Quantified Constraint Satisfaction Problem: towards the classification of complexity

Dmitriy Zhuk
joint with Barnaby Martin

Lomonosov Moscow State University
Higher School of Economics

19th International Conference on Relational and Algebraic Methods in Computer Science RAMICS 2021

European Research Council
Established by the European Commission

CoCoSym: Symmetry in Computational Complexity

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 771005)

Quantified Equality Constraints

$(\mathbb{N}; =)$

Quantified Equality Constraints

$(\mathsf{N}; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \,\hat{}\, x_3 = x_4)$,

Quantified Equality Constraints

$(N; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

Quantified Equality Constraints

$(\mathsf{N}; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$,

Quantified Equality Constraints

$(N; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \hat{} \; x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \hat{} \; x_2 = x_3 \hat{} \; x_3 = x_4)$, false

Quantified Equality Constraints

(N; =)

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP(N; $x = y$)

Given a sentence $8x_1 9x_2 ::: 8x_n \;\; {}_1 9x_n \, (x_{i_1} = x_{j_1} \wedge \qquad \wedge x_{i_s} = x_{j_s})$.
Decide whether it holds.

Quantified Equality Constraints

(N; =)

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP(N; $x = y$)

Given a sentence $8x_1 9x_2 : : : 8x_n \;{}_1 9x_n \, (x_{i_1} = x_{j_1} \wedge \quad \wedge x_{i_s} = x_{j_s})$.
Decide whether it holds.

- QCSP(N; $x = y$) is solvable in polynomial time.

Quantified Equality Constraints

$(\mathbb{N}; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP$(\mathbb{N}; R)$

Given a sentence $8x_1 9x_2 ::: 8x_{n\ \ 1} 9x_n (R(:::) \wedge \quad \wedge R(:::))$.
Decide whether it holds.

- QCSP$(\mathbb{N}; x = y)$ is solvable in polynomial time.

Quantified Equality Constraints

$(\mathbb{N}; =)$

$\forall x_1 \exists x_2 \forall x_3 \exists x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$\forall x_1 \exists x_2 \forall x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP$(\mathbb{N}; R)$

Given a sentence $\forall x_1 \exists x_2 \dots \forall x_{n-1} \exists x_n \, (R(\dots) \wedge \cdots \wedge R(\dots))$.
Decide whether it holds.

- QCSP$(\mathbb{N}; x = y)$ is solvable in polynomial time.
- QCSP$(\mathbb{N}; x = y \vee z = t)$ is NP-complete [Bodirsky, Chen 2007].

Quantified Equality Constraints

$(N; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP(N; R)

Given a sentence $8x_1 9x_2 ::: 8x_{n\ 1} 9x_n (R(:::) \wedge \quad \wedge R(:::))$.
Decide whether it holds.

- QCSP(N; $x = y$) is solvable in polynomial time.
- QCSP(N; $x = y \vee z = t$) is NP-complete [Bodirsky, Chen 2007].
- QCSP(N; $x = y\ !\ z = t$) is PSpace-complete [Bodirsky, Chen 2007].

Quantified Equality Constraints

$(\mathbb{N};=)$

$\forall x_1 \exists x_2 \forall x_3 \exists x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$\forall x_1 \exists x_2 \forall x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP$(\mathbb{N}; R)$

Given a sentence $\forall x_1 \exists x_2 \ldots \forall x_{n-1} \exists x_n \, (R(\ldots) \wedge \cdots \wedge R(\ldots))$.
Decide whether it holds.

- QCSP$(\mathbb{N}; x = y)$ is solvable in polynomial time.
- QCSP$(\mathbb{N}; x = y \vee z = t)$ is NP-complete [Bodirsky, Chen 2007].
- QCSP$(\mathbb{N}; x = y \rightarrow z = t)$ is PSpace-complete [Bodirsky, Chen 2007].

What is the complexity of QCSP$(\mathbb{N}; x = y \rightarrow y = z)$?

Quantified Equality Constraints

$(\mathrm{N}; =)$
$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \text{ ^ } x_3 = x_4)$, true
$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \text{ ^ } x_2 = x_3 \text{ ^ } x_3 = x_4)$, false

QCSP$(\mathrm{N}; R)$

Given a sentence $8x_1 9x_2 ::: 8x_{n-1} 9x_n (R(:::) \text{ ^ } \quad \text{ ^ } R(:::))$.
Decide whether it holds.

- QCSP$(\mathrm{N}; x = y)$ is solvable in polynomial time.
- QCSP$(\mathrm{N}; x = y _ z = t)$ is NP-complete [Bodirsky, Chen 2007].
- QCSP$(\mathrm{N}; x = y \; ! \; z = t)$ is PSpace-complete [Bodirsky, Chen 2007].

What is the complexity of QCSP$(\mathrm{N}; x = y \; ! \; y = z)$?

A concrete question

Quantified Equality Constraints

$(\mathrm{N}; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \hat{} \; x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \hat{} \; x_2 = x_3 \hat{} \; x_3 = x_4)$, false

QCSP$(\mathrm{N}; R)$

Given a sentence $8x_1 9x_2 ::: 8x_{n\ 1} 9x_n\ (R(:::) \hat{} \quad \hat{} \; R(:::))$.
Decide whether it holds.

- QCSP$(\mathrm{N}; x = y)$ is solvable in polynomial time.
- QCSP$(\mathrm{N}; x = y _ z = t)$ is NP-complete [Bodirsky, Chen 2007].
- QCSP$(\mathrm{N}; x = y \; ! \; z = t)$ is PSpace-complete [Bodirsky, Chen 2007].

What is the complexity of QCSP$(\mathrm{N}; x = y \; ! \; y = z)$?

A concrete question

Easy to Formulate

Quantified Equality Constraints

$(\mathrm{N}; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP$(\mathrm{N}; R)$

Given a sentence $8x_1 9x_2 ::: 8x_{n \ 1} 9x_n \ (R(:::) \wedge \quad \wedge R(:::))$.
Decide whether it holds.

- QCSP$(\mathrm{N}; x = y)$ is solvable in polynomial time.
- QCSP$(\mathrm{N}; x = y \vee z = t)$ is NP-complete [Bodirsky, Chen 2007].
- QCSP$(\mathrm{N}; x = y \ ! \ z = t)$ is PSpace-complete [Bodirsky, Chen 2007].

What is the complexity of QCSP$(\mathrm{N}; x = y \ ! \ y = z)$?

A concrete question

Open since 2007

Easy to Formulate

Quantified Equality Constraints

$(\mathbb{N}; =)$

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_3 = x_4)$, true

$8x_1 9x_2 8x_3 9x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, false

QCSP$(\mathbb{N}; R)$

Given a sentence $8x_1 9x_2 : : : 8x_{n \;\; 1} 9x_n \, (R(: : :) \wedge \quad \wedge R(: : :))$.
Decide whether it holds.

- QCSP$(\mathbb{N}; x = y)$ is solvable in polynomial time.
- QCSP$(\mathbb{N}; x = y _ z = t)$ is NP-complete [Bodirsky, Chen 2007].
- QCSP$(\mathbb{N}; x = y \; ! \; z = t)$ is PSpace-complete [Bodirsky, Chen 2007].

What is the complexity of QCSP$(\mathbb{N}; x = y \; ! \; y = z)$?

A concrete question

Open since 2007

Accessible to anyone

Easy to Formulate

Quantified Equality Constraints

What is the complexity of QCSP(N; $x = y$! $y = z$)?

Quantified Equality Constraints

What is the complexity of QCSP$(\mathrm{N}; x = y \; ! \; y = z)$?

- QCSP$(\mathrm{N}; x = y \; ! \; y = z)$ is coNP-hard [Bodirsky, Chen, 2010].

Quantified Equality Constraints

What is the complexity of $\mathrm{QCSP}(\mathbb{N}; x = y \; ! \; y = z)$?

- $\mathrm{QCSP}(\mathbb{N}; x = y \; ! \; y = z)$ is coNP-hard [Bodirsky, Chen, 2010].

Lemma [Zhuk, Martin, 2021]

$\mathrm{QCSP}(\mathbb{N}; x = y \; ! \; y = z)$ is PSpace-hard.

Quantified Equality Constraints

What is the complexity of QCSP(N; $x = y \ ! \ y = z$)?

- QCSP(N; $x = y \ ! \ y = z$) is coNP-hard [Bodirsky, Chen, 2010].

