + cost function
- cost function of arity k is $\phi : D^k \to \mathbb{Q}_+ \cup \{\infty\}$
  (cost $\approx$ violation, the smaller the better)

Valued Constraint Satisfaction Problem

- VCSP $= \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables
- domain = finite set of values
- valued constraint = scope + cost function
- cost function of arity k is $\phi : D^k \rightarrow \mathbb{Q}_+ \cup \{\infty\}$
  (cost $\approx$ violation, the smaller the better)
- goal: assignment of variables with values from the domain which minimises the total cost/violation of all constraints
  (hard vs. soft constraints)

Valued Constraint Satisfaction Problem

- VCSP $= \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables
- domain = finite set of values
- valued constraint = scope + cost function
- cost function of arity k is $\phi : D^k \to \mathbb{Q}_+ \cup \{\infty\}$
  (cost $\approx$ violation, the smaller the better)
- goal: assignment of variables with values from the domain which minimises the total cost/violation of all constraints (hard vs. soft constraints)
- obviously a very general framework (NP-hard)

Valued Constraint Satisfaction Problem

- VCSP $= \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ - **V**ariables, **D**omain, **C**onstraints
- finite number of variables
- domain = finite set of values
- valued constraint = scope + cost function
- cost function of arity k is $\phi : D^k \rightarrow \mathbb{Q}_+ \cup \{\infty\}$
  (cost $\approx$ violation, the smaller the better)
- goal: assignment of variables with values from the domain which minimises the total cost/violation of all constraints (hard vs. soft constraints)
- obviously a very general framework (NP-hard)
- VCSP(L): all cost functions from L

VCSP(L)

VCSP(L) where L is

VCSP(L)

VCSP(L) where L is

- symmetric binary relation (sbr) = GRAPH HOMOMORPHISM

VCSP(L)

VCSP(L) where L is

- symmetric binary relation (sbr) = GRAPH HOMOMORPHISM
- sbr + unary relations = GRAPH LIST HOMOMORPHISM

VCSP(L)

VCSP(L) where L is

- symmetric binary relation (sbr) = GRAPH HOMOMORPHISM
- sbr + unary relations = GRAPH LIST HOMOMORPHISM
- sbr + unary cost functions = GRAPH MIN-COST HOMOMORPHISM

VCSP(L)

VCSP(L) where L is

- symmetric binary relation (sbr) = GRAPH HOMOMORPHISM
- sbr + unary relations = GRAPH LIST HOMOMORPHISM
- sbr + unary cost functions = GRAPH MIN-COST HOMOMORPHISM
- analogously for a single binary relation (DIGRAPH (LIST/MIN-COST) HOMOMORPHISM)

VCSP(L)

VCSP(L) where L is

- symmetric binary relation (sbr) = GRAPH HOMOMORPHISM
- sbr + unary relations = GRAPH LIST HOMOMORPHISM
- sbr + unary cost functions = GRAPH MIN-COST HOMOMORPHISM
- analogously for a single binary relation (DIGRAPH (LIST/MIN-COST) HOMOMORPHISM)
- costs $\{0, \infty\}$ (hard constraints) only = CSP

VCSP(L)

VCSP(L) where L is

- symmetric binary relation (sbr) = GRAPH HOMOMORPHISM
- sbr + unary relations = GRAPH LIST HOMOMORPHISM
- sbr + unary cost functions = GRAPH MIN-COST HOMOMORPHISM
- analogously for a single binary relation (DIGRAPH (LIST/MIN-COST) HOMOMORPHISM)
- costs $\{0, \infty\}$ (hard constraints) only = CSP
- costs $\{0, 1\}$ only = MAX-CSP

MIN-CUT instance: $\mathbf{G} = (V, E)$, $w : E \to \mathbb{Q}_+$

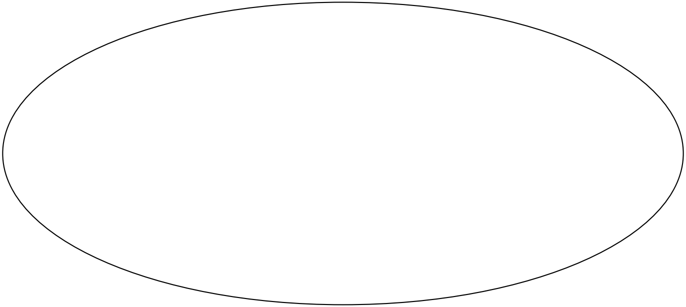
MIN-CUT

MIN-CUT instance: $\mathbf{G} = (V, E)$, $w : E \to \mathbb{Q}_+$
$D = \{0,1\}$, define: $\phi_w(x, y) = w$ if $(x, y) = (0,1)$ and 0 otherwise

