Many independent equality constraints

Michael Pinsker

jointly with Manuel Bodirsky (Paris) and Hubie Chen (Barcelona)

LMNO
Université de Caen
Caen, France

SSAOS 2008 / Třešť

Outline

1. Formulas and primitive positive definitions
2. PP-closed relational structures
3. Local clones


STAR
THE
CLONE
WARS
WARS
Executive Producer GEORGE LUCAS Director DAVE FILONI Producer CATHERINE WINDER
Writers HENRY GILROY STEVE MELCHING SCOTT MURPHY
Editor JASON TUCKER Score by KEVIN KINER Original Star Wars Themes and Score by JOHN WILLIAMS
IN THEATERS AUGUST 15
www.starwars.com

Equality constraint languages

Let ϕ be a first-order formula over the empty language,
i.e., ϕ contains no function or relation symbols except $=$.

Equality constraint languages

Let ϕ be a first-order formula over the empty language,
i.e., ϕ contains no function or relation symbols except $=$.

E.g. $\phi(x, y, z) \leftrightarrow x = y \vee (y \neq z \wedge z \neq x)$.

Equality constraint languages

Let ϕ be a first-order formula over the empty language,
i.e., ϕ contains no function or relation symbols except $=$.

E.g. $\phi(x, y, z) \leftrightarrow x = y \vee (y \neq z \wedge z \neq x)$.

Definition

ϕ is called an *equality constraint*.

Equality constraint languages

Let ϕ be a first-order formula over the empty language,
i.e., ϕ contains no function or relation symbols except $=$.

E.g. $\phi(x, y, z) \leftrightarrow x = y \vee (y \neq z \wedge z \neq x)$.

Definition

ϕ is called an *equality constraint*.

Definition

Let Σ be a set of equality constraints.
Σ is called an *equality constraint language*.

Primitive positive definitions and the problem

Definition

ϕ is *pp-definable* (*primitively positively definable*) from $\Sigma \leftrightarrow$ ϕ is logically equivalent to a formula of the form $\exists x_{j_1} \ldots \exists x_{j_n}\, \phi_1 \wedge \ldots \wedge \phi_m$, with $\phi_i \in \Sigma \cup \{x = y\}$.

Primitive positive definitions and the problem

Definition

ϕ is *pp-definable* (*primitively positively definable*) from $\Sigma \leftrightarrow$
ϕ is logically equivalent to a formula of the form
$\exists x_{j_1} \ldots \exists x_{j_n} \, \phi_1 \wedge \ldots \wedge \phi_m$, with $\phi_i \in \Sigma \cup \{x = y\}$.

Definition

Σ_0, Σ_1 are *pp-equivalent* $\leftrightarrow$
all formulas of Σ_0 are pp-definable from Σ_1, and vice-versa.

Primitive positive definitions and the problem

Definition

ϕ is *pp-definable* (*primitively positively definable*) from $\Sigma \leftrightarrow$
ϕ is logically equivalent to a formula of the form
$\exists x_{j_1} \ldots \exists x_{j_n} \phi_1 \wedge \ldots \wedge \phi_m$, with $\phi_i \in \Sigma \cup \{x = y\}$.

Definition

Σ_0, Σ_1 are *pp-equivalent* $\leftrightarrow$
all formulas of Σ_0 are pp-definable from Σ_1, and vice-versa.

Problem (Formulas)

Are there uncountably many pp-inequivalent equality constraint languages?

The problem, strong version

The problem, strong version

Definition

Σ *independent* $\leftrightarrow$
for all $\phi \in \Sigma$, ϕ is not pp-definable from $\Sigma \setminus \{\phi\}$.

The problem, strong version

Definition

Σ *independent* $\leftrightarrow$
for all $\phi \in \Sigma$, ϕ is not pp-definable from $\Sigma \setminus \{\phi\}$.

Problem, strong version

Is there an infinite independent equality constraint language?