Lemma [Zhuk, Martin, 2021]

QCSP(N; $x = y \ ! \ y = z$) is PSpace-hard.

Theorem [Zhuk, Martin, Bodirsky, Chen, 2021]

Suppose relations $R_1; : : : ; R_s$ are definable by some Boolean combination of atoms of the form $(x = y)$. Then QCSP(N; $R_1; : : : ; R_s$) is either tractable, NP-complete, or PSpace-complete.

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t (R_1(:::) \wedge \quad \wedge R_s(:::))$, where $R_1; :::; R_s$ 2 Γ.
Decide whether it holds.

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t (R_1(:::) \wedge \quad \wedge R_s(:::))$, where $R_1; :::; R_s \; 2 \; \Gamma$.
Decide whether it holds.

Examples:
$A = f0; 1; 2g; \Gamma = fx \neq yg$.

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $\exists y_1 \forall x_1 \ldots \exists y_t \forall x_t (R_1(\ldots) \wedge \cdots \wedge R_s(\ldots))$, where $R_1, \ldots, R_s \in \Gamma$.
Decide whether it holds.

Examples:
$A = \{0, 1, 2\}, \Gamma = \{x \neq y\}$. QCSP instances:

$\forall x \exists y_1 \exists y_2 (x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2)$,

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $\exists y_1 \forall x_1 \dots \exists y_t \forall x_t (R_1(\dots) \wedge \cdots \wedge R_s(\dots))$, where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

Examples:
$A = \{0, 1, 2\}, \Gamma = \{x \neq y\}$. QCSP instances:

$\forall x \exists y_1 \exists y_2 (x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2)$, true

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t (R_1(:::) \wedge \quad \wedge R_s(:::))$, where $R_1; :::; R_s$ 2 Γ.
Decide whether it holds.

Examples:
$A = f0; 1; 2g; \Gamma = fx \; 6= yg$. QCSP instances:

$8x 9y_1 9y_2 (x \; 6= y_1 \wedge x \; 6= y_2 \wedge y_1 \; 6= y_2)$, true

$8x_1 8x_2 8x_3 9y (x_1 \; 6= y \wedge x_2 \; 6= y \wedge x_3 \; 6= y)$,

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $\exists y_1 \forall x_1 \dots \exists y_t \forall x_t (R_1(\dots) \wedge \cdots \wedge R_s(\dots))$, where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

Examples:
$A = \{0, 1, 2\}, \Gamma = \{x \neq y\}$. QCSP instances:

$\forall x \exists y_1 \exists y_2 (x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2)$, true

$\forall x_1 \forall x_2 \forall x_3 \exists y (x_1 \neq y \wedge x_2 \neq y \wedge x_3 \neq y)$, false

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $\exists y_1 \forall x_1 \dots \exists y_t \forall x_t (R_1(\dots) \wedge \quad \wedge R_s(\dots))$, where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

Examples:

$A = \{0, 1, 2\}, \Gamma = \{x \neq y\}$. QCSP instances:

$\forall x \exists y_1 \exists y_2 (x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2)$, true

$\forall x_1 \forall x_2 \forall x_3 \exists y (x_1 \neq y \wedge x_2 \neq y \wedge x_3 \neq y)$, false

$\forall x_1 \exists y_1 \forall x_2 \exists y_2 (x_1 \neq y_1 \wedge y_1 \neq y_2 \wedge y_2 \neq x_2)$,

Quantified Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $\exists y_1 \forall x_1 \dots \exists y_t \forall x_t (R_1(\dots) \wedge \quad \wedge R_s(\dots))$, where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

Examples:
$A = \{0, 1, 2\}, \Gamma = \{x \neq y\}$. QCSP instances:

$\forall x \exists y_1 \exists y_2 (x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2)$, true

$\forall x_1 \forall x_2 \forall x_3 \exists y (x_1 \neq y \wedge x_2 \neq y \wedge x_3 \neq y)$, false

$\forall x_1 \exists y_1 \forall x_2 \exists y_2 (x_1 \neq y_1 \wedge y_1 \neq y_2 \wedge y_2 \neq x_2)$, true

Quantified Constraint Satisfaction Problem

A is a finite set,

Γ is a set of relations on A (a constraint language)

QCSP(Γ):

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t (R_1(:::) \wedge \quad \wedge R_s(:::))$, where $R_1; :::; R_s \; 2 \; \Gamma$.

Decide whether it holds.

Examples:

$A = f0; 1; 2g; \Gamma = fx \; 6\!\!= yg$. QCSP instances:

$8x 9y_1 9y_2 (x \; 6\!\!= y_1 \wedge x \; 6\!\!= y_2 \wedge y_1 \; 6\!\!= y_2)$, true

$8x_1 8x_2 8x_3 9y (x_1 \; 6\!\!= y \wedge x_2 \; 6\!\!= y \wedge x_3 \; 6\!\!= y)$, false

$8x_1 9y_1 8x_2 9y_2 (x_1 \; 6\!\!= y_1 \wedge y_1 \; 6\!\!= y_2 \wedge y_2 \; 6\!\!= x_2)$, true

Main Question

What is the complexity of QCSP(Γ) for different Γ?

Σ	dual-Σ	Classification	Complexity Classes

Quantified Constraint Satisfaction Problem:

Given a sentence $\exists y_1 \forall x_1 \ldots \exists y_t \forall x_t (R_1(\ldots) \wedge \cdots \wedge R_s(\ldots))$, where $R_1, \ldots, R_s \in \Gamma$.

Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
$f9; 8; \wedge g$			

Quantified Constraint Satisfaction Problem:

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t (R_1(:::) \wedge \quad \wedge R_s(:::))$,
where $R_1; :::; R_s$ 2 Γ.

Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
$f9;8;$^g	$f9;8;\_g$		

Quantified Constraint Satisfaction Problem:

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t(R_1(:::)$ ^ ^ $R_s(:::))$,
where $R_1; :::; R_s$ 2 Γ.

Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
$f9;8;\wedge g$	$f9;8;\vee g$	??????????	??????????

Quantified Constraint Satisfaction Problem:

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t (R_1(:::) \wedge \quad \wedge R_s(:::))$,
where $R_1; :::; R_s$ 2 Γ.

Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
f9; 8; ^g	*f9; 8; _g*	??????????	??????????
f9; _g	*f8; ^g*	Trivial	L

Given a sentence $9y_1 ::: 9y_t(R_1(:::) \_ \quad \_ R_s(:::))$,
where $R_1; ::: ; R_s$ 2 Γ.
Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
f9; 8; ^g	f9; 8; _g	??????????	??????????
f9; _g	f8; ^g	Trivial	L
f9; ^g	f8; _g	CSP Dichotomy	P, NP-complete

Constraint Satisfaction Problem:
Given a sentence $9y_1 ::: 9y_t(R_1(:::) \hat{} \quad \hat{} R_s(:::))$,
where $R_1; ::: ; R_s$ 2 Γ.
Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
$f9; 8; \wedge g$	$f9; 8; \_g$	??????????	??????????
$f9; \_g$	$f8; \wedge g$	Trivial	L
$f9; \wedge g$	$f8; \_g$	CSP Dichotomy	P, NP-complete
$f9; \wedge; \_g$	$f8; \wedge; \_g$	Trivial iff the core has one element	L NP-complete

Given a sentence $9y_1 ::: 9y_t((R_1(:::) \_ R_2(:::)) \wedge R_3(:::))$, where $R_1; :::; R_3$ 2 Γ.

Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
f9; 8; ^g	*f9; 8; _g*	??????????	??????????
f9; _g	*f8; ^g*	Trivial	L
f9; ^g	*f8; _g*	CSP Dichotomy	P, NP-complete
f9; ^; _g	*f8; ^; _g*	Trivial iff the core has one element	L NP-complete
f9; 8; ^; _g		Positive equality free tetrachotomy	P, NP-complete co-NP-complete PSPACE-complete

Given a sentence $9y_1 8x_1 ::: 9y_t 8x_t((R_1(:::) \_ R_2(:::)) \,\hat{}\, R_3(:::))$, where $R_1; :::; R_3$ 2 Γ.
Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
f9; 8; ^g	*f9; 8; _g*	??????????	??????????
f9; _g	*f8; ^g*	Trivial	L
f9; ^g	*f8; _g*	CSP Dichotomy	P, NP-complete
f9; ^; _g	*f8; ^; _g*	Trivial iff the core has one element	L NP-complete
f9; 8; ^; _g		Positive equality free tetrachotomy	P, NP-complete co-NP-complete PSPACE-complete
f9; 8; ^; _; :g		Trivial iff Γ is trivial	L PSPACE-complete

Given a sentence
$9y_1 8x_1 ::: 9y_t 8x_t((:R_1(:::)\ \_\ R_2(:::))\ \hat{}\ :R_3(:::))$,
where $R_1; ::: ; R_3\ 2\ \Gamma$.
Decide whether it holds.