MIN-CUT

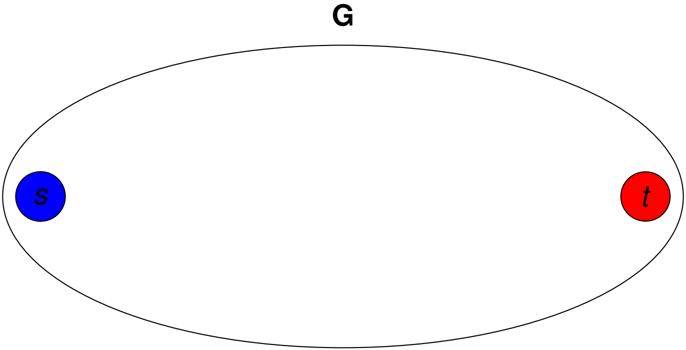
MIN-CUT instance: $\mathbf{G} = (V, E)$, $w : E \to \mathbb{Q}_+$
$D = \{0,1\}$, define: $\phi_w(x, y) = w$ if $(x, y) = (0,1)$ and 0 otherwise

G

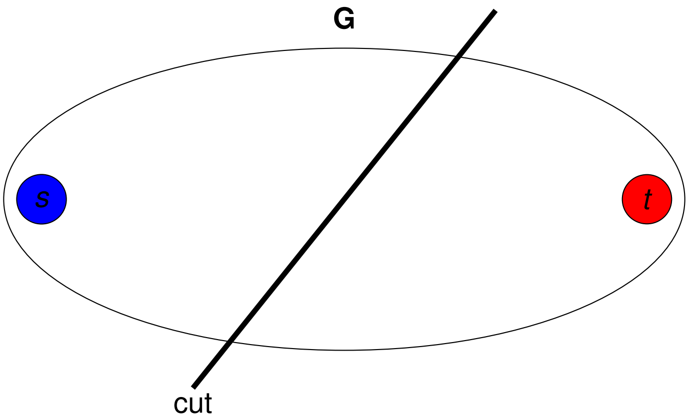


MIN-CUT

MIN-CUT instance: $\mathbf{G} = (V, E)$, $w : E \to \mathbb{Q}_+$
$D = \{0,1\}$, define: $\phi_w(x, y) = w$ if $(x, y) = (0,1)$ and 0 otherwise



MIN-CUT instance: $\mathbf{G} = (V, E)$, $w : E \to \mathbb{Q}_+$
$D = \{0,1\}$, define: $\phi_w(x, y) = w$ if $(x, y) = (0,1)$ and 0 otherwise

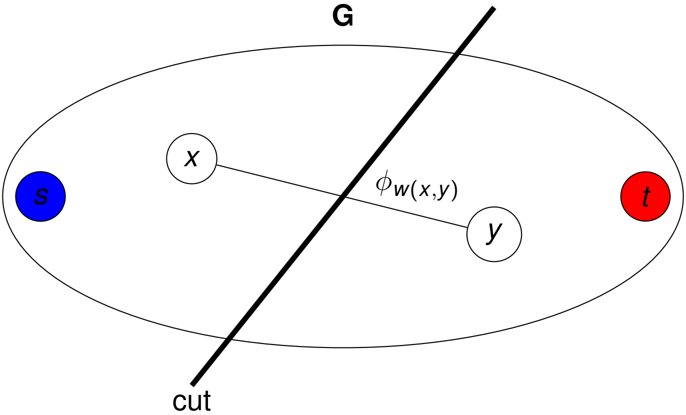


edges assigned $\langle 0, 1\rangle$ correspond to a cut

MIN-CUT

MIN-CUT instance: $\mathbf{G} = (V, E)$, $w : E \to \mathbb{Q}_+$

$D = \{0,1\}$, define: $\phi_w(x, y) = w$ if $(x, y) = (0,1)$ and 0 otherwise



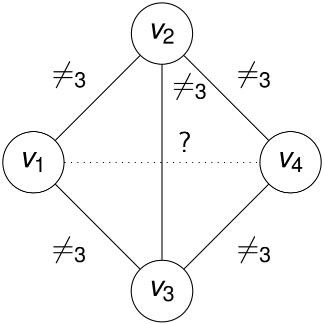
edges assigned $\langle 0, 1\rangle$ correspond to a cut

