Identifying Tractable Quantified Temporal Constraints

Žaneta Semanišinová
joint work with Jakub Rydval and Michał Wrona

Institute of Algebra
TU Dresden

Trends in Arithmetic Theories
6 Jul 2024

European Research Council
Established by the European Commission

ERC Synergy Grant POCOCOP (GA 101071674)

(Quantified) constraint satisfaction problem

(relational) structure $\mathfrak{B} = (B; R^{\mathfrak{B}} : R \in \tau)$; finite signature τ

primitive positive (pp) formula: $\exists y_1, \ldots, y_l(\psi_1 \wedge \cdots \wedge \psi_m)$, ψ_i atomic

(Quantified) constraint satisfaction problem

(relational) structure $\mathfrak{B} = (B; R^{\mathfrak{B}} : R \in \tau)$; finite signature τ
primitive positive (pp) formula: $\exists y_1, \ldots, y_l(\psi_1 \wedge \cdots \wedge \psi_m)$, ψ_i atomic

Definition (CSP)

Constraint Satisfaction Problem for $\mathfrak{B}$ (CSP($\mathfrak{B}$)):

Input: pp-formula Φ over τ
Question: $\mathfrak{B} \models \Phi$?

(Quantified) constraint satisfaction problem

(relational) structure $\mathfrak{B} = (B; R^{\mathfrak{B}} : R \in \tau)$; finite signature τ

primitive positive (pp) formula: $\exists y_1, \ldots, y_l(\psi_1 \wedge \cdots \wedge \psi_m)$, ψ_i atomic

Definition (CSP)

Constraint Satisfaction Problem for $\mathfrak{B}$ (CSP($\mathfrak{B}$)):

Input: pp-formula Φ over τ

Question: $\mathfrak{B} \models \Phi$?

quantified primitive positive (qpp) formula: both $\forall$ and $\exists$ are allowed

(Quantified) constraint satisfaction problem

(relational) structure $\mathfrak{B} = (B; R^{\mathfrak{B}} : R \in \tau)$; finite signature τ
primitive positive (pp) formula: $\exists y_1, \ldots, y_l (\psi_1 \wedge \cdots \wedge \psi_m)$, ψ_i atomic

Definition (CSP)

Constraint Satisfaction Problem for $\mathfrak{B}$ (CSP($\mathfrak{B}$)):

Input: pp-formula Φ over τ
Question: $\mathfrak{B} \models \Phi$?

quantified primitive positive (qpp) formula: both $\forall$ and $\exists$ are allowed

Definition (QCSP)

Quantified Constraint Satisfaction Problem for $\mathfrak{B}$ (QCSP($\mathfrak{B}$)):

Input: qpp-formula Φ over τ
Question: $\mathfrak{B} \models \Phi$?

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.
- EP: trying to satisfy all constraints.
- UP: trying to violate some constraint.

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.
- EP: trying to satisfy all constraints.
- UP: trying to violate some constraint.
- The EP has a winning strategy $\leftrightarrow \mathfrak{B} \models \Phi$.
- The UP has a winning strategy $\leftrightarrow \mathfrak{B} \not\models \Phi$.

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.
- EP: trying to satisfy all constraints.
- UP: trying to violate some constraint.
- The EP has a winning strategy $\leftrightarrow \mathfrak{B} \models \Phi$.
- The UP has a winning strategy $\leftrightarrow \mathfrak{B} \not\models \Phi$.

Example: $\mathrm{D} = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow y = z\}$.

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.
- EP: trying to satisfy all constraints.
- UP: trying to violate some constraint.
- The EP has a winning strategy $\leftrightarrow \mathfrak{B} \models \Phi$.
- The UP has a winning strategy $\leftrightarrow \mathfrak{B} \not\models \Phi$.

Example: $\mathrm{D} = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow y = z\}$.

- $\mathrm{CSP}(\mathbb{Q}; \mathrm{D})$ is trivial: $[\![x]\!] := 0$ for all x satisfies all constraints.

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.
- EP: trying to satisfy all constraints.
- UP: trying to violate some constraint.
- The EP has a winning strategy $\leftrightarrow \mathfrak{B} \models \Phi$.
- The UP has a winning strategy $\leftrightarrow \mathfrak{B} \not\models \Phi$.

Example: $\mathrm{D} = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow y = z\}$.

- $\mathrm{CSP}(\mathbb{Q}; \mathrm{D})$ is trivial: $[\![x]\!] := 0$ for all x satisfies all constraints.
- $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$ is PSPACE-complete (Zhuk, Martin, Wrona '23).

Quantified constraint satisfaction problem

- QCSP: a game between the universal player (UP) and the existential player (EP) assigning variables x with $[\![x]\!] \in B$.
- EP: trying to satisfy all constraints.
- UP: trying to violate some constraint.
- The EP has a winning strategy $\leftrightarrow \mathfrak{B} \models \Phi$.
- The UP has a winning strategy $\leftrightarrow \mathfrak{B} \not\models \Phi$.

Example: $\mathrm{D} = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow y = z\}$.

- $\mathrm{CSP}(\mathbb{Q}; \mathrm{D})$ is trivial: $[\![x]\!] := 0$ for all x satisfies all constraints.
- $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$ is PSPACE-complete (Zhuk, Martin, Wrona '23).

Intuition:

- UP: tries to force $u = v$ for some u, v with $[\![u]\!] \neq [\![v]\!]$
- EP: obeys the constraints, does not introduce unnecessary equalities

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)
- classification of QCSPs with all unary relations and 3-element structures with all singleton unary relations (Zhuk, Martin '22)

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)
- classification of QCSPs with all unary relations and 3-element structures with all singleton unary relations (Zhuk, Martin '22)

Equality (Q)CSPs (relations fo-definable in $(\mathbb{Q}; =)$):

- classification of CSPs (Bodirsky, Kára '08)

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)
- classification of QCSPs with all unary relations and 3-element structures with all singleton unary relations (Zhuk, Martin '22)

Equality (Q)CSPs (relations fo-definable in $(\mathbb{Q}; =)$):

- classification of CSPs (Bodirsky, Kára '08)
- partial classification of QCSPs (Bodirsky, Chen '10) – only complexity of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$ missing

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)
- classification of QCSPs with all unary relations and 3-element structures with all singleton unary relations (Zhuk, Martin '22)

Equality (Q)CSPs (relations fo-definable in $(\mathbb{Q}; =)$):

- classification of CSPs (Bodirsky, Kára '08)
- partial classification of QCSPs (Bodirsky, Chen '10) – only complexity of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$ missing
- full classification of QCSPs (Zhuk, Martin, Wrona '23)

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)
- classification of QCSPs with all unary relations and 3-element structures with all singleton unary relations (Zhuk, Martin '22)

Equality (Q)CSPs (relations fo-definable in $(\mathbb{Q}; =)$):

- classification of CSPs (Bodirsky, Kára '08)
- partial classification of QCSPs (Bodirsky, Chen '10) – only complexity of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$ missing
- full classification of QCSPs (Zhuk, Martin, Wrona '23)

