The complexity of the Constraint Satisfaction Problem and its variations

Dmitriy Zhuk

Charles University

Colloquium of Faculty of Informatics

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.

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 0 \mod 2 \\ x_2 + x_3 + x_5 = 1 \mod 2 \end{cases}$$

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 0 \mod 2 \\ x_2 + x_3 + x_5 = 1 \mod 2 \end{cases}$$

Gaussian elimination solves the problem in polynomial time.

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 0 \mod 2 \\ x_2 + x_3 + x_5 = 1 \mod 2 \end{cases}$$

Gaussian elimination solves the problem in polynomial time.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \end{cases}$$

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 & \mod 2 \\ x_1 + x_3 + x_5 = 0 & \mod 2 \\ x_2 + x_4 + x_5 = 0 & \mod 2 \\ x_2 + x_3 + x_5 = 1 & \mod 2 \end{cases}$$

Gaussian elimination solves the problem in polynomial time.

$$\begin{cases} x_1 + x_2 + x_3 = 0 & \mod 2 \\ x_1 + x_3 + x_5 = 0 & \mod 2 \\ x_2 + x_4 + x_5 = 1 & \mod 2 \\ x_2 + x_3 + x_5 = 3 & \end{cases}$$

The problem is NP-hard.

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 0 \mod 2 \\ x_2 + x_3 + x_5 = 1 \mod 2 \end{cases}$$

Gaussian elimination solves the problem in polynomial time.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \end{cases}$$

The problem is NP-hard.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \mod 24 \end{cases}$$

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 0 \mod 2 \\ x_2 + x_3 + x_5 = 1 \mod 2 \end{cases}$$

Gaussian elimination solves the problem in polynomial time.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \end{cases}$$

The problem is NP-hard.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \mod 24 \end{cases}$$

What is the complexity of this problem?

Example

Check whether there exists a solution $x_1, x_2, x_3, \ldots \in \{0, 1\}$.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 0 \mod 2 \\ x_2 + x_3 + x_5 = 1 \mod 2 \end{cases}$$

Gaussian elimination solves the problem in polynomial time.

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \end{cases}$$

The problem is NP-hard.

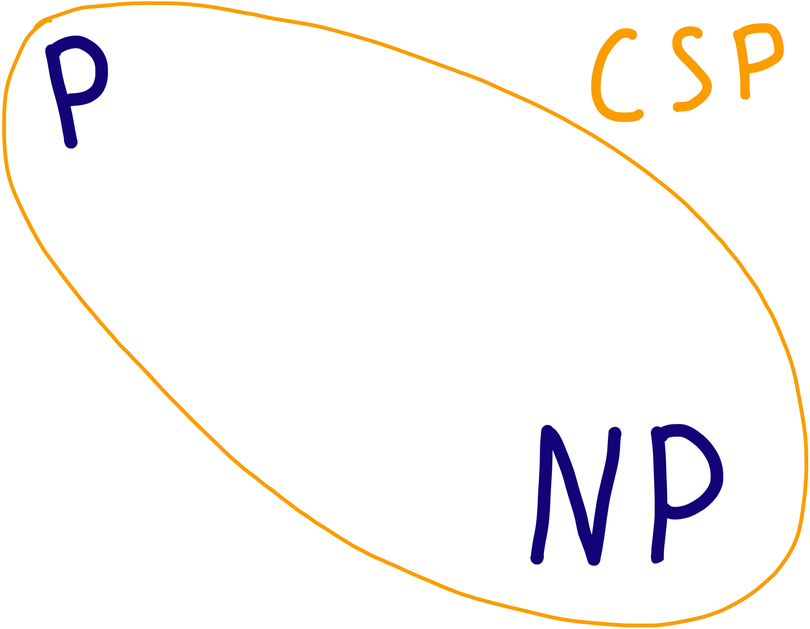
$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_1 + x_3 + x_5 = 0 \mod 2 \\ x_2 + x_4 + x_5 = 1 \mod 2 \\ x_2 + x_3 + x_5 = 3 \mod 24 \end{cases}$$

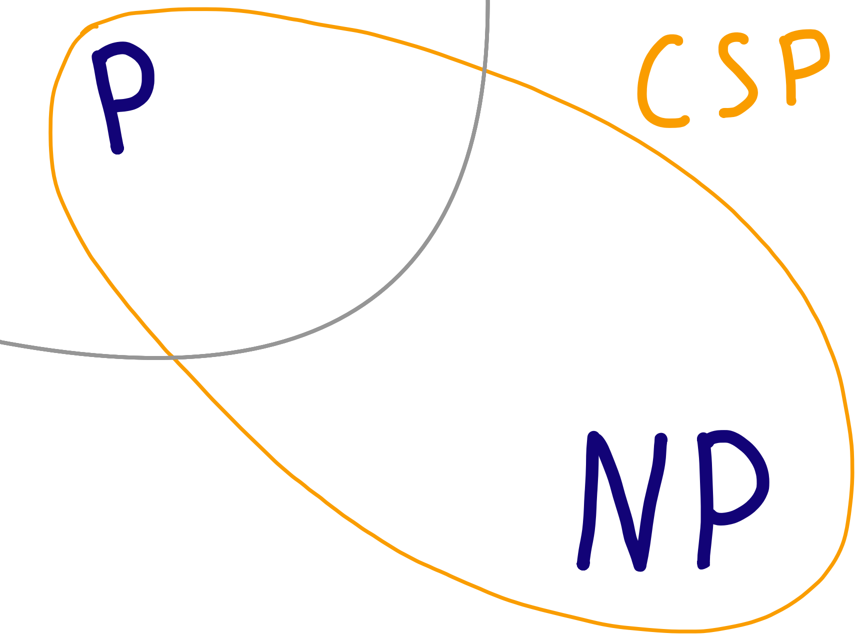
What is the complexity of this problem? Nobody knows!

P

P

NP


CSP
P
NP



P
CSP
NP

What is CSP?

What is CSP?

Constraint Satisfaction Problem
is a triple $\langle \mathbf{X}, \mathbf{D}, \mathbf{C} \rangle$, where

- $\mathbf{X} = \{x_1, \ldots, x_n\}$ is a set of variables,
- $\mathbf{D} = \{D_1, \ldots, D_n\}$ is a set of the respective domains of values, and
- $\mathbf{C} = \{C_1, \ldots, C_m\}$ is a set of constraints

What is CSP?

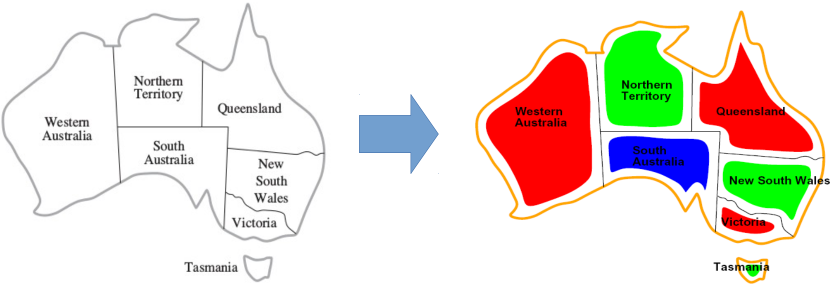
Constraint Satisfaction Problem

is a triple $\langle \mathbf{X}, \mathbf{D}, \mathbf{C} \rangle$, where

- $\mathbf{X} = \{x_1, \ldots, x_n\}$ is a set of variables,
- $\mathbf{D} = \{D_1, \ldots, D_n\}$ is a set of the respective domains of values, and
- $\mathbf{C} = \{C_1, \ldots, C_m\}$ is a set of constraints

Almost everything is CSP!!!

CSP example: map coloring



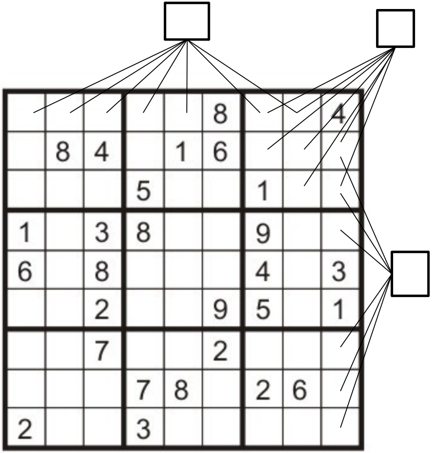
Problem: assign each territory a color such that no two adjacent territories have the same color

Variables: $X = \{WA, NT, Q, NSW, V, SA, T\}$

Domain of variables: $D = \{r, g, b\}$

Constraints: $C = \{SA \neq WA, SA \neq NT, SA \neq Q, \ldots\}$

Another example: sudoku



- Variables:
  - Each (open) square
- Domains:
  - {1,2,…,9}
- Constraints:

9-way alldiff for each column

9-way alldiff for each row

9-way alldiff for each region

(or can have a bunch of pairwise inequality constraints)

Constraint Satisfaction Problem parameterized by a constraint language

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Example:
$A = \{0, 1, 2\}, \Gamma = \{x < y, x \leq y\}$.

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Example:
$A = \{0, 1, 2\}, \Gamma = \{x < y, x \leq y\}$.
CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_4$,

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Example:
$A = \{0, 1, 2\}, \Gamma = \{x < y, x \leq y\}$.
CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_4$, No solutions

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Example:
$A = \{0, 1, 2\}, \Gamma = \{x < y, x \leq y\}$.
CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_4$, No solutions
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_1$,

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Example:
$A = \{0, 1, 2\}, \Gamma = \{x < y, x \leq y\}$.
CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_4$, No solutions
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_1$, $x_1 = x_2 = x_3 = 0$.

Constraint Satisfaction Problem parameterized by a constraint language

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

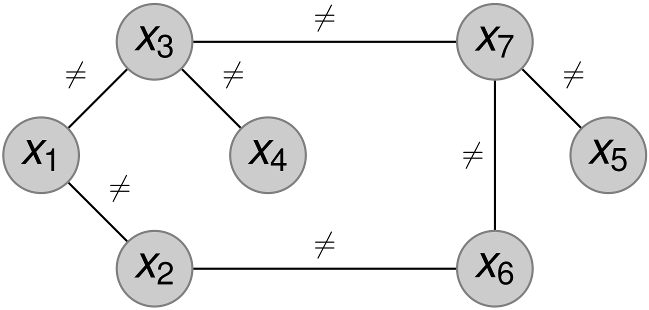
where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Example:
$A = \{0, 1, 2\}, \Gamma = \{x < y, x \leq y\}$.
CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_4$, No solutions
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_1$, $x_1 = x_2 = x_3 = 0$.

Question

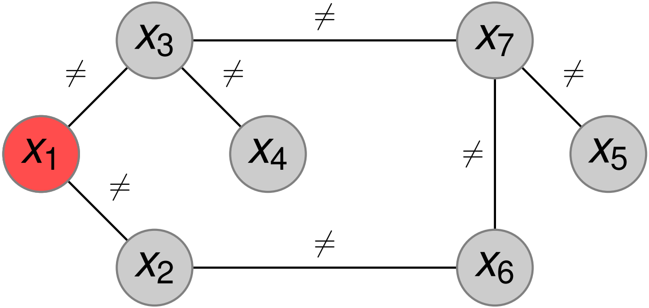
What is the complexity of CSP(Γ) for different Γ?

Graph coloring (two colors)



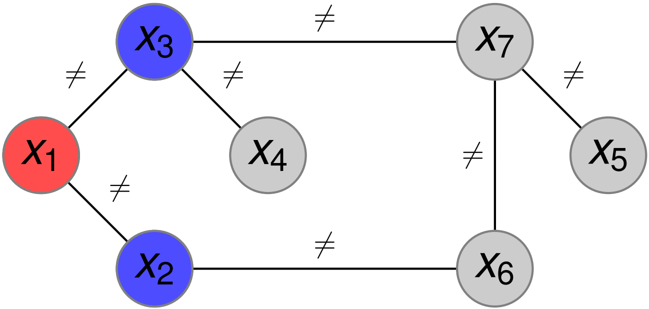
Domain $D = \{$■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (two colors)



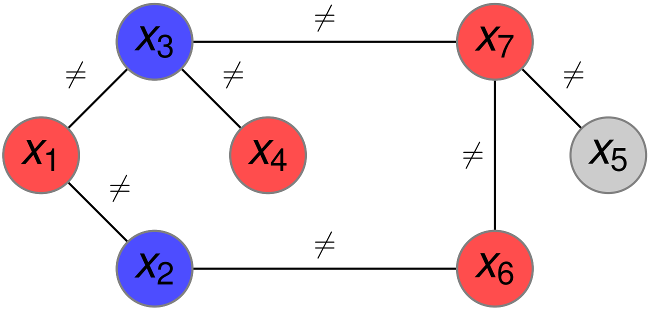
Domain $D = \{$, $\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (two colors)



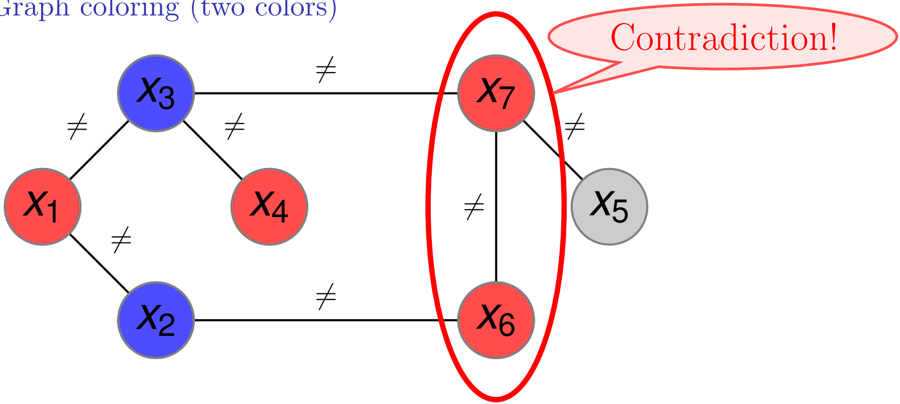
Domain $D = \{$■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (two colors)



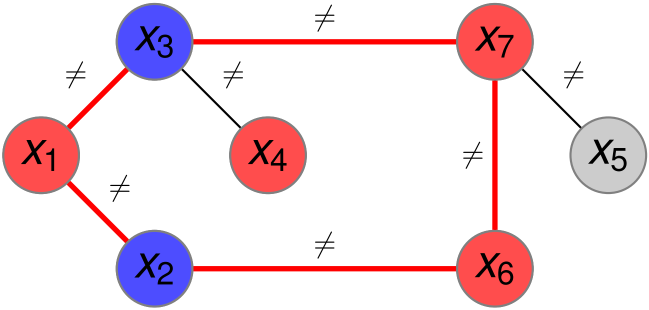
Domain $D = \{$■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (two colors)



Domain $D = \{$■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

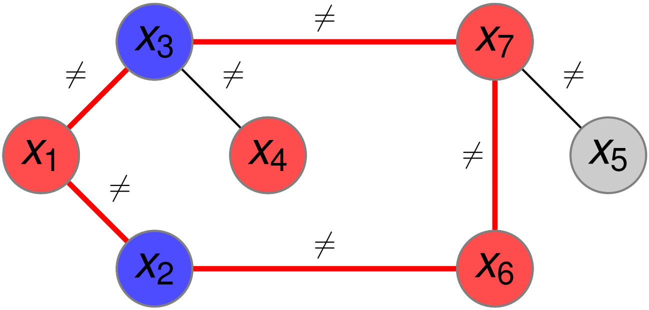
Graph coloring (two colors)



Domain $D = \{$, $\}$

Constraint language $\Gamma = \{\neq\}$.