Σ	dual-Σ	Classification	Complexity Classes
f9; 8; ^g	f9; 8; _g	??????????	??????????
f9; _g	f8; ^g	Trivial	L
f9; ^g	f8; _g	CSP Dichotomy	P, NP-complete
f9; ^; _g	f8; ^; _g	Trivial iff the core has one element	L NP-complete
f9; 8; ^; _g		Positive equality free tetrachotomy	P, NP-complete co-NP-complete PSPACE-complete
f9; 8; ^; _; :g		Trivial iff Γ is trivial	L PSPACE-complete

Quantified Constraint Satisfaction Problem:
Given a sentence $9y_1 8x_1 : : : 9y_t 8x_t (R_1(: : :) \,\hat{}\, \quad \hat{}\, R_s(: : :))$,
where $R_1; : : : ; R_s\ 2\ \Gamma$.
Decide whether it holds.

Constraint Satisfaction Problem

A is a finite set,

Γ is a set of relations on A (a constraint language)

Constraint Satisfaction Problem

A is a finite set,

Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a formula $(R_1(\ldots) \wedge \cdots \wedge R_s(\ldots))$,
where $R_1, \ldots, R_s \in \Gamma$.
Decide whether the formula is satisfiable.

Constraint Satisfaction Problem

A is a finite set,

Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $9y_1 ::: 9y_t(R_1(:::) \wedge \quad \wedge R_s(:::))$,
where $R_1; :::; R_s\ 2\ \Gamma$.
Decide whether it holds.

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $\exists y_1 \ldots \exists y_t (R_1(\ldots) \wedge \cdots \wedge R_s(\ldots))$,
where $R_1, \ldots, R_s \in \Gamma$.
Decide whether it holds.

An operation f preserves a relation R,

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $9y_1 ::: 9y_t(R_1(:::) \wedge \quad \wedge R_s(:::))$,
where $R_1; :::; R_s$ 2 Γ.
Decide whether it holds.

An operation f preserves a relation R,
(equivalently, f is a polymorphism of R, shortly f 2 Pol(R))

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $\exists y_1 \dots \exists y_t (R_1(\dots) \wedge \cdots \wedge R_s(\dots))$,
where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

An operation f preserves a relation R,
(equivalently, f is a polymorphism of R, shortly $f \in \mathrm{Pol}(R)$)

if for all $\begin{pmatrix} a_1^1 \\ \vdots \\ a_1^s \end{pmatrix}, \dots, \begin{pmatrix} a_n^1 \\ \vdots \\ a_n^s \end{pmatrix} \in R;$

$$f \begin{pmatrix} a_1^1 & \dots & a_n^1 \\ \vdots & \ddots & \vdots \\ a_1^s & \dots & a_n^s \end{pmatrix} = \begin{pmatrix} f(a_1^1, \dots, a_n^1) \\ \vdots \\ f(a_1^s, \dots, a_n^s) \end{pmatrix} \in R$$

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $\exists y_1 \dots \exists y_t (R_1(\dots) \wedge \dots \wedge R_s(\dots))$,
where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

An operation f preserves a relation R,
(equivalently, f is a polymorphism of R, shortly $f \in \mathrm{Pol}(R)$)

if for all $\begin{pmatrix} a_1^1 \\ \vdots \\ a_1^s \end{pmatrix}, \dots, \begin{pmatrix} a_n^1 \\ \vdots \\ a_n^s \end{pmatrix} \in R$;

$$f \begin{pmatrix} a_1^1 & \dots & a_n^1 \\ \vdots & \ddots & \vdots \\ a_1^s & \dots & a_n^s \end{pmatrix} = \begin{pmatrix} f(a_1^1, \dots, a_n^1) \\ \vdots \\ f(a_1^s, \dots, a_n^s) \end{pmatrix} \in R$$

f preserves Γ (equivalently $f \in \mathrm{Pol}(\Gamma)$) if f preserves every $R \in \Gamma$.

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $9y_1 ::: 9y_t(R_1(:::) \wedge \quad \wedge R_s(:::))$,
where $R_1; :::; R_s \; 2 \; \Gamma$.
Decide whether it holds.

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $\exists y_1 \dots \exists y_t (R_1(\dots) \wedge \cdots \wedge R_s(\dots))$,
where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

Theorem [Bulatov, Zhuk, 2017]

- CSP(Γ) is solvable in polynomial time (tractable) if there exists a weak near-unanimity operation preserving Γ,
- CSP(Γ) is NP-complete otherwise.

Weak near-unanimity operation (WNU) is an operation satisfying

$$w(y, x, x, \dots, x) = w(x, y, x, \dots, x) = \cdots = w(x, x, \dots, x, y)$$

Constraint Satisfaction Problem

A is a finite set,
Γ is a set of relations on A (a constraint language)

CSP(Γ):

Given a sentence $\exists y_1 \dots \exists y_t (R_1(\dots) \wedge \quad \wedge R_s(\dots))$, where $R_1, \dots, R_s \in \Gamma$.
Decide whether it holds.

Theorem [Bulatov, Zhuk, 2017]

- CSP(Γ) is solvable in polynomial time (tractable) if there exists a weak near-unanimity operation preserving Γ,
- CSP(Γ) is NP-complete otherwise.

Weak near-unanimity operation (WNU) is an operation satisfying

$$w(y, x, x, \dots, x) = w(x, y, x, \dots, x) = \quad = w(x, x, \dots, x, y)$$

Examples: $x \vee y, x \wedge y, xy \vee xz \vee yz, x + y + z, 0, \min(x, y), \dots$

Few facts about QCSP

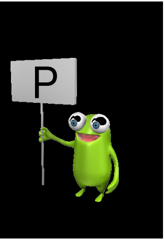
Few facts about QCSP

- If Γ contains all relations then QCSP(Γ) is PSPACE-complete.

PSPACE

Few facts about QCSP

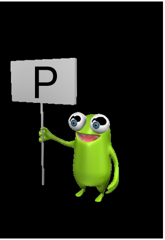
- If Γ contains all relations then QCSP(Γ) is PSPACE-complete.
- If Γ consists of linear equations in a finite field then QCSP(Γ) can be solved in polynomial time (tractable).



PSPACE

Few facts about QCSP

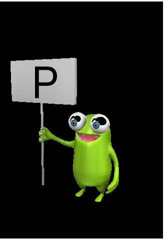
- If Γ contains all relations then QCSP(Γ) is PSPACE-complete.
- If Γ consists of linear equations in a finite field then QCSP(Γ) can be solved in polynomial time (tractable).
- For $A' = A \cup \{g\}$, Γ' an extension of Γ to A', QCSP(Γ') is equivalent to CSP(Γ).



PSPACE

Few facts about QCSP

- If Γ contains all relations then QCSP(Γ) is PSPACE-complete.
- If Γ consists of linear equations in a finite field then QCSP(Γ) can be solved in polynomial time (tractable).
- For $A' = A \cup \{\ \}$, Γ' an extension of Γ to A', QCSP(Γ') is equivalent to CSP(Γ). QCSP(Γ) can be NP-complete.



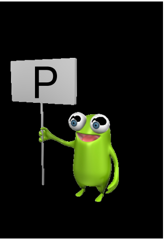
PSPACE

NP

Few facts about QCSP

- If Γ contains all relations then QCSP(Γ) is PSPACE-complete.
- If Γ consists of linear equations in a finite field then QCSP(Γ) can be solved in polynomial time (tractable).
- For $A' = A \cup \{\ \}$, Γ' an extension of Γ to A', QCSP(Γ') is equivalent to CSP(Γ). QCSP(Γ) can be NP-complete.

Are there any other complexity classes?



PSPACE

NP

QCSP classifications

QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)

QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)
- Graphs of permutations. Trichotomy P, NP-complete, Pspace-complete. (Börner et al. 2002.)

QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)
- Graphs of permutations. Trichotomy P, NP-complete, Pspace-complete. (Börner et al. 2002.)
- Various graphs Dichotomies and trichotomies P, NP-complete, Pspace-complete. (Madelaine, M. 2006, 2011, 2013)

QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)
- Graphs of permutations. Trichotomy P, NP-complete, Pspace-complete. (Börner et al. 2002.)
- Various graphs Dichotomies and trichotomies P, NP-complete, Pspace-complete. (Madelaine, M. 2006, 2011, 2013)
- Structures with 2-semilattice polymorphism. Dichotomy P, Pspace-complete. (Chen 2004 + Börner et al. 2009.)

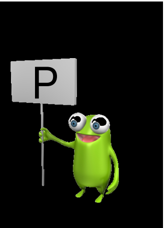
QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)
- Graphs of permutations. Trichotomy P, NP-complete, Pspace-complete. (Börner et al. 2002.)
- Various graphs Dichotomies and trichotomies P, NP-complete, Pspace-complete. (Madelaine, M. 2006, 2011, 2013)
- Structures with 2-semilattice polymorphism. Dichotomy P, Pspace-complete. (Chen 2004 + Börner et al. 2009.)
- Semicomplete digraphs. Trichotomy. P, NP-complete, Pspace-complete. (Dapic et al. 2014.)

QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)
- Graphs of permutations. Trichotomy P, NP-complete, Pspace-complete. (Börner et al. 2002.)
- Various graphs Dichotomies and trichotomies P, NP-complete, Pspace-complete. (Madelaine, M. 2006, 2011, 2013)
- Structures with 2-semilattice polymorphism. Dichotomy P, Pspace-complete. (Chen 2004 + Börner et al. 2009.)
- Semicomplete digraphs. Trichotomy. P, NP-complete, Pspace-complete. (Dapic et al. 2014.)

PSPACE



NP

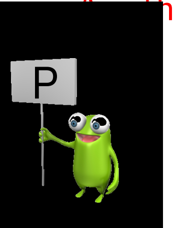
QCSP classifications

- Boolean structures. Dichotomy P, Pspace-complete. (Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.)
- Graphs of permutations. Trichotomy P, NP-complete, Pspace-complete. (Börner et al. 2002.)
- Various graphs Dichotomies and trichotomies P, NP-complete, Pspace-complete. (Madelaine, M. 2006, 2011, 2013)
- Structures with 2-semilattice polymorphism. Dichotomy P, Pspace-complete. (Chen 2004 + Börner et al. 2009.)
- Semicomplete digraphs. Trichotomy. P, NP-complete, Pspace-complete. (Dapic et al. 2014.)

A[illegible]here any other complexity classes?

PSPACE

NP



Surjective polymorphisms

Observation

Suppose each relation of Γ_1 is definable from Γ_2 using quantified conjunctive formulas

$$R(x_1, \dots, x_n) = \forall y_1 \exists y_2 \forall y_3 \exists y_4 \dots R_1(\dots) \wedge \cdots \wedge R_s(\dots).$$

Then QCSP(Γ_1) is polynomially reducible to QCSP(Γ_2).

Surjective polymorphisms

Observation

Suppose each relation of Γ_1 is definable from Γ_2 using quantified conjunctive formulas

$$R(x_1; : : : ; x_n) = 8y_1 9y_2 8y_3 9y_4 : : : R_1(: : :) \,\hat{}\qquad \hat{}\, R_s(: : :):$$

Then QCSP(Γ_1) is polynomially reducible to QCSP(Γ_2).

Theorem (Galois Correspondence, Börner, Bulatov, Chen, Jeavons, and Krokhin, 2003)

Γ_1 is de nable by quanti ed conjunctive formulas over Γ_2 IFF each surjective polymorphism of Γ_2 is a polymorphism of Γ_1.

Surjective polymorphisms

Observation

Suppose each relation of Γ_1 is definable from Γ_2 using quantified conjunctive formulas

$$R(x_1, \ldots, x_n) = \forall y_1 \exists y_2 \forall y_3 \exists y_4 \ldots R_1(\ldots) \wedge \qquad \wedge R_s(\ldots).$$

Then QCSP(Γ_1) is polynomially reducible to QCSP(Γ_2).

Theorem (Galois Correspondence, Börner, Bulatov, Chen, Jeavons, and Krokhin, 2003)

Γ_1 is definable by quantified conjunctive formulas over Γ_2 IFF each surjective polymorphism of Γ_2 is a polymorphism of Γ_1.

- The complexity of QCSP(Γ) depends only on surjective polymorphisms of Γ.

Surjective polymorphisms

Observation

Suppose each relation of Γ_1 is definable from Γ_2 using primitive positive formulas

$$R(x_1; : : : ; x_n) = 9y_1 9y_2 9y_3 9y_4 : : : R_1(: : :) \wedge \quad \wedge R_s(: : :):$$

Then QCSP(Γ_1) is polynomially reducible to QCSP(Γ_2).

Theorem (Galois Correspondence, Börner, Bulatov, Chen, Jeavons, and Krokhin, 2003)

Γ_1 is de nable by quanti ed conjunctive formulas over Γ_2 IFF each surjective polymorphism of Γ_2 is a polymorphism of Γ_1.

- The complexity of QCSP(Γ) depends only on surjective polymorphisms of Γ.

Two questions

- What makes QCSP( ) easy?
- What makes QCSP( ) hard?

Two questions

- What makes QCSP( ) easy?
- What makes QCSP( ) hard?

$_2$-restriction of QCSP.

QCSP ^{2}( ) :

Given a sentence $\forall x_1 \ldots \forall x_t \exists y_1 \ldots \exists y_q (R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$, where $R_1, \ldots, R_s \in$.

Decide whether it holds.

$_2$-restriction of QCSP.

QCSP ^{2}( ) :

Given a sentence $\forall x_1 \ldots \forall x_t \exists y_1 \ldots \exists y_q (R_1(\ldots) \wedge \qquad \wedge R_s(\ldots))$, where $R_1; \ldots ; R_s \in$.

Decide whether it holds.

- We need to check that for all evaluations of $x_1; \ldots ; x_t$ there exists a solution of the CSP $R_1(\ldots) \wedge \qquad \wedge R_s(\ldots)$.

$_2$-restriction of QCSP.

QCSP $^{2}( )$:

Given a sentence $\forall x_1 \ldots \forall x_t \exists y_1 \ldots \exists y_q (R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$, where $R_1; \ldots; R_s \in$.
Decide whether it holds.

- We need to check that for all evaluations of $x_1; \ldots; x_t$ there exists a solution of the CSP $R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$.
- How many tuples is it sufficient to check?

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

Examples

1. $A = \{0;1\}$, $F = \{x \vee y\}$. $d_F(n) = n+1$. It is sufficient to have $(0; \ldots ; 0)$ and $(0; \ldots ; 0; 1; 0; \ldots ; 0)$ for any position of 1 to generate $\{0;1\}^n$.

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

Examples

1. $A = \{0; 1\}$, $F = \{x \vee y\}$. $d_F(n) = n + 1$. It is sufficient to have $(0; \dots; 0)$ and $(0; \dots; 0; 1; 0; \dots; 0)$ for any position of 1 to generate $\{0; 1\}^n$.
2. $A = \{0; 1\}$, $F = \{\neg x\}$. $d_F(n) = 2^{n-1}$. It is sufficient to have all tuples starting with 0 to generate $\{0; 1\}^n$.

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

Examples

1. $A = \{0, 1\}$, $F = \{x \vee y\}$. $d_F(n) = n + 1$. It is sufficient to have $(0, \ldots, 0)$ and $(0, \ldots, 0, 1, 0, \ldots, 0)$ for any position of 1 to generate $\{0, 1\}^n$.
2. $A = \{0, 1\}$, $F = \{\neg x\}$. $d_F(n) = 2^{n-1}$. It is sufficient to have all tuples starting with 0 to generate $\{0, 1\}^n$.

- If $d_F(n)$ is restricted by a polynomial in n, then the algebra has the Polynomially Generated Powers (PGP) property

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

Examples

1. $A = \{0; 1\}$, $F = \{x \vee y\}$. $d_F(n) = n + 1$. It is sufficient to have $(0; \dots; 0)$ and $(0; \dots; 0; 1; 0; \dots; 0)$ for any position of 1 to generate $\{0; 1\}^n$.
2. $A = \{0; 1\}$, $F = \{\neg x\}$. $d_F(n) = 2^{n-1}$. It is sufficient to have all tuples starting with 0 to generate $\{0; 1\}^n$.

