The Complexity of Constraint Satisfaction Problems

Manuel Bodirsky

Institut für Algebra, TU Dresden

March 2015

Outline

1. Constraint Satisfaction Problems: definition and examples

Outline

1. Constraint Satisfaction Problems: definition and examples
2. Complexity classification of finite domain CSPs:
   The universal-algebraic approach

Outline

1. Constraint Satisfaction Problems: definition and examples
2. Complexity classification of finite domain CSPs:
   The universal-algebraic approach
3. The algebraic approach for infinite-domain CSPs

Outline

1. Constraint Satisfaction Problems: definition and examples
2. Complexity classification of finite domain CSPs:
   The universal-algebraic approach
3. The algebraic approach for infinite-domain CSPs
4. Complexity of CSPs over the integers, the rationals, and the reals.

Constraint Satisfaction Problems

Let Γ be a structure with a finite relational signature τ.
Γ also called the template.

Constraint Satisfaction Problems

Let Γ be a structure with a finite relational signature τ.
Γ also called the template.

Definition 1 (CSP).

$\mathrm{CSP}(\Gamma)$ is the computational problem to decide whether a given finite τ-structure A homomorphically maps to Γ.

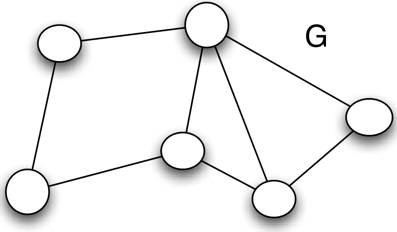
Constraint Satisfaction Problems

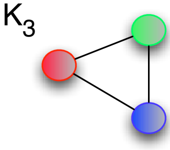
Let Γ be a structure with a finite relational signature τ.
Γ also called the template.

Definition 1 (CSP).

CSP(Γ) is the computational problem to decide whether a given finite τ-structure A homomorphically maps to Γ.

Example: 3-colorability is CSP(K_3)





More Examples of CSPs

Positive 1-in-3-3SAT

Input: A set V and a subset T of V^3.

Question: Is there a map $V \rightarrow \{0,1\}$ such that exactly one entry in each triple in T is mapped to 1?

More Examples of CSPs

Positive 1-in-3-3SAT

Input: A set V and a subset T of V^3.

Question: Is there a map $V \to \{0, 1\}$ such that exactly one entry in each triple in T is mapped to 1?

Is a CSP: Template is $\big(\{0, 1\}; \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}\big)$

More Examples of CSPs

Positive 1-in-3-3SAT

Input: A set V and a subset T of V^3.

Question: Is there a map $V \to \{0,1\}$ such that exactly one entry in each triple in T is mapped to 1?

Is a CSP: Template is $\big(\{0,1\}; \{(0,0,1),(0,1,0),(1,0,0)\}\big)$

Complexity: NP-complete (Garey and Johnson)

More Examples of CSPs

Positive 1-in-3-3SAT

Input: A set V and a subset T of V^3.

Question: Is there a map $V \to \{0, 1\}$ such that exactly one entry in each triple in T is mapped to 1?

Is a CSP: Template is $\big(\{0, 1\}; \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}\big)$

Complexity: NP-complete (Garey and Johnson)

Digraph Acyclicity

Input: A directed graph $(V; E)$

Question: Is $(V; E)$ acyclic?

More Examples of CSPs

Positive 1-in-3-3SAT

Input: A set V and a subset T of V^3.

Question: Is there a map $V \to \{0,1\}$ such that exactly one entry in each triple in T is mapped to 1?

Is a CSP: Template is $\big(\{0,1\};\ \{(0,0,1),(0,1,0),(1,0,0)\}\big)$

Complexity: NP-complete (Garey and Johnson)

Digraph Acyclicity

Input: A directed graph $(V;E)$

Question: Is $(V;E)$ acyclic?

Is a CSP: Template is $(\mathbb{Q};<)$

More Examples of CSPs

Positive 1-in-3-3SAT

Input: A set V and a subset T of V^3.

Question: Is there a map $V \to \{0,1\}$ such that exactly one entry in each triple in T is mapped to 1?

Is a CSP: Template is $\big(\{0,1\}; \{(0,0,1),(0,1,0),(1,0,0)\}\big)$

Complexity: NP-complete (Garey and Johnson)

Digraph Acyclicity

Input: A directed graph $(V;E)$

Question: Is $(V;E)$ acyclic?

Is a CSP: Template is $(\mathbb{Q};<)$

Complexity: In P (e.g. by depth-first search)

Logic Formulation of the CSP

Let $\Gamma = (D; R_1, \dots, R_l)$ be a τ-structure.

Logic Formulation of the CSP

Let $\Gamma = (D; R_1, \ldots, R_l)$ be a τ-structure.

CSP(Γ)

Input: A primitive positive τ-sentence Φ,
i.e., a first-order sentence of the form

$$\exists x_1, \ldots, x_n \, (\psi_1 \wedge \cdots \wedge \psi_l)$$

where ψ_i are atomic, i.e. of the form $R(x_{i_1}, \ldots, x_{i_k})$ for $R \in \tau$.

Logic Formulation of the CSP

Let $\Gamma = (D; R_1, \ldots, R_l)$ be a τ-structure.

CSP(Γ)

Input: A primitive positive τ-sentence Φ,
i.e., a first-order sentence of the form

$$\exists x_1, \ldots, x_n \, (\psi_1 \wedge \cdots \wedge \psi_l)$$

where ψ_i are atomic, i.e. of the form $R(x_{i_1}, \ldots, x_{i_k})$ for $R \in \tau$.

Question: Is Φ true in Γ?

Logic Formulation of the CSP

Let $\Gamma = (D; R_1, \ldots, R_l)$ be a τ-structure.

CSP(Γ)

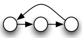
Input: A primitive positive τ-sentence Φ,
i.e., a first-order sentence of the form

$$\exists x_1, \ldots, x_n \, (\psi_1 \wedge \cdots \wedge \psi_l)$$

where ψ_i are atomic, i.e. of the form $R(x_{i_1}, \ldots, x_{i_k})$ for $R \in \tau$.

Question: Is Φ true in Γ?

Example:



not homomorphic to $(\mathbb{Q}; <)$.

$\exists x_1, x_2, x_3 \, (x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1)$ is false in $(\mathbb{Q}; <)$.

More Examples of CSPs

Betweenness:

Input: A finite set V, and a subset S of V^3.

Question: Is there a linear order $<$ on V such that for every $(u, v, w) \in S$ we have $u < v < w$ or $w < v < u$?

More Examples of CSPs

Betweenness:

Input: A finite set V, and a subset S of V^3.

Question: Is there a linear order $<$ on V such that for every $(u, v, w) \in S$ we have $u < v < w$ or $w < v < u$?

Is a CSP: template is $(\mathbb{Q}; \text{Betw})$ where $\text{Betw} := \{(x, y, z) \mid (x < y < z) \vee (z < y < x)\}$

More Examples of CSPs

Betweenness:

Input: A finite set V, and a subset S of V^3.