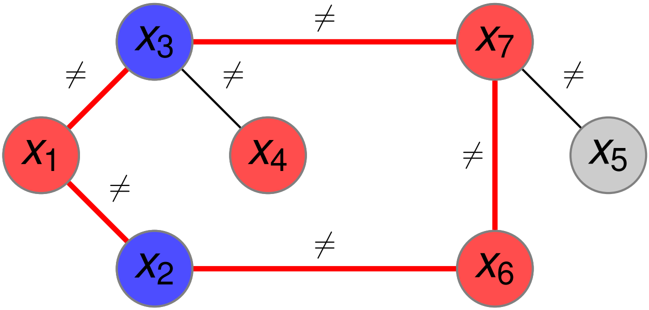
Graph coloring (two colors)



- Either we can color every vertex,
- or we can find an odd cycle.

Domain $D = \{$■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (two colors)



- Either we can color every vertex,
- or we can find an odd cycle.

Local consistency check solves the problem.

Domain $D = \{\blacksquare, \blacksquare\}$

Constraint language $\Gamma = \{\neq\}$.

System of linear equations in a finite field

$$\begin{cases} x_1 + x_2 + 2x_3 = 0 \mod 3 \\ x_1 + 2x_3 + x_5 = 0 \mod 3 \\ 2x_2 + x_4 + x_5 = 0 \mod 3 \\ x_1 + x_3 + 2x_5 = 1 \mod 3 \end{cases}$$

Domain $D = \{0, 1, 2\}$
Constraint language
$\Gamma = \{a_1 x + a_2 y + a_3 z = a_0 \mid a_0, a_1, a_2, a_3 \in D\}$.

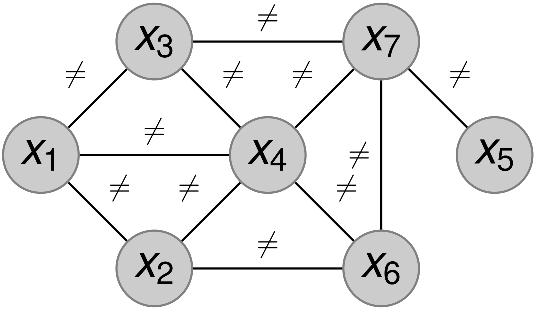
System of linear equations in a finite field

$$\begin{cases} x_1 + x_2 + 2x_3 = 0 \mod 3 \\ x_1 + 2x_3 + x_5 = 0 \mod 3 \\ 2x_2 + x_4 + x_5 = 0 \mod 3 \\ x_1 + x_3 + 2x_5 = 1 \mod 3 \end{cases}$$

Gaussian elimination solves the problem.

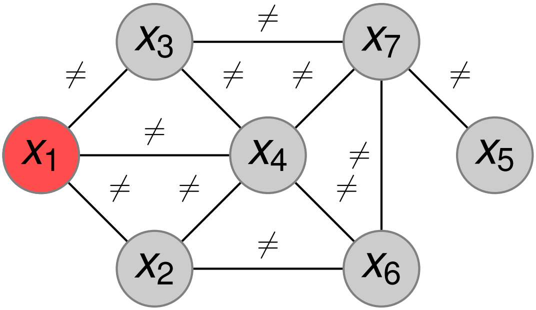
Domain $D = \{0, 1, 2\}$
Constraint language
$\Gamma = \{a_1 x + a_2 y + a_3 z = a_0 \mid a_0, a_1, a_2, a_3 \in D\}$.

Graph coloring (three colors)



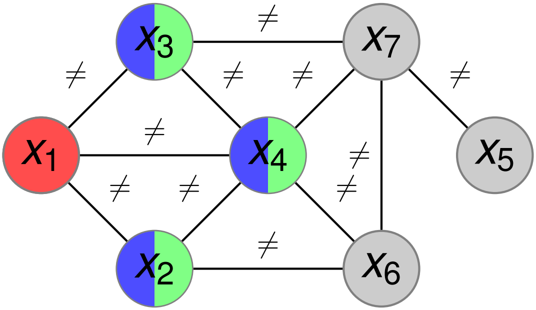
Domain $D = \{\ ,\ ,\ \}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (three colors)



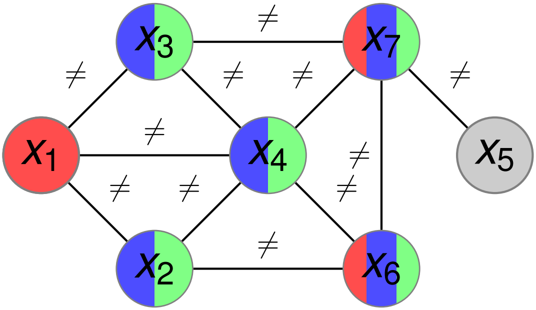
Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (three colors)



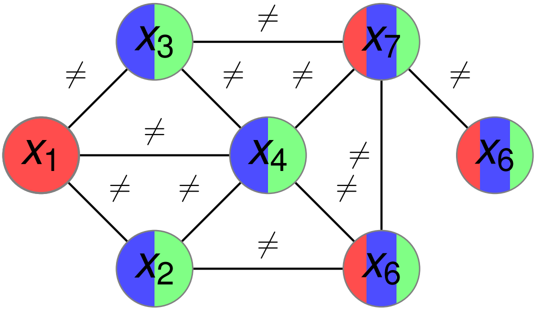
Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (three colors)



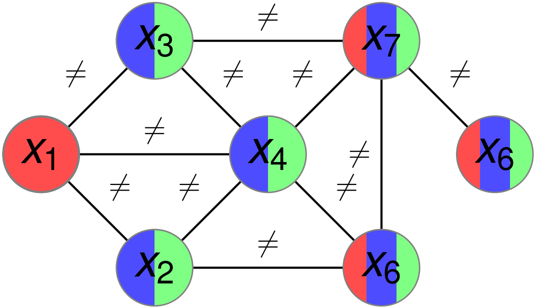
Domain $D = \{\blacksquare, \blacksquare, \blacksquare\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (three colors)



Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

Graph coloring (three colors)

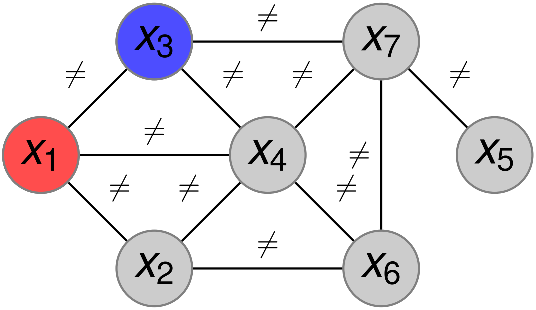


- Local consistency check doesn't give a contradiction.

Domain $D = \{$$\}$

Constraint language $\Gamma = \{\neq\}$.

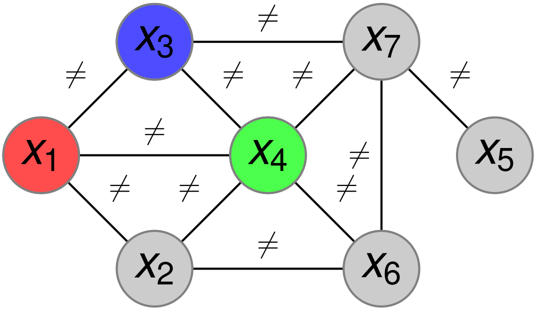
Graph coloring (three colors)



- Local consistency check doesn't give a contradiction.

Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

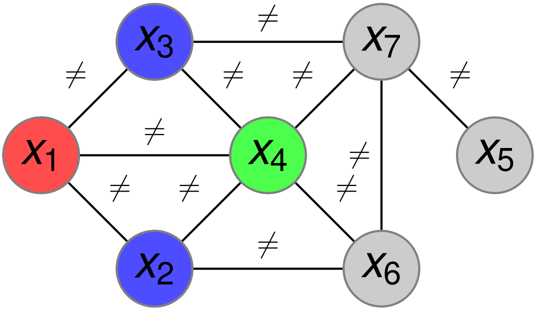
Graph coloring (three colors)



- Local consistency check doesn't give a contradiction.

Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

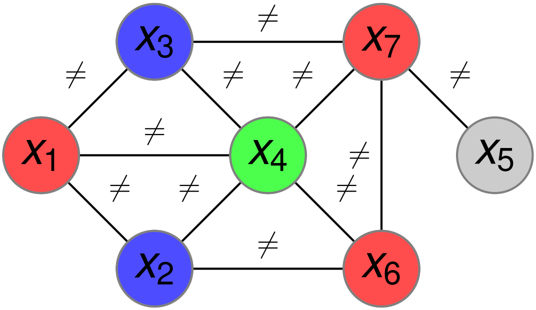
Graph coloring (three colors)



- Local consistency check doesn't give a contradiction.

Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

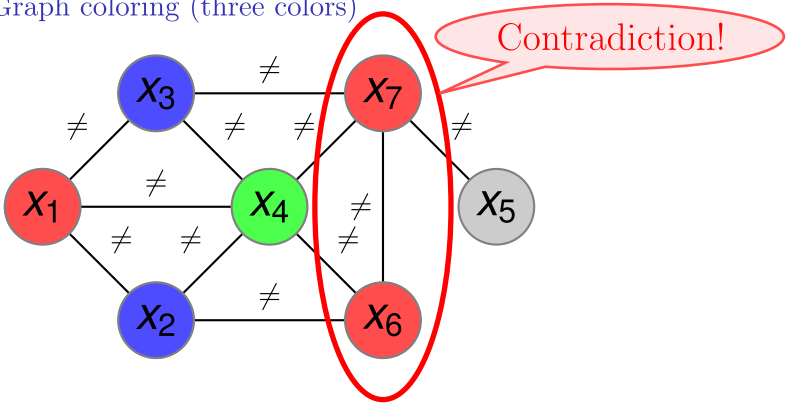
Graph coloring (three colors)



- Local consistency check doesn't give a contradiction.

Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

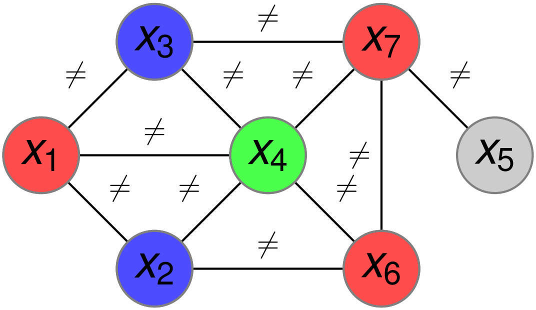
Graph coloring (three colors)



- Local consistency check doesn't give a contradiction.

Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

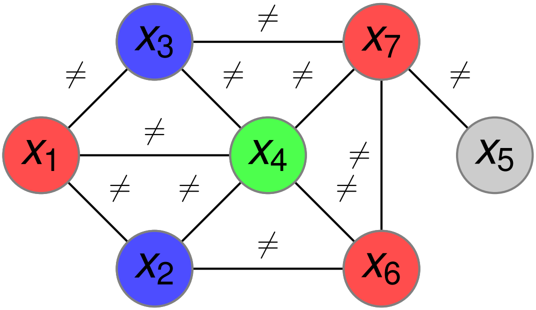
Graph coloring (three colors)



- Local consistency check doesn't give a contradiction.
- The instance has no solutions.

Domain $D = \{$, , $\}$
Constraint language $\Gamma = \{\neq\}$.

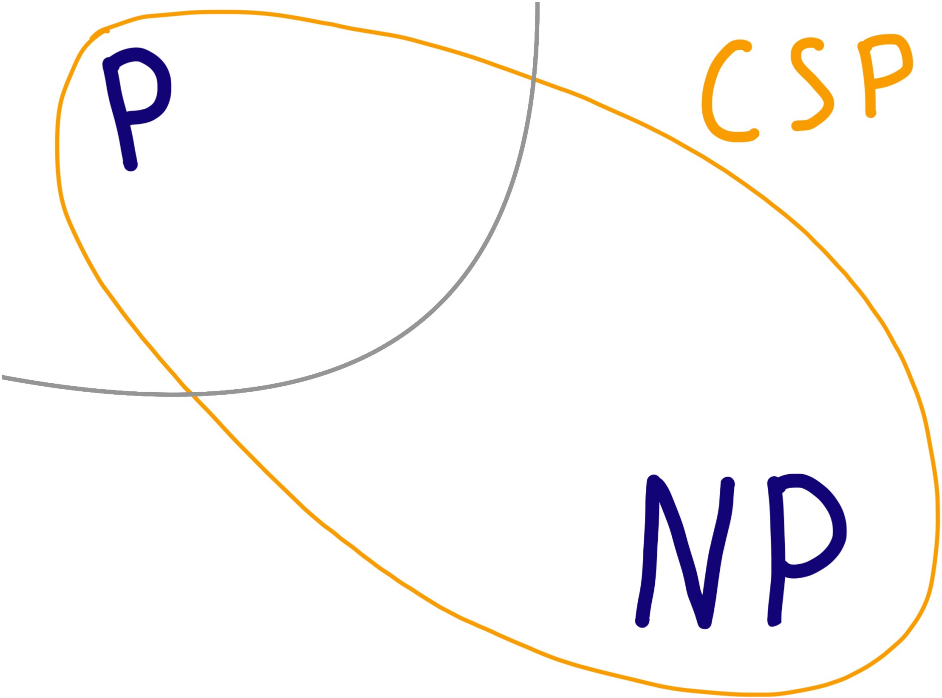
Graph coloring (three colors)

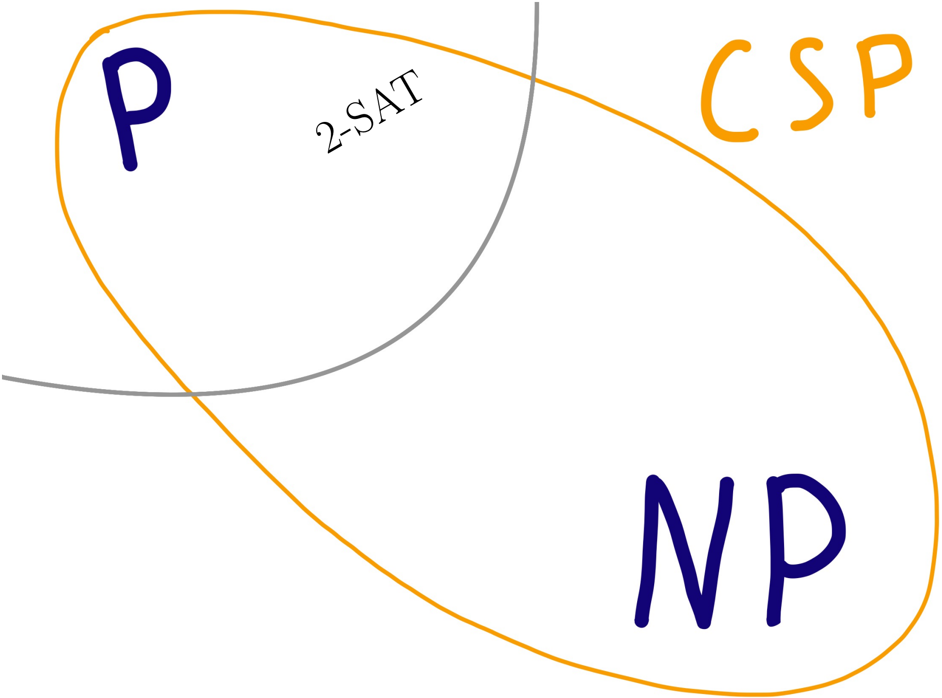


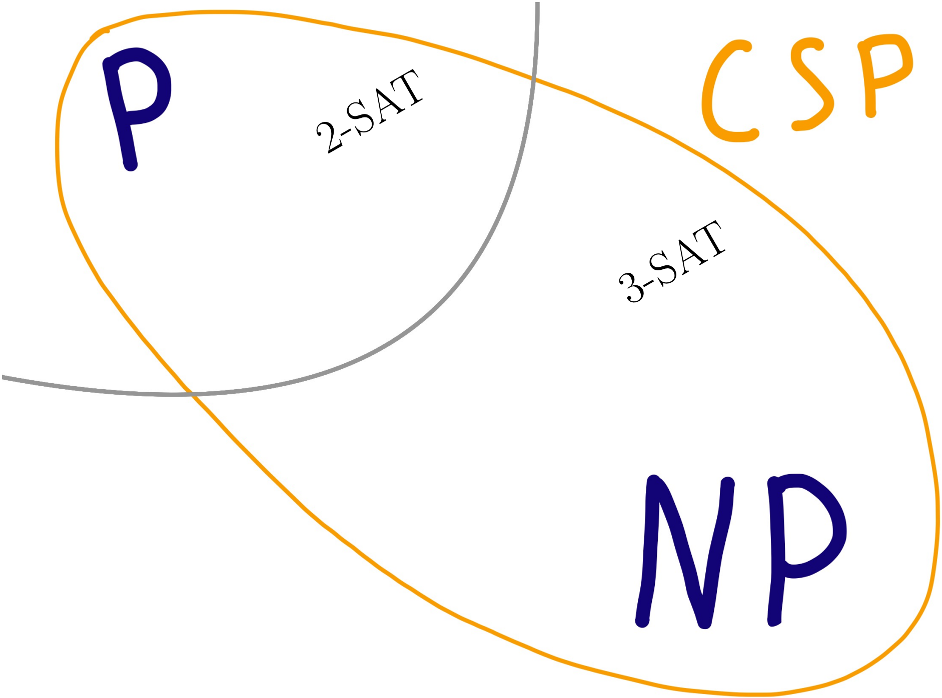
- Local consistency check doesn't give a contradiction.
- The instance has no solutions.

