Challenges in Infinite Domain Constraint Satisfaction

Manuel Bodirsky

TU Dresden

November 2014

Infinite-Domain CSPs

Two Research Directions:

Infinite-Domain CSPs

Two Research Directions:

1. For which infinite structures Γ can we still use the universal-algebraic approach to classify the complexity of CSP(Γ)?

Infinite-Domain CSPs

Two Research Directions:

1. For which infinite structures Γ can we still use the universal-algebraic approach to classify the complexity of CSP(Γ)?
   Tutorial of Michael, talks of Andras, Libor,

Infinite-Domain CSPs

Two Research Directions:

1. For which infinite structures Γ can we still use the universal-algebraic approach to classify the complexity of CSP(Γ)?
   Tutorial of Michael, talks of Andras, Libor, Jan,

Infinite-Domain CSPs

Two Research Directions:

1. For which infinite structures Γ can we still use the universal-algebraic approach to classify the complexity of CSP(Γ)?
   Tutorial of Michael, talks of Andras, Libor, Jan,
2. What can be said about the complexity of CSP(Γ) for infinite structures Γ 'of general interest'? E.g., on the domain $\mathbb{Z}$, $\mathbb{Q}$, or $\mathbb{R}$?

Infinite-Domain CSPs

Two Research Directions:

1. For which infinite structures Γ can we still use the universal-algebraic approach to classify the complexity of CSP(Γ)?
   Tutorial of Michael, talks of Andras, Libor, Jan,
2. What can be said about the complexity of CSP(Γ) for infinite structures Γ 'of general interest'? E.g., on the domain $\mathbb{Z}$, $\mathbb{Q}$, or $\mathbb{R}$?
   Talks of Barny, Johan,

Part I: ω-categorical templates

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

1. Further develop the method:

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

1. Further develop the method:
   - Homomorphisms vs continuous homomorphisms to $\mathbf{1}$

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

1. Further develop the method:
   - Homomorphisms vs continuous homomorphisms to $\mathbf{1}$
   - Taylor terms modulo endomorphisms versus
     weak near unanimity terms modulo endomorphisms versus ...

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

1. Further develop the method:
   - Homomorphisms vs continuous homomorphisms to $\mathbf{1}$
   - Taylor terms modulo endomorphisms versus
     weak near unanimity terms modulo endomorphisms versus . . .
   - Lift algorithms from finite to infinite

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

1. Further develop the method:
   - Homomorphisms vs continuous homomorphisms to $\mathbf{1}$
   - Taylor terms modulo endomorphisms versus
     weak near unanimity terms modulo endomorphisms versus …
   - Lift algorithms from finite to infinite
   - Ramsey expansion problem(s)

The Universal-Algebraic Approach

Michael's tutorial:
the universal-algebraic approach works (more or less) for ω-categorical structures.

Two subdirections:

1. Further develop the method:
   - Homomorphisms vs continuous homomorphisms to $\mathbf{1}$
   - Taylor terms modulo endomorphisms versus
     weak near unanimity terms modulo endomorphisms versus . . .
   - Lift algorithms from finite to infinite
   - Ramsey expansion problem(s)
2. **Apply the method!**

Permutations

Permutations

Finite permutations can be represented by two linear orders:

(3142) corresponds to $(\{a, b, c, d\}; <_1, <_2)$ such that

$a <_1 b <_1 c <_1 d$ and $c <_2 a <_2 d <_2 b$

Permutations

Finite permutations can be represented by two linear orders:

(3142) corresponds to $(\{a, b, c, d\}; <_1, <_2)$ such that

$$a <_1 b <_1 c <_1 d \text{ and } c <_2 a <_2 d <_2 b$$

Homogeneous countably infinite 'permutations' (Cameron'02):

Permutations

Finite permutations can be represented by two linear orders:

(3142) corresponds to $(\{a, b, c, d\}; <_1, <_2)$ such that

$a <_1 b <_1 c <_1 d$ and $c <_2 a <_2 d <_2 b$

Homogeneous countably infinite 'permutations' (Cameron'02):

Age(Γ) equals

Permutations

Finite permutations can be represented by two linear orders:

(3142) corresponds to $(\{a,b,c,d\}; <_1, <_2)$ such that

$a <_1 b <_1 c <_1 d$ and $c <_2 a <_2 d <_2 b$

Homogeneous countably infinite 'permutations' (Cameron'02):

Age(Γ) equals

- Forb(12)
- Forb(21)
- Forb$(231, 312)$
- Forb$(213, 132)$
- Forb$(\emptyset)$

Forb$(\mathcal{S})$: class of all finite permutations that do not embed any $S \in \mathcal{S}$.

The Random Permutation

Let Δ be the Fraïssé-limit of all finite permutations.

Question (Cameron'02): Classify the reducts of Δ.

The Random Permutation

Let Δ be the Fraïssé-limit of all finite permutations.

Question (Cameron'02): Classify the reducts of Δ.

Theorem (Pinsker,Linman'14): There are precisely 39 possible automorphism groups of reducts of Δ.

The Random Permutation

Let Δ be the Fraïssé-limit of all finite permutations.

Question (Cameron'02): Classify the reducts of Δ.

Theorem (Pinsker,Linman'14): There are precisely 39 possible automorphism groups of reducts of Δ.

Challenge 1: Classify CSP(Γ) for all reducts Γ of Δ.

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Hanika'14 studies the following computational problem:

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Hanika'14 studies the following computational problem:

Definition

Input: A finite τ-structure.

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Hanika'14 studies the following computational problem:

Definition

Input: A finite τ-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2)$ does not embed (3142) and (2413)?

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Hanika'14 studies the following computational problem:

Definition

Input: A finite τ-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2)$ does not embed (3142) and (2413)?

Comments.

- Problem is closed under disjoint unions and inverse homomorphisms.

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Hanika'14 studies the following computational problem:

Definition

Input: A finite τ-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2)$ does not embed (3142) and (2413)?

Comments.