Temporal (Q)CSPs (relations fo-definable in $(\mathbb{Q}; <)$):

- classification of CSPs (Bodirsky, Kára '10)

Overview of results on (Q)CSPs

Finite domains:

- classification of CSPs (Bulatov '17; Zhuk '17)
- classification of QCSPs with all unary relations and 3-element structures with all singleton unary relations (Zhuk, Martin '22)

Equality (Q)CSPs (relations fo-definable in $(\mathbb{Q}; =)$):

- classification of CSPs (Bodirsky, Kára '08)
- partial classification of QCSPs (Bodirsky, Chen '10) – only complexity of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$ missing
- full classification of QCSPs (Zhuk, Martin, Wrona '23)

Temporal (Q)CSPs (relations fo-definable in $(\mathbb{Q}; <)$):

- classification of CSPs (Bodirsky, Kára '10)
- some classification results on QCSPs (Charatonik, Wrona '08; Chen, Wrona '12; Bodirsky, Chen, Wrona '14; Wrona '14)

Ord-Horn constraints

Ord-Horn (OH) fragment: temporal structures whose relations are definable by an OH formula, i.e., a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_{k+1} \geq y_{k+1}) \text{ (last disjunct is optional).}$$

Ord-Horn constraints

Ord-Horn (OH) fragment: temporal structures whose relations are definable by an OH formula, i.e., a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_{k+1} \geq y_{k+1}) \text{ (last disjunct is optional).}$$

Example (complexity within OH):
QCSP($\mathbb{Q}$; R) where R is defined by $(x_1 \neq x_2 \vee x_3 \geq x_4) \wedge \phi$ is:

- in PTIME if $\phi = (x_3 \geq x_1) \wedge (x_1 \geq x_3) \wedge (x_3 \neq x_4)$ (Chen, Wrona '12)

Ord-Horn constraints

Ord-Horn (OH) fragment: temporal structures whose relations are definable by an OH formula, i.e., a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_{k+1} \geq y_{k+1}) \text{ (last disjunct is optional).}$$

Example (complexity within OH):
QCSP($\mathbb{Q}$; R) where R is defined by $(x_1 \neq x_2 \vee x_3 \geq x_4) \wedge \phi$ is:

- in PTIME if $\phi = (x_3 \geq x_1) \wedge (x_1 \geq x_3) \wedge (x_3 \neq x_4)$ (Chen, Wrona '12)
- coNP-complete if $\phi = (\bigwedge_{i,j \in \{1,2\}} x_i \neq x_{j+2})$ (Zhuk '22, pers. comm.)

Ord-Horn constraints

Ord-Horn (OH) fragment: temporal structures whose relations are definable by an OH formula, i.e., a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_{k+1} \geq y_{k+1}) \text{ (last disjunct is optional).}$$

Example (complexity within OH):
QCSP($\mathbb{Q}$; R) where R is defined by $(x_1 \neq x_2 \vee x_3 \geq x_4) \wedge \phi$ is:

- in PTIME if $\phi = (x_3 \geq x_1) \wedge (x_1 \geq x_3) \wedge (x_3 \neq x_4)$ (Chen, Wrona '12)
- coNP-complete if $\phi = (\bigwedge_{i,j \in \{1,2\}} x_i \neq x_{j+2})$ (Zhuk '22, pers. comm.)
- PSPACE-complete if ϕ is empty (Zhuk, Martin, Wrona '23)

Three cases for Ord-Horn constraints

Theorem (Wrona '14)

Let $\mathfrak{B}$ be an OH structure. Then one of the following holds:

- $\mathfrak{B}$ *is guarded OH.*
- QCSP($\mathfrak{B}$) *is* coNP*-hard.*
- $\mathfrak{B}$ *pp-defines* M^{+} *or* M^{-}*.*

Three cases for Ord-Horn constraints

Theorem (Wrona '14)

Let $\mathfrak{B}$ be an OH structure. Then one of the following holds:

- $\mathfrak{B}$ *is guarded OH.*
- QCSP($\mathfrak{B}$) *is* coNP*-hard.*
- $\mathfrak{B}$ *pp-defines* M^+ *or* M^-.

$$\mathrm{M}^+ \coloneqq \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$$
$$\mathrm{M}^- \coloneqq \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \leq z\}$$

Complexity of Ord-Horn constraints

Theorem (Chen, Wrona '12)

Let $\mathfrak{B}$ *be a guarded OH structure. Then* QCSP($\mathfrak{B}$) *is in* PTIME.

Complexity of Ord-Horn constraints

Theorem (Chen, Wrona '12)

Let $\mathfrak{B}$ be a guarded OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME.

Proposition (Bodirsky, Chen '10)

Let $\mathfrak{A}, \mathfrak{B}$ be structures with the same domain. If every relation of $\mathfrak{B}$ is qpp-definable in $\mathfrak{A}$, then QCSP($\mathfrak{B}$) *reduces to* QCSP($\mathfrak{A}$) *in* PTIME.

Complexity of Ord-Horn constraints

Theorem (Chen, Wrona '12)

Let $\mathfrak{B}$ be a guarded OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME.

Proposition (Bodirsky, Chen '10)

Let $\mathfrak{A}, \mathfrak{B}$ be structures with the same domain. If every relation of $\mathfrak{B}$ is qpp-definable in $\mathfrak{A}$, then QCSP($\mathfrak{B}$) *reduces to* QCSP($\mathfrak{A}$) *in* PTIME.

$\rightsquigarrow$ need to understand QCSP($\mathbb{Q}; \mathrm{M}^+$) (QCSP($\mathbb{Q}; \mathrm{M}^-$) is the dual problem)

Complexity of Ord-Horn constraints

Theorem (Chen, Wrona '12)

Let $\mathfrak{B}$ be a guarded OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME.

Proposition (Bodirsky, Chen '10)

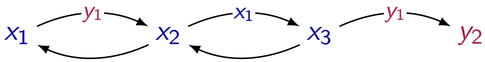
Let $\mathfrak{A}, \mathfrak{B}$ be structures with the same domain. If every relation of $\mathfrak{B}$ is qpp-definable in $\mathfrak{A}$, then QCSP($\mathfrak{B}$) *reduces to* QCSP($\mathfrak{A}$) *in* PTIME.

