CSPs with finite duality closed under primitive positive constructions

Florian Starke

joint work with Manuel Bodirsky

CSPs

Definition

$\mathbb{T}$ a relational structure

$$\mathrm{CSP}(\mathbb{T}) := \{\mathbb{I} \mid \mathbb{I} \text{ finite structure such that } \mathbb{I} \to \mathbb{T}\}$$

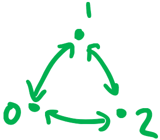
CSPs

Definition

$\mathbb{T}$ a relational structure

$$\mathrm{CSP}(\mathbb{T}) := \{\mathbb{I} \mid \mathbb{I} \text{ finite structure such that } \mathbb{I} \to \mathbb{T}\}$$

$$\mathrm{CSP}(\mathbb{K}_3) = \{\mathbb{G} \mid \mathbb{G} \text{ is 3-colourable}\}$$



CSPs

Definition

$\mathbb{T}$ a relational structure

$$\mathrm{CSP}(\mathbb{T}) := \{\mathbb{I} \mid \mathbb{I} \text{ finite structure such that } \mathbb{I} \to \mathbb{T}\}$$

$$\mathrm{CSP}(\mathbb{K}_3) = \{\mathbb{G} \mid \mathbb{G} \text{ is 3-colourable}\}$$

$$\mathrm{CSP}(\mathbb{P}_2) = \{\mathbb{G} \mid \text{no directed path of length 2 in } \mathbb{G}\}$$

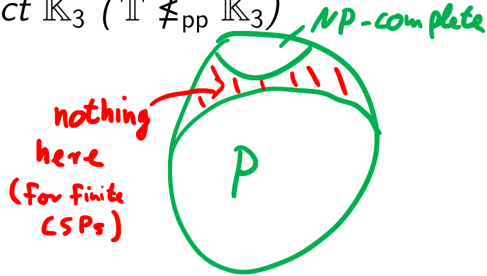
Dichotomy

$\mathbb{T}$ a finite relational structure

Theorem (Bulatov17,Zhuk17)

The following are equivalent

1. $\mathbb{T}$ *has a Siggers polymorphism and* CSP($\mathbb{T}$) *is in P*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{K}_3$ *(*$\mathbb{T} \not\leq_{\mathrm{pp}} \mathbb{K}_3$*)*



Dichotomy

$\mathbb{T}$ a finite relational structure

Theorem (Bulatov17,Zhuk17)

The following are equivalent

1. $\mathbb{T}$ *has a Siggers polymorphism and* $\mathrm{CSP}(\mathbb{T})$ *is in P*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{K}_3$ *(*$\mathbb{T} \not\leq_{\mathrm{pp}} \mathbb{K}_3$*)*

$\mathbb{T} \leq_{\mathrm{pp}} \mathbb{S} \Rightarrow \mathrm{CSP}(\mathbb{T}) \geq_{\log} \mathrm{CSP}(\mathbb{S})$

Dichotomy

$\mathbb{T}$ a finite relational structure

Theorem (Bulatov17,Zhuk17)

The following are equivalent

1. $\mathbb{T}$ *has a Siggers polymorphism and* $\mathrm{CSP}(\mathbb{T})$ *is in P*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{K}_3$ *(*$\mathbb{T} \not\leq_{\mathrm{pp}} \mathbb{K}_3$*)*

has NP-complete CSP

$\mathbb{T} \leq_{\mathrm{pp}} \mathbb{S} \Rightarrow \mathrm{CSP}(\mathbb{T}) \geq_{\log} \mathrm{CSP}(\mathbb{S})$

Dichotomy

$\mathbb{T}$ a finite relational structure

Theorem (Bulatov17,Zhuk17)

The following are equivalent

1. $\mathbb{T}$ *has a Siggers polymorphism and* $\mathrm{CSP}(\mathbb{T})$ *is in P*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{K}_3$ *(*$\mathbb{T} \not\leq_{\mathrm{pp}} \mathbb{K}_3$*)*