- Problem is closed under disjoint unions and inverse homomorphisms.
- Hence, this problem is the CSP for an infinite structure Γ.

Forbidden Permutations Problems

Let $\tau := \{E_1, E_2\}$ be a signature with two binary relations.

Hanika'14 studies the following computational problem:

Definition

Input: A finite τ-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2)$ does not embed (3142) and (2413)?

Comments.

- Problem is closed under disjoint unions and inverse homomorphisms.
- Hence, this problem is the CSP for an infinite structure Γ.
- The permutations in $\text{Forb}(3142, 2413)$ are called separable in finite combinatorics.

Separable Permutations

Illustration of (3142) and (2413):

		x	
x			
			x
	x		

	x		
			x
x			
		x	

Separable Permutations

Illustration of (3142) and (2413):

		x	
x			
			x
	x		

	x		
			x
x			
		x	

Separable permutations can be characterized recursively as follows (Bose+Buss+Lubiw’98):

Separable Permutations

Illustration of (3142) and (2413):

		x	
x			
			x
	x		

	x		
			x
x			
		x	

Separable permutations can be characterized recursively as follows (Bose+Buss+Lubiw'98):

if S_1, S_2 are separable, then

	S_2
S_1	

and

S_1	
	S_2

are separable, too.

An ω-categorical separable template

An ω-categorical separable template

Observation (Cameron'02): Expansion by ternary relation $xy|z$ defined by

$$(x <_1 y <_1 z \wedge y <_2 x <_2 z) \vee (z <_1 x <_1 y \wedge z <_2 y <_2 x)$$

(a so-called C-relation) makes $\mathrm{Forb}(3142, 2413)$ an amalgamation class.

An ω-categorical separable template

Observation (Cameron'02): Expansion by ternary relation $xy|z$ defined by

$$(x <_1 y <_1 z \wedge y <_2 x <_2 z) \vee (z <_1 x <_1 y \wedge z <_2 y <_2 x)$$

(a so-called *C*-relation) makes $\text{Forb}(3142, 2413)$ an amalgamation class.

- $\mathcal{C}$: class of all separable permutations, equipped with the C-relation $xy|z$

An ω-categorical separable template

Observation (Cameron'02): Expansion by ternary relation $xy|z$ defined by

$$(x <_1 y <_1 z \wedge y <_2 x <_2 z) \vee (z <_1 x <_1 y \wedge z <_2 y <_2 x)$$

(a so-called C-relation) makes $\mathrm{Forb}(3142, 2413)$ an amalgamation class.

- $\mathcal{C}$: class of all separable permutations, equipped with the C-relation $xy|z$
- Δ: homogeneous ω-categorical Fraïssé-limit of $\mathcal{C}$.

An ω-categorical separable template

Observation (Cameron'02): Expansion by ternary relation $xy|z$ defined by

$$(x <_1 y <_1 z \wedge y <_2 x <_2 z) \vee (z <_1 x <_1 y \wedge z <_2 y <_2 x)$$

(a so-called C-relation) makes $\text{Forb}(3142, 2413)$ an amalgamation class.

- $\mathcal{C}$: class of all separable permutations, equipped with the C-relation $xy|z$
- Δ: homogeneous ω-categorical Fraïssé-limit of $\mathcal{C}$.
- Γ: the $\{<_1, <_2\}$-reduct of Δ.

An ω-categorical separable template

Observation (Cameron'02): Expansion by ternary relation $xy|z$ defined by

$$(x <_1 y <_1 z \wedge y <_2 x <_2 z) \vee (z <_1 x <_1 y \wedge z <_2 y <_2 x)$$

(a so-called C-relation) makes $\text{Forb}(3142, 2413)$ an amalgamation class.

- $\mathcal{C}$: class of all separable permutations, equipped with the C-relation $xy|z$
- Δ: homogeneous ω-categorical Fraïssé-limit of $\mathcal{C}$.
- Γ: the $\{<_1, <_2\}$-reduct of Δ.
- Γ is an ω-categorical template for Hanika's problem.

An ω-categorical separable template

Observation (Cameron'02): Expansion by ternary relation $xy|z$ defined by

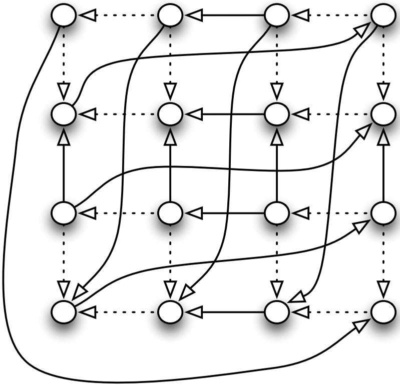
$$(x <_1 y <_1 z \wedge y <_2 x <_2 z) \vee (z <_1 x <_1 y \wedge z <_2 y <_2 x)$$

(a so-called C-relation) makes $\text{Forb}(3142, 2413)$ an amalgamation class.

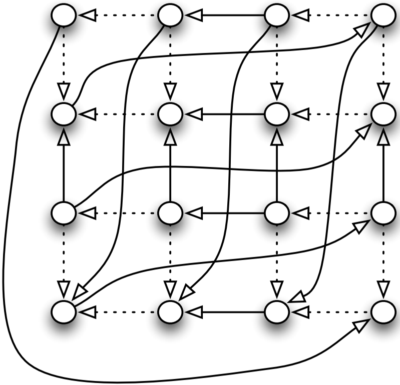
- $\mathcal{C}$: class of all separable permutations, equipped with the C-relation $xy|z$
- Δ: homogeneous ω-categorical Fraïssé-limit of $\mathcal{C}$.
- Γ: the $\{<_1, <_2\}$-reduct of Δ.
- Γ is an ω-categorical template for Hanika's problem.

What is the complexity of $\text{CSP}(\Gamma)$?

