Identifying Tractable Quantified Temporal Constraints within Ord-Horn

Žaneta Semanišinová
joint work with Jakub Rydval and Michał Wrona

Institute of Algebra
TU Dresden

AAA105
1 Jun 2024

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 signature τ

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

(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 signature τ

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

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 signature τ
Question: Does $\mathfrak{B} \models \Phi$?

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 signature τ
Question: Does $\mathfrak{B} \models \Phi$?

Quantified constraint satisfaction problem

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

Quantified constraint satisfaction problem

- QCSP can be seen as 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 can be seen as 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 can be seen as 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 can be seen as 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 can be seen as 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: $\mathrm{D} = \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow y = z\}$.

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

Quantified constraint satisfaction problem

- QCSP can be seen as 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.

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 '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)
- partial classification of QCSPs (Bodirsky, Chen '10) – only complexity of $\mathrm{QCSP}(\mathbb{Q};\mathrm{D})$ missing
- full classification of QCSPs (Zhuk, Martin, Wrona '22)

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 '22)

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

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

$\hookrightarrow$ contains QCSPs that are in PTIME, coNP- and PSPACE-complete

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

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

$\hookrightarrow$ contains QCSPs that are in PTIME, coNP- and PSPACE-complete

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

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

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

$\hookrightarrow$ contains QCSPs that are in PTIME, coNP- and PSPACE-complete

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

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^-*.*

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 $\mathrm{QCSP}(\mathfrak{B})$ *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 $\mathrm{QCSP}(\mathfrak{B})$ *reduces to* $\mathrm{QCSP}(\mathfrak{A})$ *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

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

Theorem (Rydval, S., Wrona '24)

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

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

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

Theorem (Rydval, S., Wrona '24)

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

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

$\prec :=$ the linear order on V from the order in the quantifier prefix of Φ

We write $A \prec B$ meaning $x \prec y$, $\forall x \in A, y \in B$.

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

Theorem (Rydval, S., Wrona '24)

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

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

$\prec$:= the 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

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

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 $\uparrow_u \coloneqq \{y \in \mathrm{V}_\forall \mid u \preceq y\}$

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

- reject if constraint $(x \geq z)$ 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 $\uparrow_u \coloneqq \{y \in \mathrm{V}_\forall \mid u \preceq y\}$

- reject if constraint $(x \geq z)$ or $(z \geq x)$ is derived where $x \prec z$, $z \in \mathrm{V}_\forall$
- accept if no new constraints can be derived

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

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

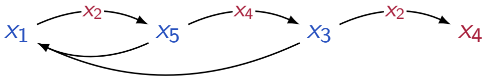
- x-z-cut comprises variables that the UP can play equal to x to trigger the constraint $x \geq z$
- z is removed so that the constraint does not become trivial

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

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

Example of the run of the algorithm

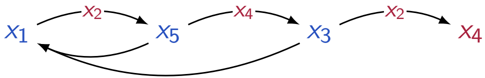
$$\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \wedge (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1) \big).$$



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

Example of the run of the algorithm

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

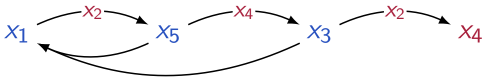


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

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

Example of the run of the algorithm

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

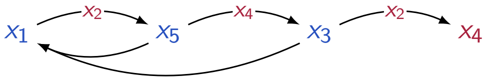


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ 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

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

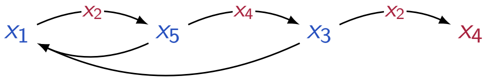


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ 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 $= \{x_4\}$ $\rightsquigarrow$ $\uparrow_{x_2} \setminus (\{x_1, x_3\} \cup x_1\text{-}x_3\text{-cut}) = \{x_2\}$

Example of the run of the algorithm

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

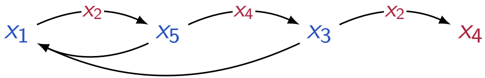


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ 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

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

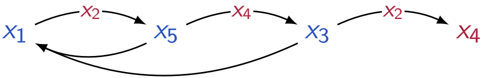


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ 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

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

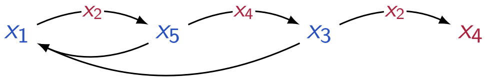


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ 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

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

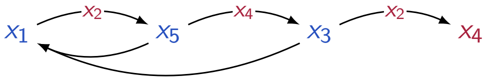


Claim: The algorithm derives $(x_1 \geq x_4)$, and thereby rejects on Φ.

- $\phi \wedge (x_1 = x_2) \wedge (x_1 = x_4)$ 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 $= \{x_4\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus(\{x_1, x_3\} \cup x_1\text{-}x_3\text{-cut}) = \{x_2\}$
- 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 $= \{x_2\} \quad \rightsquigarrow \quad \uparrow_{x_2} \setminus(\{x_1, x_4\} \cup x_1\text{-}x_4\text{-cut}) = \emptyset$

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 $(x_1 \geq x_4)$.

Tractability consequences

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

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.*

Equivalently: structures $\mathfrak{B}$ whose relations lie both in the OH fragment and the $\pi\pi$-fragment (preserved by the operation $\pi\pi$ – 'projection-projection' operation 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.*

Proof idea:

- $\mathfrak{B}$ 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.*

Proof idea:

- $\mathfrak{B}$ 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.*

Proof idea:

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

Proof idea:

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

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

Answer 'yes' to Question 2 $\Rightarrow$ tractability for QCSP($\mathfrak{B}$) for all $\mathfrak{B}$ contained in the *mi* fragment (preserved by the operation *mi* [Bodirsky, Kára '09])

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.