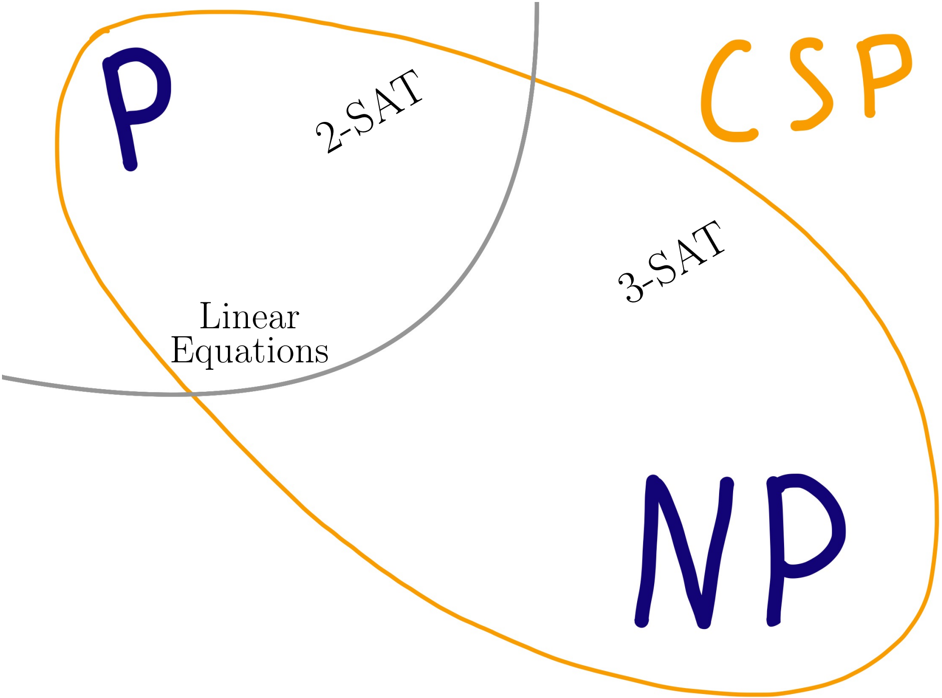
The problem is NP-hard.

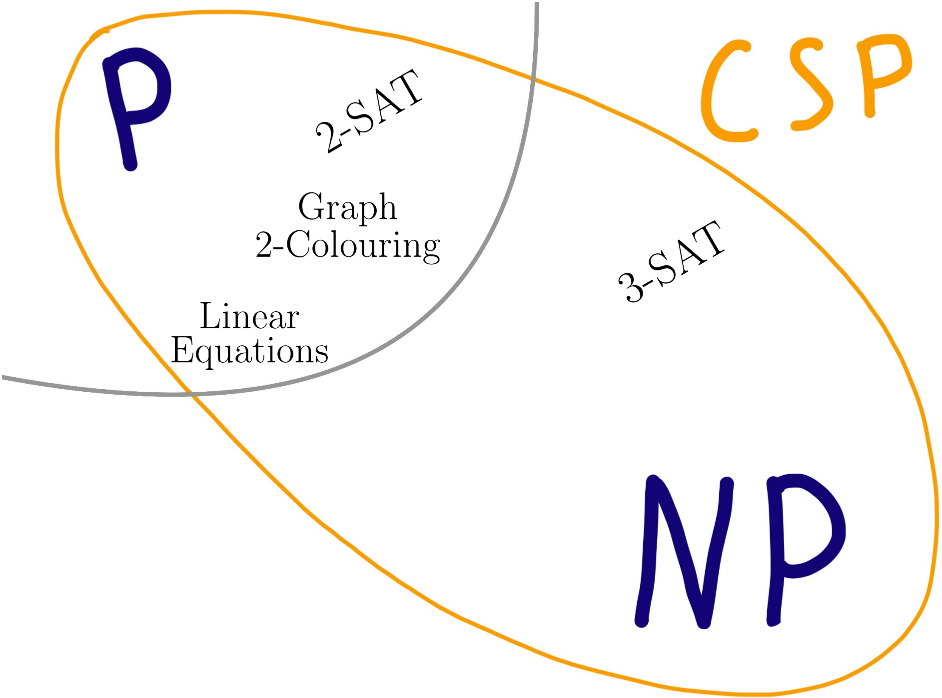
Domain $D = \{$■, ■, ■$\}$
Constraint language $\Gamma = \{\neq\}$.

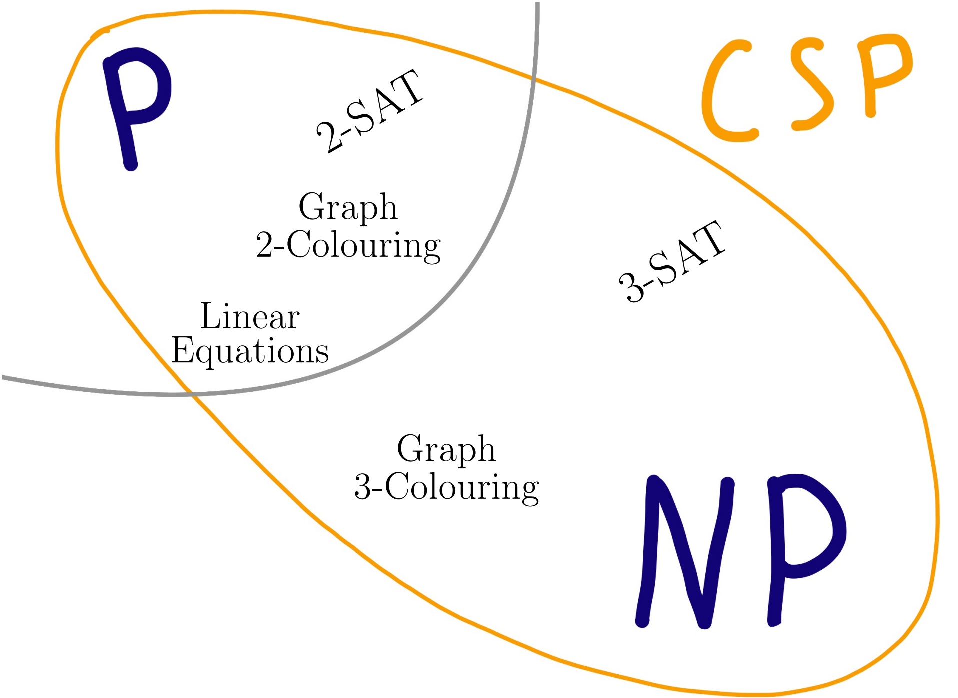

P
CSP
NP


P
2-SAT
CSP
NP


P
CSP
2-SAT
3-SAT
NP


P
2-SAT
CSP
3-SAT
Linear
Equations
NP


P
CSP
2-SAT
Graph
2-Colouring
3-SAT
Linear
Equations
NP


P
CSP
2-SAT
Graph
2-Colouring
3-SAT
Linear
Equations
Graph
3-Colouring
NP

Reduction from one language to another

Reduction from one language to another

CSP(Γ)

Given: a sentence

$$\exists x_1 \dots \exists x_n \, R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it holds.

Reduction from one language to another

CSP(Γ)

Given: a sentence

$$\exists x_1 \dots \exists x_n\ R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it holds.

primitive positive (pp) definition is

$$R(\dots) = \exists y_1 \dots \exists y_\ell\ R_1(\dots) \wedge \dots \wedge R_s(\dots)$$

Reduction from one language to another

CSP(Γ)

Given: a sentence

$$\exists x_1 \dots \exists x_n \, R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it holds.

primitive positive (pp) definition is

$$R(\dots) = \exists y_1 \dots \exists y_\ell \, R_1(\dots) \wedge \dots \wedge R_s(\dots)$$

Fact [Schaefer, 1978]

Suppose $|\Gamma_1| < \infty$, $|\Gamma_2| < \infty$, Γ_2 pp-defines Γ_1. Then CSP(Γ_1) is log-space reducible to CSP(Γ_2).

Reduction from one language to another

CSP(Γ)

Given: a sentence

$$\exists x_1 \dots \exists x_n \, R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it holds.

primitive positive (pp) definition is

$$R(\dots) = \exists y_1 \dots \exists y_\ell \, R_1(\dots) \wedge \dots \wedge R_s(\dots)$$

Fact [Schaefer, 1978]

Suppose $|\Gamma_1| < \infty$, $|\Gamma_2| < \infty$, Γ_2 pp-defines Γ_1. Then CSP(Γ_1) is log-space reducible to CSP(Γ_2).

Theorem [Bodnarchuk, Kaluzhnin, Kotov, Romov, Geiger, 1969]

Γ_2 pp-defines Γ_1 IFF every operation preserving Γ_2 preserves Γ_1

Polymorphisms

Polymorphisms

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

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

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

Polymorphisms

An operation f preserves a relation R,
(equivalently, f is a polymorphism of R)
if for all $\begin{pmatrix} a_1^1 \\ \vdots \\ a_1^s \end{pmatrix}, \ldots, \begin{pmatrix} a_n^1 \\ \vdots \\ a_n^s \end{pmatrix} \in R$,

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

Example

The relation $\leq$ on $\{0, 1, 2\}$

Polymorphisms

An operation f preserves a relation R,
(equivalently, f is a polymorphism of R)
if for all $\begin{pmatrix} a_1^1 \\ \vdots \\ a_1^s \end{pmatrix}, \ldots, \begin{pmatrix} a_n^1 \\ \vdots \\ a_n^s \end{pmatrix} \in R$,

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

Example

The relation $\leq$ on $\{0, 1, 2\}$ is $\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 2 \end{pmatrix}$

Polymorphisms

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

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

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

Example

The relation $\leq$ on $\{0, 1, 2\}$ is $\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 2 \end{pmatrix}$

An operation f preserves $\leq$ iff $f \begin{pmatrix} a_1^1 & \dots & a_n^1 \\ \wedge\!\!\backslash & \ddots & \wedge\!\!\backslash \\ a_1^2 & \dots & a_n^2 \end{pmatrix} = \begin{pmatrix} b^1 \\ \wedge\!\!\backslash \\ b^2 \end{pmatrix}$

Polymorphisms

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

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

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

Example

The relation $\leq$ on $\{0, 1, 2\}$ is $\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 2 \end{pmatrix}$

An operation f preserves $\leq$ iff $f \begin{pmatrix} a_1^1 & \ldots & a_n^1 \\ \wedge\!\!\backslash & \ddots & \wedge\!\!\backslash \\ a_1^2 & \ldots & a_n^2 \end{pmatrix} = \begin{pmatrix} b^1 \\ \wedge\!\!\backslash \\ b^2 \end{pmatrix}$

or, equivalently, f is monotone.

Polymorphisms

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

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

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

Polymorphisms

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

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

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

CSP Dichotomy Theorem [Bulatov, Zhuk, 2017]

CSP(Γ) is solvable in polynomial time if there is a WNU operation preserving Γ; it is NP-complete otherwise.

Polymorphisms

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

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

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

CSP Dichotomy Theorem [Bulatov, Zhuk, 2017]

CSP(Γ) is solvable in polynomial time if there is a WNU operation preserving Γ; it is NP-complete otherwise.

A weak near unanimity operation (WNU) is an operation f satisfying $f(x, \dots, x, y) = f(x, \dots, x, y, x) = \dots = f(y, x, \dots, x)$.

Polymorphisms

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

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

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

CSP Dichotomy Theorem [Bulatov, Zhuk, 2017]

CSP(Γ) is solvable in polynomial time if there is a WNU operation preserving Γ; it is NP-complete otherwise.

A weak near unanimity operation (WNU) is an operation f satisfying $f(x, \ldots, x, y) = f(x, \ldots, x, y, x) = \cdots = f(y, x, \ldots, x)$.

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

Hardness

Hardness

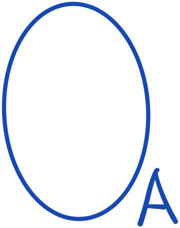
Theorem [McKenzie, Maróti, 2007]

Suppose Γ is not preserved by a WNU.

Hardness

Theorem [McKenzie, Maróti, 2007]

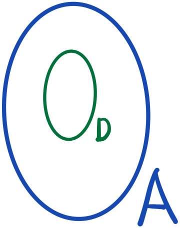
Suppose Γ is not preserved by a WNU.



Hardness

Theorem [McKenzie, Maróti, 2007]

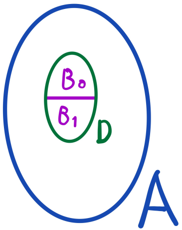
Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$,



Hardness

Theorem [McKenzie, Maróti, 2007]

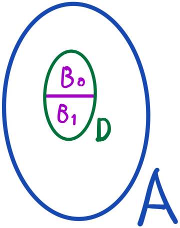
Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$, $D = B_0 \sqcup B_1$



Hardness

Theorem [McKenzie, Maróti, 2007]

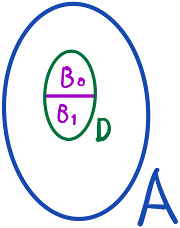
Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$, $D = B_0 \sqcup B_1$ s.t. $D^3 \setminus (B_0^3 \cup B_1^3)$ is pp-definable from Γ.



Hardness

Theorem [McKenzie, Maróti, 2007]

Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$, $D = B_0 \sqcup B_1$ s.t. $D^3 \setminus (B_0^3 \cup B_1^3)$ is pp-definable from Γ.



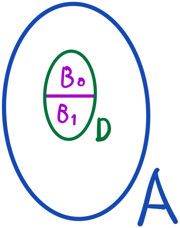
Corollary

Suppose Γ is not preserved by a WNU. Then

Hardness

Theorem [McKenzie, Maróti, 2007]

Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$, $D = B_0 \sqcup B_1$ s.t. $D^3 \setminus (B_0^3 \cup B_1^3)$ is pp-definable from Γ.