An Example

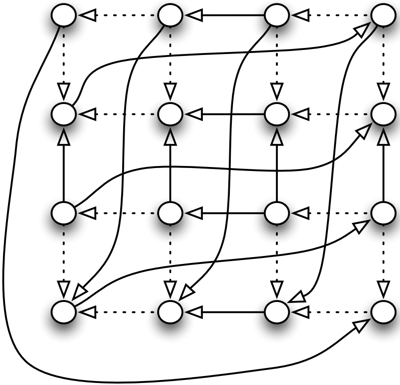


An Example



Datalog has been introduced in Alexander's talk.

An Example



Datalog has been introduced in Alexander's talk.

Hanika's problem CSP(Γ) is not in Datalog

A Polynomial-Time Algorithm

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
    Let C be an SCC.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
  Let C be an SCC.
  Recursively solve $G - C$.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
  Let C be an SCC.
  Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:
  Let C be an SCC.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.
Otherwise return 'false'.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
  Let C be an SCC.
  Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:
  Let C be an SCC.
  Recursively solve $G - C$.
Otherwise return 'false'.

Algorithm can be formulated in LFP (least fixed point logic).

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:
    Let C be an SCC.
    Recursively solve $G - C$.
Otherwise return 'false'.

Algorithm can be formulated in LFP (least fixed point logic).
Atserias+Bulatov+Dawar'07: LFP = Datalog for finite CSPs.

A Polynomial-Time Algorithm

Input: $G = (V; E_1, E_2)$
If $E_1 \cup E_2 = \emptyset$ return 'true'.
If $(V, E_1 \cup E_2)$ is not strongly connected:
  Let C be an SCC.
  Recursively solve $G - C$.
If $(V; E_1 \cup E_2^{-1})$ is not strongly connected:
  Let C be an SCC.
  Recursively solve $G - C$.
Otherwise return 'false'.

Algorithm can be formulated in LFP (least fixed point logic).
Atserias+Bulatov+Dawar'07: LFP = Datalog for finite CSPs.
For ω-categorical CSPs: $\text{CSP}(\Gamma) \in \text{LFP} \setminus \text{Datalog}$.

Forbidden Permutations CSPs

Let $\mathcal{S}$ be a class of finite permutations.

Forbidden Permutations CSPs

Let $\mathcal{S}$ be a class of finite permutations.

Definition (FPermCSP($\mathcal{S}$))

Input: A finite $\{E_1, E_2\}$-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2) \in \text{Forb}(\mathcal{S})$?

Forbidden Permutations CSPs

Let $\mathcal{S}$ be a class of finite permutations.

Definition (FPermCSP($\mathcal{S}$))

Input: A finite $\{E_1, E_2\}$-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2) \in \text{Forb}(\mathcal{S})$?

Challenge 2: Classify the computational complexity of all FPermCSPs.

Forbidden Permutations CSPs

Let $\mathcal{S}$ be a class of finite permutations.

Definition (FPermCSP($\mathcal{S}$))

Input: A finite $\{E_1, E_2\}$-structure.

Question: Are there linear extensions $<_1$ of E_1 and $<_2$ of E_2 such that $(V; <_1, <_2) \in \text{Forb}(\mathcal{S})$?

Challenge 2: Classify the computational complexity of all FPermCSPs.

Challenge 3: For which $\mathcal{S}$ can this problem be formulated as the CSP for an ω-categorical template?

SNP and Fragments

SNP and Fragments

An SNP τ-sentence is an ESO τ-sentence of the form

$$\exists R_1, \ldots, R_k. \forall x_1, \ldots, x_l. \phi$$

where ϕ is quantifier-free over the signature $\tau \cup \{R_1, \ldots, R_k\}$.

SNP and Fragments

An SNP τ-sentence is an ESO τ-sentence of the form

$$\exists R_1, \ldots, R_k.\, \forall x_1, \ldots, x_l.\, \phi$$

where ϕ is quantifier-free over the signature $\tau \cup \{R_1, \ldots, R_k\}$.

Example. $\tau = \{E\}$,

SNP and Fragments

An SNP τ-sentence is an ESO τ-sentence of the form

$$\exists R_1, \ldots, R_k.\, \forall x_1, \ldots, x_l.\, \phi$$

where ϕ is quantifier-free over the signature $\tau \cup \{R_1, \ldots, R_k\}$.

Example. $\tau = \{E\}$,

$$\begin{aligned} \exists T.\, \forall x, y, z.\, & \big(\neg E(x,y) \vee T(x,y)\big) \\ & \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \\ & \wedge \neg T(x,x) \end{aligned}$$

SNP and Fragments

An SNP τ-sentence is an ESO τ-sentence of the form

$$\exists R_1, \dots, R_k. \forall x_1, \dots, x_l. \phi$$

where ϕ is quantifier-free over the signature $\tau \cup \{R_1, \dots, R_k\}$.

Example. $\tau = \{E\}$,

$$\begin{aligned} \exists T. \forall x, y, z. & \big(\neg E(x,y) \vee T(x,y)\big) \\ & \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \\ & \wedge \neg T(x,x) \end{aligned}$$

An SNP sentence is called

- monotone if all symbols from τ appear negatively;

SNP and Fragments

An SNP τ-sentence is an ESO τ-sentence of the form

$$\exists R_1, \dots, R_k. \, \forall x_1, \dots, x_l. \, \phi$$

where ϕ is quantifier-free over the signature $\tau \cup \{R_1, \dots, R_k\}$.