Question: Is there a linear order $<$ on V such that for every $(u, v, w) \in S$ we have $u < v < w$ or $w < v < u$?

Is a CSP: template is $\big(\mathbb{Q}; \text{Betw}\big)$ where
$\text{Betw} := \{(x, y, z) \mid (x < y < z) \vee (z < y < x)\}\big)$

Complexity: NP-complete (Garey and Johnson)

More Examples of CSPs

Betweenness:

Input: A finite set V, and a subset S of V^3.

Question: Is there a linear order $<$ on V such that for every $(u, v, w) \in S$ we have $u < v < w$ or $w < v < u$?

Is a CSP: template is $\big(\mathbb{Q}; \text{Betw}\big)$ where
$\text{Betw} := \{(x, y, z) \mid (x < y < z) \vee (z < y < x)\}\big)$

Complexity: NP-complete (Garey and Johnson)

And/Or Precedence Constraints:

Input: A finite set V, and a subset T of V^3.

Question: Is there a linear order $<$ on V such that $u < v$ or $u < w$ for all $(u, v, w) \in T$.

More Examples of CSPs

Betweenness:

Input: A finite set V, and a subset S of V^3.

Question: Is there a linear order $<$ on V such that for every $(u, v, w) \in S$ we have $u < v < w$ or $w < v < u$?

Is a CSP: template is $(\mathbb{Q}; \text{Betw})$ where $\text{Betw} := \{(x, y, z) \mid (x < y < z) \vee (z < y < x)\}$

Complexity: NP-complete (Garey and Johnson)

And/Or Precedence Constraints:

Input: A finite set V, and a subset T of V^3.

Question: Is there a linear order $<$ on V such that $u < v$ or $u < w$ for all $(u, v, w) \in T$.

Is a CSP: template is $(\mathbb{Q}; \{(u, v, w) \mid u < max(v, w)\})$

More Examples of CSPs

Betweenness:

Input: A finite set V, and a subset S of V^3.

Question: Is there a linear order $<$ on V such that for every $(u, v, w) \in S$ we have $u < v < w$ or $w < v < u$?

Is a CSP: template is $(\mathbb{Q}; \text{Betw})$ where $\text{Betw} := \{(x, y, z) \mid (x < y < z) \vee (z < y < x)\}$

Complexity: NP-complete (Garey and Johnson)

And/Or Precedence Constraints:

Input: A finite set V, and a subset T of V^3.

Question: Is there a linear order $<$ on V such that $u < v$ or $u < w$ for all $(u, v, w) \in T$.

Is a CSP: template is $(\mathbb{Q}; \{(u, v, w) \mid u < max(v, w)\})$

Complexity: in P

CSPs of Open Complexity

- CSP$(\mathbb{R}; <, R_+, R_{=1}, R_{sq})$ where
  - $R_+ := \{(x, y, z) \mid x = y + z\}$,
  - $R_{=1} := \{1\}$, and
  - $R_{sq} := \{(x, y) \mid y \geq x^2\}$.

CSPs of Open Complexity

- CSP$(\mathbb{R}; <, R_+, R_{=1}, R_{sq})$ where
  - $R_+ := \{(x, y, z) \mid x = y + z\}$,
  - $R_{=1} := \{1\}$, and
  - $R_{sq} := \{(x, y) \mid y \geq x^2\}$.

  At least as hard as Sums of Square Roots.

CSPs of Open Complexity

- CSP$(\mathbb{R}; <, R_+, R_{=1}, R_{sq})$ where
  - $R_+ := \{(x, y, z) \mid x = y + z\}$,
  - $R_{=1} := \{1\}$, and
  - $R_{sq} := \{(x, y) \mid y \geq x^2\}$.

  At least as hard as Sums of Square Roots.

  Not known to be in P.

CSPs of Open Complexity

- CSP$(\mathbb{R}; <, R_+, R_{=1}, R_{sq})$ where
  - $R_+ := \{(x, y, z) \mid x = y + z\}$,
  - $R_{=1} := \{1\}$, and
  - $R_{sq} := \{(x, y) \mid y \geq x^2\}$.

  At least as hard as Sums of Square Roots.

  Not known to be in P.

- CSP$(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$ where
  - $S_{+1} := \{(x, y) \mid y = x + 1\}$,
  - $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
  - $S_{max} := \{(x, y, z) \mid x \leq max(y, z)\}$.

CSPs of Open Complexity

- CSP$(\mathbb{R}; <, R_+, R_{=1}, R_{sq})$ where
  - $R_+ := \{(x, y, z) \mid x = y + z\}$,
  - $R_{=1} := \{1\}$, and
  - $R_{sq} := \{(x, y) \mid y \geq x^2\}$.

  At least as hard as Sums of Square Roots.
  Not known to be in P.

- CSP$(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$ where
  - $S_{+1} := \{(x, y) \mid y = x + 1\}$,
  - $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
  - $S_{max} := \{(x, y, z) \mid x \leq max(y, z)\}$.

  At least as hard as Mean Payoff Games.

CSPs of Open Complexity

- CSP$(\mathbb{R}; <, R_+, R_{=1}, R_{sq})$ where
  - $R_+ := \{(x, y, z) \mid x = y + z\}$,
  - $R_{=1} := \{1\}$, and
  - $R_{sq} := \{(x, y) \mid y \geq x^2\}$.

  At least as hard as Sums of Square Roots.
  Not known to be in P.

- CSP$(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$ where
  - $S_{+1} := \{(x, y) \mid y = x + 1\}$,
  - $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
  - $S_{max} := \{(x, y, z) \mid x \leq max(y, z)\}$.

  At least as hard as Mean Payoff Games.
  Not known to be in P.

The Feder-Vardi Dichotomy Conjecture

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

Conjecture 1 (Feder-Vardi'93).

For finite Γ, the problem CSP(Γ) is either in P or NP-complete.

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

Conjecture 1 (Feder-Vardi'93).

For finite Γ, the problem CSP(Γ) is either in P or NP-complete.

This dichotomy has been confirmed in many special cases, for example

- For 2-element structures Γ (Schaefer'78)
  and 3-element structures Γ (Bulatov'06)

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

Conjecture 1 (Feder-Vardi'93).

For finite Γ, the problem CSP(Γ) is either in P or NP-complete.

This dichotomy has been confirmed in many special cases, for example

- For 2-element structures Γ (Schaefer'78)
  and 3-element structures Γ (Bulatov'06)
- For undirected graphs (Hell+Nešetřil'90)
  and digraphs without sources and sinks (Barto+Kozik+Niven'08)

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

Conjecture 1 (Feder-Vardi'93).

For finite Γ, the problem CSP(Γ) is either in P or NP-complete.

This dichotomy has been confirmed in many special cases, for example

- For 2-element structures Γ (Schaefer'78)
  and 3-element structures Γ (Bulatov'06)
- For undirected graphs (Hell+Nešetřil'90)
  and digraphs without sources and sinks (Barto+Kozik+Niven'08)
- Open for digraphs Γ

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

Conjecture 1 (Feder-Vardi'93).

For finite Γ, the problem CSP(Γ) is either in P or NP-complete.