Corollary

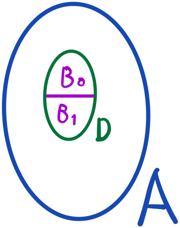
Suppose Γ is not preserved by a WNU. Then

- CSP(NAE3) is log-space reducible to CSP(Γ)

Hardness

Theorem [McKenzie, Maróti, 2007]

Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$, $D = B_0 \sqcup B_1$ s.t. $D^3 \setminus (B_0^3 \cup B_1^3)$ is pp-definable from Γ.



Corollary

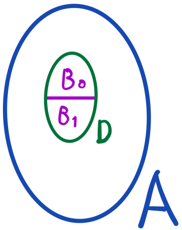
Suppose Γ is not preserved by a WNU. Then

- CSP(NAE3) is log-space reducible to CSP(Γ)
- CSP(Γ) is NP-complete.

Hardness

Theorem [McKenzie, Maróti, 2007]

Suppose Γ is not preserved by a WNU. Then there exists $D \subseteq A$, $D = B_0 \sqcup B_1$ s.t. $D^3 \setminus (B_0^3 \cup B_1^3)$ is pp-definable from Γ.



Corollary

Suppose Γ is not preserved by a WNU. Then

- CSP(NAE3) is log-space reducible to CSP(Γ)
- CSP(Γ) is NP-complete.
- CSP(Δ) is log-space reducible to CSP(Γ) for any finite constraint language Δ.

Polynomial algorithm

Polynomial algorithm

Tractable cases on $\{0, 1\}$

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \left\{ \substack{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0 \\ x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \right\}$.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \left\{ \substack{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0 \\ x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \right\}$.
- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{ {}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \}$.
- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$
- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \left\{ {}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \right\}$.

How to solve:

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0,1\}^2\}$

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0,1\}\}$.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{{}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$
- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{\substack{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0 \\ x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution:

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \{ {}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0,1\}^2\}$

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0,1\}\}$.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{{}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{ {}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{{}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

How to solve:

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \{ {}^{x_1 = 1 \vee x_2 = 0 \vee \dots \vee x_n = 0}_{\quad x_1 = 0 \vee \dots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0,1\}^2\}$

How to solve: force local consistency.

- System of linear equations: $\Gamma = \{a_1 x_1 + \dots + a_n x_n = a_0 \mid a_0, \dots, a_n \in \{0,1\}\}$.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{\begin{smallmatrix} x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0 \\ x_1 = 0 \vee \cdots \vee x_n = 0 \end{smallmatrix} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

How to solve: force local consistency.

How to find a solution:

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \left\{ {x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0 \atop x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \right\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0,1\}^2\}$

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ or $x_i := 1$ and force consistency again.

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0,1\}\}$.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \{{}^{x_1 = 1 \vee x_2 = 0 \vee \dots \vee x_n = 0}_{x_1 = 0 \vee \dots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0,1\}^2\}$

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ or $x_i := 1$ and force consistency again.

Both $\{0\}$ and $\{1\}$ are strong subsets.

- System of linear equations: $\Gamma = \{a_1 x_1 + \dots + a_n x_n = a_0 \mid a_0, \dots, a_n \in \{0,1\}\}$.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0, 1\}$

- Horn-SAT: $\Gamma = \{\substack{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0 \\ x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0\}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0, 1\}^2\}$

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ or $x_i := 1$ and force consistency again.

Both $\{0\}$ and $\{1\}$ are strong subsets.

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0, 1\}\}$.

How to find a solution:

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

Polynomial algorithm

Tractable cases on $\{0,1\}$

- Horn-SAT: $\Gamma = \{ {}^{x_1 = 1 \vee x_2 = 0 \vee \cdots \vee x_n = 0}_{x_1 = 0 \vee \cdots \vee x_n = 0} \mid n \geq 0 \}$.

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ and force consistency again.

$\{0\}$ is a strong subset.

- 2-SAT: $\Gamma = \{R \mid R \subseteq \{0,1\}^2\}$

How to solve: force local consistency.

How to find a solution: set $x_i := 0$ or $x_i := 1$ and force consistency again.

Both $\{0\}$ and $\{1\}$ are strong subsets.

- System of linear equations: $\Gamma = \{a_1 x_1 + \cdots + a_n x_n = a_0 \mid a_0, \ldots, a_n \in \{0,1\}\}$.

How to find a solution: Gaussian elimination.

$B \subseteq D_i$ is called strong if reducing $x_i \in B$ does not kill all the solution for any consistent enough instance of CSP.