Reducts of relational structures

$\Gamma = (X, \mathcal{R}) \ldots$ relational structure (X infinite).

Reducts of relational structures

$\Gamma = (X, \mathcal{R}) \ldots$ relational structure (X infinite).

Problem

Determine the reducts of Γ, i.e.,
all relational structures which are first-order definable from Γ.

Reducts of relational structures

$\Gamma = (X, \mathcal{R}) \ldots$ relational structure (X infinite).

Problem

Determine the reducts of Γ, i.e.,
all relational structures which are first-order definable from Γ.

Usually done up to first-order interdefinability, i.e.,
structures Γ_0, Γ_1 equivalent $\leftrightarrow$
Γ_i has a first-order definition in Γ_{1-i}.

Reducts of relational structures

$\Gamma = (X, \mathcal{R}) \ldots$ relational structure (X infinite).

Problem

Determine the reducts of Γ, i.e.,
all relational structures which are first-order definable from Γ.

Usually done up to first-order interdefinability, i.e.,
structures Γ_0, Γ_1 equivalent $\leftrightarrow$
Γ_i has a first-order definition in Γ_{1-i}.

Examples

- P. J. Cameron: There are 5 reducts of $(\mathbb{Q}, <)$ up to f.o.-interdefinability.

Reducts of relational structures

$\Gamma = (X, \mathcal{R}) \ldots$ relational structure (X infinite).

Problem

Determine the reducts of Γ, i.e.,
all relational structures which are first-order definable from Γ.

Usually done up to first-order interdefinability, i.e.,
structures Γ_0, Γ_1 equivalent $\leftrightarrow$
Γ_i has a first-order definition in Γ_{1-i}.

Examples

- P. J. Cameron: There are 5 reducts of $(\mathbb{Q}, <)$ up to f.o.-interdefinability.
- M. Junker and M. Ziegler: There are 116 reducts of $(\mathbb{Q}, <, 0)$ up to f.o.-interdefinability.

Reducts up to pp-interdefinability

Problem

Given a structure Γ, determine its reducts *up to primitive positive interdefinability*.

Reducts up to pp-interdefinability

Problem

Given a structure Γ, determine its reducts *up to primitive positive interdefinability*.

First step

Try with the simplest structure, $\Gamma := (X, =)$.

Reducts up to pp-interdefinability

Problem

Given a structure Γ, determine its reducts *up to primitive positive interdefinability*.

First step

Try with the simplest structure, $\Gamma := (X, =)$.

Observation

Equality constraint languages correspond to reducts of Γ.

Reducts up to pp-interdefinability

Problem

Given a structure Γ, determine its reducts *up to primitive positive interdefinability*.

First step

Try with the simplest structure, $\Gamma := (X, =)$.

Observation

Equality constraint languages correspond to reducts of Γ.

Problem (Reducts)

Is the number of reducts of Γ uncountable (up to pp-interdefinability)?

The **C**onstraint **S**atisfaction **P**roblem

The **C**onstraint **S**atisfaction **P**roblem

Reducts of $(X, =)$ as templates of CSPs.

The Constraint Satisfaction Problem

Reducts of $(X, =)$ as templates of CSPs.

Fixed structure $\Gamma = (X, \mathcal{R})$.

The Constraint Satisfaction Problem

Reducts of $(X, =)$ as templates of CSPs.

Fixed structure $\Gamma = (X, \mathcal{R})$.

Input: A finite structure Δ.

The Constraint Satisfaction Problem

Reducts of $(X, =)$ as templates of CSPs.

Fixed structure $\Gamma = (X, \mathcal{R})$.

Input: A finite structure Δ.

Question: Is there a homomorphism from Δ to Γ?

The Constraint Satisfaction Problem

Reducts of $(X, =)$ as templates of CSPs.

Fixed structure $\Gamma = (X, \mathcal{R})$.

Input: A finite structure Δ.