Expressibility Example

values for variables listed in the scope of a constraint are constrained **explicitly**; due to the combined effect of constraints, **any** subset of variables is constrained **implicitly**

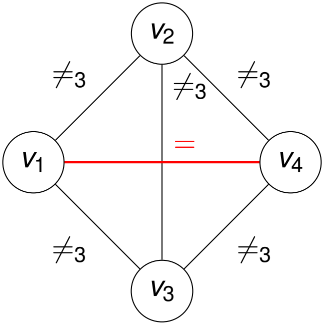
Expressibility Example

values for variables listed in the scope of a constraint are constrained **explicitly**; due to the combined effect of constraints, **any** subset of variables is constrained **implicitly**



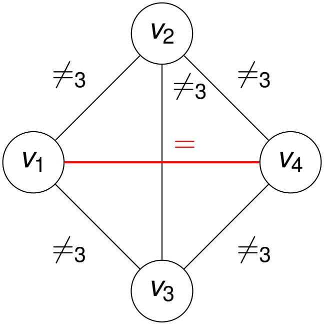
Expressibility Example

values for variables listed in the scope of a constraint are constrained **explicitly**; due to the combined effect of constraints, **any** subset of variables is constrained **implicitly**



Expressibility Example

values for variables listed in the scope of a constraint are constrained **explicitly**; due to the combined effect of constraints, **any** subset of variables is constrained **implicitly**



cost function $=$ is **expressible** over $\{\neq_3\}$

Expressibility

- $\mathcal{I} = \langle V, D, C \rangle$, $s : V \rightarrow D$, **cost** of s is

$$Cost_{\mathcal{I}}(s) = \sum_{\langle \langle v_1, v_2, \ldots, v_m \rangle, \phi \rangle \in C} \phi(\langle s(v_1), s(v_2), \ldots, s(v_m) \rangle)$$

Expressibility

- $\mathcal{I} = \langle V, D, C \rangle$, $s : V \to D$, **cost** of s is

$$Cost_{\mathcal{I}}(s) = \sum_{\langle\langle v_1, v_2, \ldots, v_m \rangle, \phi\rangle \in C} \phi(\langle s(v_1), s(v_2), \ldots, s(v_m) \rangle)$$

- $l = \langle v_1, \ldots, v_m \rangle$, **projection** of $\mathcal{I}$ onto l is

$$\pi_l(\mathcal{I})(x_1, \ldots, x_m) \ = \ \min_{s \in S} \mathrm{Cost}_{\mathcal{P}}(s)$$

where $S = \{s : V \to D \mid \langle s(v_1), \ldots, s(v_m) \rangle = \langle x_1, \ldots, x_m \rangle\}$

Expressibility

- $\mathcal{I} = \langle V, D, C\rangle$, $s : V \to D$, **cost** of s is

$$Cost_{\mathcal{I}}(s) = \sum_{\langle\langle v_1, v_2, \ldots, v_m\rangle, \phi\rangle \in C} \phi(\langle s(v_1), s(v_2), \ldots, s(v_m)\rangle)$$

- $l = \langle v_1, \ldots, v_m\rangle$, **projection** of $\mathcal{I}$ onto l is

$$\pi_l(\mathcal{I})(x_1, \ldots, x_m) \;=\; \min_{s \in S} \mathrm{Cost}_{\mathcal{P}}(s)$$

  where $S = \{s : V \to D \mid \langle s(v_1), \ldots, s(v_m)\rangle = \langle x_1, \ldots, x_m\rangle\}$

- ϕ **expressible** over $L \;\Leftrightarrow\; \phi = \pi_l(\mathcal{I})$ for some $\mathcal{I} \in \mathrm{VCSP}(L)$