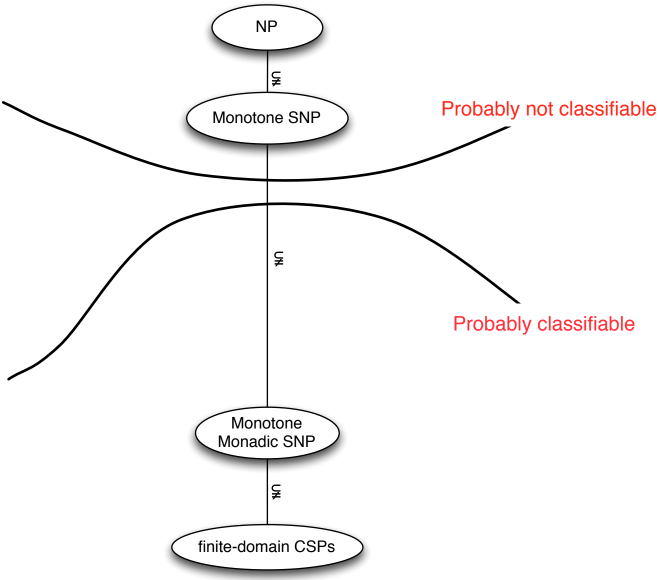
Example. $\tau = \{E\}$,

$$\begin{aligned} \exists T. \, \forall x, y, z. \, & \big(\neg E(x,y) \vee T(x,y)\big) \\ & \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \\ & \wedge \neg T(x,x) \end{aligned}$$

An SNP sentence is called

- monotone if all symbols from τ appear negatively;
- monadic if $R_1, \dots, R_k$ are unary.

The Results of Feder and Vardi



Connected SNP

What is the link between SNP and CSPs?

Connected SNP

What is the link between SNP and CSPs?

Let Φ be an SNP sentence.

Connected SNP

What is the link between SNP and CSPs?

Let Φ be an SNP sentence.
Assume wlog. that the first-order part ϕ of Φ is in CNF.

Connected SNP

What is the link between SNP and CSPs?

Let Φ be an SNP sentence.
Assume wlog. that the first-order part ϕ of Φ is in CNF.

Definition

Φ is called connected if for each clause in ϕ, the negative literals define a connected graph.

Connected SNP

What is the link between SNP and CSPs?

Let Φ be an SNP sentence.
Assume wlog. that the first-order part ϕ of Φ is in CNF.

Definition

Φ is called connected if for each clause in ϕ, the negative literals define a connected graph.

Example:

$$\begin{aligned} \exists T.\, \forall x, y, z.\, & \big(\neg E(x,y) \vee T(x,y)\big) \\ & \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \\ & \wedge \neg T(x,x) \end{aligned}$$

Connected SNP

What is the link between SNP and CSPs?

Let Φ be an SNP sentence.
Assume wlog. that the first-order part ϕ of Φ is in CNF.

Definition

Φ is called connected if for each clause in ϕ, the negative literals define a connected graph.

Example:

$$\exists T. \forall x, y, z. \big(\neg E(x,y) \vee T(x,y)\big) \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \wedge \neg T(x,x)$$

Proposition.

An SNP sentence Φ describes a problem of the form CSP(Γ) if and only if Φ is equivalent to a connected monotone SNP sentence.

MMSNP

Write MMSNP for Monotone Monadic SNP.

MMSNP

Write MMSNP for Monotone Monadic SNP.

Proposition.

- A monadic SNP sentence Φ describes a problem of the form CSP(Γ) if and only if Φ is logically equivalent to a connected monotone monadic SNP sentence

MMSNP

Write MMSNP for Monotone Monadic SNP.

Proposition.

- A monadic SNP sentence Φ describes a problem of the form CSP(Γ) if and only if Φ is logically equivalent to a connected monotone monadic SNP sentence
- Every MMSNP sentence is logically equivalent to a finite disjunction of connected MMSNP sentences.

MMSNP

Write MMSNP for Monotone Monadic SNP.

Proposition.

- A monadic SNP sentence Φ describes a problem of the form CSP(Γ) if and only if Φ is logically equivalent to a connected monotone monadic SNP sentence
- Every MMSNP sentence is logically equivalent to a finite disjunction of connected MMSNP sentences.

Easy:
every finite-domain CSP can be expressed by a connected MMSNP sentence.

MMSNP

Write MMSNP for Monotone Monadic SNP.

Proposition.

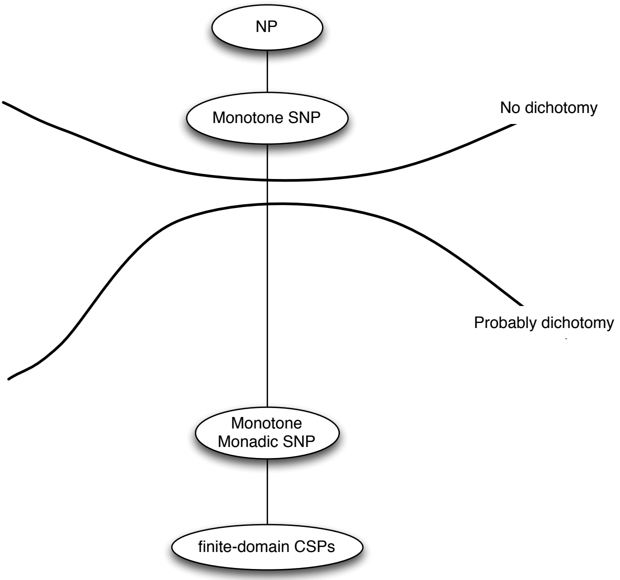
- A monadic SNP sentence Φ describes a problem of the form CSP(Γ) if and only if Φ is logically equivalent to a connected monotone monadic SNP sentence
- Every MMSNP sentence is logically equivalent to a finite disjunction of connected MMSNP sentences.

Easy:
every finite-domain CSP can be expressed by a connected MMSNP sentence.

Theorem (Feder-Vardi'97,Kun'13).

Every problem in MMSNP is computationally equivalent (under polynomial-time Turing reductions) to a finite domain CSP.

The Results of Feder and Vardi



$MMSNP_2$

Larger fragments of SNP which might permit complexity classification?

MMSNP_2

Larger fragments of SNP which might permit complexity classification?

Definition (As in Bienvenu+tenCate+Lutz+Wolter'PODS13)

An MMSNP_2 sentence (over the signature τ) is a monotone SNP sentence such that for every positive literal α there exists a negative input literal containing all variables from α.

$MMSNP_2$

Larger fragments of SNP which might permit complexity classification?

