Rounding rules and vague solutions to bounded width CSPs

Zarathustra Brady

CSPs and relational structures

- If $\mathbf{A} = (A, R_1, ..., R_n)$ is a finite relational structure, we get an associated Constraint Satisfaction Problem, CSP($\mathbf{A}$).

CSPs and relational structures

- If $\mathbf{A} = (A, R_1, ..., R_n)$ is a finite relational structure, we get an associated Constraint Satisfaction Problem, CSP($\mathbf{A}$).

- An *instance* $\mathbf{X} = (X, C_1, ..., C_n)$ of CSP($\mathbf{A}$) is a relational structure with the same signature as $\mathbf{A}$.

CSPs and relational structures

- If $\mathbf{A} = (A, R_1, ..., R_n)$ is a finite relational structure, we get an associated Constraint Satisfaction Problem, CSP($\mathbf{A}$).
- An *instance* $\mathbf{X} = (X, C_1, ..., C_n)$ of CSP($\mathbf{A}$) is a relational structure with the same signature as $\mathbf{A}$.
- The elements $x \in X$ are called the *variables* of the instance $\mathbf{X}$.

CSPs and relational structures

- If $\mathbf{A} = (A, R_1, ..., R_n)$ is a finite relational structure, we get an associated Constraint Satisfaction Problem, CSP($\mathbf{A}$).

- An *instance* $\mathbf{X} = (X, C_1, ..., C_n)$ of CSP($\mathbf{A}$) is a relational structure with the same signature as $\mathbf{A}$.

- The elements $x \in X$ are called the *variables* of the instance $\mathbf{X}$.
- The elements $c = (x_1, ..., x_k) \in C_i$ are called the *constraints* of the instance $\mathbf{X}$.

CSPs and relational structures

- If $\mathbf{A} = (A, R_1, ..., R_n)$ is a finite relational structure, we get an associated Constraint Satisfaction Problem, CSP($\mathbf{A}$).

- An *instance* $\mathbf{X} = (X, C_1, ..., C_n)$ of CSP($\mathbf{A}$) is a relational structure with the same signature as $\mathbf{A}$.

- The elements $x \in X$ are called the *variables* of the instance $\mathbf{X}$.
- The elements $c = (x_1, ..., x_k) \in C_i$ are called the *constraints* of the instance $\mathbf{X}$.
- A *solution* to the instance $\mathbf{X}$ of CSP($\mathbf{A}$) is a homomorphism $a : \mathbf{X} \to \mathbf{A}$.

CSPs and relational structures

- If $\mathbf{A} = (A, R_1, ..., R_n)$ is a finite relational structure, we get an associated Constraint Satisfaction Problem, CSP($\mathbf{A}$).

- An *instance* $\mathbf{X} = (X, C_1, ..., C_n)$ of CSP($\mathbf{A}$) is a relational structure with the same signature as $\mathbf{A}$.

- The elements $x \in X$ are called the *variables* of the instance $\mathbf{X}$.
- The elements $c = (x_1, ..., x_k) \in C_i$ are called the *constraints* of the instance $\mathbf{X}$.
- A *solution* to the instance $\mathbf{X}$ of CSP($\mathbf{A}$) is a homomorphism $a : \mathbf{X} \to \mathbf{A}$.

- If $c = (x_1, ..., x_k) \in C_i$ is a constraint, then a solution a must satisfy $(a(x_1), ..., a(x_k)) \in R_i$.

Approximate solutions

- An *approximate solution* to the instance $\mathbf{X}$ is an arbitrary map $a : X \to A$.

Approximate solutions

- An *approximate solution* to the instance **X** is an arbitrary map $a : X \to A$.
- The *value* of the approximate solution a is the fraction of the constraints which are satisfied by a:

$$\frac{\sum_i \#\{(x_1, \dots, x_k) \in C_i \mid (a(x_1), \dots, a(x_k)) \in R_i\}}{\sum_i |C_i|}.$$

Approximate solutions

- An *approximate solution* to the instance $\mathbf{X}$ is an arbitrary map $a : X \to A$.
- The *value* of the approximate solution a is the fraction of the constraints which are satisfied by a:

$$\frac{\sum_i \#\{(x_1, \ldots, x_k) \in C_i \mid (a(x_1), \ldots, a(x_k)) \in R_i\}}{\sum_i |C_i|}.$$

- The *value* of the instance $\mathbf{X}$ is the maximum value of any approximate solution $a : X \to A$.

Approximate solutions

- An *approximate solution* to the instance **X** is an arbitrary map $a : X \to A$.
- The *value* of the approximate solution a is the fraction of the constraints which are satisfied by a:

$$\frac{\sum_i \#\{(x_1, ..., x_k) \in C_i \mid (a(x_1), ..., a(x_k)) \in R_i\}}{\sum_i |C_i|}.$$

- The *value* of the instance **X** is the maximum value of any approximate solution $a : X \to A$.
- An approximate solution with value 1 is the same thing as an ordinary solution.

Robustly solvable CSPs

Definition

We say that CSP($\mathbf{A}$) is *robustly solvable* if there is a function $f : [0,1] \to [0,1]$ such that:

Robustly solvable CSPs

Definition

We say that CSP(**A**) is *robustly solvable* if there is a function $f : [0,1] \to [0,1]$ such that:

- when **X** is an instance of value $1 - \epsilon$, we can algorithmically find an approximate solution $a : X \to A$ of value $1 - f(\epsilon)$ in polynomial time,

Robustly solvable CSPs

Definition

We say that CSP(**A**) is *robustly solvable* if there is a function $f : [0,1] \to [0,1]$ such that:

- when **X** is an instance of value $1 - \epsilon$, we can algorithmically find an approximate solution $a : X \to A$ of value $1 - f(\epsilon)$ in polynomial time,
- $\lim_{\epsilon \to 0} f(\epsilon) = 0$.

Robustly solvable CSPs

Definition

We say that CSP(**A**) is *robustly solvable* if there is a function $f : [0,1] \to [0,1]$ such that:

- when **X** is an instance of value $1 - \epsilon$, we can algorithmically find an approximate solution $a : X \to A$ of value $1 - f(\epsilon)$ in polynomial time,
- $\lim_{\epsilon \to 0} f(\epsilon) = 0$.

- The main barrier to being robustly solvable is the ability to simulate *affine* CSPs.

Robustly solvable CSPs

Definition

We say that CSP(**A**) is *robustly solvable* if there is a function $f : [0, 1] \to [0, 1]$ such that:

- when **X** is an instance of value $1 - \epsilon$, we can algorithmically find an approximate solution $a : X \to A$ of value $1 - f(\epsilon)$ in polynomial time,
- $\lim_{\epsilon \to 0} f(\epsilon) = 0$.