Expressibility

- $\mathcal{I} = \langle V, D, C \rangle$, $s : V \to D$, **cost** of s is

$$Cost_{\mathcal{I}}(s) = \sum_{\langle \langle v_1, v_2, \ldots, v_m \rangle, \phi \rangle \in C} \phi(\langle s(v_1), s(v_2), \ldots, s(v_m) \rangle)$$

- $I = \langle v_1, \ldots, v_m \rangle$, **projection** of $\mathcal{I}$ onto I is

$$\pi_I(\mathcal{I})(x_1, \ldots, x_m) \; = \; \min_{s \in S} \mathrm{Cost}_{\mathcal{P}}(s)$$

where $S = \{s : V \to D \mid \langle s(v_1), \ldots, s(v_m) \rangle = \langle x_1, \ldots, x_m \rangle\}$

- ϕ **expressible** over $L \;\Leftrightarrow\; \phi = \pi_I(\mathcal{I})$ for some $\mathcal{I} \in \mathrm{VCSP}(L)$
- in CSP this corresponds to $\exists$,$\wedge$ (primitive positive formulae)

Expressive Power

- **expressive power** of L, $\langle L \rangle$, are cost functions expressible over L up to additive constant and scaling

Expressive Power

- **expressive power** of L, $\langle L \rangle$, are cost functions expressible over L up to additive constant and scaling
- L and $\langle L \rangle$ have the same complexity (log-space equiv.)

Expressive Power

- **expressive power** of L, $\langle L \rangle$, are cost functions expressible over L up to additive constant and scaling
- L and $\langle L \rangle$ have the same complexity (log-space equiv.)
- $\langle L \rangle$ is characterised by a nontrivial algebraic property called **fractional polymorphism**

Expressive Power

- **expressive power** of L, $\langle L \rangle$, are cost functions expressible over L up to additive constant and scaling
- L and $\langle L \rangle$ have the same complexity (log-space equiv.)
- $\langle L \rangle$ is characterised by a nontrivial algebraic property called **fractional polymorphism**
- CSP: Galois connection between sets of relations and sets of polymorphisms

Fractional Polymorphism Example

$F = \{f_1 = \langle 0.8, \textsc{Min} \rangle, f_2 = \langle 0.7, \textsc{Max} \rangle, f_3 = \langle 0.5, \textsc{Const} \rangle\}$ is a **fractional polymorphism** of $\phi(x, y, z) = x + 2y + 3z$ over $D = \{1, \ldots, 5\}$

E.g.

$$\begin{array}{lcccl} t_1 & \langle 1,2,3\rangle & \xrightarrow{\phi} & 14 & \\ t_2 & \langle 5,1,2\rangle & & 13 & \left.\right\} \sum = 27 \\ \hline & & & & \geq \\ f_1(t_1,t_2) & \langle 1,1,2\rangle & & 0.8 * 9 & \\ f_2(t_1,t_2) & \langle 5,2,3\rangle & \xrightarrow{\phi} & 0.7 * 18 & \left.\right\} \sum = 22.8 \\ f_3(t_1,t_2) & \langle 1,1,1\rangle & & 0.5 * 6 & \end{array}$$

Multimorphisms

- in all known cases (dichotomy for Boolean VCSP, tractable classes of MAX-CSP, tractable class of VCSP defined by a tournament pair, separation for finite-valued max-closed cost functions,...) a more restrictive form of fractional polymorphisms is used

Multimorphisms

- in all known cases (dichotomy for Boolean VCSP, tractable classes of MAX-CSP, tractable class of VCSP defined by a tournament pair, separation for finite-valued max-closed cost functions,. . .) a more restrictive form of fractional polymorphisms is used
- **multimorphisms**, mappings from D^k to D^k with natural weights (as opposed to mappings from D^k to D^n with fractional weights)

Multimorphism Example

$F = \{\textsc{Min}, \textsc{Max}\}$ is a **multimorphism** of
$\phi(x, y, z) = x + 2y + 3z$ over $D = \{1, \ldots, 5\}$
E.g.

$$\begin{array}{lcccl}
t_1 & \langle 1,2,3 \rangle & \xrightarrow{\phi} & 14 & \\
t_2 & \langle 5,1,2 \rangle & \xrightarrow{\phi} & 13 & \Big\} \sum = 27 \\
 & & & & \quad\;\; \geq \\
\textsc{Min}(t_1, t_2) & \langle 1,1,2 \rangle & \xrightarrow{\phi} & 9 & \\
\textsc{Max}(t_1, t_2) & \langle 5,2,3 \rangle & \xrightarrow{\phi} & 18 & \Big\} \sum = 27
\end{array}$$

Open Problems

- is it possible to characterise the expressive power of valued constraints in terms of multimorphisms only?