Definition (As in Bienvenu+tenCate+Lutz+Wolter'PODS13)

An $MMSNP_2$ sentence (over the signature τ) is a monotone SNP sentence such that for every positive literal α there exists a negative input literal containing all variables from α.

Non-Example:

$$\exists T. \forall x, y, z. \big(\neg E(x,y) \vee T(x,y)\big) \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \wedge \neg T(x,x)$$

MMSNP_2

Larger fragments of SNP which might permit complexity classification?

Definition (As in Bienvenu+tenCate+Lutz+Wolter'PODS13)

An MMSNP_2 sentence (over the signature τ) is a monotone SNP sentence such that for every positive literal α there exists a negative input literal containing all variables from α.

Non-Example:

$$\exists T.\, \forall x,y,z.\, \big(\neg E(x,y) \vee T(x,y)\big) \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \wedge \neg T(x,x)$$

Example: "Can we partition the edges of a given graph $(V;E)$ into two parts such that each of the parts does not contain a triangle?"

MMSNP_2

Larger fragments of SNP which might permit complexity classification?

Definition (As in Bienvenu+tenCate+Lutz+Wolter'PODS13)

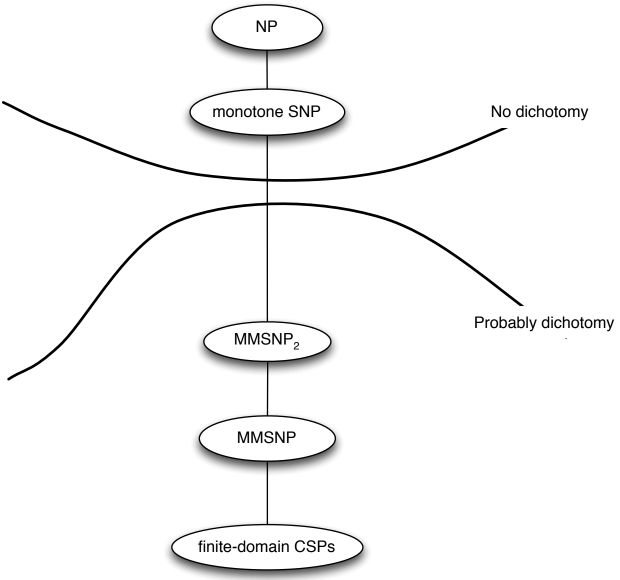
An MMSNP_2 sentence (over the signature τ) is a monotone SNP sentence such that for every positive literal α there exists a negative input literal containing all variables from α.

Non-Example:
$$\begin{aligned} \exists T.\, \forall x,y,z.\, & \big(\neg E(x,y) \vee T(x,y)\big) \\ & \wedge \big(\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\big) \\ & \wedge \neg T(x,x) \end{aligned}$$

Example: "Can we partition the edges of a given graph $(V;E)$ into two parts such that each of the parts does not contain a triangle?"

$$\begin{aligned} \exists R,G.\, \forall x,y,z.\, & \big(\neg E(x,y) \vee R(x,y) \vee G(x,y)\big) \\ & \wedge \big(\neg R(x,y) \vee \neg R(y,z) \vee \neg R(x,z)\big) \\ & \wedge \big(\neg G(x,y) \vee \neg G(y,z) \vee \neg G(x,z)\big) \end{aligned}$$

Dichotomy for $MMSNP_2$?



MMSNP$_2$ and Forbidden Patterns Problems

MMSNP_2 and Forbidden Patterns Problems

An edge-k-colored τ-structure is a τ-structure $\mathfrak{A}$ with a function from $R^{\mathfrak{A}} \to \{1, \ldots, k\}$ for every $R \in \tau$.

MMSNP_2 and Forbidden Patterns Problems

An edge-k-colored τ-structure is a τ-structure $\mathfrak{A}$ with a function from $R^{\mathfrak{A}} \to \{1, \ldots, k\}$ for every $R \in \tau$.

Definition (FPP_2)

A class $\mathcal{C}$ of finite τ-structures is in FPP_2 if there exists a $k \geq 1$ and a set of edge-k-colored connected structures $\mathcal{F}$ such that:

MMSNP_2 and Forbidden Patterns Problems

An edge-k-colored τ-structure is a τ-structure $\mathfrak{A}$ with a function from $R^{\mathfrak{A}} \to \{1, \ldots, k\}$ for every $R \in \tau$.

Definition (FPP_2)

A class $\mathcal{C}$ of finite τ-structures is in FPP_2 if there exists a $k \geq 1$ and a set of edge-k-colored connected structures $\mathcal{F}$ such that: $\mathfrak{A} \in \mathcal{C}$ if and only if $\mathfrak{A}$ has a k-edge coloring such that no structure from $\mathcal{F}$ embeds into $\mathfrak{A}$.

MMSNP_2 and Forbidden Patterns Problems

An edge-k-colored τ-structure is a τ-structure $\mathfrak{A}$ with a function from $R^{\mathfrak{A}} \to \{1, \ldots, k\}$ for every $R \in \tau$.

Definition (FPP_2)

A class $\mathcal{C}$ of finite τ-structures is in FPP_2 if there exists a $k \geq 1$ and a set of edge-k-colored connected structures $\mathcal{F}$ such that: $\mathfrak{A} \in \mathcal{C}$ if and only if $\mathfrak{A}$ has a k-edge coloring such that no structure from $\mathcal{F}$ embeds into $\mathfrak{A}$.

- The classes described by connected MMSNP_2 sentences are precisely the classes in FPP_2.

MMSNP$_2$ and Forbidden Patterns Problems

An edge-k-colored τ-structure is a τ-structure $\mathfrak{A}$ with a function from $R^{\mathfrak{A}} \to \{1, \dots, k\}$ for every $R \in \tau$.

Definition (FPP$_2$)

A class $\mathcal{C}$ of finite τ-structures is in FPP$_2$ if there exists a $k \geq 1$ and a set of edge-k-colored connected structures $\mathcal{F}$ such that: $\mathfrak{A} \in \mathcal{C}$ if and only if $\mathfrak{A}$ has a k-edge coloring such that no structure from $\mathcal{F}$ embeds into $\mathfrak{A}$.