- The main barrier to being robustly solvable is the ability to simulate *affine* CSPs.

Theorem (Håstad)

If $\mathbf{A} = (\mathbb{Z}/p, \{x + y = z\}, ..., \{x + y = z + p - 1\})$*, then it is NP-hard to find an approximate solution* $a : X \to A$ *of value* $\frac{1}{p} + \epsilon$*, even if the instance* **X** *is promised to have value* $1 - \epsilon$*.*

Characterization of robustly solvable CSPs

- We say that a relational structure $\mathbf{A}$ has *bounded width* if $\text{CSP}(\mathbf{A})$ can be solved by a local consistency algorithm.

Characterization of robustly solvable CSPs

- We say that a relational structure **A** has *bounded width* if CSP(**A**) can be solved by a local consistency algorithm.

Theorem (Conjectured by Guruswami and Zhou, proved by Barto and Kozik)

If $P \neq NP$, the following are equivalent:

Characterization of robustly solvable CSPs

- We say that a relational structure $\mathbf{A}$ has *bounded width* if CSP($\mathbf{A}$) can be solved by a local consistency algorithm.

Theorem (Conjectured by Guruswami and Zhou, proved by Barto and Kozik)

If $P \neq NP$, the following are equivalent:

- CSP($\mathbf{A}$) *is robustly solvable,*

Characterization of robustly solvable CSPs

- We say that a relational structure **A** has *bounded width* if CSP(**A**) can be solved by a local consistency algorithm.

Theorem (Conjectured by Guruswami and Zhou, proved by Barto and Kozik)

If $P \neq NP$, the following are equivalent:

- CSP(**A**) *is robustly solvable,*
- **A** *has bounded width,*

Characterization of robustly solvable CSPs

- We say that a relational structure **A** has *bounded width* if CSP(**A**) can be solved by a local consistency algorithm.

Theorem (Conjectured by Guruswami and Zhou, proved by Barto and Kozik)

If $P \neq NP$, the following are equivalent:

- CSP(**A**) *is robustly solvable,*
- **A** *has bounded width,*
- CSP(**A**) *can be robustly solved via the standard semidefinite programming relaxation.*

Characterization of robustly solvable CSPs

- We say that a relational structure **A** has *bounded width* if CSP(**A**) can be solved by a local consistency algorithm.

Theorem (Conjectured by Guruswami and Zhou, proved by Barto and Kozik)

If $P \neq NP$, the following are equivalent:

- CSP(**A**) *is robustly solvable,*
- **A** *has bounded width,*
- CSP(**A**) *can be robustly solved via the standard semidefinite programming relaxation.*

- Furthermore, Barto and Kozik's algorithm has

$$f(\epsilon) \ll \frac{\log\log(1/\epsilon)}{\log(1/\epsilon)}.$$

Bounded width

- For any finite relational structure $\mathbf{A}$, the computational complexity of $\mathrm{CSP}(\mathbf{A})$ is controlled by the set of polymorphisms $\mathrm{Pol}(\mathbf{A})$.

Bounded width

- For any finite relational structure $\mathbf{A}$, the computational complexity of $\mathrm{CSP}(\mathbf{A})$ is controlled by the set of polymorphisms $\mathrm{Pol}(\mathbf{A})$.

Theorem (Bulatov, Barto, Kozik)

If $\mathbf{A}$ is a finite core relational structure, and if $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ is the corresponding algebraic structure, then TFAE:

Bounded width

- For any finite relational structure **A**, the computational complexity of CSP(**A**) is controlled by the set of polymorphisms Pol(**A**).

Theorem (Bulatov, Barto, Kozik)

If **A** *is a finite core relational structure, and if* $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ *is the corresponding algebraic structure, then TFAE:*

- CSP(**A**) *can be solved by a local consistency algorithm,*

Bounded width

- For any finite relational structure $\mathbf{A}$, the computational complexity of $\mathrm{CSP}(\mathbf{A})$ is controlled by the set of polymorphisms $\mathrm{Pol}(\mathbf{A})$.

Theorem (Bulatov, Barto, Kozik)

If $\mathbf{A}$ *is a finite core relational structure, and if* $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ *is the corresponding algebraic structure, then TFAE:*

- $\mathrm{CSP}(\mathbf{A})$ *can be solved by a local consistency algorithm,*
- *The variety* $\mathrm{Var}(\mathbb{A})$ *generated by* $\mathbb{A}$ *contains no nontrivial quasi-affine algebras,*

Bounded width

- For any finite relational structure $\mathbf{A}$, the computational complexity of $\mathrm{CSP}(\mathbf{A})$ is controlled by the set of polymorphisms $\mathrm{Pol}(\mathbf{A})$.

Theorem (Bulatov, Barto, Kozik)

If $\mathbf{A}$ *is a finite core relational structure, and if* $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ *is the corresponding algebraic structure, then TFAE:*

- $\mathrm{CSP}(\mathbf{A})$ *can be solved by a local consistency algorithm,*
- *The variety* $\mathrm{Var}(\mathbb{A})$ *generated by* $\mathbb{A}$ *contains no nontrivial quasi-affine algebras,*
- $\mathrm{Var}(\mathbb{A})$ *is congruence meet-semidistributive,*

Bounded width

- For any finite relational structure $\mathbf{A}$, the computational complexity of $\mathrm{CSP}(\mathbf{A})$ is controlled by the set of polymorphisms $\mathrm{Pol}(\mathbf{A})$.

Theorem (Bulatov, Barto, Kozik)

If $\mathbf{A}$ is a finite core relational structure, and if $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ is the corresponding algebraic structure, then TFAE:

- $\mathrm{CSP}(\mathbf{A})$ *can be solved by a local consistency algorithm,*
- *The variety* $\mathrm{Var}(\mathbb{A})$ *generated by* $\mathbb{A}$ *contains no nontrivial quasi-affine algebras,*
- $\mathrm{Var}(\mathbb{A})$ *is congruence meet-semidistributive,*
- *every cycle-consistent instance of* $\mathrm{CSP}(\mathbf{A})$ *has a solution.*

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.
- We think of $\Delta(S)$ as the convex hull of the one-hot vectors $(0, \ldots, 0, 1, 0, \ldots, 0)$ in $\mathbb{R}^S$.

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.
- We think of $\Delta(S)$ as the convex hull of the one-hot vectors $(0, \ldots, 0, 1, 0, \ldots, 0)$ in $\mathbb{R}^S$.

Definition

A *fractional solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.
- We think of $\Delta(S)$ as the convex hull of the one-hot vectors $(0, \dots, 0, 1, 0, \dots, 0)$ in $\mathbb{R}^S$.