$\mathbb{T} \leq_{\mathrm{pp}} \mathbb{S} \Rightarrow \mathrm{CSP}(\mathbb{T}) \geq_{\log} \mathrm{CSP}(\mathbb{S})$

$\Rightarrow$ use $\leq_{\mathrm{pp}}$ to study complexity classes in P

Dichotomy

$\mathbb{T}$ a finite relational structure

Theorem (Bulatov17,Zhuk17)

The following are equivalent

1. $\mathbb{T}$ *has a Siggers polymorphism and* CSP($\mathbb{T}$) *is in P*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{K}_3$ *(*$\mathbb{T} \not\leq_{\mathrm{pp}} \mathbb{K}_3$*)*

$\mathbb{T} \leq_{\mathrm{pp}} \mathbb{S} \Rightarrow \mathrm{CSP}(\mathbb{T}) \geq_{\log} \mathrm{CSP}(\mathbb{S})$

$\Rightarrow$ use $\leq_{\mathrm{pp}}$ to study complexity classes in P

Definition later

Datalog

Theorem

The following are equivalent

1. $\mathrm{CSP}(\mathbb{T})$ *is solved by a Datalog program*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{D}_{3\mathrm{LIN}\,p}$ *for any prime p*

$$\mathbb{D}_{3\mathrm{LIN}\,p} := (\{0, \ldots, p-1\}, R_{0000}, R_{1000}, R_{2000}, \ldots)$$

$$R_{abcd} := \{(x, y, z) \mid ax + ay + cz = d\}$$

linear equations over $\mathbb{Z}_p$
with 3 variables

Datalog

Theorem

The following are equivalent

1. $\mathrm{CSP}(\mathbb{T})$ *is solved by a Datalog program*
2. $\mathbb{T}$ *cannot pp-construct* $\mathbb{D}_{\mathrm{3LIN}\,p}$ *for any prime* p

$$\mathbb{D}_{\mathrm{3LIN}\,p} := (\{0,\ldots,p-1\}, R_{0000}, R_{1000}, R_{2000}, \ldots)$$
$$R_{abcd} := \{(x,y,z) \mid ax + ay + cz = d\}$$

Lets come up with a new theorem!

Finite Duality

Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example

$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$

Finite Duality

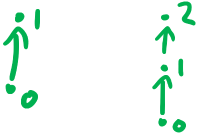
Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example

$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$



Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example

$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$
$D(\mathbb{P}_3) = \{\mathbb{P}_4\}$

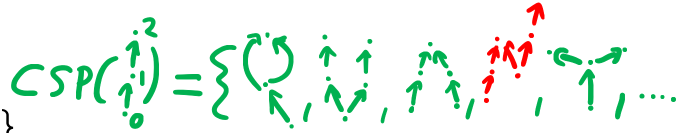
Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example



$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$
$D(\mathbb{P}_3) = \{\mathbb{P}_4\}$

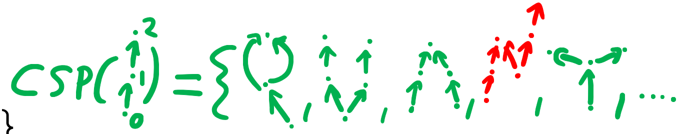
Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example



$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$

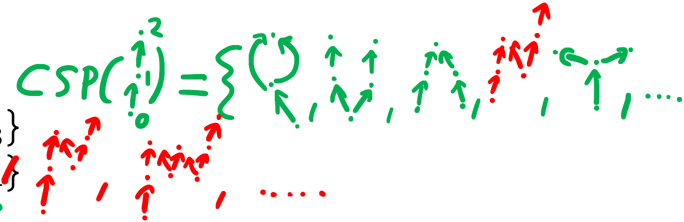
$D(\mathbb{P}_3) = \{\mathbb{P}_4\}$

Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$



Example

$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$

$D(\mathbb{P}_3) = \{\mathbb{P}_4\}$

Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example

$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$
$D(\mathbb{P}_3) = \cancel{\{\mathbb{P}_4\}}$ no finite duality