This dichotomy has been confirmed in many special cases, for example

- For 2-element structures Γ (Schaefer'78)
  and 3-element structures Γ (Bulatov'06)
- For undirected graphs (Hell+Nešetřil'90)
  and digraphs without sources and sinks (Barto+Kozik+Niven'08)
- Open for digraphs Γ
- Open for 5-element structures Γ

The Feder-Vardi Dichotomy Conjecture

What can be said about CSP(Γ) when Γ is a finite structure?

Conjecture 1 (Feder-Vardi'93).

For finite Γ, the problem CSP(Γ) is either in P or NP-complete.

This dichotomy has been confirmed in many special cases, for example

- For 2-element structures Γ (Schaefer'78)
  and 3-element structures Γ (Bulatov'06)
- For undirected graphs (Hell+Nešetřil'90)
  and digraphs without sources and sinks (Barto+Kozik+Niven'08)
- Open for digraphs Γ
- Open for 5-element structures Γ

Strongest evidence comes from the so-called universal algebraic approach.

Primitive Positive Definability

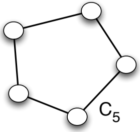
Lemma (Jeavons et al'97).

Let $\Gamma = (D; R_1, \ldots, R_k)$ be a relational structure, and
let R be a relation that has a primitive positive definition in Γ.
Then $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(D; R, R_1, \ldots, R_k)$ are polynomial-time equivalent.

Primitive Positive Definability

Lemma (Jeavons et al'97).

Let $\Gamma = (D; R_1, \ldots, R_k)$ be a relational structure, and
let R be a relation that has a primitive positive definition in Γ.
Then CSP(Γ) and CSP$(D; R, R_1, \ldots, R_k)$ are polynomial-time equivalent.

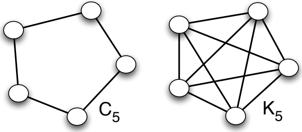


Example. Claim: CSP(C_5) is NP-hard.

Primitive Positive Definability

Lemma (Jeavons et al'97).

Let $\Gamma = (D; R_1, \ldots, R_k)$ be a relational structure, and
let R be a relation that has a primitive positive definition in Γ.
Then CSP(Γ) and CSP$(D; R, R_1, \ldots, R_k)$ are polynomial-time equivalent.

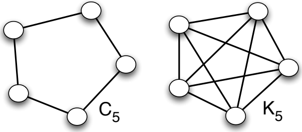


Example. Claim: CSP(C_5) is NP-hard.
Proof: $K_5 = (V; E')$ has a primitive positive definition in $C_5 = (V; E)$

Primitive Positive Definability

Lemma (Jeavons et al'97).

Let $\Gamma = (D; R_1, \ldots, R_k)$ be a relational structure, and let R be a relation that has a primitive positive definition in Γ. Then CSP(Γ) and CSP$(D; R, R_1, \ldots, R_k)$ are polynomial-time equivalent.



Example. Claim: CSP(C_5) is NP-hard.

Proof: $K_5 = (V; E')$ has a primitive positive definition in $C_5 = (V; E)$

$$E'(x,y) \equiv \exists p_1, p_2, p_3, q_1, q_2 \big(E(x,p_1) \wedge E(p_1,p_2) \wedge E(p_2,p_3) \wedge E(p_3,y) \wedge E(x,q_1) \wedge E(q_1,q_2) \wedge E(q_2,y)\big)$$

Polymorphisms

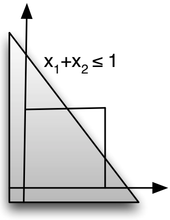
Polymorphisms

A function $f\colon D^k \to D$ preserves $R \subseteq D^m$ if $\big(f(a_1^1,\dots,a_1^k),\dots,f(a_m^1,\dots,a_m^k)\big) \in R$ whenever $(a_1^i,\dots,a_m^i) \in R$ for all $i \leq k$.

Polymorphisms

A function $f\colon D^k \to D$ preserves $R \subseteq D^m$ if
$\big(f(a_1^1, \dots, a_1^k), \dots, f(a_m^1, \dots, a_m^k)\big) \in R$ whenever $(a_1^i, \dots, a_m^i) \in R$ for all $i \leq k$.

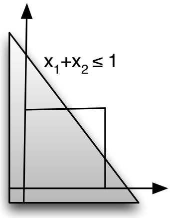
Example: $(x, y) \mapsto max(x, y)$ preserves a linear half-space given by $a_1 x_1 + \cdots + a_n x_n \leq a_0$ iff at most one of $a_1, \dots, a_n$ is positive.



Polymorphisms

A function $f\colon D^k \rightarrow D$ preserves $R \subseteq D^m$ if
$\left(f(a_1^1, \ldots, a_1^k), \ldots, f(a_m^1, \ldots, a_m^k)\right) \in R$ whenever $(a_1^i, \ldots, a_m^i) \in R$ for all $i \leq k$.

Example: $(x, y) \mapsto max(x, y)$ preserves a linear half-space given by
$a_1 x_1 + \cdots + a_n x_n \leq a_0$ iff at most one of $a_1, \ldots, a_n$ is positive.

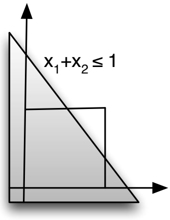


We say that f is a polymorphism of Γ if f preserves all relations of Γ.

Polymorphisms

A function $f\colon D^k \to D$ preserves $R \subseteq D^m$ if
$\big(f(a_1^1,\dots,a_1^k),\dots,f(a_m^1,\dots,a_m^k)\big) \in R$ whenever $(a_1^i,\dots,a_m^i) \in R$ for all $i \leq k$.

Example: $(x,y) \mapsto max(x,y)$ preserves a linear half-space given by $a_1x_1 + \cdots + a_nx_n \leq a_0$ iff at most one of $a_1,\dots,a_n$ is positive.



We say that f is a polymorphism of Γ if f preserves all relations of Γ.

Example: Every structure Γ has the projections as polymorphisms.

Polymorphisms and Primitive Positive Definability

Equivalent definition:
Polymorphisms are homomorphisms from Γ^k to Γ.

Polymorphisms and Primitive Positive Definability

Equivalent definition:
Polymorphisms are homomorphisms from Γ^k to Γ.

Theorem (Geiger'68, Bodnarcuk et al'69).

Let Γ be finite. Then a relation R has a primitive positive definition in Γ if and only if R is preserved by all finitary polymorphisms of Γ.

Polymorphisms and Primitive Positive Definability

Equivalent definition:
Polymorphisms are homomorphisms from Γ^k to Γ.

Theorem (Geiger'68, Bodnarcuk et al'69).

Let Γ be finite. Then a relation R has a primitive positive definition in Γ if and only if R is preserved by all finitary polymorphisms of Γ.

Polymorphisms $\leftrightarrow$ Algorithms

Weak Near Unanimities

Weak Near Unanimities

Assume that Γ has a finite domain D.

Theorem (Bulatov+Jeavons+Krokhin'05,Maroti+McKenzie'08).