Definition

A *fractional solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \Delta(A)$, together with

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.
- We think of $\Delta(S)$ as the convex hull of the one-hot vectors $(0, \ldots, 0, 1, 0, \ldots, 0)$ in $\mathbb{R}^S$.

Definition

A *fractional solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \Delta(A)$, together with
- a collection of maps $r_i : C_i \to \Delta(R_i)$, such that

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.
- We think of $\Delta(S)$ as the convex hull of the one-hot vectors $(0, \ldots, 0, 1, 0, \ldots, 0)$ in $\mathbb{R}^S$.

Definition

A *fractional solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \Delta(A)$, together with
- a collection of maps $r_i : C_i \to \Delta(R_i)$, such that
- for each constraint $c = (x_1, \ldots, x_k) \in C_i$, and for each $j \leq k$, the distribution $a(x_j)$ is the jth marginal probability distribution of $r_i(c)$.

The Linear Programming Relaxation

- For any finite set S, let $\Delta(S)$ be the collection of probability distributions on S.
- We think of $\Delta(S)$ as the convex hull of the one-hot vectors $(0, ..., 0, 1, 0, ..., 0)$ in $\mathbb{R}^S$.

Definition

A *fractional solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \Delta(A)$, together with
- a collection of maps $r_i : C_i \to \Delta(R_i)$, such that
- for each constraint $c = (x_1, ..., x_k) \in C_i$, and for each $j \leq k$, the distribution $a(x_j)$ is the jth marginal probability distribution of $r_i(c)$.

- We can define *approximate fractional solutions* similarly, with $r_i : C_i \to \Delta(A^k)$ instead of $r_i : C_i \to \Delta(R_i)$.

Rounding schemes for the Linear Programming relaxation

- An *LP rounding scheme* is just a map

$$s : \Delta(A) \to A.$$

Rounding schemes for the Linear Programming relaxation

- An *LP rounding scheme* is just a map

$$s : \Delta(A) \to A.$$

- We say that the LP rounding scheme s *solves* $\mathrm{CSP}(\mathbf{A})$ if for every instance $\mathbf{X}$, and for every fractional solution

$$a : X \to \Delta(A), \quad r_i : C_i \to \Delta(R_i),$$

  the map

$$s \circ a : X \to A$$

  defines a homomorphism $\mathbf{X} \to \mathbf{A}$.

Example of an LP rounding scheme

- Consider the relational structure

$$\mathbf{A} = (\{-1, 0, +1\}, \{x = -y\}, \{x + y + z \geq 1\}).$$

Example of an LP rounding scheme

- Consider the relational structure

$$\mathbf{A} = (\{-1, 0, +1\}, \{x = -y\}, \{x + y + z \geq 1\}).$$

- CSP($\mathbf{A}$) is solved by the LP rounding scheme s given by

$$s(p_{-1}, p_0, p_{+1}) = \begin{cases} +1 & p_{+1} > p_{-1}, \\ 0 & p_{+1} = p_{-1}, \\ -1 & p_{+1} < p_{-1}. \end{cases}$$

Example of an LP rounding scheme

- Consider the relational structure

$$\mathbf{A} = (\{-1, 0, +1\}, \{x = -y\}, \{x + y + z \geq 1\}).$$

- CSP($\mathbf{A}$) is solved by the LP rounding scheme s given by

$$s(p_{-1}, p_0, p_{+1}) = \begin{cases} +1 & p_{+1} > p_{-1}, \\ 0 & p_{+1} = p_{-1}, \\ -1 & p_{+1} < p_{-1}. \end{cases}$$

- For every n, the symmetric function s_n given by

$$s_n(x_1, \dots, x_n) = \begin{cases} +1 & \sum_i x_i > 0, \\ 0 & \sum_i x_i = 0, \\ -1 & \sum_i x_i < 0 \end{cases}$$

  is a polymorphism of $\mathbf{A}$.

Characterization of LP rounding schemes

Theorem (Kun, O'Donnell, Tamaki, Yoshida, Zhou)

For a finite relational structure $\mathbf{A}$, *TFAE:*

Characterization of LP rounding schemes

Theorem (Kun, O'Donnell, Tamaki, Yoshida, Zhou)

For a finite relational structure $\mathbf{A}$, *TFAE:*

- CSP($\mathbf{A}$) *is solved by some LP rounding scheme* s,

Characterization of LP rounding schemes

Theorem (Kun, O'Donnell, Tamaki, Yoshida, Zhou)

For a finite relational structure $\mathbf{A}$*, TFAE:*

- CSP($\mathbf{A}$) *is solved by some LP rounding scheme* s*,*
- *for every* n*, there is a symmetric* n*-ary polymorphism* $s_n \in \mathrm{Pol}(\mathbf{A})$*.*

Characterization of LP rounding schemes

Theorem (Kun, O'Donnell, Tamaki, Yoshida, Zhou)

For a finite relational structure $\mathbf{A}$*, TFAE:*

- CSP($\mathbf{A}$) *is solved by some LP rounding scheme* s*,*
- *for every* n*, there is a symmetric* n*-ary polymorphism* $s_n \in \mathrm{Pol}(\mathbf{A})$*.*

- An LP rounding scheme is a collection of polymorphisms $s_n \in \mathrm{Pol}(\mathbf{A})$ that satisfy certain height 1 identities (asserting symmetry).

Characterization of LP rounding schemes

Theorem (Kun, O'Donnell, Tamaki, Yoshida, Zhou)
For a finite relational structure $\mathbf{A}$*, TFAE:*

- $\mathrm{CSP}(\mathbf{A})$ *is solved by some LP rounding scheme* s*,*
- *for every* n*, there is a symmetric* n*-ary polymorphism* $s_n \in \mathrm{Pol}(\mathbf{A})$*.*

- An LP rounding scheme is a collection of polymorphisms $s_n \in \mathrm{Pol}(\mathbf{A})$ that satisfy certain height 1 identities (asserting symmetry).

- Unfortunately, not every bounded width CSP has an LP rounding scheme:

$$\text{2-SAT} = (\{0,1\}, \{x \neq y\}, \{x \geq y\})$$

has no binary symmetric polymorphism.

From fractional solutions to preference relations

- If $p \in \Delta(A)$ is a probability distribution over A, then we can define a *total preorder* $\preceq_p$ on the powerset $\mathcal{P}(A)$:

$$U \preceq_p V \quad \Longleftrightarrow \quad \sum_{u \in U} p_u \leq \sum_{v \in V} p_v .$$

From fractional solutions to preference relations

- If $p \in \Delta(A)$ is a probability distribution over A, then we can define a *total preorder* $\preceq_p$ on the powerset $\mathcal{P}(A)$:

$$U \preceq_p V \quad \iff \quad \sum_{u \in U} p_u \leq \sum_{v \in V} p_v .$$

- Total preorders are also known as *preference relations*.