- The classes described by connected MMSNP$_2$ sentences are precisely the classes in FPP$_2$.
- Every MMSNP$_2$ sentence describes a finite union of problems in FPP$_2$.

MMSNP_2 and Forbidden Patterns Problems

An edge-k-colored τ-structure is a τ-structure $\mathfrak{A}$ with a function from $R^{\mathfrak{A}} \to \{1, \ldots, k\}$ for every $R \in \tau$.

Definition (FPP_2)

A class $\mathcal{C}$ of finite τ-structures is in FPP_2 if there exists a $k \geq 1$ and a set of edge-k-colored connected structures $\mathcal{F}$ such that: $\mathfrak{A} \in \mathcal{C}$ if and only if $\mathfrak{A}$ has a k-edge coloring such that no structure from $\mathcal{F}$ embeds into $\mathfrak{A}$.

- The classes described by connected MMSNP_2 sentences are precisely the classes in FPP_2.
- Every MMSNP_2 sentence describes a finite union of problems in FPP_2.

Theorem (Cherlin+Shelah+Shi'98,MB+Dalmau'06,Madelaine'09).

Every problem in FPP_2 is the CSP for an ω-categorical template.

Cherlin-Shelah-Shi Structures

Let $\mathcal{F}$ be a finite set of finite relational structures.

Cherlin-Shelah-Shi Structures

Let $\mathcal{F}$ be a finite set of finite relational structures.
Forb$(\mathcal{F})$:
class of all finite structures that do not homomorphically embed any $F \in \mathcal{F}$.

Cherlin-Shelah-Shi Structures

Let $\mathcal{F}$ be a finite set of finite relational structures.
Forb$(\mathcal{F})$:
class of all finite structures that do not homomorphically embed any $F \in \mathcal{F}$.
Recall:
an ω-categorical Γ is called model complete iff Aut(Γ) is dense in Emb(Γ).

Cherlin-Shelah-Shi Structures

Let $\mathcal{F}$ be a finite set of finite relational structures.
Forb$(\mathcal{F})$:
class of all finite structures that do not homomorphically embed any $F \in \mathcal{F}$.
Recall:
an ω-categorical Γ is called model complete iff Aut(Γ) is dense in Emb(Γ).

Theorem (Cherlin-Shelah-Shi'98).

There exists an (up to isomorphism unique) model-complete ω-categorical structure Γ whose age is Forb$(\mathcal{F})$.

Cherlin-Shelah-Shi Structures

Let $\mathcal{F}$ be a finite set of finite relational structures.
Forb$(\mathcal{F})$:
class of all finite structures that do not homomorphically embed any $F \in \mathcal{F}$.
Recall:
an ω-categorical Γ is called model complete iff Aut(Γ) is dense in Emb(Γ).

Theorem (Cherlin-Shelah-Shi'98).

There exists an (up to isomorphism unique) model-complete ω-categorical structure Γ whose age is Forb$(\mathcal{F})$.

Nešetřil'11 (Bertinoro): all these structures have homogeneous Ramsey expansions in a finite relational signature.

Cherlin-Shelah-Shi Structures

Let $\mathcal{F}$ be a finite set of finite relational structures.
Forb$(\mathcal{F})$:
class of all finite structures that do not homomorphically embed any $F \in \mathcal{F}$.
Recall:
an ω-categorical Γ is called model complete iff Aut(Γ) is dense in Emb(Γ).

Theorem (Cherlin-Shelah-Shi'98).

There exists an (up to isomorphism unique) model-complete ω-categorical structure Γ whose age is Forb$(\mathcal{F})$.

Nešetřil'11 (Bertinoro): all these structures have homogeneous Ramsey expansions in a finite relational signature.

Consequence: Canonical function machine can be applied systematically to study MMSNP_2.

Further Challenges

Challenge 4: Classify the complexity of all problems in MMSNP_2.

Further Challenges

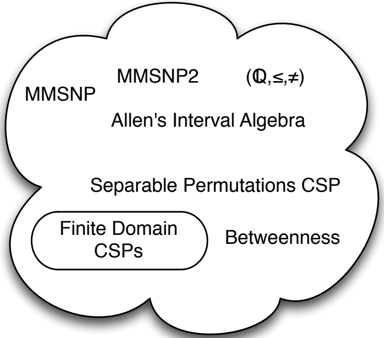
Challenge 4: Classify the complexity of all problems in $MMSNP_2$.

Challenge 5: Give a universal-algebraic characterization for tractability of problems in $MMSNP_1$.

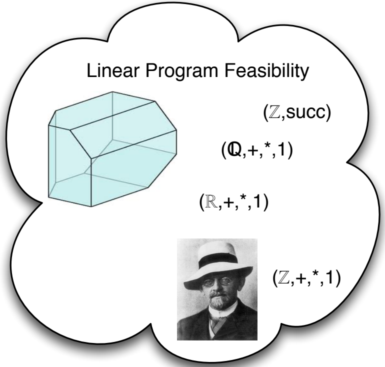
Qualitative vs Quantitative CSPs

Constraint Satisfaction Problems:

Qualitative CSPs



Quantitative CSPs



Infinite-Domain CSPs in P

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

- Linear Program Feasibility

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

- Linear Program Feasibility
- Linear Diophantine Equation Systems

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

- Linear Program Feasibility
- Linear Diophantine Equation Systems

Famous open problems:

- Max-Atoms (p-time equivalent to Mean-Payoff Games); CSP$(\mathbb{Q}; x \geq y \vee x \geq z, (\text{succ}^i)_i)$.