Let Γ be a finite structure. Then Γ has a weak near unanimity polymorphism of arity $n \geq 2$, this is, a polymorphism f such that for all elements x, y of Γ

$$f(y,x,\dots,x) = f(x,y,\dots,x) = \cdots = f(x,\dots,x,y) \; ,$$

or CSP(Γ) is NP-hard.

Weak Near Unanimities

Assume that Γ has a finite domain D.

Theorem (Bulatov+Jeavons+Krokhin'05,Maroti+McKenzie'08).

Let Γ be a finite structure. Then Γ has a weak near unanimity polymorphism of arity $n \geq 2$, this is, a polymorphism f such that for all elements x, y of Γ

$$f(y, x, \ldots, x) = f(x, y, \ldots, x) = \cdots = f(x, \ldots, x, y) ,$$

or $\mathrm{CSP}(\Gamma)$ is NP-hard.

Example:

$$(x, y) \mapsto max(x, y)$$

is a weak near unanimity polymorphism of $(\mathbb{Q}; <)$.

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then $\mathrm{CSP}(\Gamma)$ is in P.

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then $\text{CSP}(\Gamma)$ is in P.

Confirmed for the following polymorphisms:

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then $\mathrm{CSP}(\Gamma)$ is in P.

Confirmed for the following polymorphisms:

- majority, that is, satisfies $\forall x, y.\ f(x,x,y) = f(x,y,x) = f(y,x,x) = x$.

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then CSP(Γ) is in P.

Confirmed for the following polymorphisms:

- majority, that is, satisfies $\forall x, y.\ f(x, x, y) = f(x, y, x) = f(y, x, x) = x$.
- Maltsev, that is, satisfies $\forall x, y.\ f(x, y, y) = f(y, y, x) = x$.

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then CSP(Γ) is in P.

Confirmed for the following polymorphisms:

- majority, that is, satisfies $\forall x, y.\ f(x,x,y) = f(x,y,x) = f(y,x,x) = x$.
- Maltsev, that is, satisfies $\forall x, y.\ f(x,y,y) = f(y,y,x) = x$.
- semi-lattice, that is, is binary commutative, associative, idempotent.
- ...

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then CSP(Γ) is in P.

Confirmed for the following polymorphisms:

- majority, that is, satisfies $\forall x, y.\ f(x, x, y) = f(x, y, x) = f(y, x, x) = x$.
- Maltsev, that is, satisfies $\forall x, y.\ f(x, y, y) = f(y, y, x) = x$.
- semi-lattice, that is, is binary commutative, associative, idempotent.
- ...

There are two algorithmic techniques to obtain those results:

The Finite-Domain Tractability Conjecture

Bulatov+Jeavons+Krokhin'04 (in different, but equivalent form):

Conjecture 2.

If Γ has a weak near unanimity polymorphism, then CSP(Γ) is in P.

Confirmed for the following polymorphisms:

- majority, that is, satisfies $\forall x, y.\ f(x,x,y) = f(x,y,x) = f(y,x,x) = x$.
- Maltsev, that is, satisfies $\forall x, y.\ f(x,y,y) = f(y,y,x) = x$.
- semi-lattice, that is, is binary commutative, associative, idempotent.
- ...

There are two algorithmic techniques to obtain those results:

- Generalizations of Gaussian elimination
  (works for example when Γ has Maltsev polymorphism)
- 'Constraint Propagation' / Datalog

CSPs and Datalog

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI
Elegant formalism. “Datalog = Prolog without function symbols”.

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI
Elegant formalism. “Datalog = Prolog without function symbols”.

Example: CSP$(\mathbb{Q};<)$ can be solved by Datalog.

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI
Elegant formalism. “Datalog = Prolog without function symbols”.

Example: CSP$(\mathbb{Q}; <)$ can be solved by Datalog.

$$\begin{aligned} tc(x,y) &:- \; x < y \\ tc(x,y) &:- \; tc(x,z), z < y \\ false() &:- \; tc(x,x) \end{aligned}$$

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI
Elegant formalism. “Datalog = Prolog without function symbols”.

Example: CSP$(\mathbb{Q}; <)$ can be solved by Datalog.

$$\begin{aligned} tc(x,y) &:- \; x < y \\ tc(x,y) &:- \; tc(x,z), z < y \\ false() &:- \; tc(x,x) \end{aligned}$$

Question (Feder+Vardi'93)

For which finite templates Γ can CSP(Γ) be solved by Datalog?

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI
Elegant formalism. “Datalog = Prolog without function symbols”.

Example: CSP$(\mathbb{Q}; <)$ can be solved by Datalog.

$$\begin{aligned} tc(x,y) &:- \quad x < y \\ tc(x,y) &:- \quad tc(x,z), z < y \\ false() &:- \quad tc(x,x) \end{aligned}$$

Question (Feder+Vardi'93)

For which finite templates Γ can CSP(Γ) be solved by Datalog?

Larose-Zadori'07:

- If Γ interprets primitively positively linear equations over a finite field, then CSP(Γ) is not in Datalog;

CSPs and Datalog

Datalog: captures many algorithmic techniques used for CSPs in AI
Elegant formalism. "Datalog = Prolog without function symbols".

Example: CSP$(\mathbb{Q}; <)$ can be solved by Datalog.

$$\begin{aligned} tc(x,y) &:- \quad x < y \\ tc(x,y) &:- \quad tc(x,z), z < y \\ false() &:- \quad tc(x,x) \end{aligned}$$

Question (Feder+Vardi'93)

For which finite templates Γ can CSP(Γ) be solved by Datalog?

Larose-Zadori'07:

- If Γ interprets primitively positively linear equations over a finite field, then CSP(Γ) is not in Datalog;
- conjecture that CSP(Γ) is in Datalog otherwise.

CSPs in Datalog

Theorem (Barto+Kozik'09,Kozik+Krokhin+Valeriote+Willard'14).

Let Γ be a finite structure. Then $\mathrm{CSP}(\Gamma)$ is in Datalog if and only if

CSPs in Datalog

Theorem (Barto+Kozik'09,Kozik+Krokhin+Valeriote+Willard'14).

Let Γ be a finite structure. Then $\text{CSP}(\Gamma)$ is in Datalog if and only if Γ has weak near-unanimity polymorphisms f, g such that for all elements x, y of Γ

$$f(x, x, y) = g(x, x, x, y) \; .$$

CSPs in Datalog

Theorem (Barto+Kozik'09,Kozik+Krokhin+Valeriote+Willard'14).

Let Γ be a finite structure. Then $\text{CSP}(\Gamma)$ is in Datalog if and only if Γ has weak near-unanimity polymorphisms f, g such that for all elements x, y of Γ

$$f(x, x, y) = g(x, x, x, y) \; .$$

Corollary: Given Γ, we can effectively decide whether $\text{CSP}(\Gamma)$ is in Datalog.

CSPs in Datalog

Theorem (Barto+Kozik'09,Kozik+Krokhin+Valeriote+Willard'14).