- If $d_F(n)$ is restricted by a polynomial in n, then the algebra has the Polynomially Generated Powers (PGP) property
- If $d_F(n)$ is exponential in n, then the algebra has the Exponentially Generated Powers (EGP) property

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)

$d_F(n)$ is the minimal size of a generating set of A^n.

- If $d_F(n)$ is restricted by a polynomial in n, then the algebra has the Polynomially Generated Powers (PGP) property
- If $d_F(n)$ is exponential in n, then the algebra has the Exponentially Generated Powers (EGP) property

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

- If $d_F(n)$ is restricted by a polynomial in n, then the algebra has the Polynomially Generated Powers (PGP) property
- If $d_F(n)$ is exponential in n, then the algebra has the Exponentially Generated Powers (EGP) property

Theorem[Zhuk, 2015]

Every finite algebra either has PGP, or has EGP.

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

- If $d_F(n)$ is restricted by a polynomial in n, then the algebra has the Polynomially Generated Powers (PGP) property
- If $d_F(n)$ is exponential in n, then the algebra has the Exponentially Generated Powers (EGP) property

Theorem [Zhuk, 2015]

Every finite algebra either has PGP, or has EGP.

Pair $(a_i; a_{i+1})$ with $a_i \neq a_{i+1}$ is a switch in a tuple $(a_1; \ldots; a_n)$.
$(0; 0; 0; 1; 2; 2; 0; 0; 0; 0)$ has 3 switches,
$(3; 3; 3; 4; 3; 3; 3; 3; 3; 3)$ has 2 switches.

PGP vs EGP

For an algebra $(A; F)$ (a set of operations F on a set A)
$d_F(n)$ is the minimal size of a generating set of A^n.

- If $d_F(n)$ is restricted by a polynomial in n, then the algebra has the Polynomially Generated Powers (PGP) property
- If $d_F(n)$ is exponential in n, then the algebra has the Exponentially Generated Powers (EGP) property

Theorem[Zhuk, 2015]

Every finite algebra either has PGP, or has EGP.

Pair $(a_i; a_{i+1})$ with $a_i \neq a_{i+1}$ is a switch in a tuple $(a_1; \dots; a_n)$.
$(0; 0; 0; 1; 2; 2; 0; 0; 0; 0)$ has 3 switches,
$(3; 3; 3; 4; 3; 3; 3; 3; 3; 3)$ has 2 switches.

Theorem[Zhuk, 2015]

A finite algebra A has PGP IFF there exists k such that each A^n is generated by all tuples with at most k switches.

From ${}_2$ to NP

QCSP^2( ) :

Given a sentence $\forall x_1 \ldots \forall x_t \exists y_1 \ldots \exists y_q (R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$,
where $R_1, \ldots, R_s \in$.
Decide whether it holds.

From $_2$ to NP

QCSP $^2()$:

Given a sentence $\forall x_1 \dots \forall x_t \exists y_1 \dots \exists y_q (R_1(\dots) \wedge \quad \wedge R_s(\dots))$,
where $R_1, \dots, R_s \in$.
Decide whether it holds.

Example

If $x \vee y$ preserves then it is sufficient to check that
$(R_1(\dots) \wedge \quad \wedge R_s(\dots))$ is satisfiable for $(x_1, \dots, x_t) = (0, \dots, 0)$
and $(x_1, \dots, x_{i \; 1}, x_i, x_{i+1}, \dots, x_t) = (0, \dots, 0, 1, 0, \dots, 0)$ for $\forall i$.

From $_2$ to NP

QCSP^2( ) :

Given a sentence $\forall x_1 \ldots \forall x_t \exists y_1 \ldots \exists y_q (R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$, where $R_1, \ldots, R_s \in$.
Decide whether it holds.

Example

If $x \vee y$ preserves then it is sufficient to check that $(R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$ is satisfiable for $(x_1, \ldots, x_t) = (0, \ldots, 0)$ and $(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_t) = (0, \ldots, 0, 1, 0, \ldots, 0)$ for $\forall i$.

Observation

If Pol( ) has PGP, then QCSP^2( ) can be polynomially reduced to CSP($\cup \{ x = a \mid a \in A\}$).

From ${}_2$ to NP

QCSP ^{2}( ) :

Given a sentence $\forall x_1 \ldots \forall x_t \exists y_1 \ldots \exists y_q (R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$, where $R_1, \ldots, R_s \in$.
Decide whether it holds.

Example

If $x \vee y$ preserves then it is sufficient to check that $(R_1(\ldots) \wedge \quad \wedge R_s(\ldots))$ is satisfiable for $(x_1, \ldots, x_t) = (0, \ldots, 0)$ and $(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_t) = (0, \ldots, 0, 1, 0, \ldots, 0)$ for $\forall i$.

Observation

If Pol( ) has PGP, then QCSP ^{2}( ) can be polynomially reduced to CSP($\cup \{ \; x = a \mid a \in A\}$).

Proof:

From $_2$ to NP

QCSP ^{2}( ) :

Given a sentence $\forall x_1 \dots \forall x_t \exists y_1 \dots \exists y_q (R_1(\dots) \wedge \quad \wedge R_s(\dots))$, where $R_1, \dots, R_s \in$.
Decide whether it holds.

Example

If $x \vee y$ preserves then it is su cient to check that $(R_1(\dots) \wedge \quad \wedge R_s(\dots))$ is satis able for $(x_1, \dots, x_t) = (0, \dots, 0)$ and $(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_t) = (0, \dots, 0, 1, 0, \dots, 0)$ for $\forall i$.

Observation

If Pol( ) has PGP, then QCSP^2( ) can be polynomially reduced to CSP($\cup \{ x = a \mid a \in A\}$).

Proof: the instance is equivalent to the CSP instance

$$\bigwedge_{\substack{(a_1,\dots,a_t) \text{ with} \\ \text{at most } k \text{ switches}}} (R_1(\dots) \wedge \quad \wedge R_s(\dots) \wedge (x_1 = a_1) \wedge \quad \wedge (x_t = a_t))$$

From PSpace to NP

From PSpace to NP

9y8x

From PSpace to NP

$$\exists y \forall x$$

$$m$$

$$\forall x^1 \forall x^2 \dots \forall x^{|A|} \exists y \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _{|A|}$$

- $_i$ is obtained from by renaming x by x^i

From PSpace to NP

$$\exists y \forall x$$

m

$$\forall x^1 \forall x^2 \ldots \forall x^{|A|} \exists y \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _{|A|}$$

- $_i$ is obtained from by renaming x by x^i

$$\exists y_1 \forall x_1 \ldots \exists y_t \forall x_t$$

From PSpace to NP

$$9y\,8x$$

$$m$$

$$8x^1 8x^2 ::: 8x^{jAj} 9y \quad {}_1 \wedge \quad {}_2 \wedge \quad \wedge \quad {}_{jAj}$$

- $_i$ is obtained from by renaming x by x^i

$$9y_1 8x_1 ::: 9y_t 8x_t$$

$$m$$

$$8x_1^1 ::: 8x_1^{jAj}\; 8x_2^1 ::: 8x_2^{jAj^2} ::: 8x_t^1 ::: 8x_t^{jAj^t}$$

$$9y_1 9y_2^1 ::: 9y_2^{jAj} ::: \; 9y_t^1 ::: 9y_t^{jAj^{t-1}} \quad {}_1 \wedge \quad {}_2 \wedge \quad \wedge \quad {}_q$$

From PSpace to NP

$$\exists y \forall x$$
$$\Updownarrow$$
$$\forall x^1 \forall x^2 \ldots \forall x^{|A|} \exists y \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _{|A|}$$

- $_i$ is obtained from by renaming x by x^i

$$\exists y_1 \forall x_1 \ldots \exists y_t \forall x_t$$
$$\Updownarrow$$

$$\forall x_1^1 \ldots \forall x_1^{|A|} \; \forall x_2^1 \ldots \forall x_2^{|A|^2} \; \ldots \forall x_t^1 \ldots \forall x_t^{|A|^t}$$
$$\exists y_1 \exists y_2^1 \ldots \exists y_2^{|A|} \ldots \; \exists y_t^1 \ldots \exists y_t^{|A|^{t \; 1}} \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _q$$

- For the PGP case it is su cient to check tuples with at most k switches

From PSpace to NP

$$9y\,8x$$

$$m$$

$$8x^{1}8x^{2} : : : 8x^{jAj}9y \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _{jAj}$$