From fractional solutions to preference relations

- If $p \in \Delta(A)$ is a probability distribution over A, then we can define a *total preorder* $\preceq_p$ on the powerset $\mathcal{P}(A)$:

$$U \preceq_p V \quad \Longleftrightarrow \quad \sum_{u \in U} p_u \leq \sum_{v \in V} p_v .$$

- Total preorders are also known as *preference relations*.
- A total preorder $\preceq$ can be decomposed into an equivalence relation $\sim$ on $\mathcal{P}(A)$ and a total ordering $\prec$ on $\mathcal{P}(A)/\sim$.

From fractional solutions to preference relations

- If $p \in \Delta(A)$ is a probability distribution over A, then we can define a *total preorder* $\preceq_p$ on the powerset $\mathcal{P}(A)$:

$$U \preceq_p V \quad \iff \quad \sum_{u \in U} p_u \leq \sum_{v \in V} p_v .$$

- Total preorders are also known as *preference relations*.
- A total preorder $\preceq$ can be decomposed into an equivalence relation $\sim$ on $\mathcal{P}(A)$ and a total ordering $\prec$ on $\mathcal{P}(A)/\sim$.

- Consider the probability distribution $p_0 = p_1 = \frac{1}{2}$ on 2-SAT.

From fractional solutions to preference relations

- If $p \in \Delta(A)$ is a probability distribution over A, then we can define a *total preorder* $\preceq_p$ on the powerset $\mathcal{P}(A)$:

$$U \preceq_p V \iff \sum_{u \in U} p_u \leq \sum_{v \in V} p_v .$$

- Total preorders are also known as *preference relations*.
- A total preorder $\preceq$ can be decomposed into an equivalence relation $\sim$ on $\mathcal{P}(A)$ and a total ordering $\prec$ on $\mathcal{P}(A)/\sim$.

- Consider the probability distribution $p_0 = p_1 = \frac{1}{2}$ on 2-SAT.
- The corresponding preference relation is

$$\emptyset \prec_p \{0\} \sim_p \{1\} \prec_p \{0, 1\}.$$

From fractional solutions to preference relations

- If $p \in \Delta(A)$ is a probability distribution over A, then we can define a *total preorder* $\preceq_p$ on the powerset $\mathcal{P}(A)$:

$$U \preceq_p V \quad \iff \quad \sum_{u \in U} p_u \leq \sum_{v \in V} p_v .$$

- Total preorders are also known as *preference relations*.
- A total preorder $\preceq$ can be decomposed into an equivalence relation $\sim$ on $\mathcal{P}(A)$ and a total ordering $\prec$ on $\mathcal{P}(A)/\sim$.

- Consider the probability distribution $p_0 = p_1 = \frac{1}{2}$ on 2-SAT.
- The corresponding preference relation is

$$\emptyset \prec_p \{0\} \sim_p \{1\} \prec_p \{0, 1\}.$$

- We want to outlaw this sort of preference relation.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.
- (Self-duality) If $U \preceq_v V$, then $S \setminus V \preceq_v S \setminus U$.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.
- (Self-duality) If $U \preceq_v V$, then $S \setminus V \preceq_v S \setminus U$.
- (Support) If $U \sim_v S$, then $U \cap V \sim_v V$.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.
- (Self-duality) If $U \preceq_v V$, then $S \setminus V \preceq_v S \setminus U$.
- (Support) If $U \sim_v S$, then $U \cap V \sim_v V$.

  The smallest set U such that $U \sim_v S$ is called the *support* of v.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.
- (Self-duality) If $U \preceq_v V$, then $S \setminus V \preceq_v S \setminus U$.
- (Support) If $U \sim_v S$, then $U \cap V \sim_v V$.

  The smallest set U such that $U \sim_v S$ is called the *support* of v.
- (Nontriviality) $S \not\sim_v \emptyset$.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.
- (Self-duality) If $U \preceq_v V$, then $S \setminus V \preceq_v S \setminus U$.
- (Support) If $U \sim_v S$, then $U \cap V \sim_v V$.

  The smallest set U such that $U \sim_v S$ is called the *support* of v.
- (Nontriviality) $S \not\sim_v \emptyset$.
- (Weak Coherence) If $U \sim_v V \not\sim_v \emptyset$, then $U \cap V \not\sim_v \emptyset$.

Vague elements

Definition

A *vague element* v of a set S is a preference relation $\preceq_v$ on $\mathcal{P}(S)$ satisfying the following properties for all $U, V \subseteq S$:

- (Monotonicity) If $U \subseteq V$, then $U \preceq_v V$.
- (Self-duality) If $U \preceq_v V$, then $S \setminus V \preceq_v S \setminus U$.
- (Support) If $U \sim_v S$, then $U \cap V \sim_v V$.

  The smallest set U such that $U \sim_v S$ is called the *support* of v.
- (Nontriviality) $S \not\sim_v \emptyset$.
- (Weak Coherence) If $U \sim_v V \not\sim_v \emptyset$, then $U \cap V \not\sim_v \emptyset$.

- We write $\mathcal{V}(S)$ for the collection of vague elements of a set S.

Marginals of vague elements

- The map $S \mapsto \mathcal{V}(S)$ defines a *functor*.

Marginals of vague elements

- The map $S \mapsto \mathcal{V}(S)$ defines a *functor*.
- If $f : S \to T$ and $v \in \mathcal{V}(S)$, we define $f_*(v) \in \mathcal{V}(T)$ by

$$U \preceq_{f_*(v)} V \quad \Longleftrightarrow \quad f^{-1}(U) \preceq_v f^{-1}(V).$$

Marginals of vague elements

- The map $S \mapsto \mathcal{V}(S)$ defines a *functor*.
- If $f : S \to T$ and $v \in \mathcal{V}(S)$, we define $f_*(v) \in \mathcal{V}(T)$ by

$$U \preceq_{f_*(v)} V \quad \Longleftrightarrow \quad f^{-1}(U) \preceq_v f^{-1}(V).$$

- In particular, if $R \subseteq A^k$ is a relation, and $r \in \mathcal{V}(R)$, then we can define the ith marginal of r to be

$$(\pi_i \circ \iota)_*(r) \in \mathcal{V}(A),$$

where $\iota : R \hookrightarrow A^k$ is the inclusion.