Open Problems

- is it possible to characterise the expressive power of valued constraints in terms of multimorphisms only?
- what is the simplest algebraic property which characterise the expressive power of valued constraints?

Open Problems

- is it possible to characterise the expressive power of valued constraints in terms of multimorphisms only?
- what is the simplest algebraic property which characterise the expressive power of valued constraints?
- what are the closure operations of this property which gives a Galois connection (between sets of cost functions and sets of algebraic properties)?

Fixed-Arity Languages Def

Definition

For every domain size $d \geq 2$ we define the following:

Fixed-Arity Languages Def

Definition

For every domain size $d \geq 2$ we define the following:

- $\mathbf{R}_{d,m}$ denotes the set of all relations of arity at most m over a domain of size d

Fixed-Arity Languages Def

Definition

For every domain size $d \geq 2$ we define the following:

- $\mathbf{R}_{d,m}$ denotes the set of all relations of arity at most m over a domain of size d
- $\mathbf{F}_{d,m}$ denotes the set of all finite-valued cost functions of arity at most m over a domain of size d

Fixed-Arity Languages Def

Definition

For every domain size $d \geq 2$ we define the following:

- $\mathbf{R}_{d,m}$ denotes the set of all relations of arity at most m over a domain of size d
- $\mathbf{F}_{d,m}$ denotes the set of all finite-valued cost functions of arity at most m over a domain of size d
- $\mathbf{G}_{d,m}$ denotes the set of all general cost functions of arity at most m over a domain of size d

Fixed-Arity Languages Def

Definition

For every domain size $d \geq 2$ we define the following:

- $\mathbf{R}_{d,m}$ denotes the set of all relations of arity at most m over a domain of size d
- $\mathbf{F}_{d,m}$ denotes the set of all finite-valued cost functions of arity at most m over a domain of size d
- $\mathbf{G}_{d,m}$ denotes the set of all general cost functions of arity at most m over a domain of size d
- $\mathbf{R}_d = \cup_{m \geq 0} \mathbf{R}_{d,m}$ $\mathbf{F}_d = \cup_{m \geq 0} \mathbf{F}_{d,m}$ $\mathbf{G}_d = \cup_{m \geq 0} \mathbf{G}_{d,m}$

Fixed-Arity Languages Results

Cohen, Jeavons, Živný 2007 [1]

For all $d \geq 3$, and $f \geq 2$,

- $\langle \mathbf{R}_{2,1} \rangle \subsetneq \langle \mathbf{R}_{2,2} \rangle \subsetneq \langle \mathbf{R}_{2,3} \rangle = \mathbf{R}_2$
- $\langle \mathbf{R}_{d,1} \rangle \subsetneq \langle \mathbf{R}_{d,2} \rangle = \mathbf{R}_d$
- $\langle \mathbf{F}_{f,1} \rangle \subsetneq \langle \mathbf{F}_{f,2} \rangle = \mathbf{F}_f$
- $\langle \mathbf{G}_{f,1} \rangle \subsetneq \langle \mathbf{G}_{f,2} \rangle = \mathbf{G}_f$

Fixed-Arity Languages Proofs

For all $d \geq 3$, and $f \geq 2$,

- $\langle \mathbf{R}_{2,1} \rangle \subsetneq \langle \mathbf{R}_{2,2} \rangle \subsetneq \langle \mathbf{R}_{2,3} \rangle = \mathbf{R}_2$
  standard SAT to 3-SAT reduction

Fixed-Arity Languages Proofs

For all $d \geq 3$, and $f \geq 2$,

- $\langle \mathbf{R}_{2,1} \rangle \subsetneq \langle \mathbf{R}_{2,2} \rangle \subsetneq \langle \mathbf{R}_{2,3} \rangle = \mathbf{R}_2$
  standard SAT to 3-SAT reduction
- $\langle \mathbf{R}_{d,1} \rangle \subsetneq \langle \mathbf{R}_{d,2} \rangle = \mathbf{R}_d$
  adapting the reduction to multi-valued logic