Let Γ be a finite structure. Then $\text{CSP}(\Gamma)$ is in Datalog if and only if Γ has weak near-unanimity polymorphisms f, g such that for all elements x, y of Γ

$$f(x,x,y) = g(x,x,x,y) \; .$$

Corollary: Given Γ, we can effectively decide whether $\text{CSP}(\Gamma)$ is in Datalog.

In fact: Given Γ, we can efficiently decide whether $\text{CSP}(\Gamma)$ is in Datalog.

CSPs over Infinite Domains

CSPs over Infinite Domains

Observation (B+Grohe'08)

For every computational problem P, there exists an infinite structure Γ such that P and CSP(Γ) are equivalent under polynomial-time Turing reductions.

CSPs over Infinite Domains

Observation (B+Grohe'08)

For every computational problem P, there exists an infinite structure Γ such that P and CSP(Γ) are equivalent under polynomial-time Turing reductions.

Two directions:

1. For which infinite structures can we use the universal-algebraic approach?

CSPs over Infinite Domains

Observation (B+Grohe'08)

For every computational problem P, there exists an infinite structure Γ such that P and CSP(Γ) are equivalent under polynomial-time Turing reductions.

Two directions:

1. For which infinite structures can we use the universal-algebraic approach?
2. Study those infinite structures that are of particular interest in computer science and mathematics.

CSPs over Infinite Domains

Observation (B+Grohe'08)

For every computational problem P, there exists an infinite structure Γ such that P and CSP(Γ) are equivalent under polynomial-time Turing reductions.

Two directions:

1. For which infinite structures can we use the universal-algebraic approach?
2. Study those infinite structures that are of particular interest in computer science and mathematics.
   E.g. systematically study CSPs over the integers, rationals, and reals.

Reducts

Let Δ be a structure, the ‘base structure’.

Reducts

Let Δ be a structure, the 'base structure'.

Definition.

A relational structure Γ is a reduct of Δ if Γ has the same domain as Δ, and all relations of Γ are first-order definable over Δ (without parameters).

Reducts

Let Δ be a structure, the 'base structure'.

Definition.

A relational structure Γ is a reduct of Δ if Γ has the same domain as Δ, and all relations of Γ are first-order definable over Δ (without parameters).

Examples:

- $(\mathbb{Q}; \text{Betw})$ is a reduct of $(\mathbb{Q}; <)$.

Reducts

Let Δ be a structure, the 'base structure'.

Definition.

A relational structure Γ is a reduct of Δ if Γ has the same domain as Δ, and all relations of Γ are first-order definable over Δ (without parameters).

Examples:

- $(\mathbb{Q}; \text{Betw})$ is a reduct of $(\mathbb{Q}; <)$.
- $(\mathbb{Q}; \{(x, y, z) \mid x < max(y, z)\})$ is a reduct of $(\mathbb{Q}; <)$.

Reducts

Let Δ be a structure, the 'base structure'.

Definition.
A relational structure Γ is a reduct of Δ if Γ has the same domain as Δ, and all relations of Γ are first-order definable over Δ (without parameters).

Examples:

- $(\mathbb{Q}; \text{Betw})$ is a reduct of $(\mathbb{Q}; <)$.
- $(\mathbb{Q}; \{(x, y, z) \mid x < max(y, z)\})$ is a reduct of $(\mathbb{Q}; <)$.

Remarks:

- Reducts studied in model theory.

Reducts

Let Δ be a structure, the 'base structure'.

Definition.
A relational structure Γ is a reduct of Δ if Γ has the same domain as Δ, and all relations of Γ are first-order definable over Δ (without parameters).

Examples:

- $\big(\mathbb{Q}; \text{Betw}\big)$ is a reduct of $(\mathbb{Q}; <)$.
- $\big(\mathbb{Q};\ \{(x, y, z) \mid x < max(y, z)\}\big)$ is a reduct of $(\mathbb{Q}; <)$.

Remarks:

- Reducts studied in model theory.
- Every automorphism of Δ is also an automorphism of all reducts of Δ.

Reducts

Let Δ be a structure, the 'base structure'.

Definition.

A relational structure Γ is a reduct of Δ if Γ has the same domain as Δ, and all relations of Γ are first-order definable over Δ (without parameters).

Examples:

- $(\mathbb{Q}; \text{Betw})$ is a reduct of $(\mathbb{Q}; <)$.
- $(\mathbb{Q}; \{(x, y, z) \mid x < max(y, z)\})$ is a reduct of $(\mathbb{Q}; <)$.

Remarks:

- Reducts studied in model theory.
- Every automorphism of Δ is also an automorphism of all reducts of Δ.

General Goal:

for interesting base structures Δ, classify $\text{CSP}(\Gamma)$ for all reducts Γ of Δ.

Generalising the Universal-Algebraic Approach

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Definition (Equivalent to standard definition by Thm. of Ryll-Nardzewski)

A countable structure Γ is ω-categorical iff for all $n \in \mathbb{N}$, the componentwise action of $\mathrm{Aut}(\Gamma)$ on n-tuples of elements from Γ has only finitely many orbits.

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Definition (Equivalent to standard definition by Thm. of Ryll-Nardzewski)

A countable structure Γ is ω-categorical iff for all $n \in \mathbb{N}$, the componentwise action of Aut(Γ) on n-tuples of elements from Γ has only finitely many orbits.

Examples:

- $(\mathbb{Q}; <)$;

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Definition (Equivalent to standard definition by Thm. of Ryll-Nardzewski)

A countable structure Γ is ω-categorical iff for all $n \in \mathbb{N}$, the componentwise action of $\mathrm{Aut}(\Gamma)$ on n-tuples of elements from Γ has only finitely many orbits.

Examples:

- $(\mathbb{Q}; <)$;
- Reducts of ω-categorical structures are ω-categorical;

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Definition (Equivalent to standard definition by Thm. of Ryll-Nardzewski)

A countable structure Γ is ω-categorical iff for all $n \in \mathbb{N}$, the componentwise action of $\text{Aut}(\Gamma)$ on *n*-tuples of elements from Γ has only finitely many orbits.

Examples:

- $(\mathbb{Q}; <)$;
- Reducts of ω-categorical structures are ω-categorical;
- Finite structures Γ;

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Definition (Equivalent to standard definition by Thm. of Ryll-Nardzewski)

A countable structure Γ is ω-categorical iff for all $n \in \mathbb{N}$, the componentwise action of $\mathrm{Aut}(\Gamma)$ on n-tuples of elements from Γ has only finitely many orbits.

Examples:

- $(\mathbb{Q}; <)$;
- Reducts of ω-categorical structures are ω-categorical;
- Finite structures Γ;
- Fraïssé-limits of amalgamation classes with finite relational signature.

Generalising the Universal-Algebraic Approach

The universal-algebraic approach can be generalised to infinite structures Γ that are ω-categorical.

Definition (Equivalent to standard definition by Thm. of Ryll-Nardzewski)

A countable structure Γ is ω-categorical iff for all $n \in \mathbb{N}$, the componentwise action of $\text{Aut}(\Gamma)$ on n-tuples of elements from Γ has only finitely many orbits.