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

- Linear Program Feasibility
- Linear Diophantine Equation Systems

Famous open problems:

- Max-Atoms (p-time equivalent to Mean-Payoff Games); CSP$(\mathbb{Q}; x \geq y \vee x \geq z, (\text{succ}^i)_i)$.
- Feasibility of semi-definite programs

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

- Linear Program Feasibility
- Linear Diophantine Equation Systems

Famous open problems:

- Max-Atoms (p-time equivalent to Mean-Payoff Games); $\mathrm{CSP}(\mathbb{Q}; x \geq y \vee x \geq z, (\mathrm{succ}^i)_i)$.
- Feasibility of semi-definite programs

Zoo of less famous open problems:

- $\mathrm{CSP}(\mathbb{R}; 1, x + y = z, y \geq x^2)$

Infinite-Domain CSPs in P

Interesting algorithms in infinite-domain CSPs:

- Linear Program Feasibility
- Linear Diophantine Equation Systems

Famous open problems:

- Max-Atoms (p-time equivalent to Mean-Payoff Games); $\mathrm{CSP}(\mathbb{Q}; x \geq y \vee x \geq z, (\mathrm{succ}^i)_i)$.
- Feasibility of semi-definite programs

Zoo of less famous open problems:

- $\mathrm{CSP}(\mathbb{R}; 1, x + y = z, y \geq x^2)$
- $\mathrm{CSP}(\mathbb{Q}; x = 2y, x = y + 1, x \geq y \vee x \geq z)$

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

$R \subseteq \mathbb{R}^k$ is called essentially convex if for all $a, b \in R$ there are only finitely many points on the line between a and b that are not in R.

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

$R \subseteq \mathbb{R}^k$ is called essentially convex if for all $a, b \in R$ there are only finitely many points on the line between a and b that are not in R.

Example. CSP $\left(\mathbb{R}; \leq, +, 1, \neq, x = y \Rightarrow u = v\right)$

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

$R \subseteq \mathbb{R}^k$ is called essentially convex if for all $a, b \in R$ there are only finitely many points on the line between a and b that are not in R.

Example. CSP $\big(\mathbb{R}; \leq, +, 1, \neq, x = y \Rightarrow u = v\big)$

MB,Mamino'14: essential convexity can be described by preservation under a certain binary polymorphism over a sufficiently saturated model.

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

$R \subseteq \mathbb{R}^k$ is called essentially convex if for all $a, b \in R$ there are only finitely many points on the line between a and b that are not in R.

Example. CSP $\big(\mathbb{R}; \leq, +, 1, \neq, x = y \Rightarrow u = v\big)$

MB,Mamino'14: essential convexity can be described by preservation under a certain binary polymorphism over a sufficiently saturated model.

Theorem (MB+Jonsson+vonOertzen'09).

Let $\Gamma = \big(\mathbb{R}; \leq, +, 1, R_1, \ldots, R_n\big)$ where $R_1, \ldots, R_n$ are semilinear.

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

$R \subseteq \mathbb{R}^k$ is called essentially convex if for all $a, b \in R$ there are only finitely many points on the line between a and b that are not in R.

Example. CSP $(\mathbb{R}; \leq, +, 1, \neq, x = y \Rightarrow u = v)$

MB,Mamino'14: essential convexity can be described by preservation under a certain binary polymorphism over a sufficiently saturated model.

Theorem (MB+Jonsson+vonOertzen'09).

Let $\Gamma = (\mathbb{R}; \leq, +, 1, R_1, \ldots, R_n)$ where $R_1, \ldots, R_n$ are semilinear. Then either

- Γ is essentially convex, and CSP(Γ) is in P, or

Semilinear Relations

Definition.

$R \subseteq \mathbb{R}^k$ is semi-linear if R has a first-order definition in $(\mathbb{R}; \leq, +, 1)$.

$R \subseteq \mathbb{R}^k$ is called essentially convex if for all $a, b \in R$ there are only finitely many points on the line between a and b that are not in R.

Example. CSP $\big(\mathbb{R}; \leq, +, 1, \neq, x = y \Rightarrow u = v\big)$

MB,Mamino'14: essential convexity can be described by preservation under a certain binary polymorphism over a sufficiently saturated model.

Theorem (MB+Jonsson+vonOertzen'09).

Let $\Gamma = \big(\mathbb{R}; \leq, +, 1, R_1, \ldots, R_n\big)$ where $R_1, \ldots, R_n$ are semilinear. Then either

- Γ is essentially convex, and CSP(Γ) is in P, or
- there is a pp interpretation of $(\{0, 1, 2\}; \neq)$ in Γ, and CSP(Γ) is NP-complete.

Semi-algebraic Relations

Definition.

Call $R \subseteq \mathbb{R}^k$ semi-algebraic if R has a first-order definition in $(\mathbb{R}; +, *)$ with parameters from $\mathbb{R}$.

Semi-algebraic Relations

Definition.

Call $R \subseteq \mathbb{R}^k$ semi-algebraic if R has a first-order definition in $(\mathbb{R}; +, *)$ with parameters from $\mathbb{R}$.

Example. The binary relation $\{(x, y) \mid y \geq x^2\}$ is semi-algebraic.

Semi-algebraic Relations

Definition.

Call $R \subseteq \mathbb{R}^k$ semi-algebraic if R has a first-order definition in $(\mathbb{R}; +, *)$ with parameters from $\mathbb{R}$.

Example. The binary relation $\{(x, y) \mid y \geq x^2\}$ is semi-algebraic.

Theorem (Quantifier-elimination; Tarski-Seidenberg).

A relation is semi-algebraic if and only if is has a quantifier-free definition over $(\mathbb{R}; +, *, <)$ with parameters from $\mathbb{R}$.

Semi-algebraic Relations

Definition.

Call $R \subseteq \mathbb{R}^k$ semi-algebraic if R has a first-order definition in $(\mathbb{R}; +, *)$ with parameters from $\mathbb{R}$.

Example. The binary relation $\{(x, y) \mid y \geq x^2\}$ is semi-algebraic.

Theorem (Quantifier-elimination; Tarski-Seidenberg).

A relation is semi-algebraic if and only if is has a quantifier-free definition over $(\mathbb{R}; +, *, <)$ with parameters from $\mathbb{R}$.

Challenge 6: Is CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$ in P?