Fixed-Arity Languages Proofs

For all $d \geq 3$, and $f \geq 2$,

- $\langle \mathbf{R}_{2,1} \rangle \subsetneq \langle \mathbf{R}_{2,2} \rangle \subsetneq \langle \mathbf{R}_{2,3} \rangle = \mathbf{R}_2$
  standard SAT to 3-SAT reduction
- $\langle \mathbf{R}_{d,1} \rangle \subsetneq \langle \mathbf{R}_{d,2} \rangle = \mathbf{R}_d$
  adapting the reduction to multi-valued logic
- $\langle \mathbf{F}_{f,1} \rangle \subsetneq \langle \mathbf{F}_{f,2} \rangle = \mathbf{F}_f$, $\langle \mathbf{G}_{f,1} \rangle \subsetneq \langle \mathbf{G}_{f,2} \rangle = \mathbf{G}_f$
  explicit gadgets

Fixed-Arity Max-Closed Languages Def

Definition

$\phi : D^k \to \mathbb{Q}_+ \cup \{\infty\}$ is **max-closed** $\Leftrightarrow$ for every $u, v \in D^k$

$$u \leq_{coordinatewise} v \Rightarrow \phi(u) \geq \phi(v)$$

(equivalently, $\langle \textsc{Max}, \textsc{Max} \rangle$ is a multimorphism of ϕ)

Fixed-Arity Max-Closed Languages Def

Definition

$\phi : D^k \to \mathbb{Q}_+ \cup \{\infty\}$ is **max-closed** $\Leftrightarrow$ for every $u, v \in D^k$

$$u \leq_{coordinatewise} v \Rightarrow \phi(u) \geq \phi(v)$$

(equivalently, $\langle \textsc{Max}, \textsc{Max} \rangle$ is a multimorphism of ϕ)

Definition

For every domain size $d \geq 2$ we define:

Fixed-Arity Max-Closed Languages Def

Definition

$\phi : D^k \to \mathbb{Q}_+ \cup \{\infty\}$ is **max-closed** $\Leftrightarrow$ for every $u, v \in D^k$

$$u \leq_{coordinatewise} v \Rightarrow \phi(u) \geq \phi(v)$$

(equivalently, $\langle \textsc{Max}, \textsc{Max} \rangle$ is a multimorphism of ϕ)

Definition

For every domain size $d \geq 2$ we define:

- $\mathbf{R}^{\max}_{d,m}$, $\mathbf{F}^{\max}_{d,m}$ and $\mathbf{G}^{\max}_{d,m}$ analogously to $\mathbf{R}_{d,m}$, $\mathbf{F}_{d,m}$ and $\mathbf{G}_{d,m}$, but for max-closed cost functions only

Fixed-Arity Max-Closed Languages Def

Definition

$\phi : D^k \to \mathbb{Q}_+ \cup \{\infty\}$ is **max-closed** $\Leftrightarrow$ for every $u, v \in D^k$

$$u \leq_{coordinatewise} v \Rightarrow \phi(u) \geq \phi(v)$$

(equivalently, $\langle \text{MAX}, \text{MAX} \rangle$ is a multimorphism of ϕ)

Definition

For every domain size $d \geq 2$ we define:

- $\mathbf{R}^{\max}_{d,m}$, $\mathbf{F}^{\max}_{d,m}$ and $\mathbf{G}^{\max}_{d,m}$ analogously to $\mathbf{R}_{d,m}$, $\mathbf{F}_{d,m}$ and $\mathbf{G}_{d,m}$, but for max-closed cost functions only
- $\mathbf{R}^{\max}_{d} = \cup_{m \geq 0} \mathbf{R}^{\max}_{d,m}$ $\quad \mathbf{F}^{\max}_{d} = \cup_{m \geq 0} \mathbf{F}^{\max}_{d,m}$ $\quad \mathbf{G}^{\max}_{d} = \cup_{m \geq 0} \mathbf{G}^{\max}_{d,m}$

Fixed-Arity Max-Closed Languages Results

Cohen, Jeavons, Živný 2007 [1]