Examples:

- $(\mathbb{Q}; <)$;
- Reducts of ω-categorical structures are ω-categorical;
- Finite structures Γ;
- Fraïssé-limits of amalgamation classes with finite relational signature.

Theorem (B+Nešetřil'03).

Let Γ be ω-categorical. Then a relation R has a primitive positive definition in Γ if and only if R is preserved by all polymorphisms of Γ.

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**,

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**, Cyclic Ordering Problem,

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**, Cyclic Ordering Problem, Point Algebra in Temporal Reasoning,

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**, Cyclic Ordering Problem, Point Algebra in Temporal Reasoning, Nebel's Ord-Horn Class, ...

Reducts of $(\mathbb{Q};<)$: Classification

Consider reducts of $(\mathbb{Q};<)$.

Examples: Betweenness problem, **And/or scheduling**, Cyclic Ordering Problem, Point Algebra in Temporal Reasoning, Nebel's Ord-Horn Class, ...

Theorem (B+Kara'08).

Let Γ be a reduct of $(\mathbb{Q};<)$. Then $\mathrm{CSP}(\Gamma)$ is in P if Γ has polymorphisms f, e_1, e_2, e_3 such that for all $x, y \in \mathbb{Q}$

$$e_1(f(x,x,y)) = e_2(f(x,y,x)) = e_3(f(y,x,x)) \, .$$

Otherwise, $\mathrm{CSP}(\Gamma)$ is NP-hard.

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**, Cyclic Ordering Problem, Point Algebra in Temporal Reasoning, Nebel's Ord-Horn Class, ...

Theorem (B+Kara'08).

Let Γ be a reduct of $(\mathbb{Q}; <)$. Then CSP(Γ) is in P if Γ has polymorphisms f, e_1, e_2, e_3 such that for all $x, y \in \mathbb{Q}$

$$e_1(f(x,x,y)) = e_2(f(x,y,x)) = e_3(f(y,x,x)) \, .$$

Otherwise, CSP(Γ) is NP-hard.

Remarks.

- Interesting class from descriptive complexity point of view: e.g., $\text{CSP}\big(\mathbb{Q}; \{(x,y,z) \mid x < max(y,z)\}\big)$ is in LFP, but not in Datalog.

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**, Cyclic Ordering Problem, Point Algebra in Temporal Reasoning, Nebel's Ord-Horn Class, ...

Theorem (B+Kara'08).

Let Γ be a reduct of $(\mathbb{Q}; <)$. Then CSP(Γ) is in P if Γ has polymorphisms f, e_1, e_2, e_3 such that for all $x, y \in \mathbb{Q}$

$$e_1(f(x,x,y)) = e_2(f(x,y,x)) = e_3(f(y,x,x)) .$$

Otherwise, CSP(Γ) is NP-hard.

Remarks.

- Interesting class from descriptive complexity point of view: e.g., CSP$\big(\mathbb{Q}; \{(x,y,z) \mid x < max(y,z)\}\big)$ is in LFP, but not in Datalog.
- Proof makes essential use of Ramsey theory.

Reducts of $(\mathbb{Q}; <)$: Classification

Consider reducts of $(\mathbb{Q}; <)$.

Examples: **Betweenness problem**, **And/or scheduling**, Cyclic Ordering Problem, Point Algebra in Temporal Reasoning, Nebel's Ord-Horn Class, ...

Theorem (B+Kara'08).

Let Γ be a reduct of $(\mathbb{Q}; <)$. Then CSP(Γ) is in P if Γ has polymorphisms f, e_1, e_2, e_3 such that for all $x, y \in \mathbb{Q}$

$$e_1(f(x,x,y)) = e_2(f(x,y,x)) = e_3(f(y,x,x)) \; .$$

Otherwise, CSP(Γ) is NP-hard.

Remarks.

- Interesting class from descriptive complexity point of view: e.g., CSP$\big(\mathbb{Q}; \{(x,y,z) \mid x < max(y,z)\}\big)$ is in LFP, but not in Datalog.
- Proof makes essential use of Ramsey theory.
- STACS Proceedings: tractability conjecture for a large class of ω-categorical structures Γ.

Reducts of Linear Program Feasibility

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.
- Classification of CSP(Γ) for reducts Γ of Δ: open!

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.
- Classification of CSP(Γ) for reducts Γ of Δ: open!
- Δ is not ω-categorical.

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.
- Classification of CSP(Γ) for reducts Γ of Δ: open!
- Δ is not ω-categorical.
- A reduct Γ of Δ contains only convex relations if and only if Γ has the polymorphism $(x, y) \mapsto (x + y)/2$.

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.
- Classification of CSP(Γ) for reducts Γ of Δ: open!
- Δ is not ω-categorical.
- A reduct Γ of Δ contains only convex relations if and only if Γ has the polymorphism $(x, y) \mapsto (x + y)/2$.

Which relations can be added to Δ such that CSP(Δ) remains in P?

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \{(x, y, z) \mid x = y + z\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.
- Classification of CSP(Γ) for reducts Γ of Δ: open!
- Δ is not ω-categorical.
- A reduct Γ of Δ contains only convex relations if and only if Γ has the polymorphism $(x, y) \mapsto (x + y)/2$.

Which relations can be added to Δ such that CSP(Δ) remains in P?

Examples.

- CSP$(\Delta, \neq)$ is in P (but $\neq$ is not convex).

Reducts of Linear Program Feasibility

Let $\Delta := (\mathbb{Q}; <, R_+, R_{=1})$ where $R_+ := \big\{(x,y,z) \mid x = y + z\big\}$ and $R_{=1} := \{1\}$.

Remarks.

- CSP(Δ) is equivalent to the feasibility problem for linear programming.
- Classification of CSP(Γ) for reducts Γ of Δ: open!
- Δ is not ω-categorical.
- A reduct Γ of Δ contains only convex relations if and only if Γ has the polymorphism $(x,y) \mapsto (x+y)/2$.

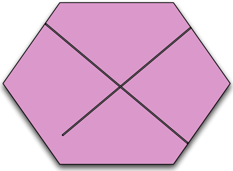
Which relations can be added to Δ such that CSP(Δ) remains in P?

Examples.

- CSP$(\Delta, \neq)$ is in P (but $\neq$ is not convex).
- CSP$(\Delta, \{(u,v,x,y) \mid u = v \Rightarrow x = y\})$ is in P (Bäckström,Jonsson'98).

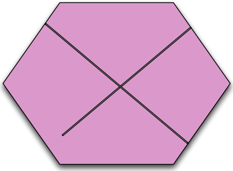
Tractable Expansions of Linear Programming

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



Tractable Expansions of Linear Programming

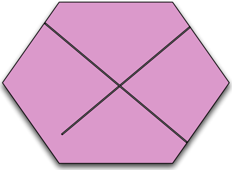
Definition:
$R \subseteq \mathbb{Q}^k$ is called essentially convex
if for all $a, b \in R$ there are only finitely many points on the line segment between a and b that are not in R.



Theorem (B+Jonsson+vonOertzen'09).
Let Γ be a reduct of Δ of the form $(\mathbb{Q}; <, R_+, R_{=1}, S_1, \ldots, S_n)$.