- $_i$ is obtained from by renaming x by x^i

$$9y_1 8x_1 : : : 9y_t\, 8x_t$$

$$m$$

$$1\ 1\ 1 : : : 1\ 1 \quad 1\ 1\ 2 : : : 2 \quad : : : \quad 0\ 0\ 0\ 0 : : : 0$$

$$8x_1^1 : : : 8x_1^{jAj}\ 8x_2^1 : : : 8x_2^{jAj^2}\ : : : 8x_t^1 : : : 8x_t^{jAj^t}$$

$$9y_1 9y_2^1 : : : 9y_2^{jAj} : : : \ 9y_t^1 : : : 9y_t^{jAj^t\ 1} \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _q$$

- For the PGP case it is su cient to check tuples with at most k switches

From PSpace to NP

$$9y\,8x$$
$$m$$
$$8x^1 8x^2 :::8x^{jAj} 9y \quad {}_1 \wedge \quad {}_2 \wedge \quad \wedge \quad {}_{jAj}$$

- ${}_i$ is obtained from by renaming x by x^i

$$9y_1 8x_1 :::9y_t 8x_t$$
$$m$$
$$1\ 1\ 1:::1\ 1 \quad 1\ 1\ 2:::2 \quad ::: \quad 0\ 0\ 0\ 0:::0$$
$$8x_1^1 :::8x_1^{jAj}\ 8x_2^1 :::8x_2^{jAj^2} :::8x_t^1 :::8x_t^{jAj^t}$$
$$9y_1 9y_2^1 :::9y_2^{jAj} ::: \ 9y_t^1 :::9y_t^{jAj^{t}\ 1} \quad {}_1 \wedge \quad {}_2 \wedge \quad \wedge \quad {}_q$$

- For the PGP case it is su cient to check tuples with at most k switches
- We keep variables with the switches

From PSpace to NP

$$\exists y \forall x$$

$$\Leftrightarrow$$

$$\forall x^1 \forall x^2 \dots \forall x^{|A|} \exists y \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _{|A|}$$

- $_i$ is obtained from by renaming x by x^i

$$\exists y_1 \forall x_1 \dots \exists y_t \forall x_t$$

$$\Leftrightarrow$$

$$1\ 1\ 1 \dots 1\ 1 \quad 1\ 1\ 2 \dots 2 \quad \dots \quad 0\ 0\ 0\ 0 \dots 0$$

$$\forall x_1^1 \dots \forall x_1^{|A|} \; \forall x_2^1 \dots \forall x_2^{|A|^2} \dots \forall x_t^1 \dots \forall x_t^{|A|^t}$$

$$\exists y_1 \exists y_2^1 \dots \exists y_2^{|A|} \dots \; \exists y_t^1 \dots \exists y_t^{|A|^{t-1}} \quad _1 \wedge \quad _2 \wedge \quad \wedge \quad _q$$

- For the PGP case it is su cient to check tuples with at most k switches
- We keep variables with the switches
- We assign $x_1^1 = \quad = x_1^{|A|} = 1; \dots; x_t^1 = \quad = x_t^{|A|^t} = 0$

From PSpace to NP

Theorem

Suppose Pol( ) has PGP. Then QCSP( ) is polynomially reducible to CSP($\cup \{ \ x = a \mid a \in A\}$).

From PSpace to NP

Theorem*

Suppose Pol( ) has PGP. Then QCSP( ) is polynomially reducible to CSP([f x = a j a 2 Ag).

* For   containing all constants relations this was shown earlier by Chen, Martin, Carvalho, and Madelaine.

From PSpace to NP

Theorem*

Suppose Pol( ) has PGP. Then QCSP( ) is polynomially reducible to CSP([f x = a j a 2 Ag).

* For containing all constants relations this was shown earlier by Chen, Martin, Carvalho, and Madelaine.

Corollary 1

Suppose Pol( ) has PGP. Then QCSP( ) is in NP.

From PSpace to NP

Theorem*

Suppose Pol( ) has PGP. Then QCSP( ) is polynomially reducible to CSP([f x = a j a 2 Ag).

* For containing all constants relations this was shown earlier by Chen, Martin, Carvalho, and Madelaine.

Corollary 1

Suppose Pol( ) has PGP. Then QCSP( ) is in NP.

Corollary 2

Suppose Pol( ) has PGP. Then QCSP( ) is either tractable, or NP-complete.

Chen Conjecture

Suppose contains $\{x = a \mid a \in A\}$. Then QCSP( )

Chen Conjecture

Suppose contains $\{x = a \mid a \in A\}$. Then QCSP( )

- is in P, if Pol( ) has PGP and WNU

P

Chen Conjecture

Suppose contains $\{x = a \mid a \in A\}$. Then QCSP( )

- is in P, if Pol( ) has PGP and WNU
- is NP-complete, if Pol( ) has PGP and has no WNU

P

NP

Chen Conjecture (QCSP Trichotomy Conjecture)

Suppose contains $\{x = a \mid a \in A\}$. Then QCSP( )

- is in P, if Pol( ) has PGP and WNU
- is NP-complete, if Pol( ) has PGP and has no WNU
- is PSPACE-complete, if Pol( ) has no PGP

P

PSPACE

NP

Chen Conjecture

Weak Chen Conjecture

If Pol( ) has EGP, then QCSP( ) is coNP-hard.

Chen Conjecture

Weak Chen Conjecture

If Pol( ) has EGP, then QCSP( ) is coNP-hard.

Almost a proof of Weak Chen Conjecture

Chen Conjecture

Weak Chen Conjecture

If Pol( ) has EGP, then QCSP( ) is coNP-hard.

Almost a proof of Weak Chen Conjecture

1. If Pol( ) has EGP then we can de ne (encode) by a positive primitive formula the compliment to 3-CNF.

Chen Conjecture

Weak Chen Conjecture

If Pol( ) has EGP, then QCSP( ) is coNP-hard.

Almost a proof of Weak Chen Conjecture

1. If Pol( ) has EGP then we can de ne (encode) by a positive primitive formula the compliment to 3-CNF.
2. If this de nition is e ciently computable, then QCSP( ) is coNP-hard.

Chen Conjecture

Weak Chen Conjecture

If Pol( ) has EGP, then QCSP( ) is coNP-hard.

Almost a proof of Weak Chen Conjecture

1. If Pol( ) has EGP then we can de ne (encode) by a positive primitive formula the compliment to 3-CNF.
2. If this de nition is e ciently computable, then QCSP( ) is coNP-hard.

Lemma (Classi cation for the conservative case) [Zhuk, Martin, 2018]

Chen Conjecture holds for containing all unary relations.

QCSP Monsters

P

PSPACE

NP

QCSP Monsters

- there exists   on a 3-element domain such that QCSP( ) is coNP-complete.

P

PSPACE

coNP

NP

QCSP Monsters

- there exists   on a 3-element domain such that QCSP( ) is coNP-complete.
- there exists   on a 4-element domain such that QCSP( ) is DP-complete, where DP = NP^ coNP.

DP

coNP

P

PSPACE

NP

QCSP Monsters

- there exists on a 3-element domain such that QCSP( ) is coNP-complete.
- there exists on a 4-element domain such that QCSP( ) is DP-complete, where DP = NP^ coNP.
- there exists on a 10-element domain such that QCSP( ) is $^{P}_{2}$-complete.

DP

coNP

P

PSPACE

NP

$^{P}_{2}$

QCSP Monsters

- there exists on a 3-element domain such that QCSP( ) is coNP-complete.
- there exists on a 4-element domain such that QCSP( ) is DP-complete, where DP = NP^ coNP.
- there exists on a 10-element domain such that QCSP( ) is $^{P}_{2}$-complete.
- there exists having EGP such that QCSP( ) is in P.

DP

coNP

P

PSPACE

NP

$^{P}_{2}$

QCSP Monsters

- there exists on a 3-element domain such that QCSP( ) is coNP-complete.
- there exists on a 4-element domain such that QCSP( ) is DP-complete, where DP = NP^ coNP.
- there exists on a 10-element domain such that QCSP( ) is $^{P}_{2}$-complete.
- there exists having EGP such that QCSP( ) is in P.

DP

Are there any other monsters???

coNP

P

PSPACE

NP

$^{P}_{2}$

Classi cation for a 3-element-domain

Classification for a 3-element-domain

Theorem (Classification for a 3-element domain)

Suppose Γ is a constraint language on $\{0,1,2\}$ containing $\{x = a \mid a \in \{0,1,2\}\}$. Then QCSP($\Gamma$) is

- in P, or
- NP-complete, or
- coNP-complete, or
- PSPACE-complete.