For all $d \geq 3$ and $f \geq 2$,

- $\langle \mathbf{R}^{\text{max}}_{2,1} \rangle \subsetneq \langle \mathbf{R}^{\text{max}}_{2,2} \rangle \subsetneq \langle \mathbf{R}^{\text{max}}_{2,3} \rangle = \mathbf{R}^{\text{max}}_{2}$
- $\langle \mathbf{R}^{\text{max}}_{d,1} \rangle \subsetneq \langle \mathbf{R}^{\text{max}}_{d,2} \rangle = \mathbf{R}^{\text{max}}_{d}$
- $\langle \mathbf{F}^{\text{max}}_{f,1} \rangle \subsetneq \langle \mathbf{F}^{\text{max}}_{f,2} \rangle \subsetneq \langle \mathbf{F}^{\text{max}}_{f,3} \rangle \subsetneq \langle \mathbf{F}^{\text{max}}_{f,4} \rangle \cdots$
- $\langle \mathbf{G}^{\text{max}}_{2,1} \rangle \subsetneq \langle \mathbf{G}^{\text{max}}_{2,2} \rangle \subsetneq \langle \mathbf{G}^{\text{max}}_{2,3} \rangle = \mathbf{G}^{\text{max}}_{2}$
- $\langle \mathbf{G}^{\text{max}}_{d,1} \rangle \subsetneq \langle \mathbf{G}^{\text{max}}_{d,2} \rangle = \mathbf{G}^{\text{max}}_{d}$

Fixed-Arity Max-Closed Languages Proofs

For all $d \geq 3$ and $f \geq 2$,

- $\langle \mathbf{R}^{\max}_{2,1} \rangle \subsetneq \langle \mathbf{R}^{\max}_{2,2} \rangle \subsetneq \langle \mathbf{R}^{\max}_{2,3} \rangle = \mathbf{R}^{\max}_{2}$
  HORN 3-SAT

Fixed-Arity Max-Closed Languages Proofs

For all $d \geq 3$ and $f \geq 2$,

- $\langle \mathbf{R}^{\max}_{2,1} \rangle \subsetneq \langle \mathbf{R}^{\max}_{2,2} \rangle \subsetneq \langle \mathbf{R}^{\max}_{2,3} \rangle = \mathbf{R}^{\max}_{2}$
  HORN 3-SAT
- $\langle \mathbf{R}^{\max}_{d,1} \rangle \subsetneq \langle \mathbf{R}^{\max}_{d,2} \rangle = \mathbf{R}^{\max}_{d}$
  polymorphisms

Fixed-Arity Max-Closed Languages Proofs

For all $d \geq 3$ and $f \geq 2$,

- $\langle \mathbf{R}_{2,1}^{\max} \rangle \subsetneq \langle \mathbf{R}_{2,2}^{\max} \rangle \subsetneq \langle \mathbf{R}_{2,3}^{\max} \rangle = \mathbf{R}_{2}^{\max}$
  HORN 3-SAT
- $\langle \mathbf{R}_{d,1}^{\max} \rangle \subsetneq \langle \mathbf{R}_{d,2}^{\max} \rangle = \mathbf{R}_{d}^{\max}$
  polymorphisms
- $\langle \mathbf{F}_{f,1}^{\max} \rangle \subsetneq \langle \mathbf{F}_{f,2}^{\max} \rangle \subsetneq \langle \mathbf{F}_{f,3}^{\max} \rangle \subsetneq \langle \mathbf{F}_{f,4}^{\max} \rangle \cdots$
  separating multimorphisms
  (1. separation; 2. playground for multimorphism)