Finite Duality

Definition

$\mathbb{T}$ has *finite duality* if there is a finite set $D(\mathbb{T})$ of finite structures such that for all $\mathbb{I}$

$$\mathbb{I} \not\to \mathbb{T} \Leftrightarrow (\exists \mathbb{F} \in D(\mathbb{T}) : \mathbb{F} \to \mathbb{I})$$

Example

$D(\mathbb{P}_2) = \{\mathbb{P}_3\}$
$D(\mathbb{P}_3) = \cancel{\{\mathbb{P}_4\}}$ no finite duality

$$\mathrm{FD} := \{\mathrm{CSP}(\mathbb{T}) \mid \mathbb{T} \text{ has finite duality}\}$$

Finite Duality

Theorem (Atserias05)

$$\mathrm{FD} = \mathrm{FO}$$

Finite Duality

Theorem (Atserias05)

$$\mathrm{FD} = \mathrm{FO}$$

$$\begin{array}{c} \mathrm{NL} \\ | \\ \mathrm{L} \\ | \\ \mathrm{FO} = AC_0 \end{array}$$

Finite Duality

Theorem (Atserias05)

$$\mathrm{FD} = \mathrm{FO}$$

$$\begin{array}{c} \mathrm{NL} \\ | \\ \mathrm{L} \\ | \\ \mathrm{FO} \end{array}$$

Can we get a theorem for FD similar to the one for Datalog?

GOAL: find structures $\mathbb{A}_1, \mathbb{A}_2, \ldots$ such that
The following are equivalent

1. CSP($\mathbb{T}$) is in FD
2. $\mathbb{T}$ cannot pp-construct $\mathbb{A}_1, \mathbb{A}_2, \ldots$

GOAL: find structures $\mathbb{A}_1, \mathbb{A}_2, \ldots$ such that
The following are equivalent

1. CSP($\mathbb{T}$) is in FD
2. $\mathbb{T}$ cannot pp-construct $\mathbb{A}_1, \mathbb{A}_2, \ldots$

Definition?

Define pp-construction

homomorphic equivalence

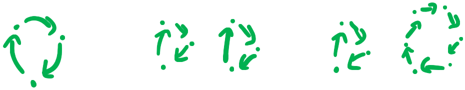
If $\mathbb{T} \to \mathbb{S}$ and $\mathbb{S} \to \mathbb{T}$, then $\mathbb{T} =_{\mathrm{pp}} \mathbb{S}$.

Define pp-construction

homomorphic equivalence

If $\mathbb{T} \to \mathbb{S}$ and $\mathbb{S} \to \mathbb{T}$, then $\mathbb{T} =_{\text{pp}} \mathbb{S}$.

Example

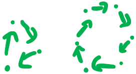


$$\mathbb{C}_3 \leftrightarrow \mathbb{C}_{3,3} \leftrightarrow \mathbb{C}_{3,6}$$

Define pp-construction

homomorphic equivalence

If $\mathbb{T} \to \mathbb{S}$ and $\mathbb{S} \to \mathbb{T}$, then $\mathbb{T} =_{\mathrm{pp}} \mathbb{S}$.

Example   

$$\mathbb{C}_3 \leftrightarrow \mathbb{C}_{3,3} \leftrightarrow \mathbb{C}_{3,6}$$

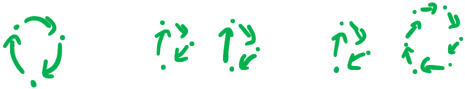
any graph with a loop $\leftrightarrow \mathbb{C}_1$ 

Define pp-construction

homomorphic equivalence

If $\mathbb{T} \to \mathbb{S}$ and $\mathbb{S} \to \mathbb{T}$, then $\mathbb{T} =_{\text{pp}} \mathbb{S}$.