Convex Semialgebraic CSPs

Convex Semialgebraic CSPs

Proposition.

A semi-algebraic relation is convex if and only if

Convex Semialgebraic CSPs

Proposition.

A semi-algebraic relation is convex if and only if it is preserved by $(x, y) \mapsto (x + y)/2$.

Convex Semialgebraic CSPs

Proposition.

A semi-algebraic relation is convex if and only if it is preserved by $(x, y) \mapsto (x + y)/2$.

Many interesting open problems concerning convexity:

Convex Semialgebraic CSPs

Proposition.

A semi-algebraic relation is convex if and only if it is preserved by $(x, y) \mapsto (x + y)/2$.

Many interesting open problems concerning convexity:

- Conjecture of Helton+Nie'09: a semi-algebraic relation is convex if and only if it is primitive positive definable over relations definable by semi-definite programs

Convex Semialgebraic CSPs

Proposition.

A semi-algebraic relation is convex if and only if it is preserved by $(x, y) \mapsto (x + y)/2$.

Many interesting open problems concerning convexity:

- Conjecture of Helton+Nie'09: a semi-algebraic relation is convex if and only if it is primitive positive definable over relations definable by semi-definite programs
- Ramana'97: Is feasibility of semi-definite programs in P?

Convex Semialgebraic CSPs

Proposition.

A semi-algebraic relation is convex if and only if it is preserved by $(x, y) \mapsto (x + y)/2$.

Many interesting open problems concerning convexity:

- Conjecture of Helton+Nie'09: a semi-algebraic relation is convex if and only if it is primitive positive definable over relations definable by semi-definite programs
- Ramana'97: Is feasibility of semi-definite programs in P?
- More general than CSP $\left(\mathbb{R}; \leq, +, 1, y \geq x^2\right)$

Numerical Complications

Can the ellipsoid algorithm be applied to solve CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$?

A well-known numerical problem:

Numerical Complications

Can the ellipsoid algorithm be applied to solve CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$?

A well-known numerical problem:

Definition (Sums-of-Square-Roots)

Input: $a_0, a_1, \ldots, a_n \in \mathbb{N}$ (represented in binary)

Question: $\sum_{i=1}^{n} \sqrt{a_i} \geq a_0$?

Numerical Complications

Can the ellipsoid algorithm be applied to solve CSP $\left(\mathbb{R}; \leq, +, 1, y \geq x^2\right)$?

A well-known numerical problem:

Definition (Sums-of-Square-Roots)

Input: $a_0, a_1, \ldots, a_n \in \mathbb{N}$ (represented in binary)

Question: $\sum_{i=1}^{n} \sqrt{a_i} \geq a_0$?

Reduction from Sums-of-Square-Roots to CSP $\left(\mathbb{R}; \leq, +, 1, y \geq x^2\right)$:

Numerical Complications

Can the ellipsoid algorithm be applied to solve CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$?

A well-known numerical problem:

Definition (Sums-of-Square-Roots)

Input: $a_0, a_1, \ldots, a_n \in \mathbb{N}$ (represented in binary)

Question: $\sum_{i=1}^n \sqrt{a_i} \geq a_0$?

Reduction from Sums-of-Square-Roots to CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$:

$$\sum_{i=1}^{n} \sqrt{a_i} \geq a_0 \Leftrightarrow \exists b_1, \ldots, b_n \left[\bigwedge_{i \geq 1}^{n} a_i \geq b_i^2 \wedge b_1 + \cdots + b_n \geq a_0 \right]$$

Numerical Complications

Can the ellipsoid algorithm be applied to solve CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$?

A well-known numerical problem:

Definition (Sums-of-Square-Roots)

Input: $a_0, a_1, \ldots, a_n \in \mathbb{N}$ (represented in binary)

Question: $\sum_{i=1}^{n} \sqrt{a_i} \geq a_0$?

Reduction from Sums-of-Square-Roots to CSP $(\mathbb{R}; \leq, +, 1, y \geq x^2)$:

$$\sum_{i=1}^{n} \sqrt{a_i} \geq a_0 \Leftrightarrow \exists b_1, \ldots, b_n \left[\bigwedge_{i \geq 1}^{n} a_i \geq b_i^2 \wedge b_1 + \cdots + b_n \geq a_0 \right]$$

Challenge 7: Does every semi-algebraic extension of linear program feasibility which is non semilinear simulate a sums-of-roots problem?

Non-Dichotomies

Non-Dichotomies

Known:

- Feder'93: every problem in NP is polynomial-time equivalent to a problem in monotone SNP

Non-Dichotomies

Known:

- Feder'93: every problem in NP is polynomial-time equivalent to a problem in monotone SNP
- MB'12: every decidable problem is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *, 1)$.

Non-Dichotomies

Known:

- Feder'93: every problem in NP is polynomial-time equivalent to a problem in monotone SNP
- MB'12: every decidable problem is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *, 1)$.

It would be good to have more non-dichotomy results.

Non-Dichotomies

Known:

- Feder'93: every problem in NP is polynomial-time equivalent to a problem in monotone SNP
- MB'12: every decidable problem is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *, 1)$.

It would be good to have more non-dichotomy results.

Challenge 8: Do CSPs for reducts of $(\mathbb{Z}; \text{succ}, y = 2x)$ have a non-dichotomy?

Non-Dichotomies

Known:

- Feder'93: every problem in NP is polynomial-time equivalent to a problem in monotone SNP
- MB'12: every decidable problem is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *, 1)$.

It would be good to have more non-dichotomy results.

Challenge 8: Do CSPs for reducts of $(\mathbb{Z}; \text{succ}, y = 2x)$ have a non-dichotomy?

Challenge 9: Do CSPs for reducts of $(\mathbb{Z}; 0, \text{succ})$ have a non-dichotomy?

AAA89 in Dresden

89th Workshop On General Algebra: February 26 (arrival) - March 1, 2015

Invited speakers:

Libor Barto, Georg Grätzer, Maja Pech, Gernot Salzer, John Truss