Question: Is there a homomorphism from Δ to Γ?

Complexity unchanged if pp-definable relations are added.

The Constraint Satisfaction Problem

Reducts of $(X, =)$ as templates of CSPs.

Fixed structure $\Gamma = (X, \mathcal{R})$.

Input: A finite structure Δ.

Question: Is there a homomorphism from Δ to Γ?

Complexity unchanged if pp-definable relations are added.

Hence the name *equality constraint languages*.

The Constraint Satisfaction Problem

Reducts of $(X, =)$ as templates of CSPs.

Fixed structure $\Gamma = (X, \mathcal{R})$.

Input: A finite structure Δ.

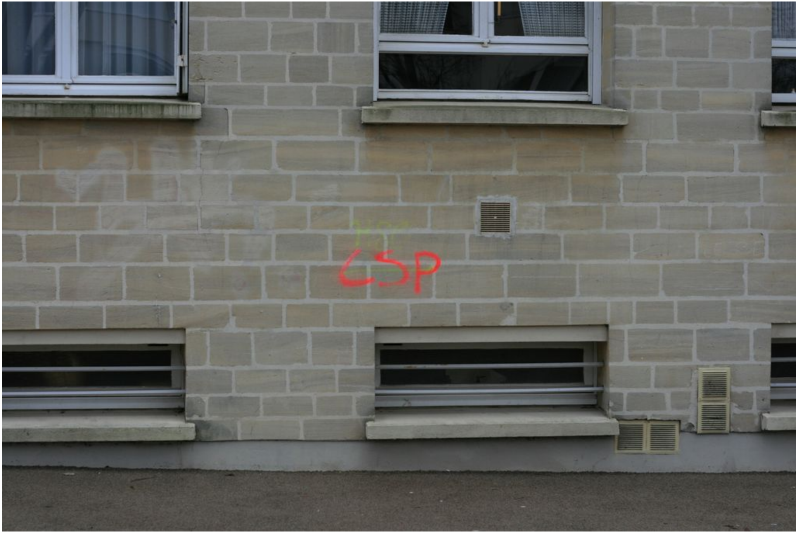
Question: Is there a homomorphism from Δ to Γ?

Complexity unchanged if pp-definable relations are added.

Hence the name *equality constraint languages*.

(Bodirsky – Chen – Kara)

CSP fanatics in Caen



pp-definability and operations

pp-definability and operations

Definitions

$\mathcal{O}$ . . . set of all finitary operations on X.

pp-definability and operations

Definitions

$\mathcal{O}$ ... set of all finitary operations on X.

Let $f \in \mathcal{O}$ be n-ary operation and $R \subseteq X^m$ relation.
f preserves R $\leftrightarrow f(r_1, \ldots, r_n) \in R$ for all $r_1, \ldots, r_n \in R$.

pp-definability and operations

Definitions

$\mathcal{O} \ldots$ set of all finitary operations on X.

Let $f \in \mathcal{O}$ be n-ary operation and $R \subseteq X^m$ relation.
f preserves R $\leftrightarrow f(r_1, \ldots, r_n) \in R$ for all $r_1, \ldots, r_n \in R$.

$\mathrm{Pol}(\Gamma) := \{f \in \mathcal{O} : f$ preserves all relations of $\Gamma\}$.
$\mathrm{Inv}(\mathcal{F}) := \{R : R$ is preserved by all $f \in \mathcal{F}\}$ (for $\mathcal{F} \subseteq \mathcal{O}$).

pp-definability and operations

Definitions

$\mathcal{O} \ldots$ set of all finitary operations on X.

Let $f \in \mathcal{O}$ be n-ary operation and $R \subseteq X^m$ relation.
f preserves R $\leftrightarrow f(r_1, \ldots, r_n) \in R$ for all $r_1, \ldots, r_n \in R$.