Tractable Expansions of Linear Programming

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

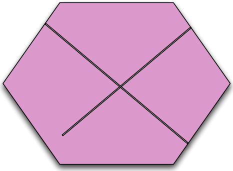


Theorem (B+Jonsson+vonOertzen'09).

Let Γ be a reduct of Δ of the form $(\mathbb{Q}; <, R_+, R_{=1}, S_1, \dots, S_n)$.
Then: all relations in Γ are essentially convex, and CSP(Γ) is in P,

Tractable Expansions of Linear Programming

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

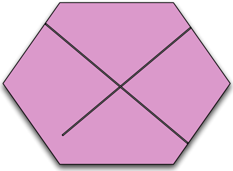


Theorem (B+Jonsson+vonOertzen'09).

Let Γ be a reduct of Δ of the form $(\mathbb{Q}; <, R_+, R_{=1}, S_1, \ldots, S_n)$.
Then: all relations in Γ are essentially convex, and CSP(Γ) is in P,
or CSP(Γ) is NP-hard.

Tractable Expansions of Linear Programming

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



Theorem (B+Jonsson+vonOertzen'09).

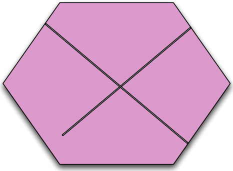
Let Γ be a reduct of Δ of the form $(\mathbb{Q}; <, R_+, R_{=1}, S_1, \ldots, S_n)$.
Then: all relations in Γ are essentially convex, and CSP(Γ) is in P,
or CSP(Γ) is NP-hard.

Remarks.

- Essential convexity is not a polymorphism condition.

Tractable Expansions of Linear Programming

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



Theorem (B+Jonsson+vonOertzen'09).
Let Γ be a reduct of Δ of the form $(\mathbb{Q}; <, R_+, R_{=1}, S_1, \dots, S_n)$.
Then: all relations in Γ are essentially convex, and CSP(Γ) is in P,
or CSP(Γ) is NP-hard.

Remarks.

- Essential convexity is not a polymorphism condition.
- But: essential convexity is a polymorphism condition in a saturated elementary extension of Γ (B.,Mamino'14).

Max-Closed Constraints

Max-Closed Constraints

Revisit

$$\mathrm{CSP}(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$$

where

- $S_{+1} := \{(x, y) \mid y = x + 1\}$,
- $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
- $S_{max} := \{(x, y, z) \mid x \leq y \vee x \leq z)\}$.

Max-Closed Constraints

Revisit

$$\text{CSP}(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$$

where

- $S_{+1} := \{(x, y) \mid y = x + 1\}$,
- $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
- $S_{max} := \{(x, y, z) \mid x \leq y \vee x \leq z)\}$.

Remarks:

- *max* is a polymorphism.

Max-Closed Constraints

Revisit

$$\text{CSP}(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$$

where

- $S_{+1} := \{(x, y) \mid y = x + 1\}$,
- $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
- $S_{max} := \{(x, y, z) \mid x \leq y \vee x \leq z)\}$.

Remarks:

- *max* is a polymorphism.
- At least as hard as determining the winner in mean payoff games,

Max-Closed Constraints

Revisit

$$\text{CSP}(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$$

where

- $S_{+1} := \{(x, y) \mid y = x + 1\}$,
- $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
- $S_{max} := \{(x, y, z) \mid x \leq y \vee x \leq z)\}$.

Remarks:

- *max* is a polymorphism.
- At least as hard as determining the winner in mean payoff games, which is in NP $\cap$ coNP, but not known to be in P.

Max-Closed Constraints

Revisit

$$\mathrm{CSP}(\mathbb{Q}; S_{+1}, S_{\cdot 2}, S_{max})$$

where

- $S_{+1} := \{(x, y) \mid y = x + 1\}$,
- $S_{\cdot 2} := \{(x, y) \mid y = 2x\}$, and
- $S_{max} := \{(x, y, z) \mid x \leq y \vee x \leq z)\}$.

Remarks:

- *max* is a polymorphism.
- At least as hard as determining the winner in mean payoff games, which is in NP $\cap$ coNP, but not known to be in P.

Theorem (Möhring, Skutella, Stork'04).

Mean payoff games are polynomial-time equivalent to deciding satisfiability of constraints of the form $x \leq max(y, z) + c$ where $c \in \mathbb{Z}$ is represented in binary.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where $\text{succ} := \{(x, y) \mid y = x + 1\}$.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where $\text{succ} := \{(x, y) \mid y = x + 1\}$.

- Simplest finite-signature structure which is not ω-categorical.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where succ $:= \{(x, y) \mid y = x + 1\}$.

- Simplest finite-signature structure which is not ω-categorical.
- $(\mathbb{Z}; \text{succ})$ satisfies the same first-order sentences as $\big(\mathbb{Q}; \{(x, y) \mid y = x + 1\}\big)$ and $\big(\mathbb{R}; \{(x, y) \mid y = x + 1\}\big)$.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where $\text{succ} := \{(x, y) \mid y = x + 1\}$.

- Simplest finite-signature structure which is not ω-categorical.
- $(\mathbb{Z}; \text{succ})$ satisfies the same first-order sentences as $\big(\mathbb{Q}; \{(x, y) \mid y = x + 1\}\big)$ and $\big(\mathbb{R}; \{(x, y) \mid y = x + 1\}\big)$.
- CSPs for reducts of $(\mathbb{Z}; \text{succ})$ appear as CSPs for reducts of $(\mathbb{Q}; R_+, R_{=1})$ and $(\mathbb{R}; R_+, R_{=1})$.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where $\text{succ} := \{(x, y) \mid y = x + 1\}$.

- Simplest finite-signature structure which is not ω-categorical.
- $(\mathbb{Z}; \text{succ})$ satisfies the same first-order sentences as $\big(\mathbb{Q}; \{(x, y) \mid y = x + 1\}\big)$ and $\big(\mathbb{R}; \{(x, y) \mid y = x + 1\}\big)$.
- CSPs for reducts of $(\mathbb{Z}; \text{succ})$ appear as CSPs for reducts of $(\mathbb{Q}; R_+, R_{=1})$ and $(\mathbb{R}; R_+, R_{=1})$.

Examples:

- $(\mathbb{Z}; \neq, \{(x, y) : |x - y| = 2\})$.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where $\text{succ} := \{(x, y) \mid y = x + 1\}$.

- Simplest finite-signature structure which is not ω-categorical.
- $(\mathbb{Z}; \text{succ})$ satisfies the same first-order sentences as $\big(\mathbb{Q}; \{(x, y) \mid y = x + 1\}\big)$ and $\big(\mathbb{R}; \{(x, y) \mid y = x + 1\}\big)$.
- CSPs for reducts of $(\mathbb{Z}; \text{succ})$ appear as CSPs for reducts of $(\mathbb{Q}; R_+, R_{=1})$ and $(\mathbb{R}; R_+, R_{=1})$.

Examples:

- $(\mathbb{Z}; \neq, \{(x, y) : |x - y| = 2\})$.
- $(\mathbb{Z}; F)$ where $F := \big\{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\big\}$.