Example



$$\mathbb{C}_3 \leftrightarrow \mathbb{C}_{3,3} \leftrightarrow \mathbb{C}_{3,6}$$

any graph with a loop $\leftrightarrow \mathbb{C}_1$ 

Note that: $\mathbb{T} \leftrightarrow \mathbb{S}$ implies $\mathrm{CSP}(\mathbb{T}) = \mathrm{CSP}(\mathbb{S})$

Define pp-construction

homomorphic equivalence

If $\mathbb{T} \to \mathbb{S}$ and $\mathbb{S} \to \mathbb{T}$, then $\mathbb{T} =_{\text{pp}} \mathbb{S}$.

Example

CSP( ) = { G | every cycle in G has a net length divisible by 3 }

$$\mathbb{C}_3 \leftrightarrow \mathbb{C}_{3,3} \leftrightarrow \mathbb{C}_{3,6}$$

any graph with a loop $\leftrightarrow \mathbb{C}_1$

CSP( ) = { all graphs }

Note that: $\mathbb{T} \leftrightarrow \mathbb{S}$ implies $\mathrm{CSP}(\mathbb{T}) = \mathrm{CSP}(\mathbb{S})$

Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(k)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\mathrm{pp}} \mathbb{S}$.

Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(k)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{pp} \mathbb{S}$.

$$\Phi(\bar{x}) = \exists \bar{y}.\ R_1(\bar{z}_1) \wedge \ldots \wedge R_\ell(\bar{z}_\ell) \wedge (x_{i_1} = x_{j_1}) \wedge \ldots$$

Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(k)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\text{pp}} \mathbb{S}$.

$$\Phi(\bar{x}) = \exists \bar{y}.\ R_1(\bar{z}_1) \wedge \ldots \wedge R_\ell(\bar{z}_\ell) \wedge (x_{i_1} = x_{j_1}) \wedge \ldots$$

T and S can have different signatures

Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(K)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\mathrm{pp}} \mathbb{S}$.

Example

$\mathbb{P}_3 \leq_{\mathrm{pp}} \mathbb{P}_2$

$\Phi_{\uparrow}(x,y) = \exists z.\ x \to y \wedge y \to z$

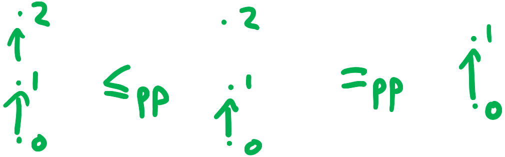
Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(K)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\text{pp}} \mathbb{S}$.

Example

$\mathbb{P}_3 \leq_{\text{pp}} \mathbb{P}_2$



$$\Phi_{\uparrow}(x,y) = \exists z.\ x \to y \wedge y \to z$$

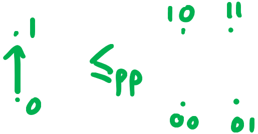
Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(K)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{pp} \mathbb{S}$.

Example

$\mathbb{P}_3 \leq_{pp} \mathbb{P}_2$



$\mathbb{P}_2 \leq_{pp} \mathbb{P}_3$

$\Phi_{\uparrow}(x_1, x_2, y_1, y_2) =$

Define pp-construction

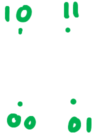
pp-power

If $S = T^n$ and every relation $R^{(K)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\text{pp}} \mathbb{S}$.

Example

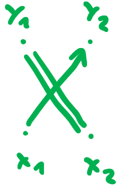
$\mathbb{P}_3 \leq_{\text{pp}} \mathbb{P}_2$

$\leq_{pp}$



$\mathbb{P}_2 \leq_{\text{pp}} \mathbb{P}_3$

$\Phi_{\uparrow}(x_1, x_2, y_1, y_2) =$



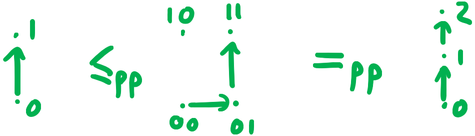
Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(K)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\text{pp}} \mathbb{S}$.

