Datalog reductions between constraint satisfaction problems

Jakub Opršal (ISTA)

Joined work with V. Dalmau (UPF) and M. Wrochna (MIMUW).

European Research Council
Established by the European Commission

This project has received funding under the European Union's Horizon 2020 research and innovation programme (ERC grant agreement No 714532 & MSCA grant agreement No 101034413).

Part I

Why do we care about reductions?

A reduction from a problem A to a problem B is an (efficiently computable) function ϕ that maps instances of A to instances of B and preserves the answer, i.e.,

- if $i \in A$ then $\phi(i) \in B$, and
- if $i \notin A$ then $\phi(i) \notin B$.

the class NP under P-time reductions

if $P = NP$

if $P \neq NP$

the constraint satisfaction problem(s)

CSP Given a list of constraints over some domain D involving variables from V where each constraint is of the form $(v_1, \ldots, v_k) \in R$ for some $R \subseteq D^k$, decide whether there is a satisfying assignment $V \to D$.

the constraint satisfaction problem(s)

CSP Given a list of constraints over some domain D involving variables from V where each constraint is of the form $(v_1, \dots, v_k) \in R$ for some $R \subseteq D^k$, decide whether there is a satisfying assignment $V \to D$.

CSP($\mathbf{A}$) Fix a relational structure $\mathbf{A}$ (e.g., a graph). Given a relational structure $\mathbf{Q}$ of the same type, decide if there is a homomoprhism $h\colon \mathbf{Q} \to \mathbf{A}$.

the constraint satisfaction problem(s)

CSP Given a list of constraints over some domain D involving variables from V where each constraint is of the form $(v_1, \dots, v_k) \in R$ for some $R \subseteq D^k$, decide whether there is a satisfying assignment $V \to D$.

CSP($\mathbf{A}$) Fix a relational structure $\mathbf{A}$ (e.g., a graph). Given a relational structure $\mathbf{Q}$ of the same type, decide if there is a homomoprhism $h \colon \mathbf{Q} \to \mathbf{A}$.

examples

- CSP(K_3) is the 3-colouring.
- 3-SAT is expressible as CSP$(\mathbf{S}_3)$ for a suitable $\mathbf{S}_3$.
- Solving systems of linear equations mod p is CSP$(\mathbb{Z}_p)$.

a brief introduction to algebraic approach

a brief introduction to algebraic approach

- assigns to $\mathrm{CSP}(\mathbf{A})$ an 'algebraic' structure $\mathrm{pol}(\mathbf{A})$
- shows that up to log-space reductions, the complexity of $\mathrm{CSP}(\mathbf{A})$ depends only on $\mathrm{pol}(\mathbf{A})$

a brief introduction to algebraic approach

- assigns to $\mathrm{CSP}(\mathbf{A})$ an 'algebraic' structure $\mathrm{pol}(\mathbf{A})$
- shows that up to log-space reductions, the complexity of $\mathrm{CSP}(\mathbf{A})$ depends only on $\mathrm{pol}(\mathbf{A})$

In fact, the algebraic approach is a characterisation of gadget reductions.

a brief introduction to algebraic approach

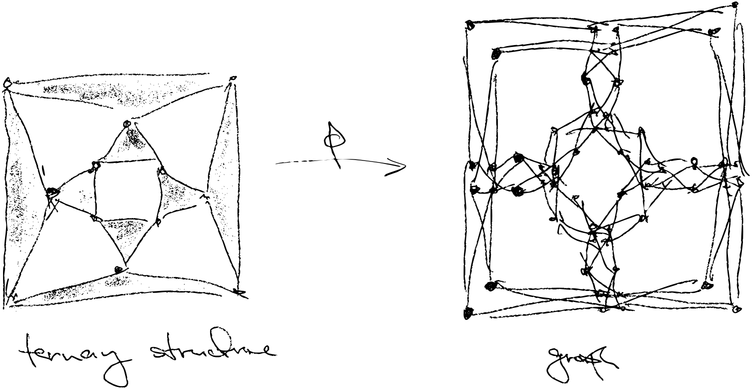
- assigns to $\mathrm{CSP}(\mathbf{A})$ an 'algebraic' structure $\mathrm{pol}(\mathbf{A})$
- shows that up to log-space reductions, the complexity of $\mathrm{CSP}(\mathbf{A})$ depends only on $\mathrm{pol}(\mathbf{A})$

In fact, the algebraic approach is a characterisation of gadget reductions.