Classification for a 3-element-domain

Theorem (Classification for a 3-element domain)

Suppose is a constraint language on $\{0; 1; 2\}$ containing $\{x = a \mid a \in \{0; 1; 2\}\}$. Then QCSP( ) is

- in P, or
- NP-complete, or
- coNP-complete, or
- PSPACE-complete.

PSPACE

NP

coNP

P

Two questions

- What makes QCSP( ) easy?
- What makes QCSP( ) hard?

Two questions

- What makes QCSP( ) easy?
- What makes QCSP( ) hard?

Two questions

- What makes QCSP( ) easy?
- What makes QCSP( ) PSpace-hard?

What makes QCSP( ) PSpace-hard?

What makes QCSP( ) PSpace-hard?

Let A = f + ; ; 0, 1 g

What makes QCSP( ) PSpace-hard?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

What makes QCSP( ) PSpace-hard?

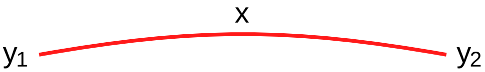
Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$R_0(y_1; y_2; x)$ = ($y_1; y_2$ 2 f + ; g) ^ (y_1 = y_2 _ x 6= 0)

What makes QCSP( ) PSpace-hard?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

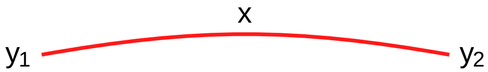
$R_0(y_1; y_2; x)$ = ($y_1; y_2$ 2 f + ; g) ^ (y_1 = y_2 _ x 6= 0)



What makes QCSP( ) PSpace-hard?

Let A = f + ; ; 0, 1 g, = f $R_0; R_1$; f + g; f gg .

$R_0(y_1; y_2; x)$ = ($y_1; y_2$ 2 f + ; g) ^ (y_1 = y_2 _ x 6= 0)



$R_1(y_1; y_2; x)$ = ($y_1; y_2$ 2 f + ; g) ^ (y_1 = y_2 _ x 6= 1)

What makes QCSP( ) PSpace-hard?

Let $A = \{+, \ , 0, 1\}$, $\ = \{R_0, R_1, \{+\}, \{\ \}\}$.

$R_0(y_1, y_2, x) = (\ y_1, y_2 \in \{+, \ \}\ ) \wedge (y_1 = \ y_2 \vee x \neq 0)$

x

y_1  y_2

$R_1(y_1, y_2, x) = (\ y_1, y_2 \in \{+, \ \}\ ) \wedge (y_1 = \ y_2 \vee x \neq 1)$

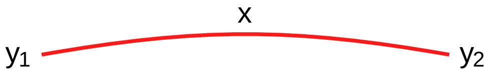
x

y_1  y_2

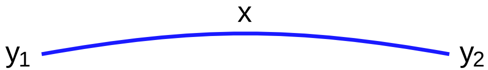
What makes QCSP( ) PSpace-hard?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

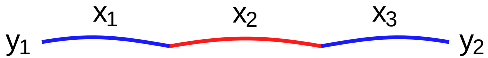
$R_0(y_1; y_2; x)$ = ($y_1; y_2$ 2 f + ; g) ^ (y_1 = y_2 _ x 6= 0)



$R_1(y_1; y_2; x)$ = ($y_1; y_2$ 2 f + ; g) ^ (y_1 = y_2 _ x 6= 1)



9$u_1$9$u_2$$R_1(y_1; u_1; x_1)$ ^ $R_0(u_1; u_2; x_2)$ ^ $R_1(u_2; y_2; x_3)$

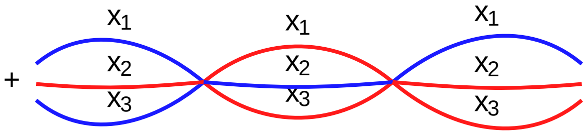


How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

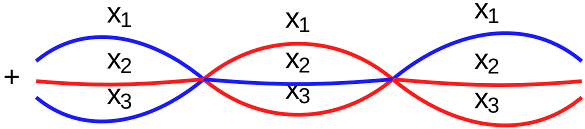
How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .



How to prove PSpace-hardness?

Let A = f + ; ; 0 , 1 g, = f R_0; R_1; f + g; f gg .

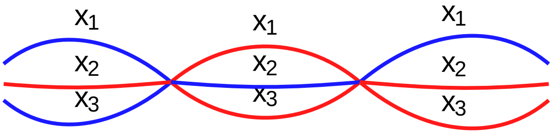


$: ((x_1 \_ \overline{x}_2 \_ x_3)\ \hat{}\ (\overline{x}_1 \_ x_2 \_ \overline{x}_3)\ \hat{}\ (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$8x_1 8x_2 8x_3$ +

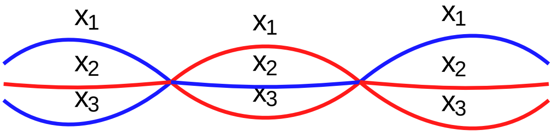


: ((x_1 _ $\overline{x}_2$ _ x_3) ^ ($\overline{x}_1$ _ x_2 _ $\overline{x}_3$) ^ (x_1 _ $\overline{x}_2$ _ $\overline{x}_3$))

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$8x_1 8x_2 8x_3$ +

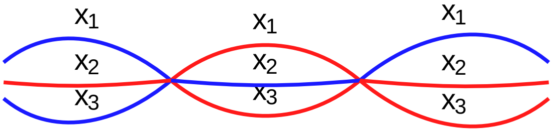


$8x_1 8x_2 8x_3$: ((x_1 _ $\overline{x}_2$ _ x_3) ^ ($\overline{x}_1$ _ x_2 _ $\overline{x}_3$) ^ (x_1 _ $\overline{x}_2$ _ $\overline{x}_3$))

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$8x_1 8x_2 8x_3$ +



$8x_1 8x_2 8x_3$: $((x_1 \_ \overline{x}_2 \_ x_3)$ ^ $(\overline{x}_1 \_ x_2 \_ \overline{x}_3)$ ^ $(x_1 \_ \overline{x}_2 \_ \overline{x}_3))$

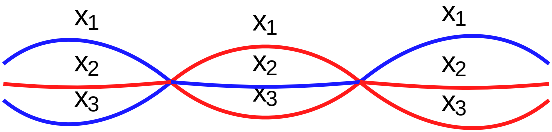
m

: $(9x_1 9x_2 9x_3$ $((x_1 \_ \overline{x}_2 \_ x_3)$ ^ $(\overline{x}_1 \_ x_2 \_ \overline{x}_3)$ ^ $(x_1 \_ \overline{x}_2 \_ \overline{x}_3))$

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$8x_1 8x_2 8x_3$ +



$$8x_1 8x_2 8x_3 \;\; : \; ((x_1 \_ \overline{x}_2 \_ x_3) \text{ ^ } (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \text{ ^ } (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

m

$$: \; (9x_1 9x_2 9x_3 \;\; ((x_1 \_ \overline{x}_2 \_ x_3) \text{ ^ } (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \text{ ^ } (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

Claim

QCSP( ) is coNP-hard.

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

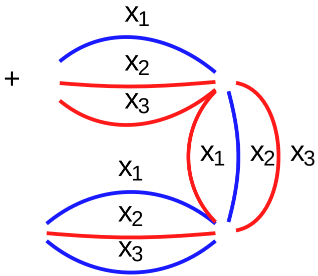
How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$$: ((x_1 \_ \overline{x}_2 \_ x_3) \wedge (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \wedge (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

How to prove PSpace-hardness?

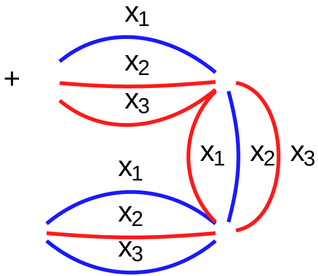
Let $A = f + ; \quad ; \boxed{0}, \boxed{1}\ g, \quad = f\, R_0; R_1; f + g; f\, gg \quad .$



$: ((x_1 \_ \overline{x}_2 \_ x_3) \wedge (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \wedge (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

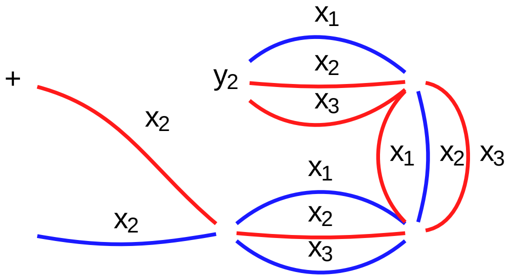


$8x_1 9x_2 8x_3$: $((x_1 \_ \overline{x}_2 \_ x_3)$ ^ $(\overline{x}_1 \_ x_2 \_ \overline{x}_3)$ ^ $(x_1 \_ \overline{x}_2 \_ \overline{x}_3))$

How to prove PSpace-hardness?