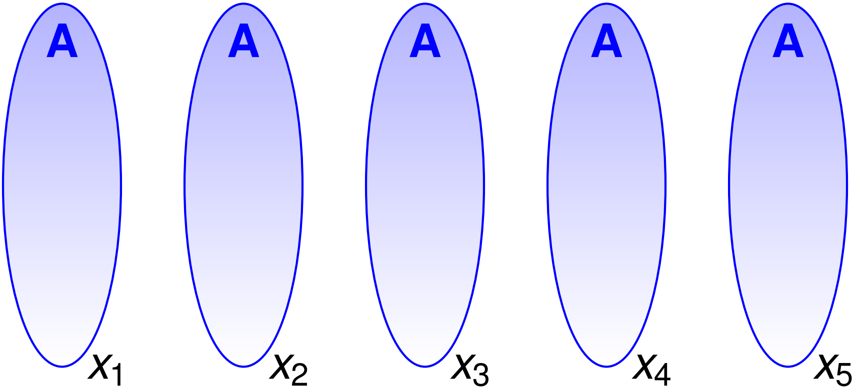
Polynomial algorithm

Polynomial algorithm

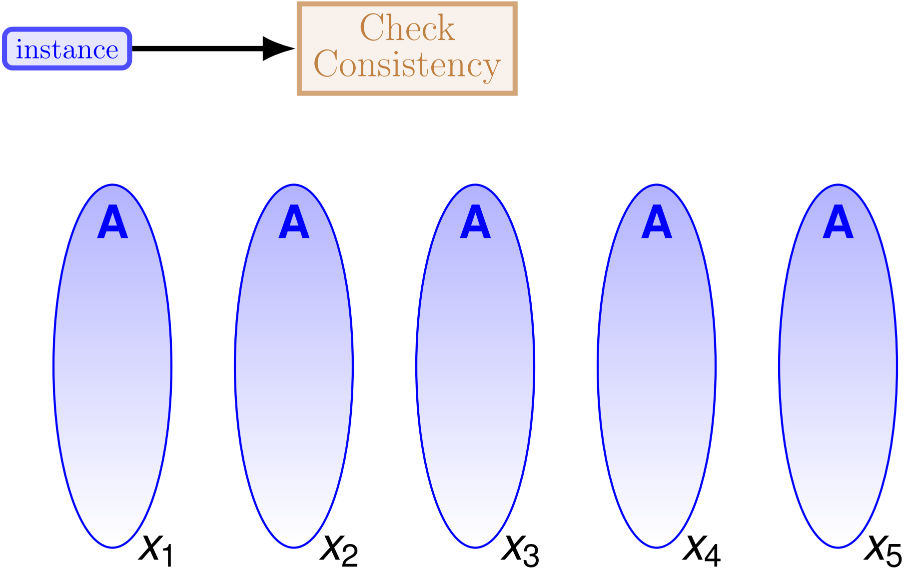
$\boxed{\text{instance}}$

Polynomial algorithm

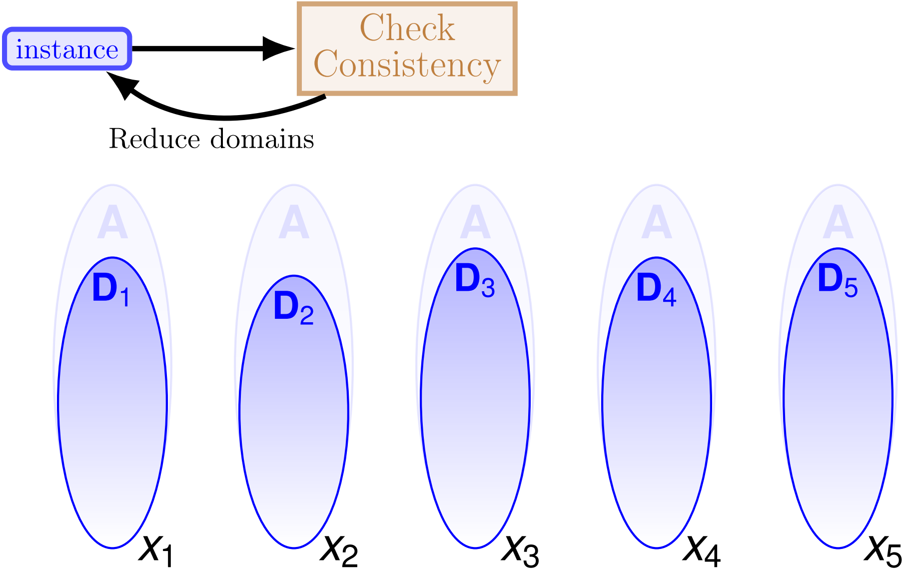
instance



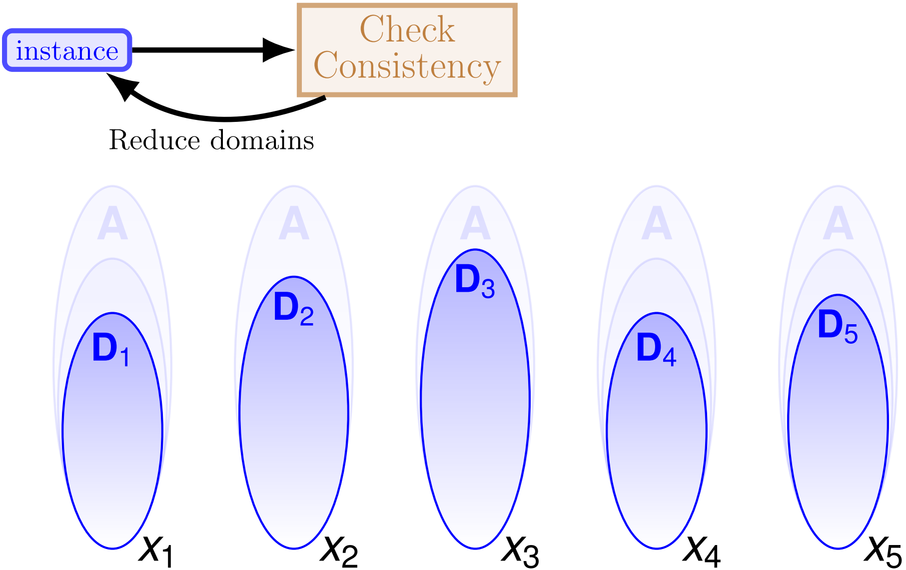
Polynomial algorithm



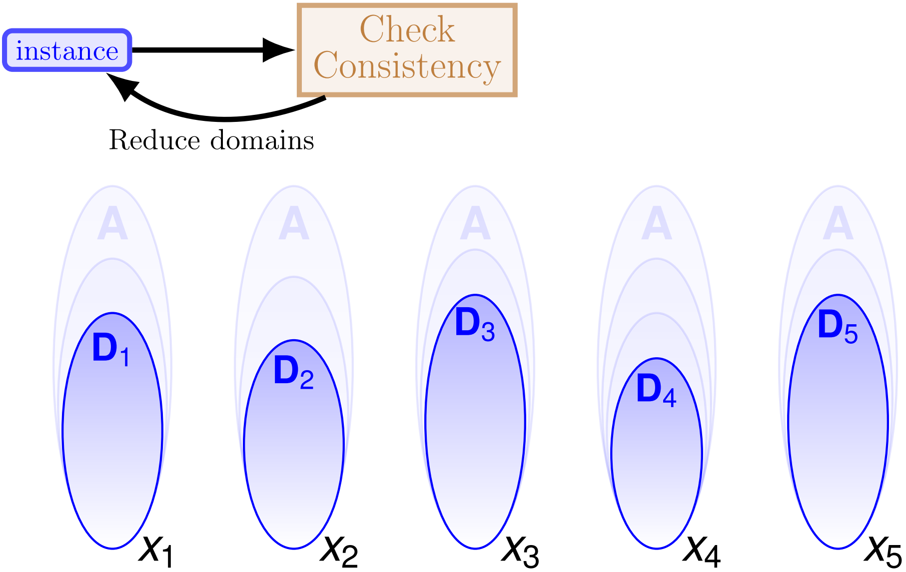
Polynomial algorithm



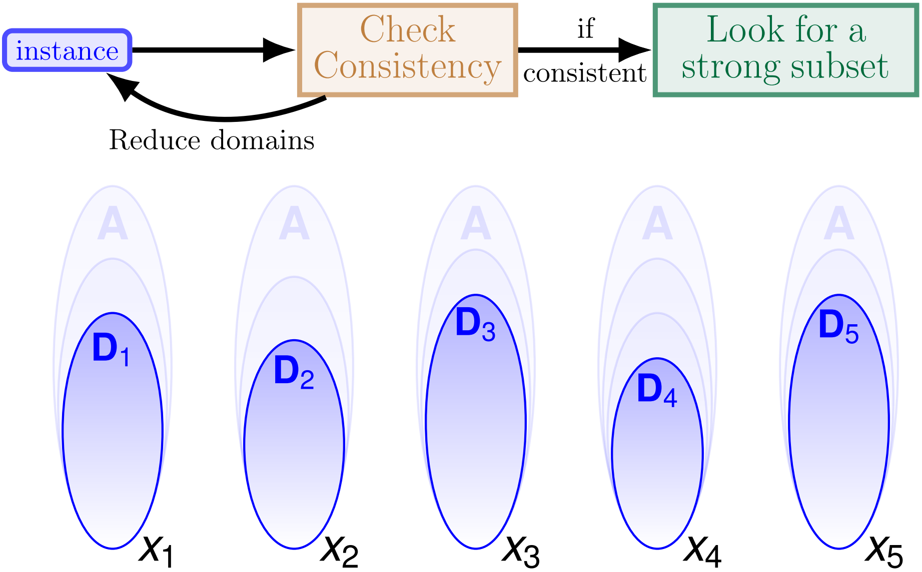
Polynomial algorithm



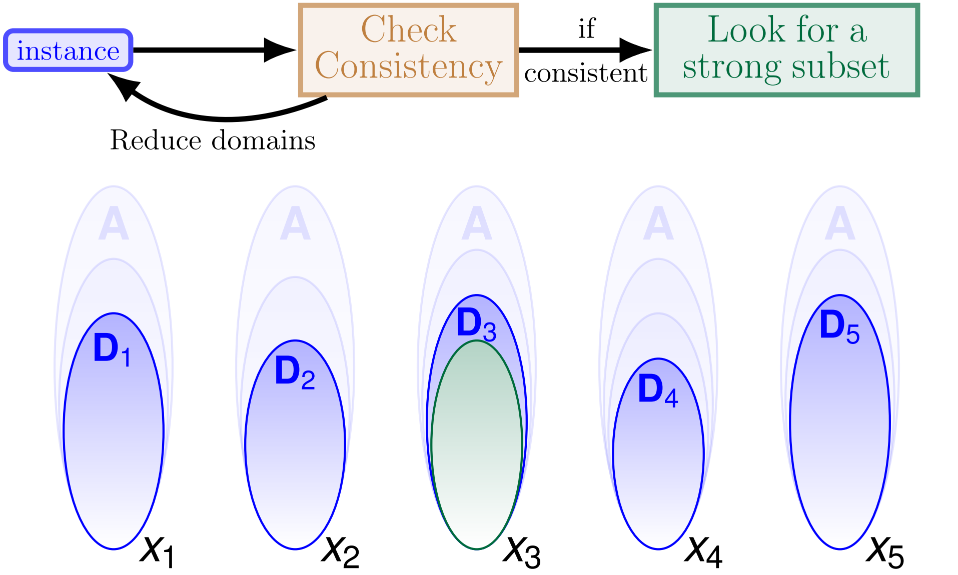
Polynomial algorithm

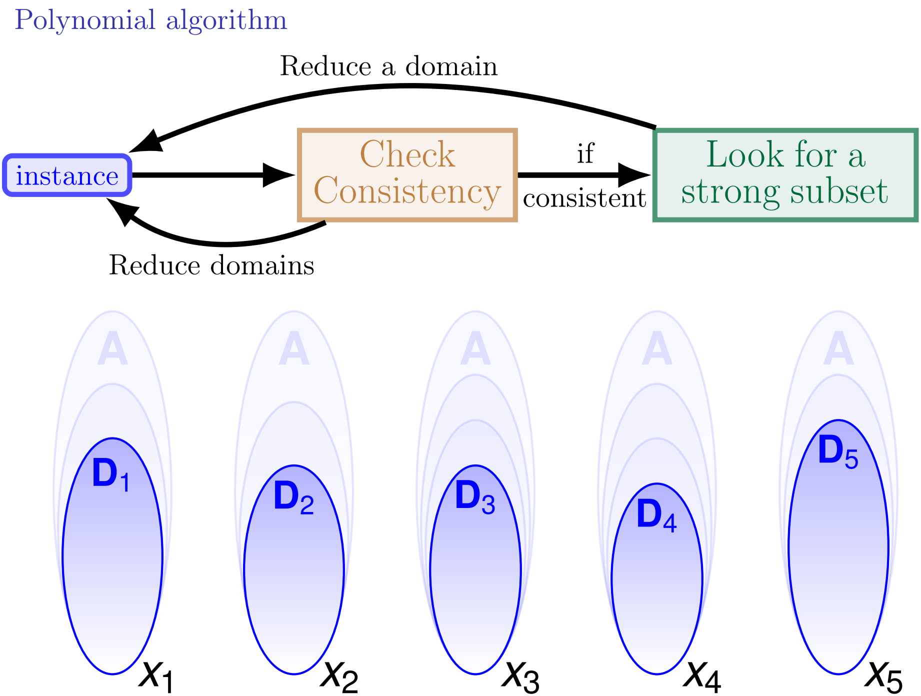


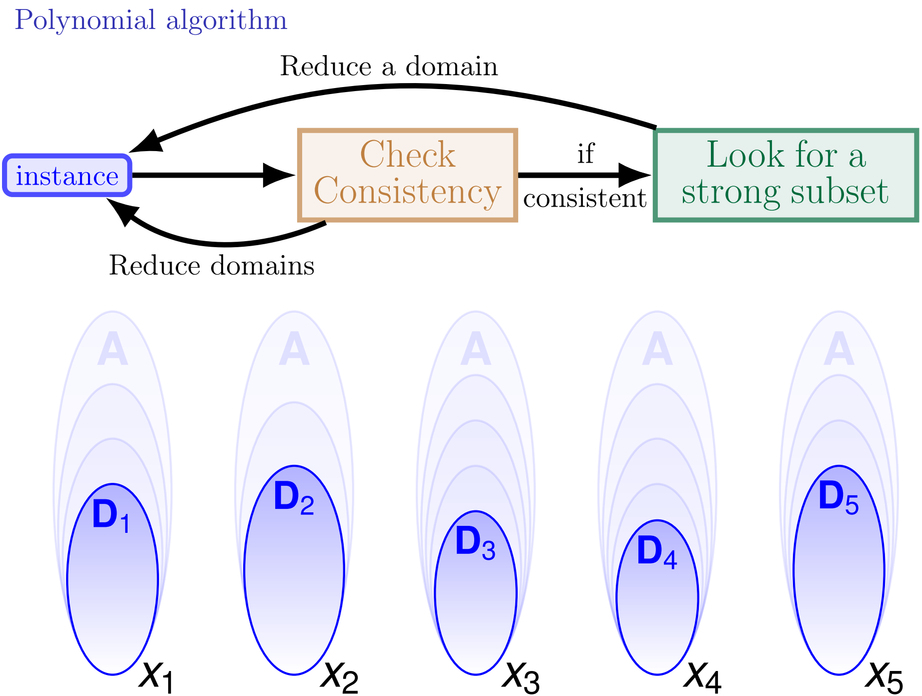
Polynomial algorithm



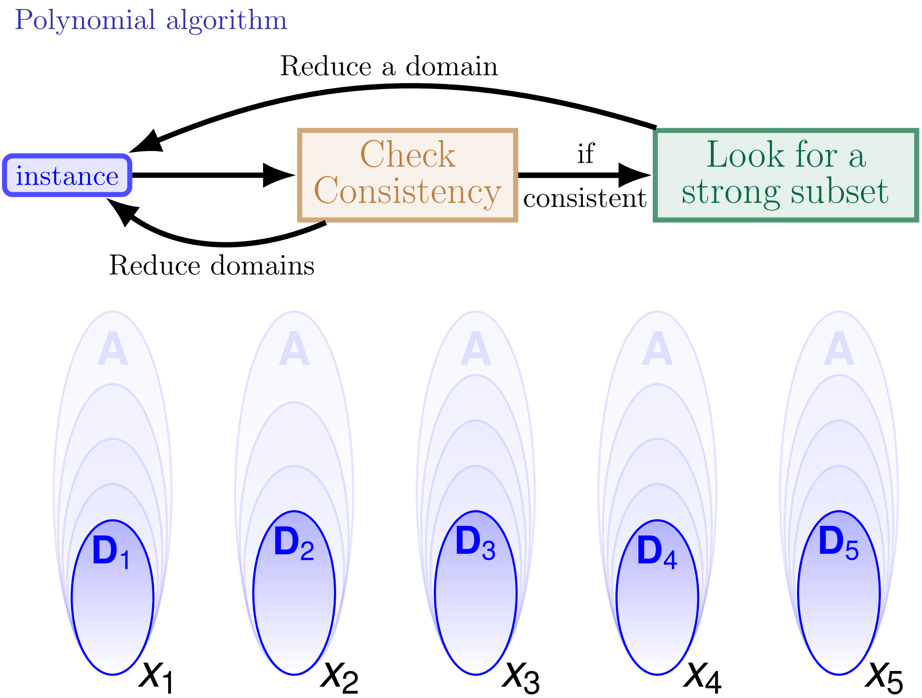
Polynomial algorithm

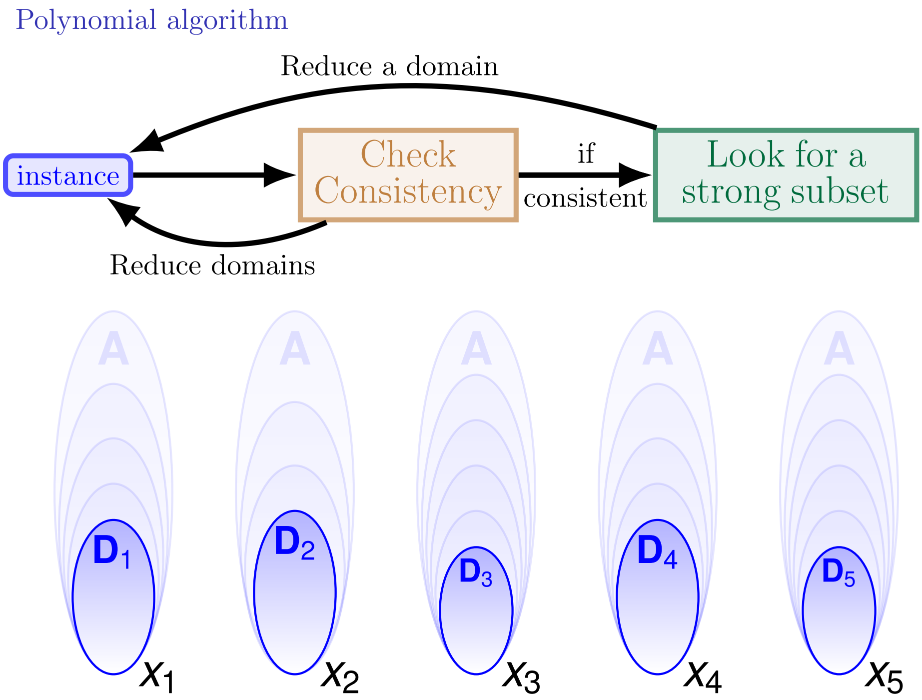


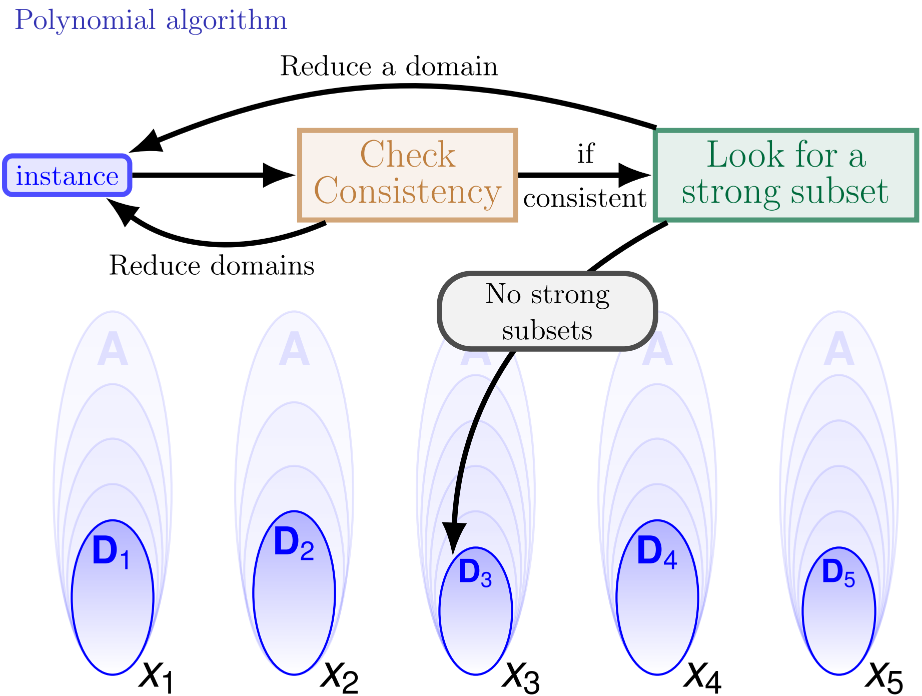

Polynomial algorithm
Reduce a domain
instance
Check Consistency
if consistent
Look for a strong subset
Reduce domains
A
D1
D2
D3
D4
D5
x1
x2
x3
x4
x5



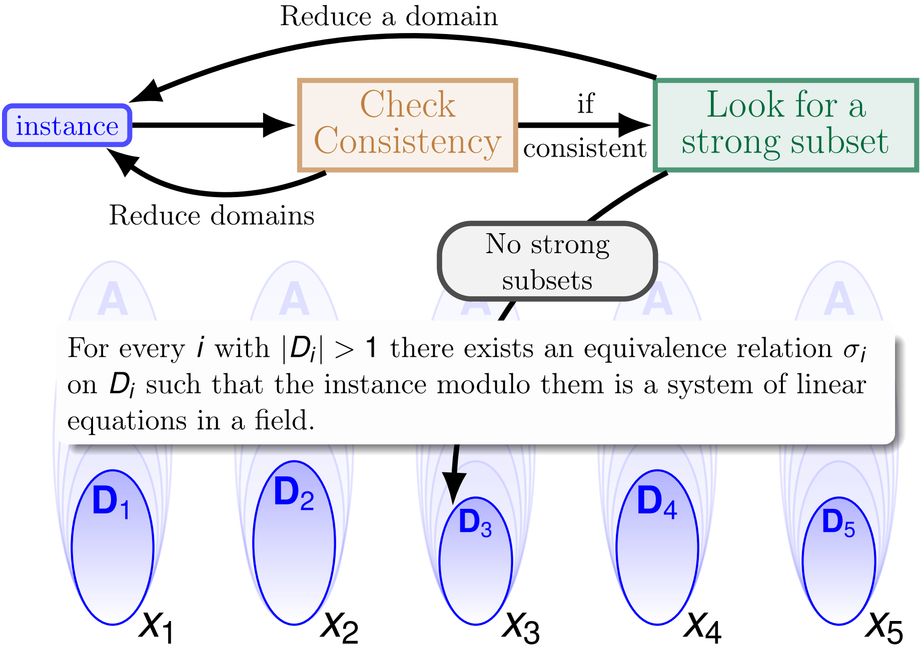
Polynomial algorithm
Reduce a domain
instance
Check Consistency
if consistent
Look for a strong subset
Reduce domains
A
A
A
A
A
D_1
D_2
D_3
D_4
D_5
x_1
x_2
x_3
x_4
x_5


Polynomial algorithm
Reduce a domain
instance
Check Consistency
if consistent
Look for a strong subset
Reduce domains
A
A
A
A
A
D_1
D_2
D_3
D_4
D_5
x_1
x_2
x_3
x_4
x_5


Polynomial algorithm
Reduce a domain
instance
Check Consistency
if consistent
Look for a strong subset
Reduce domains
A
A
A
A
A
D1
D2
D3
D4
D5
x1
x2
x3
x4
x5

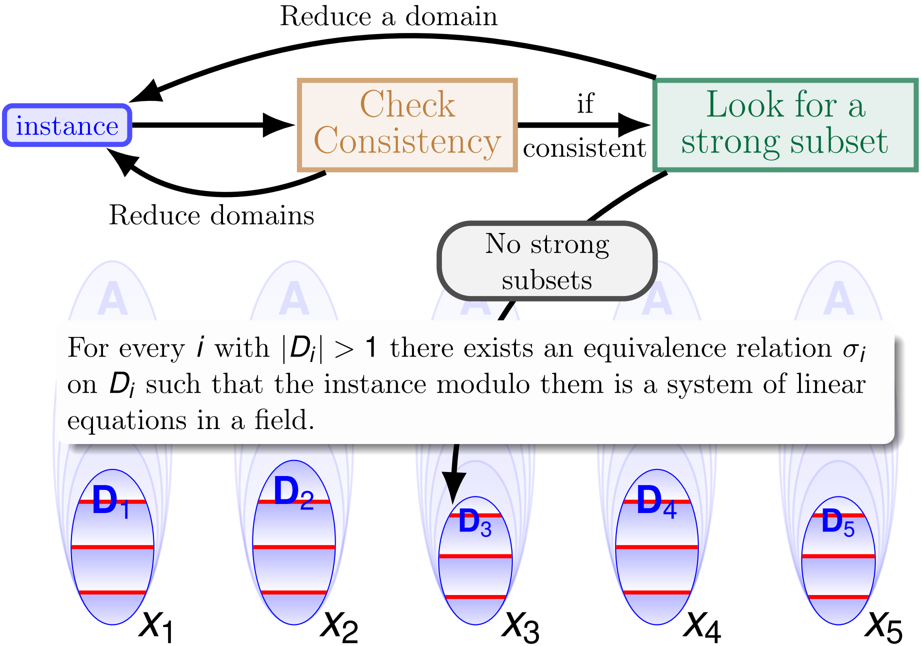

Polynomial algorithm
Reduce a domain
instance
Check Consistency
if consistent
Look for a strong subset
Reduce domains
No strong subsets
A
A
A
A
A
D_1
D_2
D_3
D_4
D_5
x_1
x_2
x_3
x_4
x_5

Polynomial algorithm



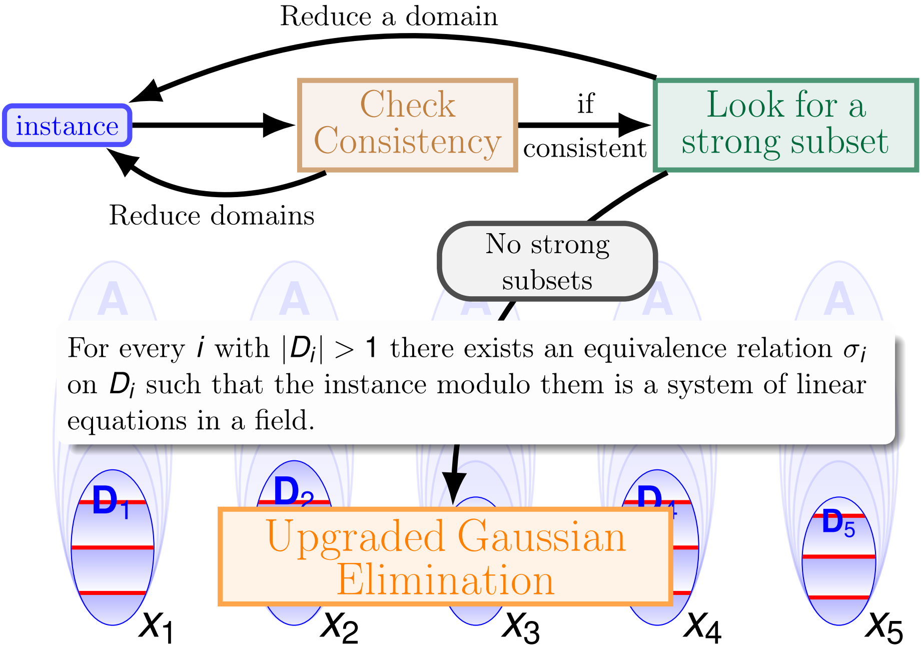
For every i with $|D_i| > 1$ there exists an equivalence relation σ_i on D_i such that the instance modulo them is a system of linear equations in a field.