$\rightsquigarrow$ need to understand QCSP($\mathbb{Q}; \mathrm{M}^+$) (QCSP($\mathbb{Q}; \mathrm{M}^-$) is the dual problem)

Complexity of QCSP($\mathbb{Q}; \mathrm{M}^+$): left open in [Bodirsky, Chen, Wrona '14]
$\hookrightarrow$ could have been anywhere between PTIME and PSPACE

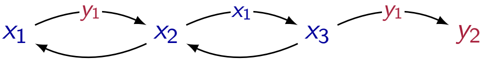
Example instance of QCSP($\mathbb{Q}; \mathrm{M}^+$)

$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \\ \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



Example instance of QCSP($\mathbb{Q}; \mathrm{M}^+$)

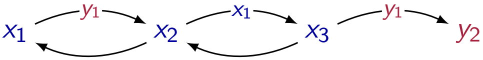
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1)\big).$$



- the EP plays $[\![x_1]\!] = 0$

Example instance of QCSP($\mathbb{Q}$; M^+)

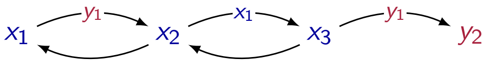
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$

Example instance of QCSP($\mathbb{Q}$; M^+)

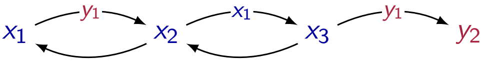
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \\ \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$
- the EP plays $[\![x_2]\!] = 0$, obeying $x_1 \geq x_2$ and $x_2 \geq x_1$

Example instance of QCSP($\mathbb{Q}$; M^+)

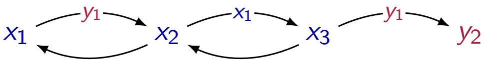
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \\ \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$
- the EP plays $[\![x_2]\!] = 0$, obeying $x_1 \geq x_2$ and $x_2 \geq x_1$
- this triggers the constraint $x_2 \geq x_3$

Example instance of QCSP($\mathbb{Q}$; M^+)

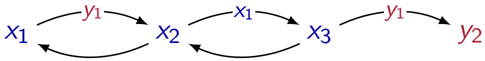
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \\ \wedge\, (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$
- the EP plays $[\![x_2]\!] = 0$, obeying $x_1 \geq x_2$ and $x_2 \geq x_1$
- this triggers the constraint $x_2 \geq x_3$
- the UP plays $[\![y_2]\!] = 1$

Example instance of QCSP($\mathbb{Q}; \mathrm{M}^+$)

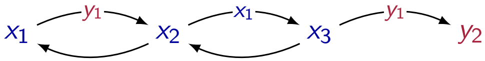
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \\ \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$
- the EP plays $[\![x_2]\!] = 0$, obeying $x_1 \geq x_2$ and $x_2 \geq x_1$
- this triggers the constraint $x_2 \geq x_3$
- the UP plays $[\![y_2]\!] = 1$
- the EP has to choose $[\![x_3]\!] = 0$, because of $x_2 \geq x_3$ and $x_3 \geq x_2$

Example instance of QCSP($\mathbb{Q}$; M^+)

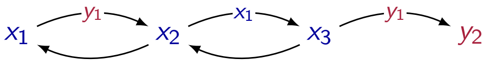
$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1)\big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$
- the EP plays $[\![x_2]\!] = 0$, obeying $x_1 \geq x_2$ and $x_2 \geq x_1$
- this triggers the constraint $x_2 \geq x_3$
- the UP plays $[\![y_2]\!] = 1$
- the EP has to choose $[\![x_3]\!] = 0$, because of $x_2 \geq x_3$ and $x_3 \geq x_2$
- $(x_3 = y_1 \Rightarrow x_3 \geq y_2)$ is now falsified

Example instance of QCSP($\mathbb{Q}; \mathrm{M}^+$)

$$\Phi = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 \big((x_1 = y_1 \Rightarrow x_1 \geq x_2) \wedge (x_2 = x_1 \Rightarrow x_2 \geq x_3) \\ \wedge (x_3 = y_1 \Rightarrow x_3 \geq y_2) \wedge (x_3 \geq x_2) \wedge (x_2 \geq x_1) \big).$$



- the EP plays $[\![x_1]\!] = 0$
- the UP plays $[\![y_1]\!] = 0$, triggering the constraint $x_1 \geq x_2$
- the EP plays $[\![x_2]\!] = 0$, obeying $x_1 \geq x_2$ and $x_2 \geq x_1$
- this triggers the constraint $x_2 \geq x_3$
- the UP plays $[\![y_2]\!] = 1$
- the EP has to choose $[\![x_3]\!] = 0$, because of $x_2 \geq x_3$ and $x_3 \geq x_2$
- $(x_3 = y_1 \Rightarrow x_3 \geq y_2)$ is now falsified
- the UP has a winning strategy on this instance $\Rightarrow$ Φ is false

The setup

Wanted: PTIME-algorithm for QCSP($\mathbb{Q}; \mathrm{M}^+$)
$\mathrm{M}^+ = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$

The setup

Wanted: PTIME-algorithm for QCSP$(\mathbb{Q}; \mathrm{M}^+)$
$\mathrm{M}^+ = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$

- fix instance Φ of QCSP$(\mathbb{Q}; \mathrm{M}^+)$ over variables $\mathrm{V} = \mathrm{V}_\exists \cup \mathrm{V}_\forall$

The setup

Wanted: PTIME-algorithm for $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$
$\mathrm{M}^+ = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$

- fix instance Φ of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ over variables $\mathrm{V} = \mathrm{V}_\exists \cup \mathrm{V}_\forall$
- $\phi :=$ quantifier-free part of Φ

The setup

Wanted: PTIME-algorithm for QCSP$(\mathbb{Q}; \mathrm{M}^+)$
$\mathrm{M}^+ = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$

- fix instance Φ of QCSP$(\mathbb{Q}; \mathrm{M}^+)$ over variables $\mathrm{V} = \mathrm{V}_\exists \cup \mathrm{V}_\forall$
- $\phi :=$ quantifier-free part of Φ
- $\prec :=$ linear order on V from the order in the quantifier prefix of Φ

The setup

Wanted: PTIME-algorithm for $\text{QCSP}(\mathbb{Q}; \mathrm{M}^+)$
$\mathrm{M}^+ = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$

- fix instance Φ of $\text{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ over variables $\mathrm{V} = \mathrm{V}_\exists \cup \mathrm{V}_\forall$
- $\phi :=$ quantifier-free part of Φ
- $\prec :=$ linear order on V from the order in the quantifier prefix of Φ
- we write $A \prec B$ meaning $x \prec y$, $\forall x \in A, y \in B$

The setup

Wanted: PTIME-algorithm for QCSP($\mathbb{Q}; \mathrm{M}^+$)
$\mathrm{M}^+ = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\}$

- fix instance Φ of QCSP($\mathbb{Q}; \mathrm{M}^+$) over variables $\mathrm{V} = \mathrm{V}_\exists \cup \mathrm{V}_\forall$
- $\phi :=$ quantifier-free part of Φ
- $\prec :=$ linear order on V from the order in the quantifier prefix of Φ
- we write $A \prec B$ meaning $x \prec y$, $\forall x \in A, y \in B$

Fact: It is possible to pp-define from M^+ constraints of the form

$$\left(\bigwedge_{v \in A} x = v\right) \Rightarrow x \geq z$$

by definitions of linear length.