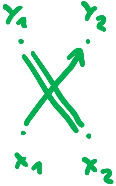
Example

$\mathbb{P}_3 \leq_{\text{pp}} \mathbb{P}_2$



$\mathbb{P}_2 \leq_{\text{pp}} \mathbb{P}_3$

$\Phi_{\uparrow}(x_1, x_2, y_1, y_2) =$



Define pp-construction

pp-power

If $S = T^n$ and every relation $R^{(K)}$ of $\mathbb{S}$ is (as a nk-ary relation) pp-definable in $\mathbb{T}$, then $\mathbb{T} \leq_{\text{pp}} \mathbb{S}$.

Example

$\mathbb{P}_3 \leq_{\text{pp}} \mathbb{P}_2$

$\Rightarrow$ $=_{\text{pp}}$

$\mathbb{P}_2 \leq_{\text{pp}} \mathbb{P}_3$

GOAL: find structures $\mathbb{A}_1, \mathbb{A}_2, \ldots$ such that
The following are equivalent

1. CSP($\mathbb{T}$) is in FD
2. $\mathbb{T}$ cannot pp-construct $\mathbb{A}_1, \mathbb{A}_2, \ldots$

OH NO! CSP($\mathbb{P}_2$) $\in$ FD, CSP($\mathbb{P}_3$) $\notin$ FD, and
$\mathbb{P}_2 =_{\text{pp}} \mathbb{P}_3$

GOAL: find structures $\mathbb{A}_1, \mathbb{A}_2, \ldots$ such that
The following are equivalent

1. CSP($\mathbb{T}$) is in FD
2. $\mathbb{T}$ cannot pp-construct $\mathbb{A}_1, \mathbb{A}_2, \ldots$

OH NO! CSP($\mathbb{P}_2$) $\in$ FD, CSP($\mathbb{P}_3$) $\notin$ FD, and $\mathbb{P}_2 =_{\mathrm{pp}} \mathbb{P}_3$

$\Rightarrow$ FD is not closed under pp-constructions

NEW GOAL: find structures $\mathbb{A}_1, \mathbb{A}_2, \ldots$ such that
The following are equivalent

1. CSP($\mathbb{T}$) is in PP(FD)
2. $\mathbb{T}$ cannot pp-construct $\mathbb{A}_1, \mathbb{A}_2, \ldots$

PP(FD) - First observations

PP(FD) - First observations

$$\mathrm{PP(FD)} \subseteq \mathrm{PP(L)} = \mathrm{L}$$

$\mathrm{FD} \subset \mathrm{L}$

PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

$\mathrm{CSP}(\mathbb{O})$ is NL-hard

with constants

PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

$\mathrm{CSP}(\mathbb{O})$ is NL-hard

with constants

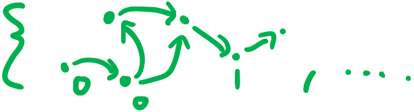
$(\{0,1\}, \leq, 0, 1)$

PP(FD) - First observations

$\mathrm{PP(FD)} \subseteq \mathrm{PP(L)} = \mathrm{L}$

CSP($\mathbb{O}$) is NL-hard

with constants $(\{0,1\}, \leq, 0, 1)$



PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

CSP($\mathbb{O}$) is NL-hard

with constants $(\{0,1\}, \leq, 0, 1)$

$\Vert$

$\{$ all partially labelled graphs with no directed path from 1 to 0 $\}$

PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

$\mathrm{CSP}(\mathbb{O})$ is NL-hard

with constants $(\{0,1\}, \leq, 0, 1)$

$\|$

$\{$all partially labelled graphs with no directed path from 1 to 0$\}$ coNL-hard

PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

$\mathrm{CSP}(\mathbb{O})$ is NL-hard
$\Longrightarrow$ If $\mathrm{L} \neq \mathrm{NL}$, then no problem in PP(FD) can pp-construct $\mathbb{O}$

PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

CSP($\mathbb{O}$) is NL-hard
$\Longrightarrow$ If $\mathrm{L} \neq \mathrm{NL}$, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency

PP(FD) - First observations

$\mathrm{PP(FD)} \subseteq \mathrm{PP(L)} = \mathrm{L}$

$\mathrm{CSP}(\mathbb{O})$ is NL-hard
$\Longrightarrow$ If $\mathrm{L} \neq \mathrm{NL}$, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency

can solve exactly the CSPs with tree duality

PP(FD) - First observations

PP(FD) $\subseteq$ PP(L) = L

CSP($\mathbb{O}$) is NL-hard
$\Longrightarrow$ If L $\neq$ NL, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency
AC is closed under pp-constructions

PP(FD) - First observations

PP(FD) $\subseteq$ PP(L) = L

CSP($\mathbb{O}$) is NL-hard
$\Longrightarrow$ If L $\neq$ NL, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency
AC is closed under pp-constructions
AC cannot solve CSP($\mathbb{C}_p$) for any prime p

PP(FD) - First observations

$\mathrm{PP}(\mathrm{FD}) \subseteq \mathrm{PP}(\mathrm{L}) = \mathrm{L}$

CSP($\mathbb{O}$) is NL-hard
$\Longrightarrow$ If L $\neq$ NL, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency
AC is closed under pp-constructions
AC cannot solve CSP($\mathbb{C}_p$) for any prime p

does not have tree duality

PP(FD) - First observations

PP(FD) $\subseteq$ PP(L) = L

CSP($\mathbb{O}$) is NL-hard
$\Rightarrow$ If L $\neq$ NL, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency
AC is closed under pp-constructions
AC cannot solve CSP($\mathbb{C}_p$) for any prime p
$\Rightarrow$ no problem in PP(FD) can pp-construct $\mathbb{C}_p$ for any p

PP(FD) - First observations

$\text{PP(FD)} \subseteq \text{PP(L)} = \text{L}$

CSP($\mathbb{O}$) is NL-hard
$\Longrightarrow$ If L $\neq$ NL, then no problem in PP(FD) can pp-construct $\mathbb{O}$

any problem in FD can be solved by Arc Consistency
AC is closed under pp-constructions
AC cannot solve CSP($\mathbb{C}_p$) for any prime p
$\Longrightarrow$ no problem in PP(FD) can pp-construct $\mathbb{C}_p$ for any p

Fun Fact CSP($\mathbb{C}_2$) = 2Colorability $\in$ L

Conjecture

The following are equivalent

1. CSP($\mathbb{T}$) is in PP(FD)
2. $\mathbb{T}$ cannot pp-construct $\mathbb{O}, \mathbb{C}_2, \mathbb{C}_3, \ldots$

Conjecture

The following are equivalent

1. CSP($\mathbb{T}$) is in PP(FD)
2. $\mathbb{T}$ cannot pp-construct $\mathbb{O}, \mathbb{C}_2, \mathbb{C}_3, \ldots$

What to do now?

Conjecture

The following are equivalent

1. CSP($\mathbb{T}$) is in PP(FD)
2. $\mathbb{T}$ cannot pp-construct $\mathbb{O}, \mathbb{C}_2, \mathbb{C}_3, \ldots$

What to do now?
Find another equivalent statement.