$\mathrm{Pol}(\Gamma) := \{f \in \mathcal{O} : f$ preserves all relations of $\Gamma\}$.
$\mathrm{Inv}(\mathcal{F}) := \{R : R$ is preserved by all $f \in \mathcal{F}\}$ (for $\mathcal{F} \subseteq \mathcal{O}$).

Fact

- Inv Pol= closure operator on the relational structures
- Pol Inv= closure operator on the sets of operations

pp-definability and operations

Definitions

$\mathcal{O}\ldots$ set of all finitary operations on X.

Let $f \in \mathcal{O}$ be n-ary operation and $R \subseteq X^m$ relation.
f preserves R $\leftrightarrow f(r_1, \ldots, r_n) \in R$ for all $r_1, \ldots, r_n \in R$.

$\mathrm{Pol}(\Gamma) := \{f \in \mathcal{O} : f$ preserves all relations of $\Gamma\}$.
$\mathrm{Inv}(\mathcal{F}) := \{R : R$ is preserved by all $f \in \mathcal{F}\}$ (for $\mathcal{F} \subseteq \mathcal{O}$).

Fact

- Inv Pol= closure operator on the relational structures
- Pol Inv= closure operator on the sets of operations

Theorem (Bodirsky-Nešetřil)

Let Γ *be* ω*-categorical. Then* $\mathrm{Inv}\,\mathrm{Pol}(\Gamma) = pp(\Gamma)$.

Local clones

Definition

$\mathcal{C} \subseteq \mathcal{O}$ is a *clone* $\leftrightarrow$

- $\mathcal{C}$ is closed under composition
  I.e. $f(g_1, \ldots, g_n) \in \mathcal{C}$ for all $f, g_1, \ldots, g_n \in \mathcal{C}$, and
- $\mathcal{C}$ contains the projections
  I.e. for all $1 \leq k \leq n$ the operation $\pi^n_k(x_1, \ldots, x_n) = x_k$.

Local clones

Definition

$\mathcal{C} \subseteq \mathcal{O}$ is a *clone* $\leftrightarrow$

- $\mathcal{C}$ is closed under composition
  I.e. $f(g_1, \ldots, g_n) \in \mathcal{C}$ for all $f, g_1, \ldots, g_n \in \mathcal{C}$, and
- $\mathcal{C}$ contains the projections
  I.e. for all $1 \leq k \leq n$ the operation $\pi^n_k(x_1, \ldots, x_n) = x_k$.

Definition

A clone $\mathcal{C}$ is *locally closed* or *local* $\leftrightarrow$
$\mathcal{C}$ is closed in the product topology on X^X (where X is discrete)

Local clones

Definition

$\mathcal{C} \subseteq \mathcal{O}$ is a *clone* $\leftrightarrow$

- $\mathcal{C}$ is closed under composition
  I.e. $f(g_1, \ldots, g_n) \in \mathcal{C}$ for all $f, g_1, \ldots, g_n \in \mathcal{C}$, and
- $\mathcal{C}$ contains the projections
  I.e. for all $1 \leq k \leq n$ the operation $\pi_k^n(x_1, \ldots, x_n) = x_k$.

Definition

A clone $\mathcal{C}$ is *locally closed* or *local* $\leftrightarrow$
$\mathcal{C}$ is closed in the product topology on X^X (where X is discrete)

Fact

The local clones are exactly the Pol Inv-closed subsets of $\mathcal{O}$.

Local clones and pp-closed reducts

Observations

Via $\mathrm{Pol} - \mathrm{Inv}$, the reducts of $(X, =)$ correspond to local clones.

Local clones and pp-closed reducts

Observations

Via Pol $-$ Inv, the reducts of $(X,=)$ correspond to local clones.
Those clones contain the group S_X of all permutations of X.

Local clones and pp-closed reducts

Observations

Via Pol – Inv, the reducts of $(X, =)$ correspond to local clones.
Those clones contain the group S_X of all permutations of X.
Conversely, if a clone contains S_X, then it induces a reduct of $(X, =)$.