x-z-cut

For $x, z \in \mathrm{V}$:

$$x\text{-}z\text{-cut} := \{u \in \mathrm{V}_\forall \mid (\mathrm{V}_\exists \cap \{x, z\}) \prec u\} \setminus \{z\}$$

- x-z-cut comprises variables that the UP can play equal to x to trigger the constraint $x \geq z$
- z is removed so that the constraint does not become trivial

x-z-cut

For $x, z \in \mathrm{V}$:

$$x\text{-}z\text{-cut} \coloneqq \{u \in \mathrm{V}_\forall \mid (\mathrm{V}_\exists \cap \{x, z\}) \prec u\} \setminus \{z\}$$

- x-z-cut comprises variables that the UP can play equal to x to trigger the constraint $x \geq z$
- z is removed so that the constraint does not become trivial

Example: $\Phi \coloneqq \exists u \forall v \exists w \forall x \forall y \; \phi(u, v, w, x, y)$

- u-w-cut $= \{x, y\}$
- u-x-cut $= \{v, y\}$
- v-x-cut $= \{v, y\}$

Sketch of the algorithm

- expand ϕ by constraints ψ of the form

$$\left(\bigwedge_{v \in A \setminus x\text{-}z\text{-cut}} x = v\right) \Rightarrow x \geq z$$

if $\phi \wedge \left(\bigwedge_{v \in A} x = v\right) \wedge (x < z)$ is unsatisfiable

Sketch of the algorithm

- expand ϕ by constraints ψ of the form

$$\left(\bigwedge_{v \in A \setminus x\text{-}z\text{-cut}} x = v\right) \Rightarrow x \geq z$$

  if $\phi \wedge \left(\bigwedge_{v \in A} x = v\right) \wedge (x < z)$ is unsatisfiable
- $A \subseteq \mathrm{V}_\forall$ is of the form

$$\uparrow_u \setminus \{x, z\} \quad \text{for } x, z, u \in \mathrm{V},$$

Sketch of the algorithm

- expand ϕ by constraints ψ of the form

$$\left(\bigwedge_{v \in A \setminus x\text{-}z\text{-cut}} x = v\right) \Rightarrow x \geq z$$

  if $\phi \wedge \left(\bigwedge_{v \in A} x = v\right) \wedge (x < z)$ is unsatisfiable
- $A \subseteq \mathrm{V}_\forall$ is of the form

$$\uparrow_u \setminus \{x, z\} \quad \text{for } x, z, u \in \mathrm{V},$$

  where $\uparrow_u := \{y \in \mathrm{V}_\forall \mid u \preceq y\}$

Sketch of the algorithm

- expand ϕ by constraints ψ of the form

$$\left(\bigwedge_{v \in A \setminus x\text{-}z\text{-cut}} x = v\right) \Rightarrow x \geq z$$

  if $\phi \wedge \left(\bigwedge_{v \in A} x = v\right) \wedge (x < z)$ is unsatisfiable
- $A \subseteq \mathrm{V}_\forall$ is of the form

$$\uparrow_u \setminus \{x, z\} \quad \text{for } x, z, u \in \mathrm{V},$$

  where $\uparrow_u := \{y \in \mathrm{V}_\forall \mid u \preceq y\}$
  $\hookrightarrow$ polynomially many such sets

Sketch of the algorithm

- expand ϕ by constraints ψ of the form

$$\left(\bigwedge_{v \in A \setminus x\text{-}z\text{-cut}} x = v\right) \Rightarrow x \geq z$$

  if $\phi \wedge \left(\bigwedge_{v \in A} x = v\right) \wedge (x < z)$ is unsatisfiable
- $A \subseteq \mathrm{V}_\forall$ is of the form

$$\uparrow_u \setminus \{x, z\} \quad \text{for } x, z, u \in \mathrm{V},$$

  where $\uparrow_u := \{y \in \mathrm{V}_\forall \mid u \preceq y\}$
  $\hookrightarrow$ polynomially many such sets
- reject if $(x \geq z)$ or $(z \geq x)$ is derived where $x \prec z$, $z \in \mathrm{V}_\forall$

Sketch of the algorithm

- expand ϕ by constraints ψ of the form

$$\left(\bigwedge_{v \in A \setminus x\text{-}z\text{-cut}} x = v\right) \Rightarrow x \geq z$$

  if $\phi \wedge \left(\bigwedge_{v \in A} x = v\right) \wedge (x < z)$ is unsatisfiable
- $A \subseteq \mathrm{V}_\forall$ is of the form

$$\uparrow_u \setminus \{x, z\} \quad \text{for } x, z, u \in \mathrm{V},$$

  where $\uparrow_u \coloneqq \{y \in \mathrm{V}_\forall \mid u \preceq y\}$
  $\hookrightarrow$ polynomially many such sets
- reject if $(x \geq z)$ or $(z \geq x)$ is derived where $x \prec z$, $z \in \mathrm{V}_\forall$
- accept if no new constraints can be derived

Algorithm for QCSP($\mathbb{Q}; \mathrm{M}^+$)

Input: an instance Φ of QCSP($\mathbb{Q}; \mathrm{M}^+$) with the quantifier-free part ϕ
Output: *true* or *false*

while ϕ *changes* **do**
  for $x, z, u \in \mathrm{V}$ **do**
    if ϕ *contains the clause* $(x \geq z)$ *or* $(z \geq x)$, *where* $x \prec z$ *and* $z \in \mathrm{V}_\forall$ **then**
      return *false*;
    if $\phi \wedge (\bigwedge_{v \in \uparrow_u \setminus \{x,z\}} x = v) \wedge (x < z)$ *is unsatisfiable* **then**
      expand ϕ by the clause $((\bigwedge_{v \in \uparrow_u \setminus (\{x,z\} \cup x\text{-}z\text{-cut})} x = v) \Rightarrow x \geq z)$;
return *true*;

Algorithm for QCSP($\mathbb{Q}; \mathrm{M}^+$)

Input: an instance Φ of QCSP($\mathbb{Q}; \mathrm{M}^+$) with the quantifier-free part ϕ
Output: *true* or *false*
while ϕ *changes* **do**
    for $x, z, u \in \mathrm{V}$ **do**
        if ϕ *contains the clause* $(x \geq z)$ *or* $(z \geq x)$*, where* $x \prec z$ *and* $z \in \mathrm{V}_\forall$ **then**
            return *false*;
        if $\phi \wedge (\bigwedge_{v \in \uparrow_u \setminus \{x,z\}} x = v) \wedge (x < z)$ *is unsatisfiable* **then**
            expand ϕ by the clause
            $((\bigwedge_{v \in \uparrow_u \setminus (\{x,z\} \cup x\text{-}z\text{-cut})} x = v) \Rightarrow x \geq z)$;
return *true*;

$\hookrightarrow$ CSP($\mathbb{Q}; <, \mathrm{M}^+$) is in PTIME $\Rightarrow$ the satisfiability test runs in PTIME