Dusl Programms

Dusl Programms

Definition

a *Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(\overline{x}) \dashv \exists \overline{y} : R_1(\overline{z_{11}}) \wedge \ldots \wedge S_1(\overline{z_{21}}) \wedge \ldots$

$S_1, \ldots \in \tau$, $R, R_1, \ldots \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Datalog programm for $\mathrm{CSP}(\mathbb{P}_2)$

Dusl Programms

Definition

a *Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(\overline{x}) \dashv \exists \overline{y} : R_1(\overline{z_{11}}) \wedge \ldots \wedge S_1(\overline{z_{21}}) \wedge \ldots$

$S_1, \ldots \in \tau$, $R, R_1, \ldots \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Datalog programm for CSP($\mathbb{P}_2$)

$$G \dashv \exists Y_1, Y_2, Y_3 : Y_1 \to Y_2 \wedge Y_2 \to Y_3$$

Dusl Programms

Definition

a *Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(\overline{x}) \dashv \exists \overline{y} : R_1(\overline{z_{11}}) \wedge \ldots \wedge S_1(\overline{z_{21}}) \wedge \ldots$

$S_1, \ldots \in \tau$, $R, R_1, \ldots \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Datalog programm for $\mathrm{CSP}(\mathbb{P}_2)$

$G \dashv$

Dusl Programms

Definition

a *Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(\overline{x}) \dashv \exists \overline{y} : R_1(\overline{z_{11}}) \wedge \ldots \wedge S_1(\overline{z_{21}}) \wedge \ldots$

$S_1, \ldots \in \tau$, $R, R_1, \ldots \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Datalog programm for CSP($\mathbb{T}$) $\in$ FD

$G \dashv \mathbb{F}$ for any $\mathbb{F} \in D(\mathbb{T})$

Dusl Programms

Definition

a *linear* *Datalog Program* consists of two signatures τ, σ, and a finite set of rules
Rules: $R(\overline{x}) \dashv \exists \overline{y} : R_1(\overline{z_1}) \wedge S_1(\overline{z_{21}}) \wedge \ldots$
$S_1, \ldots \in \tau$, $R, R_1 \in \sigma$
Input: a finite structure with signature τ
Output: can the program derive $G \in \sigma$

---

linear Datalog programm for $\mathrm{CSP}(\mathbb{C}_2)$

Dusl Programms

Definition

a *linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(\overline{x}) \dashv \exists \overline{y} : R_1(\overline{z_1}) \wedge S_1(\overline{z_{21}}) \wedge \dots$

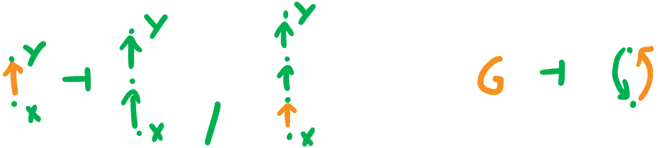
$S_1, \dots \in \tau,\ R, R_1 \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

---

linear Datalog programm for CSP$(\mathbb{C}_2)$



Dusl Programms

Definition

a *unary linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(x) \dashv \exists \overline{y} : R_1(z) \land S_1(\overline{z_{11}}) \land \ldots$

$S_1, \ldots \in \tau$, $R, R_1 \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

---

unary linear Datalog programm for CSP($\mathbb{O}$)

Dusl Programms

Definition

a *unary linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules
Rules: $R(x) \dashv \exists \overline{y} : R_1(z) \wedge S_1(\overline{z_{11}}) \wedge \dots$
$S_1, \dots \in \tau,\ R, R_1 \in \sigma$
Input: a finite structure with signature τ
Output: can the program derive $G \in \sigma$

---

unary linear Datalog programm for CSP($\mathbb{O}$)

$1(\overset{x}{\cdot}) \dashv 1(\overset{x}{\cdot}),\quad \begin{array}{l} \bullet\, x \\ \uparrow \\ \bullet\, 1 \end{array}$ $\qquad G \dashv \begin{array}{l} \bullet\, 0 \\ \uparrow \\ \bullet\, 1 \end{array}$

Dusl Programms

Definition

a *symmetric* *unary linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(x) \dashv \exists \overline{y} : R_1(z) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