Theorem [Bulatov, Jeavons, & Krokhin '05 and Barto, __, & Pinsker '17].

$$\mathrm{CSP}(\mathbf{A}) \leq_{\text{gadget}} \mathrm{CSP}(\mathbf{B}) \quad \text{iff} \quad \mathrm{pol}(\mathbf{B}) \to \mathrm{pol}(\mathbf{A})$$

a gadget reduction



a failure of algebraic approach

Widen the scope of CSPs to promise problems.

a failure of algebraic approach

Widen the scope of CSPs to promise problems.

PCSP($\mathbf{A}, \mathbf{B}$) Fix relational structures $\mathbf{A} \to \mathbf{B}$. Given a relational structure $\mathbf{Q}$ of the same type, decide between two cases:

- there is a homomorphism $h\colon \mathbf{Q} \to \mathbf{A}$, or
- there is **no** homomorphism $h'\colon \mathbf{Q} \to \mathbf{B}$.

a failure of algebraic approach

Widen the scope of CSPs to promise problems.

PCSP($\mathbf{A}$, $\mathbf{B}$) Fix relational structures $\mathbf{A} \to \mathbf{B}$. Given a relational structure $\mathbf{Q}$ of the same type, decide between two cases:

- there is a homomorphism $h\colon \mathbf{Q} \to \mathbf{A}$, or
- there is **no** homomorphism $h'\colon \mathbf{Q} \to \mathbf{B}$.

Example. Given a graph G decide whether it is 3-colourable or not even 5-colourable.

a failure of algebraic approach

Widen the scope of CSPs to promise problems.

PCSP($\mathbf{A}, \mathbf{B}$) Fix relational structures $\mathbf{A} \to \mathbf{B}$. Given a relational structure $\mathbf{Q}$ of the same type, decide between two cases:

- there is a homomorphism $h\colon \mathbf{Q} \to \mathbf{A}$, or
- there is **no** homomorphism $h'\colon \mathbf{Q} \to \mathbf{B}$.

Example. Given a graph G decide whether it is 3-colourable or not even 5-colourable.

Not all NP-hardness of PCSPs is shown by gadget reductions, e.g., $\mathrm{PCSP}(K_3, K_5)$ is NP-hard, but

$$\text{3-SAT} \not\leq_{\text{gadget}} \mathrm{PCSP}(K_3, K_5)$$

Part II

The reduction

Datalog programs

Datalog program ϕ with input signature τ is a finite set of rules of the form

$$R(x_1, \dots, x_n) \leftarrow S_1(x_{i_1}, \dots, x_{i_{k_1}}), \dots, S_r(x_{i_{*+1}}, \dots, x_{i_{*+k_r}})$$

where the symbols come from $\tau' \supseteq \tau$. We design one symbol $O \in \tau'$ as an output — we call it's arity m the arity of the Datalog program.

Datalog programs

Datalog program ϕ with input signature τ is a finite set of rules of the form

$$R(x_1, \dots, x_n) \leftarrow S_1(x_{i_1}, \dots, x_{i_{k_1}}), \dots, S_r(x_{i_{*+1}}, \dots, x_{i_{*+k_r}})$$

where the symbols come from $\tau' \supseteq \tau$. We design one symbol $O \in \tau'$ as an output — we call it's arity m the arity of the Datalog program.

For a τ-structure $\phi(\mathbf{A})$ is then computed as follows:

1. initialise: $R^\tau = R^{\mathbf{A}}$ if $R \in \tau$ and $R^\tau = \emptyset$ otherwise.
2. repeat until stabilises: whenever for some $x_1, \dots, x_k \in A$ match the body, add $(x_1, \dots, x_n)$ into R.
3. output: $O^\tau \subseteq A^m$.

Datalog programs

Datalog program ϕ with input signature τ is a finite set of rules of the form

$$R(x_1, \dots, x_n) \leftarrow S_1(x_{i_1}, \dots, x_{i_{k_1}}), \dots, S_r(x_{i_{*+1}}, \dots, x_{i_{*+k_r}})$$

where the symbols come from $\tau' \supseteq \tau$. We design one symbol $O \in \tau'$ as an output — we call it's arity m the arity of the Datalog program.

For a τ-structure $\phi(\mathbf{A})$ is then computed as follows:

1. initialise: $R^\tau = R^{\mathbf{A}}$ if $R \in \tau$ and $R^\tau = \emptyset$ otherwise.
2. repeat until stabilises: whenever for some $x_1, \dots, x_k \in A$ match the body, add $(x_1, \dots, x_n)$ into R.
3. output: $O^\tau \subseteq A^m$.