Marginals of vague elements

- The map $S \mapsto \mathcal{V}(S)$ defines a *functor*.
- If $f : S \to T$ and $v \in \mathcal{V}(S)$, we define $f_*(v) \in \mathcal{V}(T)$ by

$$U \preceq_{f_*(v)} V \quad \iff \quad f^{-1}(U) \preceq_v f^{-1}(V).$$

- In particular, if $R \subseteq A^k$ is a relation, and $r \in \mathcal{V}(R)$, then we can define the ith marginal of r to be

$$(\pi_i \circ \iota)_*(r) \in \mathcal{V}(A),$$

  where $\iota : R \hookrightarrow A^k$ is the inclusion.
- Note that $\iota_*(r)$ is a vague element of A^k with support contained in R.

Vague solutions, take one

- Vague solutions could be defined exactly analogously to fractional solutions:

Vague solutions, take one

- Vague solutions could be defined exactly analogously to fractional solutions:

Definition

A *strong vague solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

Vague solutions, take one

- Vague solutions could be defined exactly analogously to fractional solutions:

Definition

A *strong vague solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \mathcal{V}(A)$, together with

Vague solutions, take one

- Vague solutions could be defined exactly analogously to fractional solutions:

Definition

A *strong vague solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \mathcal{V}(A)$, together with
- a collection of maps $r_i : C_i \to \mathcal{V}(R_i)$, such that

Vague solutions, take one

- Vague solutions could be defined exactly analogously to fractional solutions:

Definition

A *strong vague solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \mathcal{V}(A)$, together with
- a collection of maps $r_i : C_i \to \mathcal{V}(R_i)$, such that
- for each constraint $c = (x_1, ..., x_k) \in C_i$, and for each $j \leq k$, the vague element $a(x_j)$ is the jth marginal of $r_i(c)$.

Vague solutions, take one

- Vague solutions could be defined exactly analogously to fractional solutions:

Definition

A *strong vague solution* to an instance $\mathbf{X}$ of CSP($\mathbf{A}$) is the following:

- a map $a : X \to \mathcal{V}(A)$, together with
- a collection of maps $r_i : C_i \to \mathcal{V}(R_i)$, such that
- for each constraint $c = (x_1, ..., x_k) \in C_i$, and for each $j \leq k$, the vague element $a(x_j)$ is the jth marginal of $r_i(c)$.

- But describing a vague element of R_i sounds very onerous. We will make a simpler (weaker) definition.

Vague solutions, take two

Definition

If $R \subseteq A_1 \times \cdots \times A_k$, then a collection of vague elements $v_i \in \mathcal{V}(A_i)$ *vaguely satisfies* the relation R if there exists a preorder $\preceq_r$ on the disjoint union

$$\mathcal{P}(A_1) \sqcup \cdots \sqcup \mathcal{P}(A_k)$$

such that

Vague solutions, take two

Definition

If $R \subseteq A_1 \times \cdots \times A_k$, then a collection of vague elements $v_i \in \mathcal{V}(A_i)$ *vaguely satisfies* the relation R if there exists a preorder $\preceq_r$ on the disjoint union

$$\mathcal{P}(A_1) \sqcup \cdots \sqcup \mathcal{P}(A_k)$$

such that

- for each i, the restriction of $\preceq_r$ to $\mathcal{P}(A_i)$ is $\preceq_{v_i}$,

Vague solutions, take two

Definition

If $R \subseteq A_1 \times \cdots \times A_k$, then a collection of vague elements $v_i \in \mathcal{V}(A_i)$ *vaguely satisfies* the relation R if there exists a preorder $\preceq_r$ on the disjoint union

$$\mathcal{P}(A_1) \sqcup \cdots \sqcup \mathcal{P}(A_k)$$

such that

- for each i, the restriction of $\preceq_r$ to $\mathcal{P}(A_i)$ is $\preceq_{v_i}$,
- for each i, j and each $U \subseteq A_i$, we have

  $$U \preceq_r U + \pi_{ij}(R \cap (S_1 \times \cdots \times S_k)),$$

  where the S_i are the supports of the vague elements v_i.

Vague rounding schemes

- A *vague rounding scheme* is just a map

$$s : \mathcal{V}(A) \to A.$$

Vague rounding schemes

- A *vague rounding scheme* is just a map
$$s : \mathcal{V}(A) \to A.$$
- We say that the vague rounding scheme s *solves* $\mathrm{CSP}(\mathbf{A})$ if for every instance $\mathbf{X}$, and for every vague solution
$$a : X \to \mathcal{V}(A)$$
such that $(a(x_1), \ldots, a(x_k))$ vaguely satisfies R_i for each constraint $c = (x_1, \ldots, x_k) \in C_i$, the map
$$s \circ a : X \to A$$
defines a homomorphism $\mathbf{X} \to \mathbf{A}$.

Main result

Theorem (Z.)

For a finite relational structure $\mathbf{A}$, *TFAE:*

Main result

Theorem (Z.)

For a finite relational structure $\mathbf{A}$*, TFAE:*

- $\mathbf{A}$ *has bounded width,*

Main result

Theorem (Z.)

For a finite relational structure $\mathbf{A}$*, TFAE:*

- $\mathbf{A}$ *has bounded width,*
- *there is a vague rounding scheme* $s : \mathcal{V}(A) \to A$ *which solves* $\mathrm{CSP}(\mathbf{A})$,

Main result

Theorem (Z.)

For a finite relational structure $\mathbf{A}$*, TFAE:*

- $\mathbf{A}$ *has bounded width,*
- *there is a vague rounding scheme* $s : \mathcal{V}(A) \to A$ *which solves* $\mathrm{CSP}(\mathbf{A})$*,*
- *for every* n*, and for every vague element* $v \in \mathcal{V}(\{1, ..., n\})$*, there is an* n*-ary polymorphism* $s_v \in \mathrm{Pol}(\mathbf{A})$*, such that for all*

$$f : \{1, ..., n\} \to \{1, ..., m\}$$

the height 1 identity

$$s_v(x_{f(1)}, ..., x_{f(n)}) \approx s_{f_*(v)}(x_1, ..., x_m)$$

is satisfied.

Multisorted CSPs

- In order to prove this result, it is more convenient to work in the framework of *multisorted* CSPs.

Multisorted CSPs

- In order to prove this result, it is more convenient to work in the framework of *multisorted* CSPs.
- Let $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ be the algebraic structure corresponding to $\mathbf{A}$.

Multisorted CSPs

- In order to prove this result, it is more convenient to work in the framework of *multisorted* CSPs.
- Let $\mathbb{A} = (A, \text{Pol}(\mathbf{A}))$ be the algebraic structure corresponding to $\mathbf{A}$.
- We let each variable $x \in X$ have a different domain $\mathbb{A}_x$, with $\mathbb{A}_x$ an arbitrary finite algebra in $\text{Var}(\mathbb{A})$.