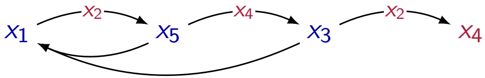
Algorithm for QCSP($\mathbb{Q}; \mathrm{M}^+$)

Input: an instance Φ of QCSP($\mathbb{Q}; \mathrm{M}^+$) with the quantifier-free part ϕ
Output: *true* or *false*
while ϕ *changes* **do**
  for $x, z, u \in \mathrm{V}$ **do**
    if ϕ *contains the clause* $(x \geq z)$ *or* $(z \geq x)$*, where* $x \prec z$ *and* $z \in \mathrm{V}_\forall$ **then**
      return *false*;
    if $\phi \wedge (\bigwedge_{v \in \uparrow_u \setminus \{x,z\}} x = v) \wedge (x < z)$ *is unsatisfiable* **then**
      expand ϕ by the clause
        $((\bigwedge_{v \in \uparrow_u \setminus (\{x,z\} \cup x\text{-}z\text{-cut})} x = v) \Rightarrow x \geq z)$;
return *true*;

$\hookrightarrow$ CSP($\mathbb{Q}; <, \mathrm{M}^+$) is in PTIME $\Rightarrow$ the satisfiability test runs in PTIME
$\hookrightarrow$ the algorithm runs in PTIME

Example of the run of the algorithm

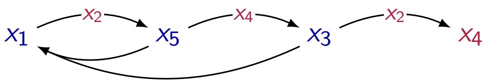
$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1) \big).$$



Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \\ \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$

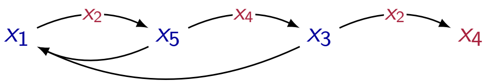


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$

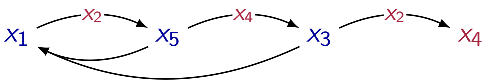


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$

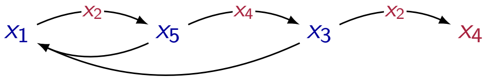


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.
- x_1-x_3-cut $= \{x_4\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1$-$x_3$-cut$) = \{x_2\}$

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1) \big).$$

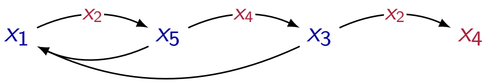


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.
- x_1-x_3-cut $= \{x_4\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1\text{-}x_3\text{-cut}) = \{x_2\}$
- Hence, the algorithm expands ϕ by $(x_1 = x_2 \Rightarrow x_1 \geq x_3)$.

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \\ \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1) \big).$$

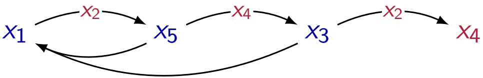


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.
- x_1-x_3-cut $= \{x_4\}$ $\rightsquigarrow$ $\uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1\text{-}x_3\text{-cut}) = \{x_2\}$
- Hence, the algorithm expands ϕ by $(x_1 = x_2 \Rightarrow x_1 \geq x_3)$.
- Now $\phi \wedge (x_1 = x_2)$ implies $x_1 = x_2 = x_3 \geq x_4$.

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$

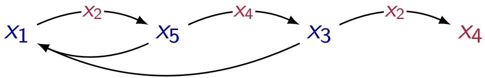


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.
- x_1-x_3-cut $= \{x_4\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1\text{-}x_3\text{-cut}) = \{x_2\}$
- Hence, the algorithm expands ϕ by $(x_1 = x_2 \Rightarrow x_1 \geq x_3)$.
- Now $\phi \wedge (x_1 = x_2)$ implies $x_1 = x_2 = x_3 \geq x_4$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_4\}} x_1 = v) \wedge (x_1 < x_4)$ is not satisfiable.

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$

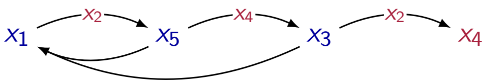


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.
- x_1-x_3-cut $= \{x_4\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1$-$x_3$-cut$) = \{x_2\}$
- Hence, the algorithm expands ϕ by $(x_1 = x_2 \Rightarrow x_1 \geq x_3)$.
- Now $\phi \wedge (x_1 = x_2)$ implies $x_1 = x_2 = x_3 \geq x_4$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_4\}} x_1 = v) \wedge (x_1 < x_4)$ is not satisfiable.
- x_1-x_4-cut $= \{x_2\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_4\} \cup x_1$-$x_4$-cut$) = \emptyset$

Example of the run of the algorithm

$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \\ \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$



Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ implies $x_1 = x_2 = x_4 = x_5 = x_3$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable.
- x_1-x_3-cut $= \{x_4\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1\text{-}x_3\text{-cut}) = \{x_2\}$
- Hence, the algorithm expands ϕ by $(x_1 = x_2 \Rightarrow x_1 \geq x_3)$.
- Now $\phi \wedge (x_1 = x_2)$ implies $x_1 = x_2 = x_3 \geq x_4$.
- Hence, $\phi \wedge (\bigwedge_{v \in \uparrow_{x_2} \setminus \{x_1, x_4\}} x_1 = v) \wedge (x_1 < x_4)$ is not satisfiable.
- x_1-x_4-cut $= \{x_2\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus (\{x_1, x_4\} \cup x_1\text{-}x_4\text{-cut}) = \emptyset$
- Hence, the algorithm expands ϕ by $(x_1 \geq x_4)$.

The algorithm on false instances

Lemma (Rydval, S., Wrona '24)

If the algorithm derives from Φ a constraint ψ, then Φ is true iff Φ expanded by ψ is true.

The algorithm on false instances

Lemma (Rydval, S., Wrona '24)

If the algorithm derives from Φ a constraint ψ, then Φ is true iff Φ expanded by ψ is true.

Proof: (almost) straightforward induction

The algorithm on false instances

Lemma (Rydval, S., Wrona '24)

If the algorithm derives from Φ a constraint ψ, then Φ is true iff Φ expanded by ψ is true.

Proof: (almost) straightforward induction

Whenever the algorithm rejects, it derived

$$x \geq z \text{ or } z \geq x \text{ where } x \prec z, z \in \mathrm{V}_{\forall}.$$

Lemma $\Rightarrow$ Φ is false $\Rightarrow$ the algorithm rejects false instances

The algorithm on true instances

Proof system $\mathcal{P}$:

The algorithm on true instances

Proof system $\mathcal{P}$:

- $\mathcal{P}$ derives predicates of the form $\mathcal{P}(x, z; A)$, $x, z \in \mathrm{V}$, $A \subseteq \mathrm{V}_\forall$.

The algorithm on true instances

Proof system $\mathcal{P}$:

- $\mathcal{P}$ derives predicates of the form $\mathcal{P}(x, z; A)$, $x, z \in \mathrm{V}$, $A \subseteq \mathrm{V}_\forall$.
- Intuitive interpretation: $\left(\bigwedge_{v \in A} x = v\right) \Rightarrow x \geq z$.