Datalog can be viewed as a fragment of $\exists^+\mathcal{L}^k_{\infty,\omega}$.

local reductions

Datalog interpretation. Fix a signature σ, a τ-structure $\mathbf{A}$, and Datalog programs ϕ and ϕ_R for $R \in \sigma$ of arities m and $m\,\mathrm{ar}(R)$ for $R \in \sigma$.

$$\mathbf{B} = (\phi(\mathbf{A}); \phi_R(\mathbf{A}), \dots)$$

where $\phi_R(\mathbf{A})$ is interpreted as $\mathrm{ar}(R)$-ary relation on $\phi(\mathbf{A})$.

local reductions

Datalog interpretation. Fix a signature σ, a τ-structure $\mathbf{A}$, and Datalog programs ϕ and ϕ_R for $R \in \sigma$ of arities m and $m\,\mathrm{ar}(R)$ for $R \in \sigma$.

$$\mathbf{B} = (\phi(\mathbf{A}); \phi_R(\mathbf{A}), \dots)$$

where $\phi_R(\mathbf{A})$ is interpreted as $\mathrm{ar}(R)$-ary relation on $\phi(\mathbf{A})$.

Definition

A local construction is arbitrary composition of Datalog interpretations and gadget replacement.
We say that CSP($\mathbf{A}$) locally reduces to CSP($\mathbf{B}$) if there is a local construction that is a reduction between these two problems.

local reductions

Datalog interpretation. Fix a signature σ, a τ-structure $\mathbf{A}$, and Datalog programs ϕ and ϕ_R for $R \in \sigma$ of arities m and $m\,\mathrm{ar}(R)$ for $R \in \sigma$.

$$\mathbf{B} = (\phi(\mathbf{A}); \phi_R(\mathbf{A}), \dots)$$

where $\phi_R(\mathbf{A})$ is interpreted as $\mathrm{ar}(R)$-ary relation on $\phi(\mathbf{A})$.

Definition

A local construction is arbitrary composition of Datalog interpretations and gadget replacement.
We say that $\mathrm{CSP}(\mathbf{A})$ locally reduces to $\mathrm{CSP}(\mathbf{B})$ if there is a local construction that is a reduction between these two problems.

Example. $\mathrm{CSP}(\mathbf{K}_2)$ locally reduces to $\mathrm{CSP}(\mathbf{T})$ where $\mathbf{T} = (\{*\}; \perp)$ with $\perp$ being the nullary empty relation.

k-consistency as a reduction

Assume we are reducing from $\mathrm{CSP}(\mathbf{A})$ to $\mathrm{CSP}(\mathbf{B})$. Given an instance $\mathbf{Q}$ of $\mathrm{CSP}(\mathbf{A})$:

k-consistency as a reduction

Assume we are reducing from $\mathrm{CSP}(\mathbf{A})$ to $\mathrm{CSP}(\mathbf{B})$. Given an instance $\mathbf{Q}$ of $\mathrm{CSP}(\mathbf{A})$:

1. for each $K \in \binom{Q}{\leq k}$, let $\mathcal{F}_K$ be the set of partial homomorphisms $h\colon K \to \mathbf{A}$.

k-consistency as a reduction

Assume we are reducing from CSP($\mathbf{A}$) to CSP($\mathbf{B}$). Given an instance $\mathbf{Q}$ of CSP($\mathbf{A}$):

1. for each $K \in \binom{Q}{\leq k}$, let $\mathcal{F}_K$ be the set of partial homomorphisms $h\colon K \to \mathbf{A}$.
2. while anything changes: for each $L \subset K$
   - $\mathcal{F}_L \leftarrow \mathcal{F}_L \cap \{h|_L : h \in \mathcal{F}_K\}$,
   - remove from $\mathcal{F}_K$ all h, s.t., $h|_L \notin \mathcal{F}_L$.

k-consistency as a reduction

Assume we are reducing from CSP($\mathbf{A}$) to CSP($\mathbf{B}$). Given an instance $\mathbf{Q}$ of CSP($\mathbf{A}$):

1. for each $K \in \binom{Q}{\leq k}$, let $\mathcal{F}_K$ be the set of partial homomorphisms $h\colon K \to \mathbf{A}$.
2. while anything changes: for each $L \subset K$
   - $\mathcal{F}_L \leftarrow \mathcal{F}_L \cap \{h|_L : h \in \mathcal{F}_K\}$,
   - remove from $\mathcal{F}_K$ all h, s.t., $h|_L \notin \mathcal{F}_L$.
3. create the output instance $\phi(\mathbf{Q})$ of CSP($\mathbf{B}$):
   - for each K, introduce to $\phi(\mathbf{Q})$ a copy of $\mathbf{B}^{\mathcal{F}_K}$.
   - for each $L \subset K$, identify each element $b\colon \mathcal{F}_L \to B$ of $\mathbf{B}^{\mathcal{F}_K}$ with the element b' of $\mathbf{B}^{\mathcal{F}_K}$ defined as $b'(h) = b(h|_L)$.