Multisorted CSPs

- In order to prove this result, it is more convenient to work in the framework of *multisorted* CSPs.
- Let $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ be the algebraic structure corresponding to $\mathbf{A}$.
- We let each variable $x \in X$ have a different domain $\mathbb{A}_x$, with $\mathbb{A}_x$ an arbitrary finite algebra in $\mathrm{Var}(\mathbb{A})$.
- A constraint c now consists of a tuple $(x_1, \ldots, x_k)$ of variables, together with a *constraint relation*

$$\mathbb{R} \leq \mathbb{A}_{x_1} \times \cdots \times \mathbb{A}_{x_k}.$$

Multisorted CSPs

- In order to prove this result, it is more convenient to work in the framework of *multisorted* CSPs.
- Let $\mathbb{A} = (A, \mathrm{Pol}(\mathbf{A}))$ be the algebraic structure corresponding to $\mathbf{A}$.
- We let each variable $x \in X$ have a different domain $\mathbb{A}_x$, with $\mathbb{A}_x$ an arbitrary finite algebra in $\mathrm{Var}(\mathbb{A})$.
- A constraint c now consists of a tuple $(x_1, ..., x_k)$ of variables, together with a *constraint relation*

$$\mathbb{R} \leq \mathbb{A}_{x_1} \times \cdots \times \mathbb{A}_{x_k}.$$

- A *solution* is a map $x \mapsto a_x$ such that for each constraint c as above, we have

$$(a_{x_1}, ..., a_{x_k}) \in \mathbb{R}.$$

Paths

- A *step* from y to z is a constraint

$$((x_1, \ldots, x_k), \mathbb{R})$$

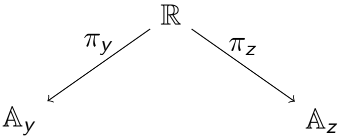
and a pair i, j such that $x_i = y$ and $x_j = z$.

Paths

- A *step* from y to z is a constraint

$$((x_1, ..., x_k), \mathbb{R})$$

and a pair i, j such that $x_i = y$ and $x_j = z$.

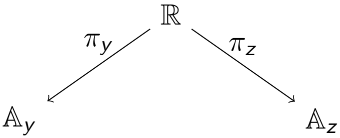


Paths

- A *step* from y to z is a constraint

$$((x_1, ..., x_k), \mathbb{R})$$

and a pair i, j such that $x_i = y$ and $x_j = z$.



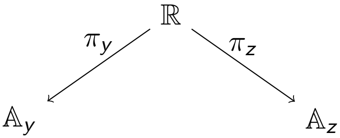
- A *path* is a sequence of steps where the endpoints match up.

Paths

- A *step* from y to z is a constraint

$$((x_1, ..., x_k), \mathbb{R})$$

  and a pair i, j such that $x_i = y$ and $x_j = z$.



- A *path* is a sequence of steps where the endpoints match up.
- We use additive notation for combining paths: $p + q$ means "first follow p, then q".

Propagating information along paths

- If $B \subseteq \mathbb{A}_y$ and p is a step from y to z through a relation $\mathbb{R}$, we write

$$B + p = B + \pi_{yz}(\mathbb{R}) = \pi_z(\pi_y^{-1}(B) \cap \mathbb{R}) \subseteq \mathbb{A}_z.$$

Propagating information along paths

- If $B \subseteq \mathbb{A}_y$ and p is a step from y to z through a relation $\mathbb{R}$, we write

$$B + p = B + \pi_{yz}(\mathbb{R}) = \pi_z(\pi_y^{-1}(B) \cap \mathbb{R}) \subseteq \mathbb{A}_z.$$

- This encodes the implication: "if $a_y \in B$, then $a_z \in B + p$".

Propagating information along paths

- If $B \subseteq \mathbb{A}_y$ and p is a step from y to z through a relation $\mathbb{R}$, we write

$$B + p = B + \pi_{yz}(\mathbb{R}) = \pi_z(\pi_y^{-1}(B) \cap \mathbb{R}) \subseteq \mathbb{A}_z.$$

- This encodes the implication: "if $a_y \in B$, then $a_z \in B + p$".

- Extend this notation to paths in the obvious way:

$$B + (p_1 + p_2) = (B + p_1) + p_2, \text{ etc.}$$

Propagating information along paths

- If $B \subseteq \mathbb{A}_y$ and p is a step from y to z through a relation $\mathbb{R}$, we write

$$B + p = B + \pi_{yz}(\mathbb{R}) = \pi_z(\pi_y^{-1}(B) \cap \mathbb{R}) \subseteq \mathbb{A}_z.$$

- This encodes the implication: "if $a_y \in B$, then $a_z \in B + p$".
- Extend this notation to paths in the obvious way:

$$B + (p_1 + p_2) = (B + p_1) + p_2, \text{ etc.}$$

- If $\mathbb{B} \leq \mathbb{A}_y$ is a subalgebra, then $\mathbb{B} + p \leq \mathbb{A}_z$ is also a subalgebra.

Consistency

- An instance is *arc-consistent* if for all paths p from x to y, we have
$$\mathbb{A}_x + p = \mathbb{A}_y.$$

Consistency

- An instance is *arc-consistent* if for all paths p from x to y, we have
$$\mathbb{A}_x + p = \mathbb{A}_y.$$
- Arc-consistency is equivalent to: for all constraint relations $\mathbb{R}$, the projections $\pi_i : \mathbb{R} \to \mathbb{A}_{x_i}$ are surjective.

Consistency

- An instance is *arc-consistent* if for all paths p from x to y, we have

$$\mathbb{A}_x + p = \mathbb{A}_y.$$

- Arc-consistency is equivalent to: for all constraint relations $\mathbb{R}$, the projections $\pi_i : \mathbb{R} \to \mathbb{A}_{x_i}$ are surjective.

- An instance is *cycle-consistent* if for all paths p from x to x, and for all $a \in \mathbb{A}_x$, we have

$$a \in \{a\} + p.$$

Consistency

- An instance is *arc-consistent* if for all paths p from x to y, we have
$$\mathbb{A}_x + p = \mathbb{A}_y.$$
- Arc-consistency is equivalent to: for all constraint relations $\mathbb{R}$, the projections $\pi_i : \mathbb{R} \to \mathbb{A}_{x_i}$ are surjective.

- An instance is *cycle-consistent* if for all paths p from x to x, and for all $a \in \mathbb{A}_x$, we have
$$a \in \{a\} + p.$$
- Beginner Sudoku players start by establishing arc-consistency, then they move on to establishing cycle-consistency.

Weaker consistency!

- I call an instance *weakly consistent* if it satisfies:
  - (P1) arc-consistency, and
  - (W) $A + p + q = A$ implies $A \cap (A + p) \neq \emptyset$.