Polynomial algorithm



For every i with $|D_i| > 1$ there exists an equivalence relation σ_i on D_i such that the instance modulo them is a system of linear equations in a field.

Polynomial algorithm

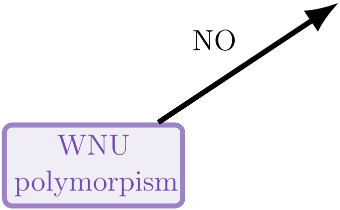


To sum up

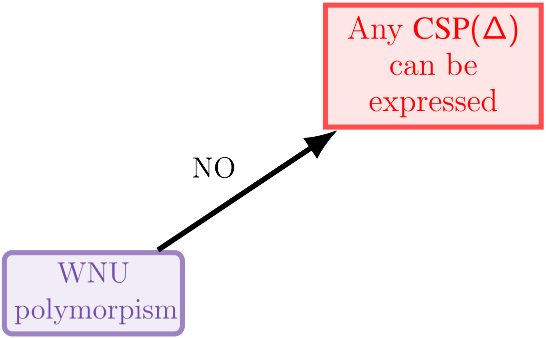
To sum up

WNU
polymorpism

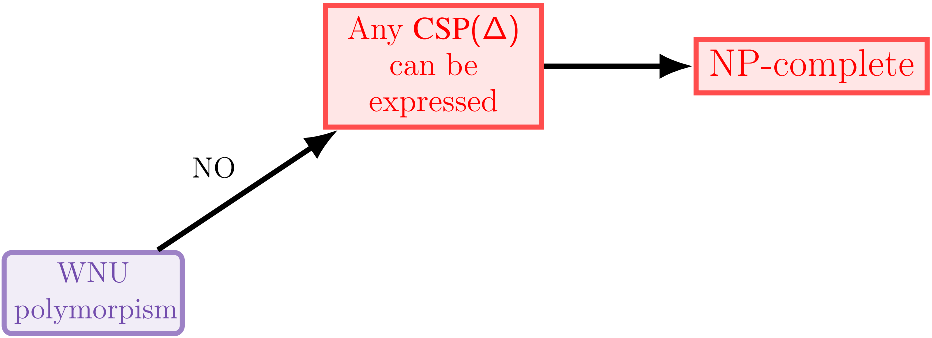
To sum up



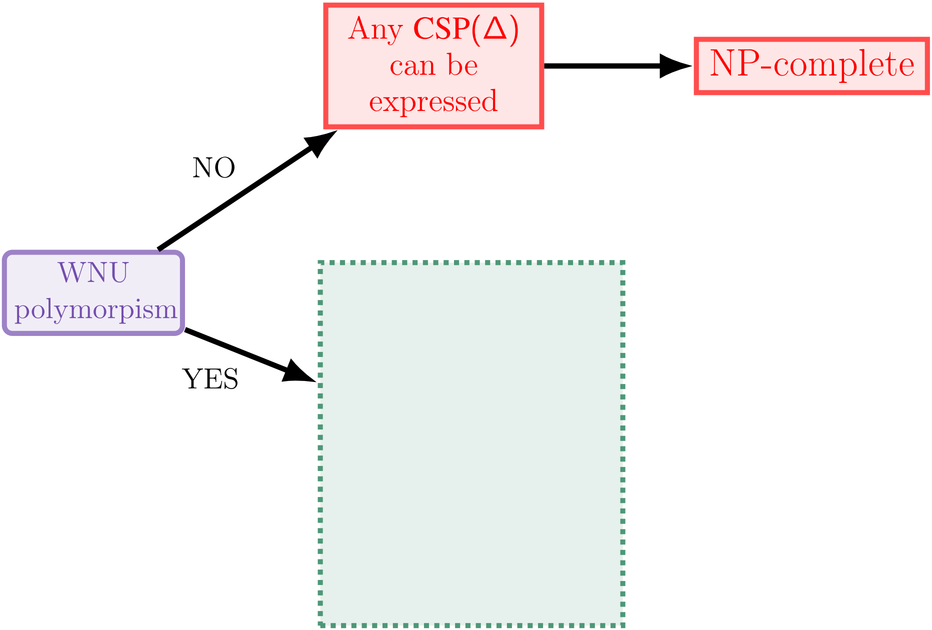
To sum up



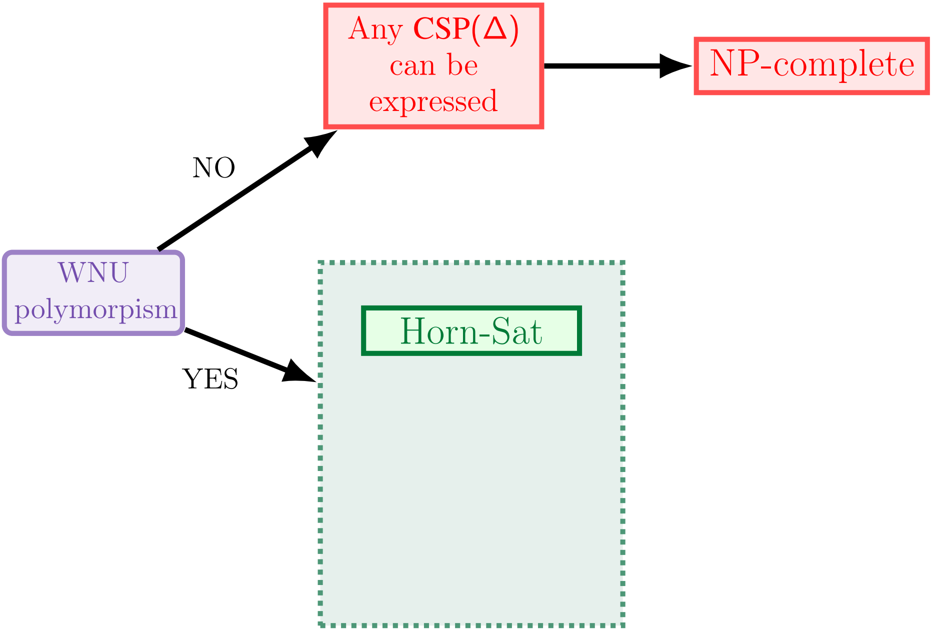
To sum up



To sum up

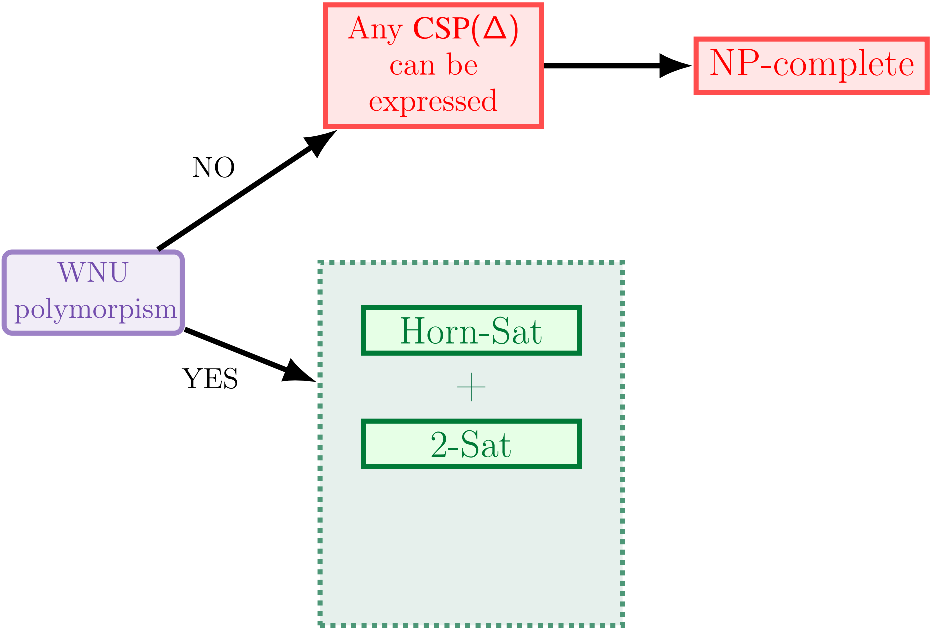

Any $\mathbf{CSP}(\Delta)$ can be expressed
NP-complete
NO
WNU polymorpism
YES

To sum up



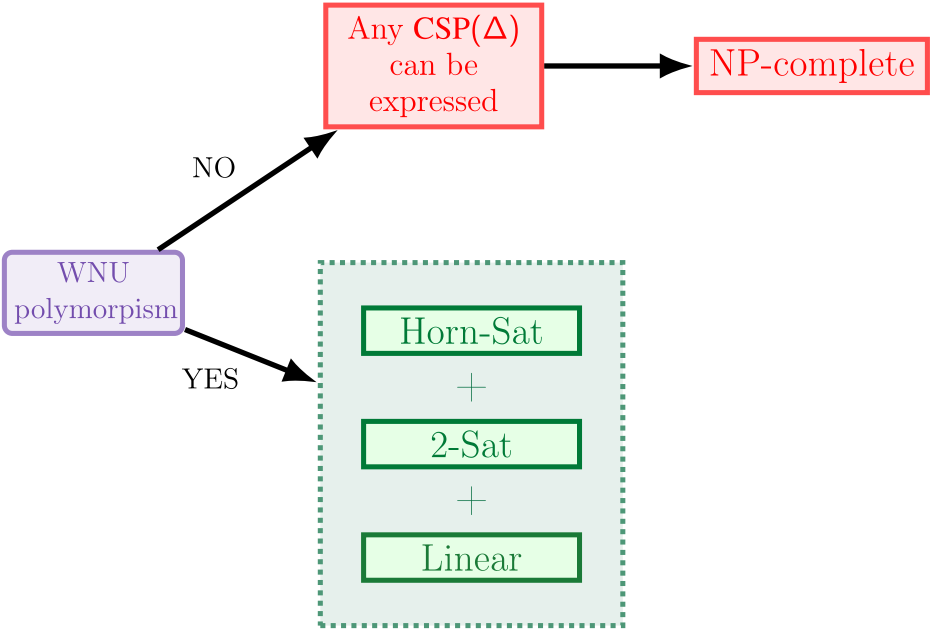
Any **CSP**(Δ) can be expressed
NP-complete
NO
WNU polymorpism
YES
Horn-Sat

To sum up



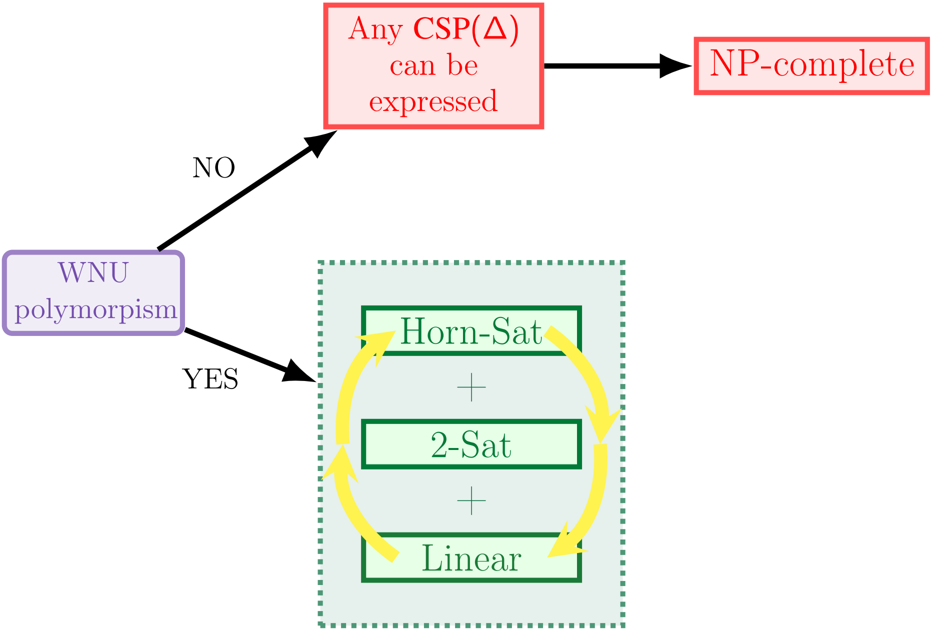
Any $\mathbf{CSP}(\Delta)$ can be expressed
NP-complete
NO
WNU polymorpism
YES
Horn-Sat
+
2-Sat

To sum up



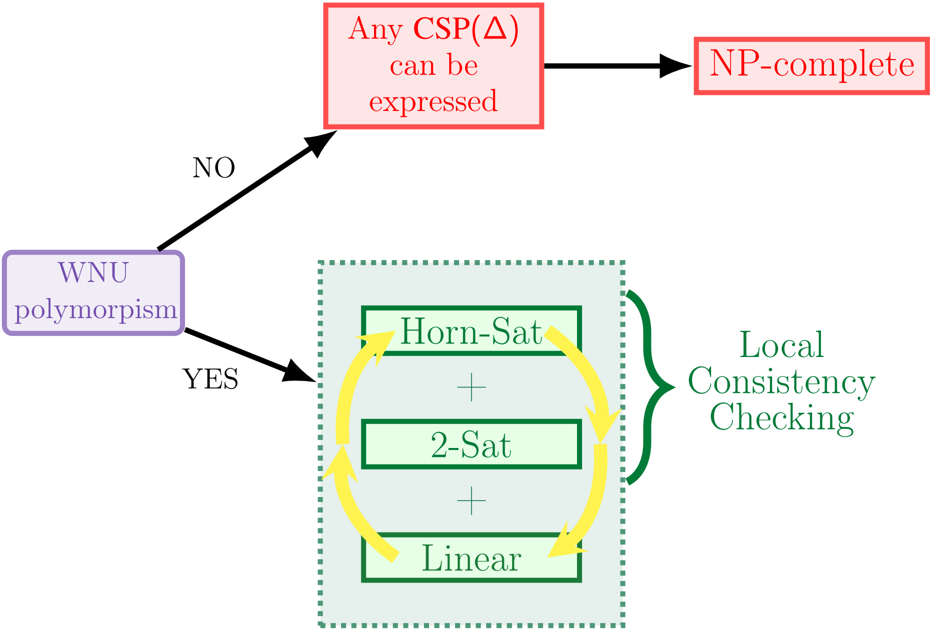
Any **CSP**(Δ) can be expressed
NP-complete
NO
WNU polymorpism
YES
Horn-Sat
+
2-Sat
+
Linear