Fixed-Arity Max-Closed Languages Proofs

For all $d \geq 3$ and $f \geq 2$,

- $\langle \mathbf{R}^{\max}_{2,1} \rangle \subsetneq \langle \mathbf{R}^{\max}_{2,2} \rangle \subsetneq \langle \mathbf{R}^{\max}_{2,3} \rangle = \mathbf{R}^{\max}_{2}$
  HORN 3-SAT
- $\langle \mathbf{R}^{\max}_{d,1} \rangle \subsetneq \langle \mathbf{R}^{\max}_{d,2} \rangle = \mathbf{R}^{\max}_{d}$
  polymorphisms
- $\langle \mathbf{F}^{\max}_{f,1} \rangle \subsetneq \langle \mathbf{F}^{\max}_{f,2} \rangle \subsetneq \langle \mathbf{F}^{\max}_{f,3} \rangle \subsetneq \langle \mathbf{F}^{\max}_{f,4} \rangle \cdots$
  separating multimorphisms
  (1. separation; 2. playground for multimorphism)
- $\langle \mathbf{G}^{\max}_{2,1} \rangle \subsetneq \langle \mathbf{G}^{\max}_{2,2} \rangle \subsetneq \langle \mathbf{G}^{\max}_{2,3} \rangle = \mathbf{G}^{\max}_{2}$
  $\langle \mathbf{G}^{\max}_{d,1} \rangle \subsetneq \langle \mathbf{G}^{\max}_{d,2} \rangle = \mathbf{G}^{\max}_{d}$
  fractional polymorphisms (MIN-CUT MAX-FLOW)

Submodular Function Minimisation

- let V, $|V| = n$, be a finite set and ϕ a set function on subset of V, $\phi : 2^V \to \mathbb{Q}_+ \cup \{\infty\}$

Submodular Function Minimisation

- let V, $|V| = n$, be a finite set and ϕ a set function on subset of V, $\phi : 2^V \rightarrow \mathbb{Q}_+ \cup \{\infty\}$
- finding $U \subseteq V$ which minimises ϕ is NP-hard in general

Submodular Function Minimisation

- let V, $|V| = n$, be a finite set and ϕ a set function on subset of V, $\phi : 2^V \rightarrow \mathbb{Q}_+ \cup \{\infty\}$
- finding $U \subseteq V$ which minimises ϕ is NP-hard in general
- if ϕ is submodular, that is, for all $S, T \subseteq V$,

$$\phi(S \cup T) + \phi(S \cap T) \leq \phi(S) + \phi(T),$$

  we can find U in $O(n^6)$; this is a a central problem in discrete optimisation (SUBMODULAR FUNCTION MINIMISATION (SFM))

Submodular Function Minimisation

- let V, $|V| = n$, be a finite set and ϕ a set function on subset of V, $\phi : 2^V \to \mathbb{Q}_+ \cup \{\infty\}$
- finding $U \subseteq V$ which minimises ϕ is NP-hard in general
- if ϕ is submodular, that is, for all $S, T \subseteq V$,

$$\phi(S \cup T) + \phi(S \cap T) \leq \phi(S) + \phi(T),$$

  we can find U in $O(n^6)$; this is a a central problem in discrete optimisation (SUBMODULAR FUNCTION MINIMISATION (SFM))
- special cases of SFM are solvable in cubic time by reducing to MIN-CUT

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?
- can valued constraints of MIN-CUT express any submodular function?

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?
- can valued constraints of MIN-CUT express any submodular function?
- YES, in case of relations (costs $\{0, \infty\}$ only)

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?
- can valued constraints of MIN-CUT express any submodular function?
- YES, in case of relations (costs $\{0, \infty\}$ only)
- in case of finite-valued cost functions:

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?
- can valued constraints of MIN-CUT express any submodular function?
- YES, in case of relations (costs $\{0, \infty\}$ only)
- in case of finite-valued cost functions:
  - if YES, a faster algorithm for SFM of bounded arity

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?
- can valued constraints of MIN-CUT express any submodular function?
- YES, in case of relations (costs $\{0, \infty\}$ only)
- in case of finite-valued cost functions:
  - if YES, a faster algorithm for SFM of bounded arity
  - if NO, indirect evidence that the general SFM problem is more difficult than the special cases

SFM vs. MIN-CUT

- can the MIN-CUT algorithm solve SFM?
- can valued constraints of MIN-CUT express any submodular function?
- YES, in case of relations (costs $\{0, \infty\}$ only)
- in case of finite-valued cost functions:
  - if YES, a faster algorithm for SFM of bounded arity
  - if NO, indirect evidence that the general SFM problem is more difficult than the special cases
- to answer the question, it is worthy studying the algebraic properties of submodular functions which characterise the expressive power

Details

stanislav.zivny@comlab.ox.ac.uk

http://zivny.cz

[1] D. A. Cohen, P. G. Jeavons, S. Živný, The expressive power of valued constraints: Hierarchies and collapses, In Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming (CP 2007), RI, USA, 2007.