The algorithm on true instances

Proof system $\mathcal{P}$:

- $\mathcal{P}$ derives predicates of the form $\mathcal{P}(x, z; A)$, $x, z \in \mathrm{V}$, $A \subseteq \mathrm{V}_\forall$.
- Intuitive interpretation: $\left(\bigwedge_{v \in A} x = v\right) \Rightarrow x \geq z$.

Lemma (Rydval, S., Wrona '24)

- *If $\mathcal{P}(x, z; A)$ is derived and $z \notin A$, then the algorithm expands ϕ by*

$$\Big(\big(\bigwedge_{v \in \uparrow_A \setminus (\{x,z\} \cup x\text{-}z\text{-}cut)} x = v\big) \Rightarrow x \geq z\Big).$$

The algorithm on true instances

Proof system $\mathcal{P}$:

- $\mathcal{P}$ derives predicates of the form $\mathcal{P}(x, z; A)$, $x, z \in \mathrm{V}$, $A \subseteq \mathrm{V}_\forall$.
- Intuitive interpretation: $\left(\bigwedge_{v \in A} x = v\right) \Rightarrow x \geq z$.

Lemma (Rydval, S., Wrona '24)

- *If $\mathcal{P}(x, z; A)$ is derived and $z \notin A$, then the algorithm expands ϕ by*

$$\left(\left(\bigwedge_{v \in \uparrow_A \setminus (\{x,z\} \cup x\text{-}z\text{-}cut)} x = v\right) \Rightarrow x \geq z\right).$$

- *In particular, it expands ϕ by $(x \geq z)$ for every $\mathcal{P}(x, z; \emptyset)$.*

The algorithm on true instances

Proof system $\mathcal{P}$:

- $\mathcal{P}$ derives predicates of the form $\mathcal{P}(x, z; A)$, $x, z \in \mathrm{V}$, $A \subseteq \mathrm{V}_\forall$.
- Intuitive interpretation: $\left(\bigwedge_{v \in A} x = v\right) \Rightarrow x \geq z$.

Lemma (Rydval, S., Wrona '24)

- *If $\mathcal{P}(x, z; A)$ is derived and $z \notin A$, then the algorithm expands ϕ by*

$$\Big(\big(\bigwedge_{v \in \uparrow_A \setminus (\{x,z\} \cup x\text{-}z\text{-}cut)} x = v\big) \Rightarrow x \geq z\Big).$$

- *In particular, it expands ϕ by $(x \geq z)$ for every $\mathcal{P}(x, z; \emptyset)$.*
- *If the proof system does not derive $\mathcal{P}(x, z; \emptyset)$ or $\mathcal{P}(z, x; \emptyset)$ for $x \prec z$, $z \in \mathrm{V}_\forall$, then Φ is true.*

$\rightsquigarrow$ the algorithm accepts correctly

The algorithm on true instances

Proof system $\mathcal{P}$:

- $\mathcal{P}$ derives predicates of the form $\mathcal{P}(x, z; A)$, $x, z \in \mathrm{V}$, $A \subseteq \mathrm{V}_\forall$.
- Intuitive interpretation: $\left(\bigwedge_{v \in A} x = v\right) \Rightarrow x \geq z$.

Lemma (Rydval, S., Wrona '24)

- *If $\mathcal{P}(x, z; A)$ is derived and $z \notin A$, then the algorithm expands ϕ by*

$$\Big(\big(\bigwedge_{v \in \uparrow_A \setminus (\{x,z\} \cup x\text{-}z\text{-}cut)} x = v\big) \Rightarrow x \geq z\Big).$$

- *In particular, it expands ϕ by $(x \geq z)$ for every $\mathcal{P}(x, z; \emptyset)$.*
- *If the proof system does not derive $\mathcal{P}(x, z; \emptyset)$ or $\mathcal{P}(z, x; \emptyset)$ for $x \prec z$, $z \in \mathrm{V}_\forall$, then Φ is true.*

$\hookrightarrow$ conditional constraints are necessary for this to be true

The proof system $\mathcal{P}$

"Trial version" of $\mathcal{P}$

Initialize	$\mathcal{P}(x,x;\emptyset)$:− $x \in V$
Simplify	$\mathcal{P}(x,z;A \setminus x\text{-}z\text{-cut})$:− $\mathcal{P}(x,z;A)$
Transitivity	$\mathcal{P}(x,z;A)$:− $\mathcal{P}(x,y;A) \wedge \mathcal{P}(y,z;\emptyset)$
Constraint	$\mathcal{P}(x,z;y)$:− $(x = y \Rightarrow x \geq z) \wedge y \in \mathrm{V}_{\forall}$

The proof system $\mathcal{P}$

"Trial version" of $\mathcal{P}$

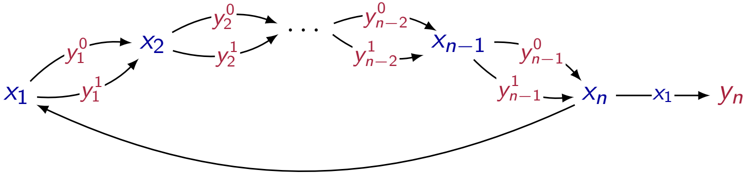
Initialize	$\mathcal{P}(x,x;\emptyset)$:− $x \in V$
Simplify	$\mathcal{P}(x,z;A \setminus x\text{-}z\text{-cut})$:− $\mathcal{P}(x,z;A)$
Transitivity	$\mathcal{P}(x,z;A)$:− $\mathcal{P}(x,y;A) \wedge \mathcal{P}(y,z;\emptyset)$
Constraint	$\mathcal{P}(x,z;y)$:− $(x = y \Rightarrow x \geq z) \wedge y \in \mathrm{V}_\forall$

Example (transitivity): $\mathcal{P}(x,z;A)$:− $\mathcal{P}(x,y;A) \wedge \mathcal{P}(y,z;\emptyset)$

$$\left(\left(\bigwedge_{v \in A} x = v \right) \Rightarrow x \geq y \right) \wedge (y \geq z)$$
$$\rightsquigarrow \left(\bigwedge_{v \in A} x = v \right) \Rightarrow x \geq z$$

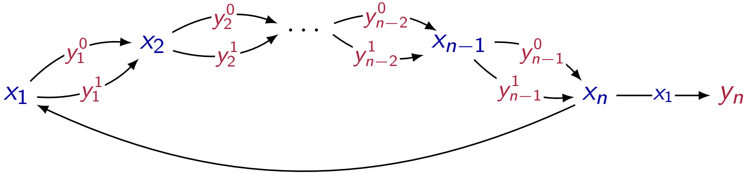
$\mathcal{P}$ does not give a PTIME-algorithm

Example: $\Phi = \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n$



$\mathcal{P}$ does not give a PTIME-algorithm

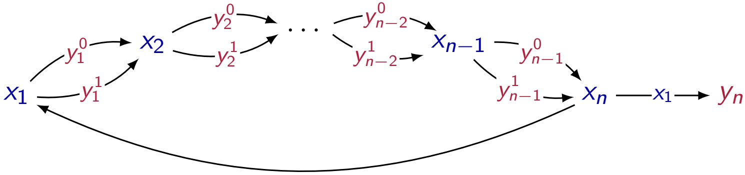
Example: $\Phi = \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n$