Weaker consistency!

- I call an instance *weakly consistent* if it satisfies:

  (P1) arc-consistency, and

  (W) $A + p + q = A$ implies $A \cap (A + p) \neq \emptyset$.

- I will use this result, from a previous AAA conference:

Theorem (Z.)

If $\mathrm{Var}(\mathbb{A})$ *is* $\mathrm{SD}(\wedge)$*, then every weakly consistent instance of* $\mathrm{CSP}(\mathrm{Var}_{fin}(\mathbb{A}))$ *has a solution.*

Connection to vague solutions

Proposition

If an instance $\mathbf{X}$ *of a multisorted CSP is weakly consistent, then it has a vague solution*

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x)$$

such that each a_x *has support equal to* $\mathbb{A}_x$*.*

Connection to vague solutions

Proposition

If an instance $\mathbf{X}$ *of a multisorted CSP is weakly consistent, then it has a vague solution*

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x)$$

such that each a_x *has support equal to* $\mathbb{A}_x$*.*

- Define a preorder $\preceq$ on $\bigsqcup_x \mathcal{P}(\mathbb{A}_x)$ by $(x, A) \preceq (y, B)$ if there is some path p from x to y such that $A + p \subseteq B$.

Connection to vague solutions

Proposition

If an instance $\mathbf{X}$ *of a multisorted CSP is weakly consistent, then it has a vague solution*

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x)$$

such that each a_x *has support equal to* $\mathbb{A}_x$*.*

- Define a preorder $\preceq$ on $\bigsqcup_x \mathcal{P}(\mathbb{A}_x)$ by $(x, A) \preceq (y, B)$ if there is some path p from x to y such that $A + p \subseteq B$.
- Extend $\preceq$ to a total preorder $\preceq'$ without changing the associated equivalence relation $\sim$.

Connection to vague solutions

Proposition

If an instance $\mathbf{X}$ *of a multisorted CSP is weakly consistent, then it has a vague solution*

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x)$$

such that each a_x *has support equal to* $\mathbb{A}_x$*.*

- Define a preorder $\preceq$ on $\bigsqcup_x \mathcal{P}(\mathbb{A}_x)$ by $(x, A) \preceq (y, B)$ if there is some path p from x to y such that $A + p \subseteq B$.
- Extend $\preceq$ to a total preorder $\preceq'$ without changing the associated equivalence relation $\sim$.
- Let $\preceq_{a_x}$ be the restriction of $\preceq'$ to $\mathcal{P}(\mathbb{A}_x)$.

From a vague solution to a weakly consistent instance

- Now suppose that we have a vague solution

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x).$$

This doesn't necessarily mean that our instance $\mathbf{X}$ is weakly consistent.

From a vague solution to a weakly consistent instance

- Now suppose that we have a vague solution

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x).$$

  This doesn't necessarily mean that our instance $\mathbf{X}$ is weakly consistent.

- We will produce a weakly consistent instance $\mathbf{X}_a^*$ which has many copies of each variable and relation from $\mathbf{X}$, in order to apply Ramsey's Theorem.

From a vague solution to a weakly consistent instance

- Now suppose that we have a vague solution

$$x \mapsto a_x \in \mathcal{V}(\mathbb{A}_x).$$

  This doesn't necessarily mean that our instance $\mathbf{X}$ is weakly consistent.

- We will produce a weakly consistent instance $\mathbf{X}_a^*$ which has many copies of each variable and relation from $\mathbf{X}$, in order to apply Ramsey's Theorem.

- The trick is to exploit the fact that everything is stated in terms of *total* preorders.

Compatibility between vague elements and functions

- If $f : \mathcal{P}(A) \to \mathbb{N}$ and $v \in \mathcal{V}(A)$, we say f is *compatible* with v if

$$U \preceq_v V \iff f(U) \leq f(V).$$

Compatibility between vague elements and functions

- If $f : \mathcal{P}(A) \to \mathbb{N}$ and $v \in \mathcal{V}(A)$, we say f is *compatible* with v if
$$U \preceq_v V \iff f(U) \leq f(V).$$
- Note that f is determined by v and $\mathrm{im}(f) \subseteq \mathbb{N}$.

Compatibility between vague elements and functions

- If $f : \mathcal{P}(A) \to \mathbb{N}$ and $v \in \mathcal{V}(A)$, we say f is *compatible* with v if
$$U \preceq_v V \iff f(U) \leq f(V).$$
- Note that f is determined by v and $\mathrm{im}(f) \subseteq \mathbb{N}$.
- If $f : \mathcal{P}(A_1) \sqcup \cdots \sqcup \mathcal{P}(A_k) \to \mathbb{N}$, and if $R \subseteq A_1 \times \cdots \times A_k$, we say f is *compatible* with R if
$$f(U) \leq f(U + \pi_{ij}(R))$$
for all $i, j \leq k$ and all $U \subseteq A_i$.

Constructing the weakly consistent instance

- $\mathbf{X}_a^*$ is constructed as follows:

Constructing the weakly consistent instance

- $\mathbf{X}_a^*$ is constructed as follows:

- For $x \in \mathbf{X}$ and $f : \mathcal{P}(\mathbb{A}_x) \to \mathbb{N}$ compatible with a_x, we introduce a variable (x, f) of $\mathbf{X}_a^*$ with domain $\mathbb{A}_x$.

Constructing the weakly consistent instance

- $\mathbf{X}_a^*$ is constructed as follows:

- For $x \in \mathbf{X}$ and $f : \mathcal{P}(\mathbb{A}_x) \to \mathbb{N}$ compatible with a_x, we introduce a variable (x, f) of $\mathbf{X}_a^*$ with domain $\mathbb{A}_x$.
- For $c = ((x_1, ..., x_k), \mathbb{R})$ and compatible $f : \mathcal{P}(\mathbb{A}_{x_1}) \sqcup \cdots \sqcup \mathcal{P}(\mathbb{A}_{x_k}) \to \mathbb{N}$, we introduce the constraint

$$\left(\left((x_1, f|_{\mathcal{P}(A_{x_1})}), ..., (x_k, f|_{\mathcal{P}(A_{x_k})})\right), \mathbb{R}\right)$$

  of $\mathbf{X}_a^*$.

Constructing the weakly consistent instance

- $\mathbf{X}_a^*$ is constructed as follows:

- For $x \in \mathbf{X}$ and $f : \mathcal{P}(\mathbb{A}_x) \to \mathbb{N}$ compatible with a_x, we introduce a variable (x, f) of $\mathbf{X}_a^*$ with domain $\mathbb{A}_x$.
- For $c = ((x_1, ..., x_k), \mathbb{R})$ and compatible $f : \mathcal{P}(\mathbb{A}_{x_1}) \sqcup \cdots \sqcup \mathcal{P}(\mathbb{A}_{x_k}) \to \mathbb{N}$, we introduce the constraint

$$\left(\left((x_1, f|_{\mathcal{P}(A_{x_1})}), ..., (x_k, f|_{\mathcal{P}(A_{x_k})})\right), \mathbb{R}\right)$$

of $\mathbf{X}_a^*$.