Let A = f + ; ; 0, 1 g, = f R_0; R_1; f + g; f gg .

$8x_1 9y_2 8x_2 8x_3$

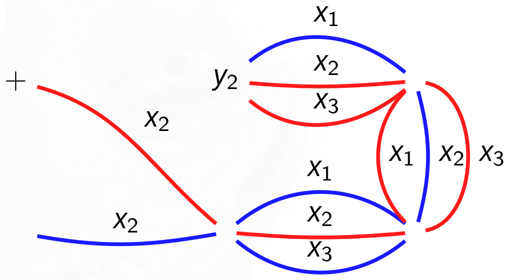


$8x_1 9x_2 8x_3$: ((x_1 _ $\overline{x}_2$ _ x_3) ^ ($\overline{x}_1$ _ x_2 _ $\overline{x}_3$) ^ (x_1 _ $\overline{x}_2$ _ $\overline{x}_3$))

How to prove PSpace-hardness?

Let $A = f+; \;\; ;$ 0, 1 g, $\Gamma = fR_0; R_1; f+g; f \;\; gg$.

$8x_1 9y_2 8x_2 8x_3$



$$8x_1 9x_2 8x_3 \;\; : ((x_1 \_ \overline{x}_2 \_ x_3) \wedge (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \wedge (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

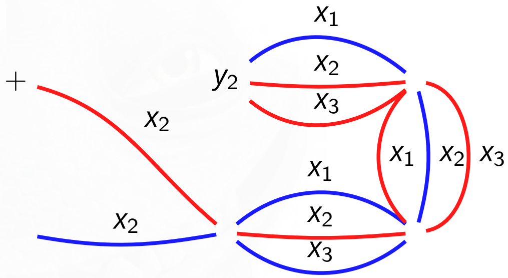
$$m$$

$$: (9x_1 8x_2 9x_3 \;\; ((x_1 \_ \overline{x}_2 \_ x_3) \wedge (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \wedge (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

How to prove PSpace-hardness?

Let $A = f+;\ \ ;$ 0, 1 g, $\Gamma = fR_0; R_1; f+g; f\ \ gg$.

$8x_1 9y_2 8x_2 8x_3$



$$8x_1 9x_2 8x_3\ \ :((x_1 \_ \overline{x}_2 \_ x_3) \wedge (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \wedge (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

$$m$$

$$:(9x_1 8x_2 9x_3\ \ ((x_1 \_ \overline{x}_2 \_ x_3) \wedge (\overline{x}_1 \_ x_2 \_ \overline{x}_3) \wedge (x_1 \_ \overline{x}_2 \_ \overline{x}_3))$$

Claim

QCSP(Γ) is PSpace-hard.

Theorem (Π_2^P vs PSpace)

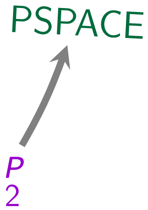
Suppose Γ *contains* $\{x = a \mid a \in A\}$. *Then* QCSP(Γ)

- *is PSpace-hard if there exists a reflexive relation* $S \subsetneq A^n$ *and a nontrivial equivalence relation* η *on* $D \subseteq A$ *such that* $R(y_1, y_2, x_1, \ldots, x_n) = \eta(y_1, y_2) \vee S(x_1, \ldots, x_n)$ *is definable by a positive primitive formula over* Γ
- *in* Π_2^P *otherwise.*

Theorem (Π_2^P vs PSpace)

Suppose Γ *contains* $f x = a \; j \; a \; 2 \; Ag$. *Then* QCSP($\Gamma$)

- *is PSpace-hard if there exists a re exive relation S* (A^n *and a nontrivial equivalence relation on D A such that* $R(y_1; y_2; x_1; \ldots; x_n) = (y_1; y_2) \_ S(x_1; \ldots; x_n)$ *is de nable by a positive primitive formula over* Γ
- *in* Π_2^P *otherwise.*



Theorem (Π_2^P vs PSpace)

Suppose Γ *contains* $fx = a\ j\ a\ 2\ Ag$. *Then* QCSP(Γ)

- *is PSpace-hard if there exists a re exive relation* S (A^n *and a nontrivial equivalence relation* on D A *such that* $R(y_1; y_2; x_1; :::; x_n) = (y_1; y_2) \_ S(x_1; :::; x_n)$ *is de nable by a positive primitive formula over* Γ
- *in* Π_2^P *otherwise.*

Theorem (Π_2^P vs PSpace)

Suppose Γ *contains* $\{x = a \mid a \in A\}$*. Then* QCSP(Γ)

- *is PSpace-hard if there exists a reflexive relation* $S \subsetneq A^n$ *and a nontrivial equivalence relation* $\sim$ *on* $D \subseteq A$ *such that* $R(y_1, y_2, x_1, \ldots, x_n) = \sim(y_1, y_2) \vee S(x_1, \ldots, x_n)$ *is definable by a positive primitive formula over* Γ
- *in* Π_2^P *otherwise.*

Theorem (Π_2^P vs PSpace)

Suppose Γ contains $\{x = a \mid a \in A\}$. Then QCSP(Γ)

- *is PSpace-hard if there exists a reflexive relation $S \subsetneq A^n$ and a nontrivial equivalence relation $\sim$ on $D \subseteq A$ such that $R(y_1, y_2, x_1, \ldots, x_n) = \sim(y_1, y_2) \vee S(x_1, \ldots, x_n)$ is definable by a positive primitive formula over Γ*
- *in Π_2^P otherwise.*

Theorem (Π_2^P vs PSpace)

Suppose Γ *contains* $f x = a \; j \; a \; 2 \; A g$. *Then* QCSP($\Gamma$)

- *is PSpace-hard if there exists a re exive relation* $S \;\; (\;\; A^n$ *and a nontrivial equivalence relation* $\quad$ *on* $D \quad\; A$ *such that* $R(y_1; y_2; x_1; : : : ; x_n) = \quad (y_1; y_2) \_ S(x_1; : : : ; x_n)$ *is de nable by a positive primitive formula over* Γ
- *in* Π_2^P *otherwise.*

Theorem (Π_2^P vs PSpace)

Suppose Γ *contains* $fx = a \; j \; a \; 2 \; Ag$. *Then* QCSP($\Gamma$)

- *is PSpace-hard if there exists a re exive relation* S (A^n *and a nontrivial equivalence relation on* D A *such that* $R(y_1; y_2; x_1; \ldots; x_n) = (y_1; y_2) \_ S(x_1; \ldots; x_n)$ *is de nable by a positive primitive formula over* Γ
- *in* Π_2^P *otherwise.*

Lemma

There exists Γ on a 6-element set such that QCSP(Γ) is Π_2^P-complete.


P
NP
coNP
DP
P 2
P 2
PSPACE

QCSP Hepta-chotomy to prove

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard
3. coNP vs NP-hard

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard
3. coNP vs NP-hard
4. NP [coNP vs DP-hard

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard
3. coNP vs NP-hard
4. NP [coNP vs DP-hard
5. DP vs Θ_2^P-hard

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard
3. coNP vs NP-hard
4. NP [coNP vs DP-hard
5. DP vs Θ_2^P-hard
6. Θ_2^P vs Π_2^P-hard

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard
3. coNP vs NP-hard
4. NP [coNP vs DP-hard
5. DP vs Θ_2^P-hard
6. Θ_2^P vs Π_2^P-hard
7. Π_2^P vs PSpace-hard

QCSP Hepta-chotomy to prove

1. P vs NP-hard (under Turing reductions).
2. NP vs coNP-hard
3. coNP vs NP-hard
4. NP *[* coNP vs DP-hard
5. DP vs Θ_2^P-hard
6. Θ_2^P vs Π_2^P-hard
7. Π_2^P vs PSpace-hard (proved for Γ containing $fx = a\ j\ a\ 2\ Ag$)

Thank you for your attention

Thank you for your attention

Thank you for your attention

Thank you for your attention