To sum up

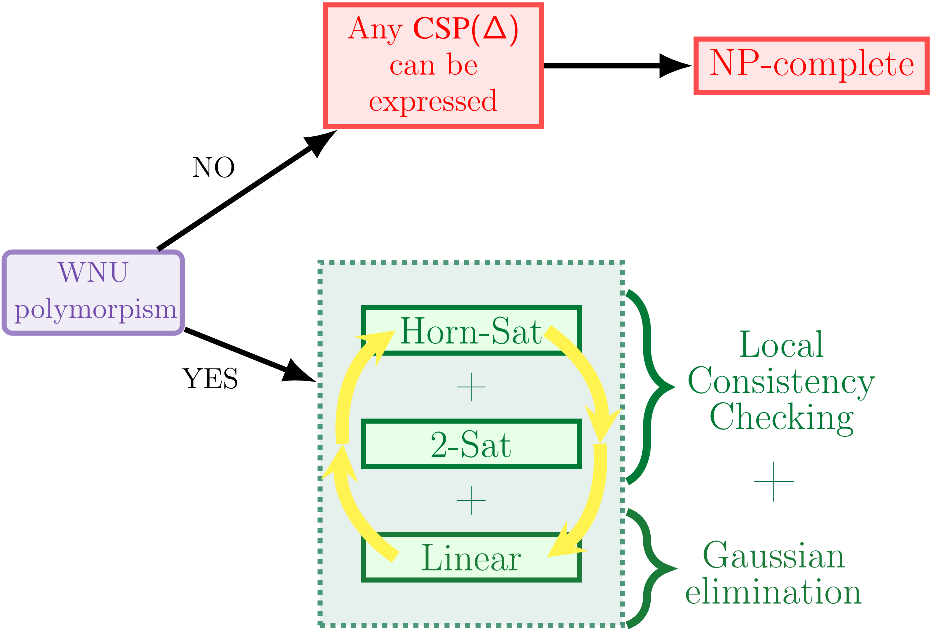


Any $\mathbf{CSP}(\Delta)$
can be
expressed
NP-complete
NO
WNU
polymorpism
YES
Horn-Sat
+
2-Sat
+
Linear

To sum up


Any **CSP(Δ)** can be expressed
NP-complete
NO
WNU polymorpism
YES
Horn-Sat
+
2-Sat
+
Linear
Local Consistency Checking

To sum up



Any CSP(Δ) can be expressed
NP-complete
NO
WNU polymorpism
YES
Horn-Sat
+
2-Sat
+
Linear
Local Consistency Checking
+
Gaussian elimination

		CSP					
Domain	finite						
	infinite						

		CSP					
Domain	finite						
	infinite						

		CSP					
Domain	finite						
	infinite						

Full classification

		CSP					
Domain	finite						
	infinite						

Full classification

Infinite Domain CSP

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

Infinite Domain CSP

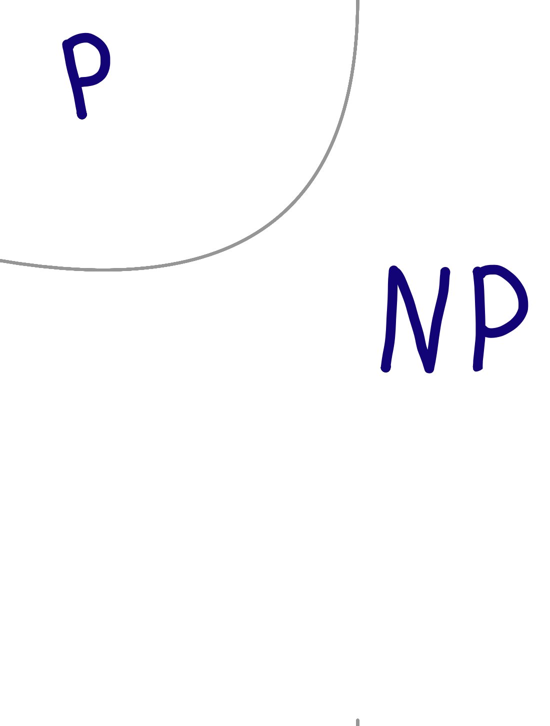
Γ is a set of relations on $\mathbb{Q}$.

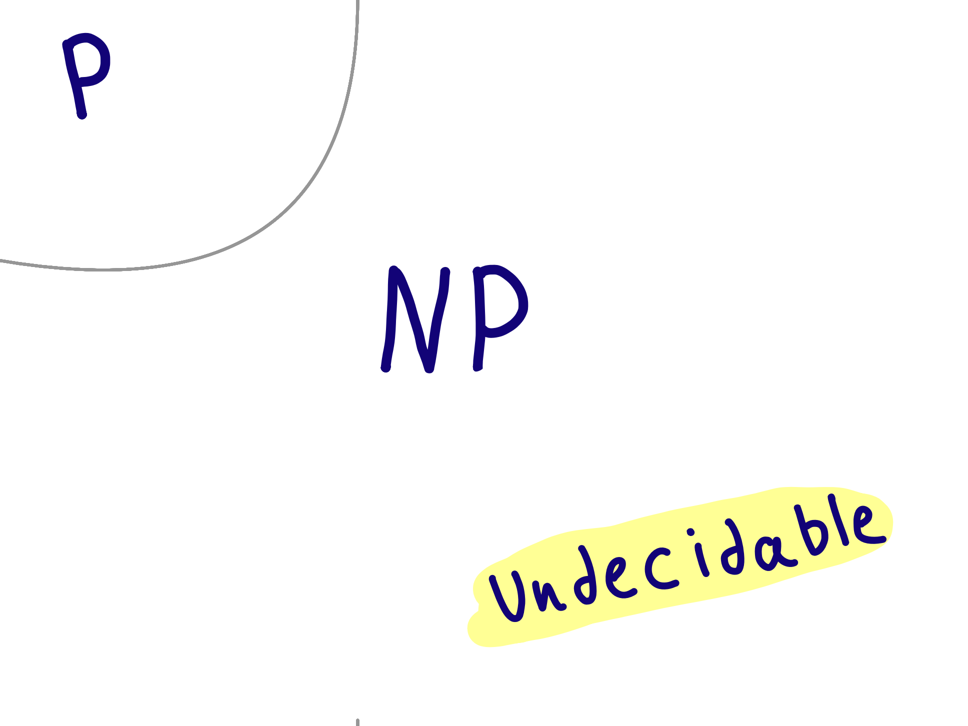
CSP(Γ)

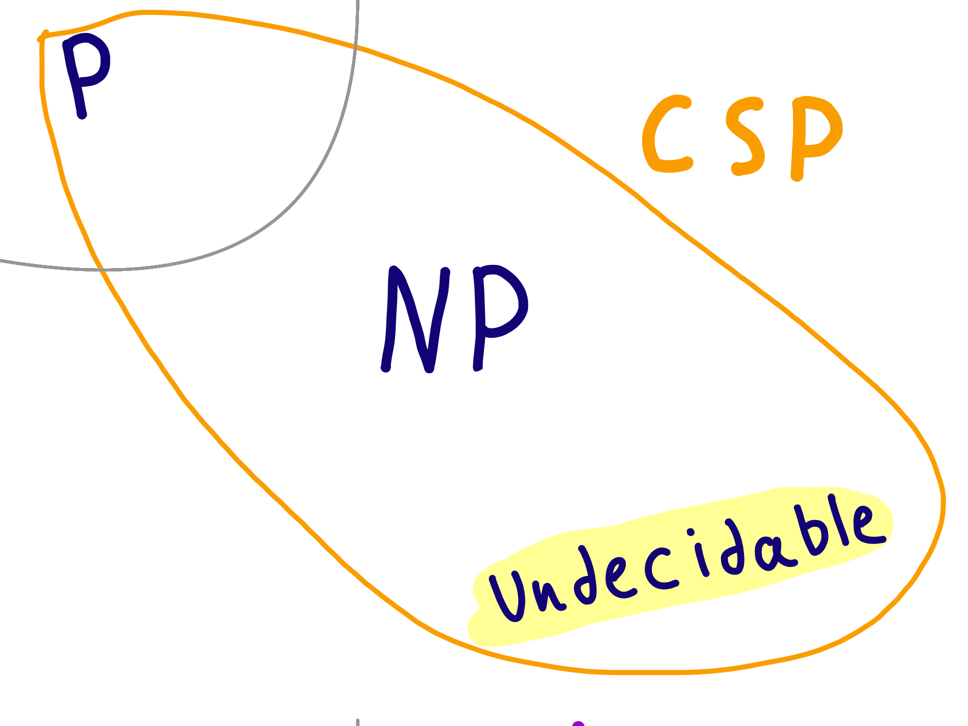
Given: a conjunction of relations, i.e. a formula

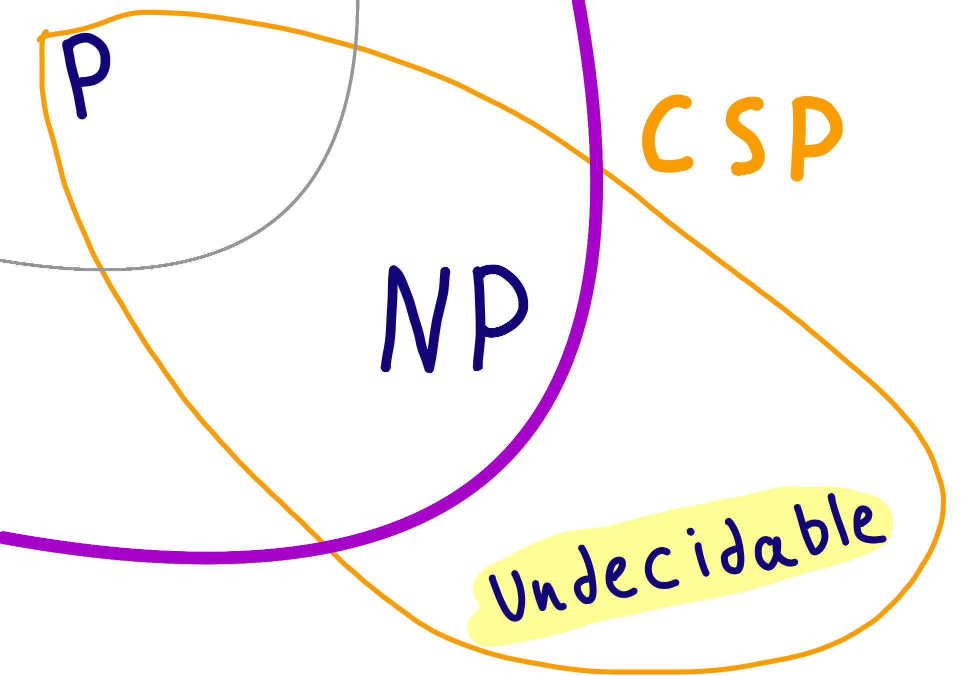
$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.


P
NP


P
NP
Undecidable


P
CSP
NP
Undecidable


P
NP
CSP
Undecidable

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

CSP instances:

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_1 < x_3$,

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_1 < x_3$, has a solution

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_1 < x_3$, has a solution
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$,

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_1 < x_3$, has a solution
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$, has no solutions

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$)

CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_1 < x_3$, has a solution
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$, has no solutions
The instance has a solution IFF there is no oriented cycle.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.

CSP instances:
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_1 < x_3$, has a solution
$x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$, has no solutions
The instance has a solution IFF there is no oriented cycle.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$)

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$)

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$) is in P.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$) is in P.
4. CSP($\{x = y < z \vee x = z < y\}$)

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$) is in P.
4. CSP($\{x = y < z \vee x = z < y\}$) is NP-complete.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$) is in P.
4. CSP($\{x = y < z \vee x = z < y\}$) is NP-complete.
5. CSP($\{x = y < z \vee x = z < y \vee y = z < x, x = y + 1\}$)

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$) is in P.
4. CSP($\{x = y < z \vee x = z < y\}$) is NP-complete.
5. CSP($\{x = y < z \vee x = z < y \vee y = z < x, x = y + 1\}$) is NP-complete.

Infinite Domain CSP

Γ is a set of relations on $\mathbb{Q}$.

CSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}}),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether the formula is satisfiable.

Examples

1. CSP($\{x < y\}$) is in P.
2. CSP($\{x < y < z \vee z < y < x\}$) is NP-complete.
3. CSP($\{x = y < z \vee x = z < y \vee y = z < x\}$) is in P.
4. CSP($\{x = y < z \vee x = z < y\}$) is NP-complete.
5. CSP($\{x = y < z \vee x = z < y \vee y = z < x, x = y + 1\}$) is NP-complete.

Classification for temporal constraint languages [Bodirsky, Kára, 2008]

A full classification of the complexity for constraint languages admitting a first-order definition in$(\mathbb{Q}; <)$ (P vs NP-complete).

		CSP					
Domain	finite						
	infinite						

Full classification

		CSP					
Domain	finite						
	infinite						

Full classification

Some classifications

		CSP	Quantified CSP				
Domain	finite						
	infinite						

Full classification

Some classifications

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

CSP(Γ)

Given: a sentence

$$\exists x_1 \ldots \exists x_n \; R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it holds.

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

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.

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

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

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

QCSP(Γ)

Given: a sentence

$$\exists y_1 \forall x_1 \dots \exists y_t \forall x_t (R_1(\dots) \wedge \dots \wedge R_s(\dots)),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it holds.

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

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

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

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:

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

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

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

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

Quantified Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

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:

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

Γ is a set of relations on a finite set A.

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:

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

Γ is a set of relations on a finite set A.

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:

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

Γ is a set of relations on a finite set A.

QCSP(Γ)

Given: a sentence

$$\exists y_1 \forall x_1 \dots \exists y_t \forall x_t (R_1(\dots) \wedge \dots \wedge R_s(\dots)),$$

where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it holds.