- $\mathcal{P}$ follows shortest derivation sequences

$\mathcal{P}$ does not give a PTIME-algorithm

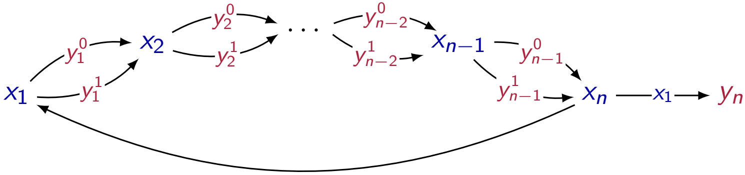
Example: $\Phi = \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n$



- $\mathcal{P}$ follows shortest derivation sequences
- $\mathcal{P}$ derives $\mathcal{P}(x_1, x_n; \{y_1^{i_1}, \dots, y_{n-1}^{i_{n-1}}\})$ for all $i_1, \dots, i_{n-1} \in \{0, 1\}$

$\mathcal{P}$ does not give a PTIME-algorithm

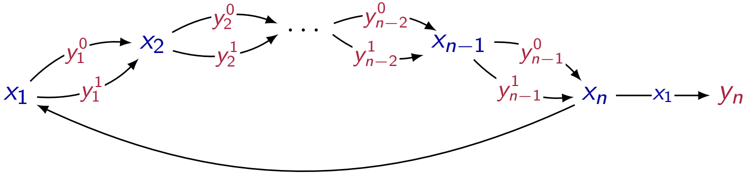
Example: $\Phi = \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n$



- $\mathcal{P}$ follows shortest derivation sequences
- $\mathcal{P}$ derives $\mathcal{P}(x_1, x_n; \{y_1^{i_1}, \ldots, y_{n-1}^{i_{n-1}}\})$ for all $i_1, \ldots, i_{n-1} \in \{0, 1\}$
- the algorithm derives $(x_{n-1} \geq x_1), \ldots, (x_2 \geq x_1)$, $(x_1 \geq y_n)$

$\mathcal{P}$ does not give a PTIME-algorithm

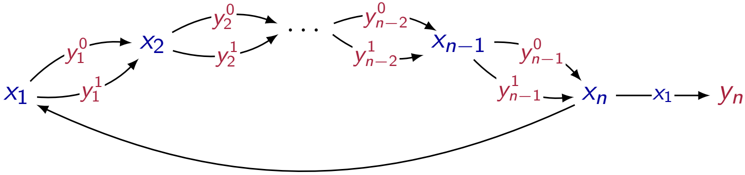
Example: $\Phi = \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n$



- $\mathcal{P}$ follows shortest derivation sequences
- $\mathcal{P}$ derives $\mathcal{P}(x_1, x_n; \{y_1^{i_1}, \ldots, y_{n-1}^{i_{n-1}}\})$ for all $i_1, \ldots, i_{n-1} \in \{0, 1\}$
- the algorithm derives $(x_{n-1} \geq x_1), \ldots, (x_2 \geq x_1)$, $(x_1 \geq y_n)$
- the algorithm rejects because of $(x_1 \geq y_n)$

$\mathcal{P}$ does not give a PTIME-algorithm

Example: $\Phi = \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n$



- $\mathcal{P}$ follows shortest derivation sequences
- $\mathcal{P}$ derives $\mathcal{P}(x_1, x_n; \{y_1^{i_1}, \ldots, y_{n-1}^{i_{n-1}}\})$ for all $i_1, \ldots, i_{n-1} \in \{0, 1\}$
- the algorithm derives $(x_{n-1} \geq x_1), \ldots, (x_2 \geq x_1)$, $(x_1 \geq y_n)$
- the algorithm rejects because of $(x_1 \geq y_n)$

$\mathcal{P}$ may derive exponentially many predicates
$\Rightarrow$ does not give a PTIME-algorithm

Tractability consequences

Theorem (Rydval, S., Wrona '24)

QCSP$(\mathbb{Q}; \mathrm{M}^{+})$ *is in* PTIME.

Tractability consequences

Theorem (Rydval, S., Wrona '24)

QCSP($\mathbb{Q}; \mathrm{M}^+$) *is in* PTIME.

Corollary

QCSP($\mathfrak{B}$) *is in* PTIME *if* $\mathfrak{B}$ *is a structure whose relations are definable by a conjunction of clauses of the form*

$$(x \neq y_1 \vee \cdots \vee x \neq y_k \vee x \geq z)$$

for $k \geq 0$ *and where the last disjunct* $(x \geq z)$ *may be omitted.*

Tractability consequences

Theorem (Rydval, S., Wrona '24)

$\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ *is in* PTIME.

Corollary

$\mathrm{QCSP}(\mathfrak{B})$ *is in* PTIME *if* $\mathfrak{B}$ *is a structure whose relations are definable by a conjunction of clauses of the form*

$$(x \neq y_1 \vee \cdots \vee x \neq y_k \vee x \geq z)$$

for $k \geq 0$ *and where the last disjunct* $(x \geq z)$ *may be omitted.*

Equivalently: structures $\mathfrak{B}$ whose relations lie both in the OH fragment and the $\pi\pi$ fragment (pp fragment from [Bodirsky, Kára '09]).

Complexity dichotomy for Ord-Horn constraints

Lemma (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment and pp-defines M^{+}. Then QCSP($\mathfrak{B}$) *is* coNP-*hard.*

Complexity dichotomy for Ord-Horn constraints

Lemma (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment and pp-defines M^+. Then QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Theorem (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME *if $\mathfrak{B}$ is guarded OH, contained in the $\pi\pi$ fragment, or in the dual $\pi\pi$ fragment. Otherwise,* QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Complexity dichotomy for Ord-Horn constraints

Lemma (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment and pp-defines M^+. Then QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Theorem (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME *if $\mathfrak{B}$ is guarded OH, contained in the $\pi\pi$ fragment, or in the dual $\pi\pi$ fragment. Otherwise,* QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Proof idea (lemma):

- $\mathfrak{B}$ pp-defines R of arity ≤ 4 outside of the $\pi\pi$ fragment.

Complexity dichotomy for Ord-Horn constraints

Lemma (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment and pp-defines M^+. Then QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Theorem (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME *if $\mathfrak{B}$ is guarded OH, contained in the $\pi\pi$ fragment, or in the dual $\pi\pi$ fragment. Otherwise,* QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Proof idea (lemma):

- $\mathfrak{B}$ pp-defines R of arity ≤ 4 outside of the $\pi\pi$ fragment.
- R qpp-defines D ($\Rightarrow$ PSPACE-hardness) or a certain relation $\check{\mathrm{Z}}$.