- By construction, if there is a path p from (x, f) to (x, f) in $\mathbf{X}_a^*$, and if $A \subseteq \mathbb{A}_x$, then

$$f(A) \le f(A + p), \quad \text{so} \quad A \preceq_{a_x} A + p.$$

Applying Ramsey's Theorem

- Let s be a solution to the weakly consistent instance $\mathbf{X}_a^*$.

Applying Ramsey's Theorem

- Let s be a solution to the weakly consistent instance $\mathbf{X}_a^*$.
- By Ramsey's Theorem, there is an infinite subset $S \subseteq \mathbb{N}$ such that for each $x \in \mathbf{X}$ there is some $\hat{s}_x$ with

$$s_{(x,f)} = \hat{s}_x$$

for all $(x, f) \in \mathbf{X}_a^*$ with $\mathrm{im}(f) \subseteq S$.

Applying Ramsey's Theorem

- Let s be a solution to the weakly consistent instance $\mathbf{X}_a^*$.
- By Ramsey's Theorem, there is an infinite subset $S \subseteq \mathbb{N}$ such that for each $x \in \mathbf{X}$ there is some $\hat{s}_x$ with
$$s_{(x,f)} = \hat{s}_x$$
for all $(x, f) \in \mathbf{X}_a^*$ with $\mathrm{im}(f) \subseteq S$.
- If $a_{x_1}, \ldots, a_{x_k}$ vaguely satisfy the relation $\mathbb{R}$, then there is some compatible $f : \mathcal{P}(\mathbb{A}_{x_1}) \sqcup \cdots \sqcup \mathcal{P}(\mathbb{A}_{x_k}) \to S$, so
$$(\hat{s}_{x_1}, \ldots, \hat{s}_{x_k}) = \left(s_{(x_1, f|_{\mathcal{P}(A_{x_1})})}, \ldots, s_{(x_k, f|_{\mathcal{P}(A_{x_k})})}\right) \in \mathbb{R}.$$

Applying Ramsey's Theorem

- Let s be a solution to the weakly consistent instance $\mathbf{X}_a^*$.
- By Ramsey's Theorem, there is an infinite subset $S \subseteq \mathbb{N}$ such that for each $x \in \mathbf{X}$ there is some $\hat{s}_x$ with

$$s_{(x,f)} = \hat{s}_x$$

  for all $(x, f) \in \mathbf{X}_a^*$ with $\mathrm{im}(f) \subseteq S$.
- If $a_{x_1}, \ldots, a_{x_k}$ vaguely satisfy the relation $\mathbb{R}$, then there is some compatible $f : \mathcal{P}(\mathbb{A}_{x_1}) \sqcup \cdots \sqcup \mathcal{P}(\mathbb{A}_{x_k}) \to S$, so

$$(\hat{s}_{x_1}, \ldots, \hat{s}_{x_k}) = (s_{(x_1, f|_{\mathcal{P}(A_{x_1})})}, \ldots, s_{(x_k, f|_{\mathcal{P}(A_{x_k})})}) \in \mathbb{R}.$$

- So $\hat{s}$ is a solution to $\mathbf{X}$!

Existence of the vague rounding scheme

- To obtain the vague rounding scheme

$$s : \mathcal{V}(A) \to A,$$

we apply this argument to the "most generic" instance **X** which has a vague solution.

Existence of the vague rounding scheme

- To obtain the vague rounding scheme

$$s : \mathcal{V}(A) \to A,$$

we apply this argument to the "most generic" instance **X** which has a vague solution.

- The variables of this **X** correspond to the elements v of $\mathcal{V}(A)$, with variable domain $\mathbb{A}_v$ equal to the support of v.

Existence of the vague rounding scheme

- To obtain the vague rounding scheme

$$s : \mathcal{V}(A) \to A,$$

we apply this argument to the "most generic" instance $\mathbf{X}$ which has a vague solution.

- The variables of this $\mathbf{X}$ correspond to the elements v of $\mathcal{V}(A)$, with variable domain $\mathbb{A}_v$ equal to the support of v.

- We impose a constraint $((v_1, \ldots, v_k), \mathbb{R})$ in $\mathbf{X}$ whenever $\mathbb{R} \leq_{sd} \mathbb{A}_{v_1} \times \cdots \times \mathbb{A}_{v_k}$ is vaguely satisfied by $v_1, \ldots, v_k$.

Back to robust satisfaction

Theorem (Z.)

If the semidefinite programming relaxation of an instance $\mathbf{X}$ *of* $\mathrm{CSP}(\mathbf{A})$ *has value* $1-\epsilon$*, then we can algorithmically find a vague solution to* $\mathbf{X}$ *which vaguely satisfies a* $1-f(\epsilon)$ *fraction of the constraints in polynomial time, where*

$$f(\epsilon) \ll_{\mathbf{A}} \frac{1}{\log(1/\epsilon)}.$$

Back to robust satisfaction

Theorem (Z.)

If the semidefinite programming relaxation of an instance $\mathbf{X}$ *of* CSP($\mathbf{A}$) *has value* $1 - \epsilon$*, then we can algorithmically find a vague solution to* $\mathbf{X}$ *which vaguely satisfies a* $1 - f(\epsilon)$ *fraction of the constraints in polynomial time, where*

$$f(\epsilon) \ll_{\mathbf{A}} \frac{1}{\log(1/\epsilon)}.$$

- Once we have the (approx.) vague solution, we apply a vague rounding scheme to get an actual (approx.) solution.

Back to robust satisfaction

Theorem (Z.)

If the semidefinite programming relaxation of an instance $\mathbf{X}$ *of* CSP($\mathbf{A}$) *has value* $1-\epsilon$*, then we can algorithmically find a vague solution to* $\mathbf{X}$ *which vaguely satisfies a* $1-f(\epsilon)$ *fraction of the constraints in polynomial time, where*

$$f(\epsilon) \ll_{\mathbf{A}} \frac{1}{\log(1/\epsilon)}.$$

- Once we have the (approx.) vague solution, we apply a vague rounding scheme to get an actual (approx.) solution.
- This is best possible: we can't robustly solve HORN-SAT with $f(\epsilon) = o(1/\log(1/\epsilon))$ unless the Unique Games Conjecture is false, by a result of Guruswami and Zhou.

Thank you for your attention.