Local clones and pp-closed reducts

Observations

Via Pol – Inv, the reducts of $(X, =)$ correspond to local clones.
Those clones contain the group S_X of all permutations of X.
Conversely, if a clone contains S_X, then it induces a reduct of $(X, =)$.

More observations

Inv (or Pol) is an antiisomorphism between

- the lattice of local clones above S_X
- and the pp-closed reducts of $(X, =)$

Local clones and pp-closed reducts

Observations

Via Pol – Inv, the reducts of $(X, =)$ correspond to local clones.
Those clones contain the group S_X of all permutations of X.
Conversely, if a clone contains S_X, then it induces a reduct of $(X, =)$.

More observations

Inv (or Pol) is an antiisomorphism between

- the lattice of local clones above S_X
- and the pp-closed reducts of $(X, =)$

Problem (Clones)

Is the number of local clones containing S_X uncountable?

All our problems

All our problems

Problem (Formulas)

Is the number of pp-inequivalent equality constraint languages uncountable?

All our problems

Problem (Formulas)

Is the number of pp-inequivalent equality constraint languages uncountable?

Problem (Structures)

Is the number of pp-closed reducts of $(X, =)$ uncountable?

All our problems

Problem (Formulas)

Is the number of pp-inequivalent equality constraint languages uncountable?

Problem (Structures)

Is the number of pp-closed reducts of $(X, =)$ uncountable?

Problem (Clones)

Is the number of local clones containing S_X uncountable?

All our problems

Problem (Formulas)

Is the number of pp-inequivalent equality constraint languages uncountable?

Problem (Structures)

Is the number of pp-closed reducts of $(X, =)$ uncountable?

Problem (Clones)

Is the number of local clones containing S_X uncountable?

Problem (Formulas strong)

Is there an infinite independent equality constraint language?

Answer to: Formulas (strong)

Problem (Formulas strong)

Is there an infinite independent equality constraint language?

Answer to: Formulas (strong)

Problem (Formulas strong)

Is there an infinite independent equality constraint language?

Answer (Formulas strong)

Yes.

The formulas

For all $n \geq 3$, write

$$\delta_n := x_1 \neq y_1 \vee \ldots \vee x_n \neq y_n.$$

For all $A \subseteq \{1, \ldots, n\}$ with $1 < |A| < n$, writing $A = \{j_1, \ldots, j_r\}$ with $j_1 < j_2 < \ldots < j_r$, we set

$$\kappa_A := y_{j_1} \neq x_{j_2} \vee y_{j_2} \neq x_{j_3} \vee \ldots \vee y_{j_r} \neq x_{j_1}.$$

Set

$$\phi_n := \delta_n \wedge \bigwedge_{A \subseteq \{1,\ldots,n\}, 1 < |A| < n} \kappa_A.$$

The formulas

For all $n \geq 3$, write

$$\delta_n := x_1 \neq y_1 \vee \ldots \vee x_n \neq y_n.$$

For all $A \subseteq \{1, \ldots, n\}$ with $1 < |A| < n$, writing $A = \{j_1, \ldots, j_r\}$ with $j_1 < j_2 < \ldots < j_r$, we set

$$\kappa_A := y_{j_1} \neq x_{j_2} \vee y_{j_2} \neq x_{j_3} \vee \ldots \vee y_{j_r} \neq x_{j_1}.$$

Set

$$\phi_n := \delta_n \wedge \bigwedge_{A \subseteq \{1,\ldots,n\}, 1<|A|<n} \kappa_A.$$

Theorem (M. Bodirsky, H. Chen, MP 2008)

$\{\phi_n : n \geq 3\}$ *is independent.*

Reference

M. Bodirsky, H. Chen, M. Pinsker,
The reducts of equality up to primitive positive interdefinability,
Preprint September 15, 2008