The Integers with Successor

Consider CSPs for reducts Γ of $(\mathbb{Z}; \text{succ})$ where $\text{succ} := \{(x,y) \mid y = x+1\}$.

- Simplest finite-signature structure which is not ω-categorical.
- $(\mathbb{Z}; \text{succ})$ satisfies the same first-order sentences as $\big(\mathbb{Q}; \{(x,y) \mid y = x+1\}\big)$ and $\big(\mathbb{R}; \{(x,y) \mid y = x+1\}\big)$.
- CSPs for reducts of $(\mathbb{Z}; \text{succ})$ appear as CSPs for reducts of $(\mathbb{Q}; R_+, R_{=1})$ and $(\mathbb{R}; R_+, R_{=1})$.

Examples:

- $(\mathbb{Z}; \neq, \{(x,y) : |x-y| = 2\})$.
- $(\mathbb{Z}; F)$ where $F := \big\{(x,y,u,v) \mid y = x+1 \Leftrightarrow v = u+1\big\}$.

Theorem (B,Martin,Mottet'15).

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$. Then $\text{CSP}(\Gamma)$ is in P, or NP-complete, or equals a finite-domain CSP.

Ideas from the proof

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then CSP(Γ) equals CSP(Δ) where Δ is one of the following:

1. a finite structure;

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then CSP(Γ) equals CSP(Δ) where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then CSP(Γ) equals CSP(Δ) where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x,y,u,v) \mid y = x+1 \Leftrightarrow v = u+1\}\big)$ where $|x-y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then CSP(Γ) equals CSP(Δ) where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$ where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then $\text{CSP}(\Gamma)$ equals $\text{CSP}(\Delta)$ where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$
   where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Examples.

- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : |x - y| \in \{1, 2\}\}\big)$

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then CSP(Γ) equals CSP(Δ) where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$ where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Examples.

- CSP$\big(\mathbb{Z}; \{(x, y) : |x - y| \in \{1, 2\}\}\big)$ equals CSP(K_3).

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then $\text{CSP}(\Gamma)$ equals $\text{CSP}(\Delta)$ where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$ where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Examples.

- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : |x - y| \in \{1, 2\}\}\big)$ equals $\text{CSP}(K_3)$.
- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : x - y = 2\}\big)$

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then $\text{CSP}(\Gamma)$ equals $\text{CSP}(\Delta)$ where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$ where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Examples.

- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : |x - y| \in \{1, 2\}\}\big)$ equals $\text{CSP}(K_3)$.
- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : x - y = 2\}\big)$ equals $\text{CSP}(\mathbb{Z}; \text{succ})$.

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then $\text{CSP}(\Gamma)$ equals $\text{CSP}(\Delta)$ where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$ where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Examples.

- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : |x - y| \in \{1, 2\}\}\big)$ equals $\text{CSP}(K_3)$.
- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : x - y = 2\}\big)$ equals $\text{CSP}(\mathbb{Z}; \text{succ})$.
- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : x - y \neq 1\}\big)$

Ideas from the proof

Half-way:

Lemma.

Let Γ be a reduct of $(\mathbb{Z}; \text{succ})$.
Then $\text{CSP}(\Gamma)$ equals $\text{CSP}(\Delta)$ where Δ is one of the following:

1. a finite structure;
2. a reduct of $(\mathbb{Z}; =)$;
3. a reduct of $\big(\mathbb{Z}; \{(x, y, u, v) \mid y = x + 1 \Leftrightarrow v = u + 1\}\big)$ where $|x - y| = d$ is primitive positive definable for all $d \in \mathbb{Z}$;
4. a reduct of $(\mathbb{Z}; \text{succ})$ where succ is primitive positive definable.

Examples.

- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : |x - y| \in \{1, 2\}\}\big)$ equals $\text{CSP}(K_3)$.
- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : x - y = 2\}\big)$ equals $\text{CSP}(\mathbb{Z}; \text{succ})$.
- $\text{CSP}\big(\mathbb{Z}; \{(x, y) : x - y \neq 1\}\big)$ equals $\text{CSP}(\mathbb{Z}; \neq)$.

A Non-Dichotomy

A Non-Dichotomy

Proposition (B'12).

Every recursively enumerable language is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *)$.

A Non-Dichotomy

Proposition (B'12).

Every recursively enumerable language is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *)$.

- Use a theorem of Davis, Matiyasevich, Putnam, Robinson:

A Non-Dichotomy

Proposition (B'12).

Every recursively enumerable language is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *)$.

- Use a theorem of Davis, Matiyasevich, Putnam, Robinson:
  $U \subseteq \mathbb{Z}$ is recursively enumerable if and only if
  it has a primitive positive definition in $(\mathbb{Z}; *, +, 1)$.

A Non-Dichotomy

Proposition (B'12).

Every recursively enumerable language is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *)$.

- Use a theorem of Davis, Matiyasevich, Putnam, Robinson: $U \subseteq \mathbb{Z}$ is recursively enumerable if and only if it has a primitive positive definition in $(\mathbb{Z}; *, +, 1)$.
- Consider $\Gamma = (\mathbb{Z}; U, S_{+1}, S_{\cdot 2})$. There is polynomial-time reduction from "$n \in U$?" to CSP(Γ).

A Non-Dichotomy

Proposition (B'12).

Every recursively enumerable language is polynomial-time equivalent to the CSP of a reduct of $(\mathbb{Z}; +, *)$.

- Use a theorem of Davis, Matiyasevich, Putnam, Robinson:
  $U \subseteq \mathbb{Z}$ is recursively enumerable if and only if
  it has a primitive positive definition in $(\mathbb{Z}; *, +, 1)$.
- Consider $\Gamma = (\mathbb{Z}; U, S_{+1}, S_{\cdot 2})$.
  There is polynomial-time reduction from "$n \in U$?" to CSP(Γ).
- Need to modify Γ and use more coding tricks so that CSP(Γ) is polynomial-time equivalent to "$n \in U$?" ...

Open Problems

Open Problems

- Classify CSP(Γ) for all reducts of $(\mathbb{Z}; <)$.

Open Problems

- Classify CSP(Γ) for all reducts of $(\mathbb{Z}; <)$.
  Generalizes classification for $(\mathbb{Q}; <)$ and for $(\mathbb{Z}; \text{succ})$.

Open Problems

- Classify CSP(Γ) for all reducts of $(\mathbb{Z};<)$.
  Generalizes classification for $(\mathbb{Q};<)$ and for $(\mathbb{Z};\text{succ})$.

- Non-dichotomy for CSPs of reducts of $\big(\mathbb{Q}; S_{+1}, S_{\cdot 2}\big)$?

Open Problems

- Classify CSP(Γ) for all reducts of $(\mathbb{Z}; <)$.
  Generalizes classification for $(\mathbb{Q}; <)$ and for $(\mathbb{Z}; \mathrm{succ})$.

- Non-dichotomy for CSPs of reducts of $\big(\mathbb{Q}; S_{+1}, S_{\cdot 2}\big)$?
  These problems are all in NP.