Part III

What can we prove?

some observations

- Every CSP whose complement is definable in Datalog (i.e., every CSP that is solvable by local consistency) is reducible to the trivial CSP.

some observations

- Every CSP whose complement is definable in Datalog (i.e., every CSP that is solvable by local consistency) is reducible to the trivial CSP.
- $\mathrm{CSP}(\mathbf{A})$ is solvable by some level of Sherali-Adams relaxation iff it is locally reducible to Linear Programming.

some observations

- Every CSP whose complement is definable in Datalog (i.e., every CSP that is solvable by local consistency) is reducible to the trivial CSP.
- CSP($\mathbf{A}$) is solvable by some level of Sherali-Adams relaxation iff it is locally reducible to Linear Programming.
- CSP($\mathbb{Z}_p$) is not locally reducible to CSP($\mathbb{Z}_q$) for any primes $p \neq q$. [Atserias, Bulatov, & Dawar '09 and Grädel & Pakusa '19]

some observations

- Every CSP whose complement is definable in Datalog (i.e., every CSP that is solvable by local consistency) is reducible to the trivial CSP.
- CSP($\mathbf{A}$) is solvable by some level of Sherali-Adams relaxation iff it is locally reducible to Linear Programming.
- CSP($\mathbb{Z}_p$) is not locally reducible to CSP($\mathbb{Z}_q$) for any primes $p \neq q$. [Atserias, Bulatov, & Dawar '09 and Grädel & Pakusa '19]
- 3-SAT is not locally reducible to CSP($\mathbb{Z}_p$) for any prime p.

what can we prove?

- Local reductions have the same power as k-consistency reductions.

what can we prove?

- Local reductions have the same power as k-consistency reductions.
- [Barto & Kozik '22] give a sufficient (but not necessary) condition for a local reduction.

what can we prove?

- Local reductions have the same power as k-consistency reductions.
- [Barto & Kozik '22] give a sufficient (but not necessary) condition for a local reduction.
- 3-SAT $\leq_{\text{local}} \text{PCSP}(K_3, K_5)$ [Barto & Kozik '22 and Wrochna '22].

what can we prove?

- Local reductions have the same power as k-consistency reductions.
- [Barto & Kozik '22] give a sufficient (but not necessary) condition for a local reduction.
- 3-SAT $\leq_{\text{local}}$ $\text{PCSP}(K_3, K_5)$ [Barto & Kozik '22 and Wrochna '22].
- We have a characterisation of arc-consistency reduction by the means of certain co-monad μ acting on polymorphisms:

Theorem
$\text{CSP}(\mathbf{A})$ *reduces to* $\text{CSP}(\mathbf{B})$ *by the arc-consistency reduction iff*

$$\mu(\text{pol}(\mathbf{B})) \to \text{pol}(\mathbf{A}).$$

what can’t we prove?

what can't we prove?

- Is 3-SAT locally reducible to $\mathrm{CSP}(\mathbb{Z})$? (Not unless P $\neq$ NP.)

what can't we prove?

- Is 3-SAT locally reducible to $\mathrm{CSP}(\mathbb{Z})$? (Not unless P $\neq$ NP.)
- Is every tractable CSP reducible to $\mathrm{CSP}(\mathbb{Z})$? Or to $\mathrm{CSP}(\mathbb{Z}_n)$ for some n?

what can't we prove?

- Is 3-SAT locally reducible to CSP($\mathbb{Z}$)? (Not unless P $\neq$ NP.)
- Is every tractable CSP reducible to CSP($\mathbb{Z}$)? Or to CSP($\mathbb{Z}_n$) for some n?

Conjecture.

For all finite structures $\mathbf{A}$, if 3-SAT $\not\leq_{\text{gadget}}$ CSP($\mathbf{A}$), then

$$\mathrm{CSP}(\mathbf{A}) \leq_{\text{local}} \mathrm{CSP}(\mathbb{Z})$$

what can't we prove?

- Is 3-SAT locally reducible to CSP($\mathbb{Z}$)? (Not unless P $\neq$ NP.)
- Is every tractable CSP reducible to CSP($\mathbb{Z}$)? Or to CSP($\mathbb{Z}_n$) for some n?

Conjecture.

For all finite structures $\mathbf{A}$, if 3-SAT $\not\leq_{\text{gadget}}$ CSP($\mathbf{A}$), then

$$\mathrm{CSP}(\mathbf{A}) \leq_{\text{local}} \mathrm{CSP}(\mathbb{Z})$$

■