Complexity dichotomy for Ord-Horn constraints

Lemma (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment and pp-defines M^+. Then QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Theorem (Rydval, S., Wrona '24)

Let $\mathfrak{B}$ be an OH structure. Then QCSP($\mathfrak{B}$) *is in* PTIME *if $\mathfrak{B}$ is guarded OH, contained in the $\pi\pi$ fragment, or in the dual $\pi\pi$ fragment. Otherwise,* QCSP($\mathfrak{B}$) *is* coNP*-hard.*

Proof idea (lemma):

- $\mathfrak{B}$ pp-defines R of arity ≤ 4 outside of the $\pi\pi$ fragment.
- R qpp-defines D ($\Rightarrow$ PSPACE-hardness) or a certain relation $\check{\mathrm{Z}}$.
- QCSP($\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}}$) is coNP-hard.

coNP-hardness of QCSP($\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}}$)

$$\check{\mathrm{Z}} \coloneqq \{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid (x_1 \neq y_1 \vee x_2 \neq y_2) \wedge (y_1 < y_2)\},$$

Lemma

QCSP($\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}}$) *is* coNP-*hard.*

coNP-hardness of QCSP$(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$

$$\check{\mathrm{Z}} \coloneqq \{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid (x_1 \neq y_1 \vee x_2 \neq y_2) \wedge (y_1 < y_2)\},$$

Lemma

QCSP$(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$ *is* coNP*-hard.*

Proof idea:

- build a similar gadget as for $(\mathbb{Q}; \mathrm{D})$ using constraints of the form $\mathrm{M}^+(x, y, z) \wedge \mathrm{M}^+(z, z, x)$ instead of $\mathrm{D}(x, y, z)$, that is,

$$(x = y \Rightarrow x \geq z) \wedge (z \geq x) \text{ instead of } x = y \Rightarrow x = z$$

coNP-hardness of QCSP$(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$

$$\check{\mathrm{Z}} := \{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid (x_1 \neq y_1 \vee x_2 \neq y_2) \wedge (y_1 < y_2)\},$$

Lemma

QCSP$(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$ *is coNP-hard.*

Proof idea:

- build a similar gadget as for $(\mathbb{Q}; \mathrm{D})$ using constraints of the form $\mathrm{M}^+(x, y, z) \wedge \mathrm{M}^+(z, z, x)$ instead of $\mathrm{D}(x, y, z)$, that is,

$$(x = y \Rightarrow x \geq z) \wedge (z \geq x) \text{ instead of } x = y \Rightarrow x = z$$

- use a $\check{\mathrm{Z}}$-constraint as a control mechanism for the choices of the EP

coNP-hardness of QCSP$(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$

$$\check{\mathrm{Z}} := \{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid (x_1 \neq y_1 \vee x_2 \neq y_2) \wedge (y_1 < y_2)\},$$

Lemma

QCSP$(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$ *is coNP-hard.*

Proof idea:

- build a similar gadget as for $(\mathbb{Q}; \mathrm{D})$ using constraints of the form $\mathrm{M}^+(x, y, z) \wedge \mathrm{M}^+(z, z, x)$ instead of $\mathrm{D}(x, y, z)$, that is,

$$(x = y \Rightarrow x \geq z) \wedge (z \geq x) \text{ instead of } x = y \Rightarrow x = z$$

- use a $\check{\mathrm{Z}}$-constraint as a control mechanism for the choices of the EP

Remark: the constraints $\mathrm{M}^+(z, z, x)$ give unconditional constraints $z \geq x$

coNP-hardness of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$

$$\check{\mathrm{Z}} := \{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid (x_1 \neq y_1 \vee x_2 \neq y_2) \wedge (y_1 < y_2)\},$$

Lemma

$\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+, \check{\mathrm{Z}})$ *is coNP-hard.*

Proof idea:

- build a similar gadget as for $(\mathbb{Q}; \mathrm{D})$ using constraints of the form $\mathrm{M}^+(x, y, z) \wedge \mathrm{M}^+(z, z, x)$ instead of $\mathrm{D}(x, y, z)$, that is,

$$(x = y \Rightarrow x \geq z) \wedge (z \geq x) \text{ instead of } x = y \Rightarrow x = z$$

- use a $\check{\mathrm{Z}}$-constraint as a control mechanism for the choices of the EP

Remark: the constraints $\mathrm{M}^+(z, z, x)$ give unconditional constraints $z \geq x$
$\rightsquigarrow$ we can prove only coNP-hardness

Open questions

Question 1: Do Ord-Horn QCSPs exhibit a dichotomy between coNP- and PSPACE-hardness?

Open questions

Question 1: Do Ord-Horn QCSPs exhibit a dichotomy between coNP- and PSPACE-hardness?

Question 2: Is $\mathrm{QCSP}(\mathbb{Q}; x \neq y \vee x \geq z_1 \vee x \geq z_2)$ in NP?

Answer 'yes' to Question 2 $\Rightarrow$ membership in NP for $\mathrm{QCSP}(\mathfrak{B})$ for all $\mathfrak{B}$ contained in the $\pi\pi$ fragment

Open questions

Question 1: Do Ord-Horn QCSPs exhibit a dichotomy between coNP- and PSPACE-hardness?

Question 2: Is $\mathrm{QCSP}(\mathbb{Q}; x \neq y \vee x \geq z_1 \vee x \geq z_2)$ in NP?

Answer 'yes' to Question 2 $\Rightarrow$ membership in NP for $\mathrm{QCSP}(\mathfrak{B})$ for all $\mathfrak{B}$ contained in the $\pi\pi$ fragment

Question 3: Is $\mathrm{QCSP}(\mathbb{Q}; x \neq y \vee x \geq z \vee x > w)$ in PTIME?

Answer 'yes' to Question 3 $\Rightarrow$ tractability for $\mathrm{QCSP}(\mathfrak{B})$ for all $\mathfrak{B}$ contained in the *mi* fragment [Bodirsky, Kára '09]

Open questions

Question 1: Do Ord-Horn QCSPs exhibit a dichotomy between coNP- and PSPACE-hardness?

Question 2: Is QCSP$(\mathbb{Q}; x \neq y \vee x \geq z_1 \vee x \geq z_2)$ in NP?

Answer 'yes' to Question 2 $\Rightarrow$ membership in NP for QCSP$(\mathfrak{B})$ for all $\mathfrak{B}$ contained in the $\pi\pi$ fragment

Question 3: Is QCSP$(\mathbb{Q}; x \neq y \vee x \geq z \vee x > w)$ in PTIME?

Answer 'yes' to Question 3 $\Rightarrow$ tractability for QCSP$(\mathfrak{B})$ for all $\mathfrak{B}$ contained in the *mi* fragment [Bodirsky, Kára '09]

$\hookrightarrow$ a maximal tractable fragment for CSPs

$\hookrightarrow$ the last such fragment where it is unknown whether it is a maximal tractable fragment for QCSPs

Thank you for your attention

Funding statement: Funded by the European Union (ERC, POCOCOP, 101071674).
Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.