$R_1(z) \dashv \exists \overline{y'} : R(x) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

$S_1, \ldots \in \tau$, $R, R_1 \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Dusl programm for $\mathrm{CSP}(\mathbb{P}_3)$

Dusl Programms

Definition

a *symmetric* *unary linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(x) \dashv \exists \overline{y} : R_1(z) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

$R_1(z) \dashv \exists \overline{y'} : R(x) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

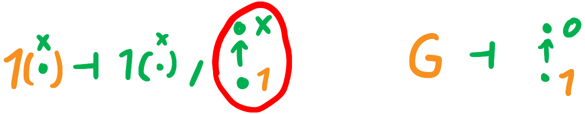
$S_1, \ldots \in \tau,\ R, R_1 \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

---

was not symmetric, $1(y) \dashv$  is missing



Dusl Programms

Definition

a *symmetric unary linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(x) \dashv \exists \overline{y} : R_1(z) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

$R_1(z) \dashv \exists \overline{y'} : R(x) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

$S_1, \ldots \in \tau$, $R, R_1 \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Dusl programm for $\mathrm{CSP}(\mathbb{P}_3)$

Dusl Programms

Definition

a *symmetric unary linear Datalog Program* consists of two signatures τ, σ, and a finite set of rules

Rules: $R(x) \dashv \exists \overline{y} : R_1(z) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

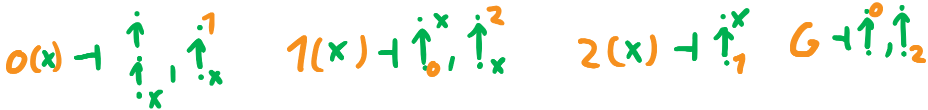
$R_1(z) \dashv \exists \overline{y'} : R(x) \wedge S_1(\overline{z_{11}}) \wedge \ldots$

$S_1, \ldots \in \tau,\ R, R_1 \in \sigma$

Input: a finite structure with signature τ

Output: can the program derive $G \in \sigma$

Dusl programm for $\mathrm{CSP}(\mathbb{P}_3)$



Conjecture

The following are equivalent

1. CSP($\mathbb{T}$) is in PP(FD)
2. $\mathbb{T}$ cannot pp-construct $\mathbb{O}, \mathbb{C}_2, \mathbb{C}_3, \ldots$
3. CSP($\mathbb{T}$) is solved by some Dusl program

What we know

1. CSP($\mathbb{T}$) is in PP(FD)

if $L \neq NL$

2. $\mathbb{T}$ cannot pp-construct $\mathbb{O}, \overbrace{\mathbb{C}_2, \mathbb{C}_3, \ldots}^{\text{Solved by AC}}$

3. CSP($\mathbb{T}$) is solved by some Dusl program

What we know

1. CSP($\mathbb{T}$) is in PP(FD)
2. $\mathbb{T}$ cannot pp-construct $\underbrace{\mathbb{O}, \mathbb{C}_2, \mathbb{C}_3, \ldots}$
3. CSP($\mathbb{T}$) is solved by some Dusl program

solved by AC

if $L \neq NL$

if $L \neq NL$

What we know

1. CSP($\mathbb{T}$) is in PP(FD)

solved by AC

if $L \neq NL$

2. $\mathbb{T}$ cannot pp-construct $\mathbb{O}, \mathbb{C}_2, \mathbb{C}_3, \ldots$

if $L \neq NL$

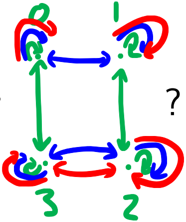
if Dusl is closed under $\leq_{pp}$

3. CSP($\mathbb{T}$) is solved by some Dusl program

Open Questions

1. are dusl programms closed under pp constructions?
2. Is $\mathbb{N}_{123}$ in PP(FD)?

3. Is there a Dusl program for



?

with constants