Examples:

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

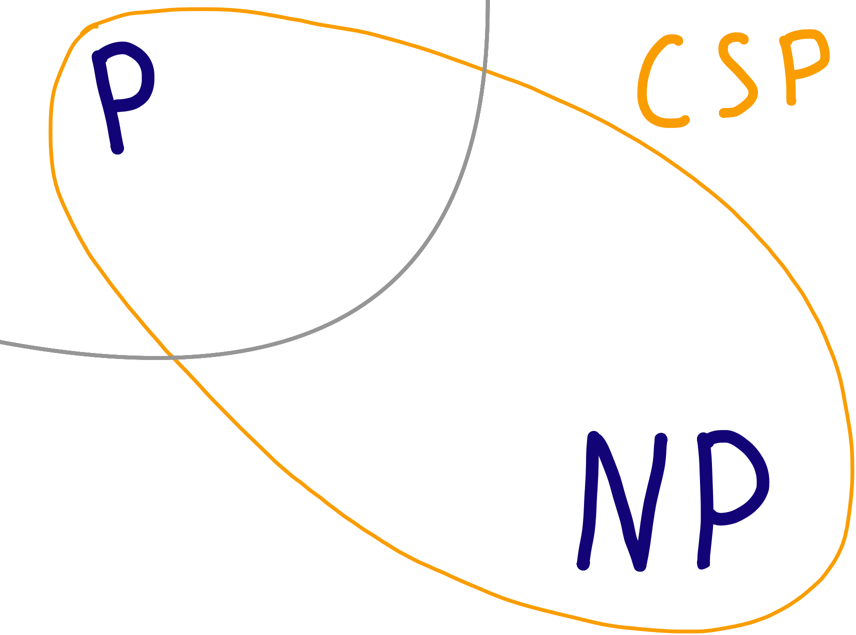
$\forall x \exists y_1 \exists y_2 (x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2)$, true

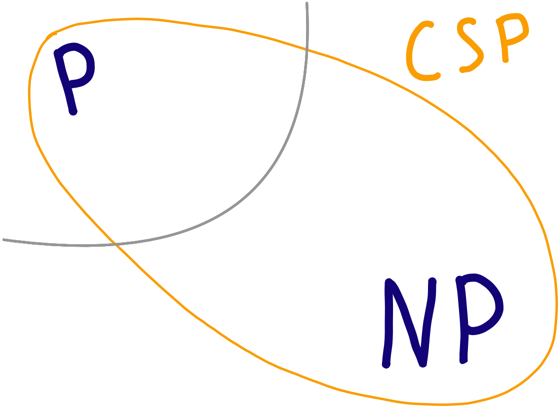
$\forall x_1 \forall x_2 \forall x_3 \exists y (x_1 \neq y \wedge x_2 \neq y \wedge x_3 \neq y)$, false

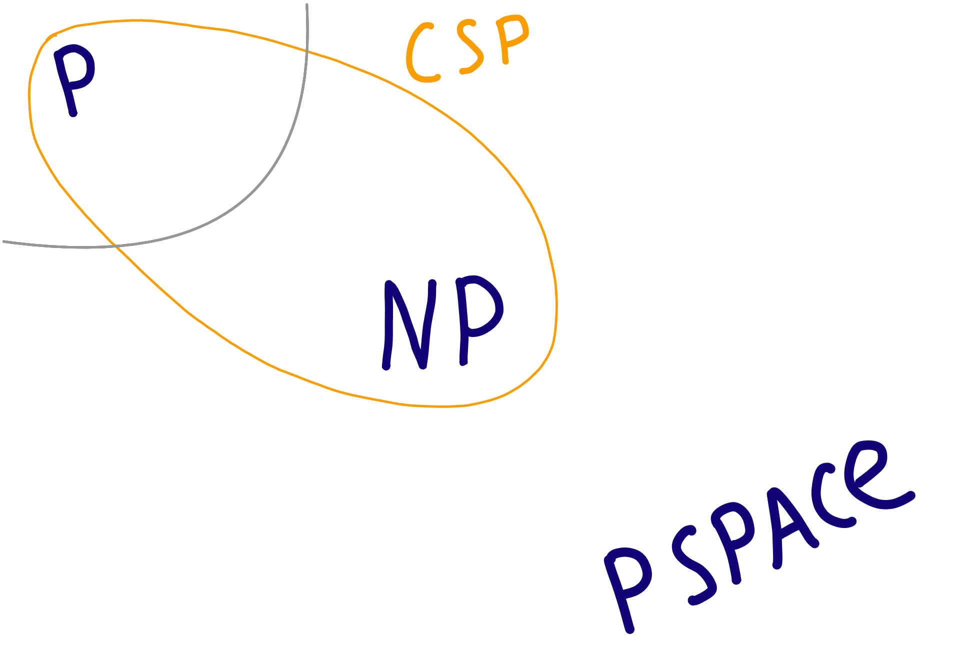
$\forall x_1 \exists y_1 \forall x_2 \exists y_2 (x_1 \neq y_1 \wedge y_1 \neq y_2 \wedge y_2 \neq x_2)$, true

Question

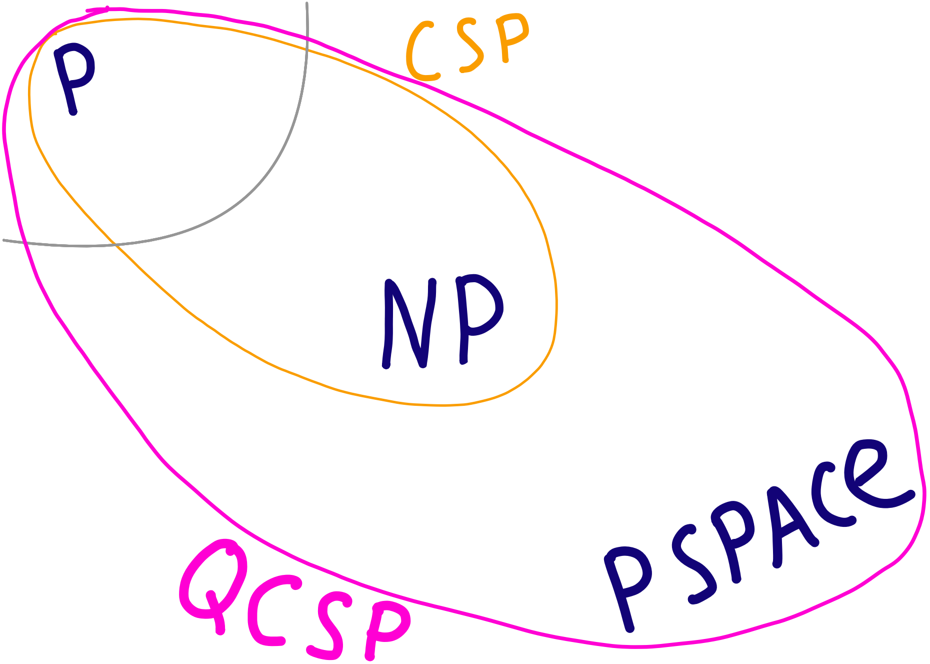
What is the complexity of QCSP(Γ) for different Γ?


CSP
P
NP


CSP
P
NP



CSP
P
NP
PSPACE



P
CSP
NP
PSPACE
QCSP

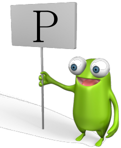
QCSP Complexity Classes

QCSP Complexity Classes

- If Γ contains all predicates then $\mathsf{QCSP}(\Gamma)$ is PSPACE-complete.

QCSP Complexity Classes

- If Γ contains all predicates then **QCSP(Γ)** is PSPACE-complete.
- If Γ consists of linear equations in a finite field then **QCSP(Γ)** is in P.



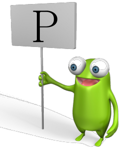
QCSP Complexity Classes

- If Γ contains all predicates then $\mathsf{QCSP}(\Gamma)$ is PSPACE-complete.
- If Γ consists of linear equations in a finite field then $\mathsf{QCSP}(\Gamma)$ is in P.

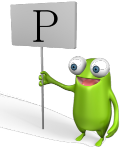
Theorem [Schaefer 1978 + Creignou et al. 2001/ Dalmau 1997.]

Suppose Γ is a constraint language on $\{0, 1\}$. Then

- $\mathsf{QCSP}(\Gamma)$ is in P if Γ is preserved by an idempotent WNU operation,
- $\mathsf{QCSP}(\Gamma)$ is PSPACE-complete otherwise.

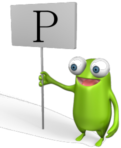


QCSP Complexity Classes



QCSP Complexity Classes

- Put $A' = A \cup \{*\}$, Γ' is Γ extended to A'. Then $\mathbf{QCSP}(\Gamma')$ is equivalent to $\mathbf{CSP}(\Gamma)$.



QCSP Complexity Classes

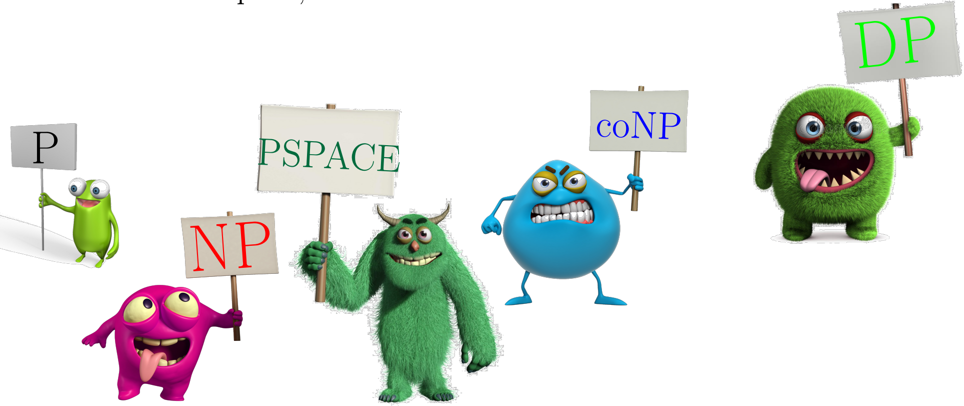
- Put $A' = A \cup \{*\}$, Γ' is Γ extended to A'. Then $\mathbf{QCSP}(\Gamma')$ is equivalent to $\mathbf{CSP}(\Gamma)$.

QCSP Complexity Classes

- Put $A' = A \cup \{*\}$, Γ' is Γ extended to A'. Then $\mathsf{QCSP}(\Gamma')$ is equivalent to $\mathsf{CSP}(\Gamma)$.
- there exists Γ on a 3-element domain such that $\mathsf{QCSP}(\Gamma)$ is coNP-complete.

QCSP Complexity Classes

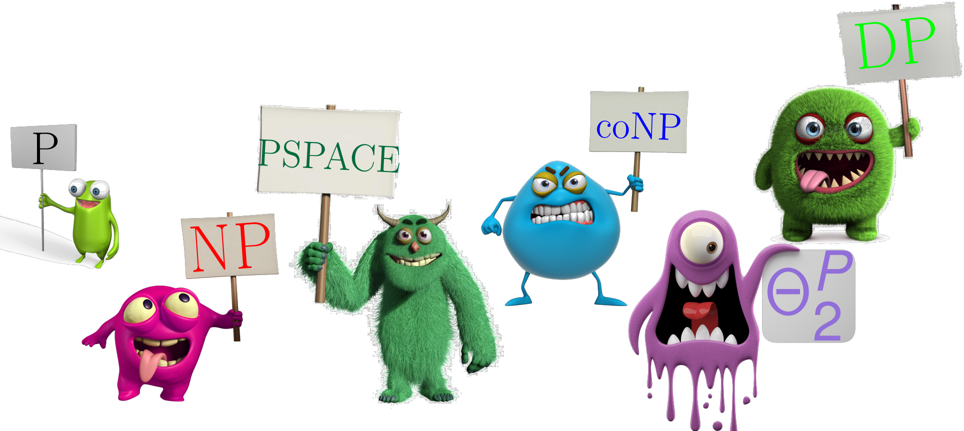
- Put $A' = A \cup \{*\}$, Γ' is Γ extended to A'. Then $\mathbf{QCSP}(\Gamma')$ is equivalent to $\mathbf{CSP}(\Gamma)$.
- there exists Γ on a 3-element domain such that $\mathbf{QCSP}(\Gamma)$ is coNP-complete.
- there exists Γ on a 4-element domain such that $\mathbf{QCSP}(\Gamma)$ is DP-complete, where DP $=$ NP $\wedge$ coNP.



QCSP Complexity Classes

- Put $A' = A \cup \{*\}$, Γ' is Γ extended to A'. Then $\mathrm{QCSP}(\Gamma')$ is equivalent to $\mathrm{CSP}(\Gamma)$.
- there exists Γ on a 3-element domain such that $\mathrm{QCSP}(\Gamma)$ is coNP-complete.
- there exists Γ on a 4-element domain such that $\mathrm{QCSP}(\Gamma)$ is DP-complete, where DP $=$ NP $\wedge$ coNP.
- there exists Γ on a 10-element domain such that $\mathrm{QCSP}(\Gamma)$ is Θ_2^P-complete.

QCSP Complexity Classes



QCSP Complexity Classes

Theorem [Zhuk, Martin, 2019]

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.

QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

PSPACE

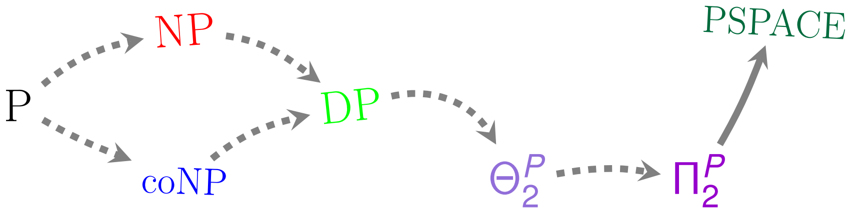
Π_2^P

QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

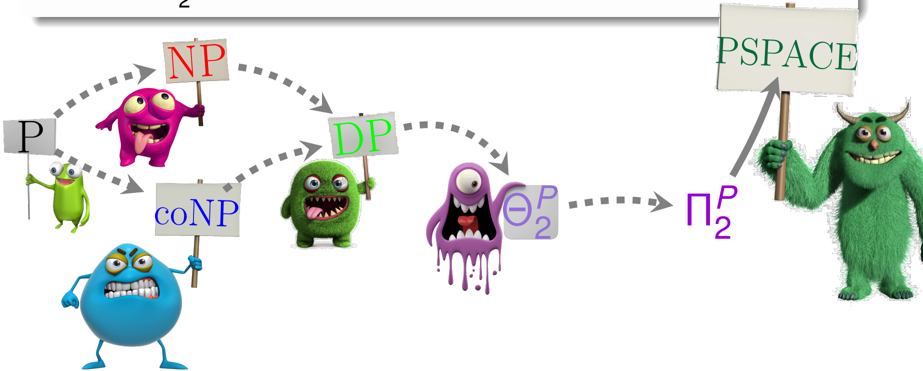


QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

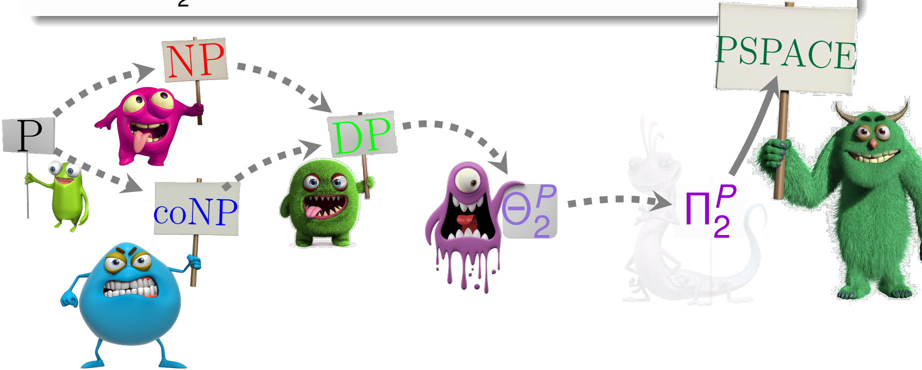


QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

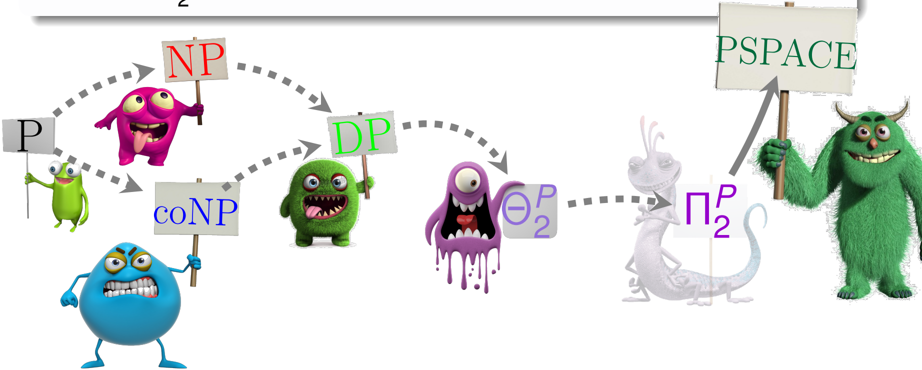


QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

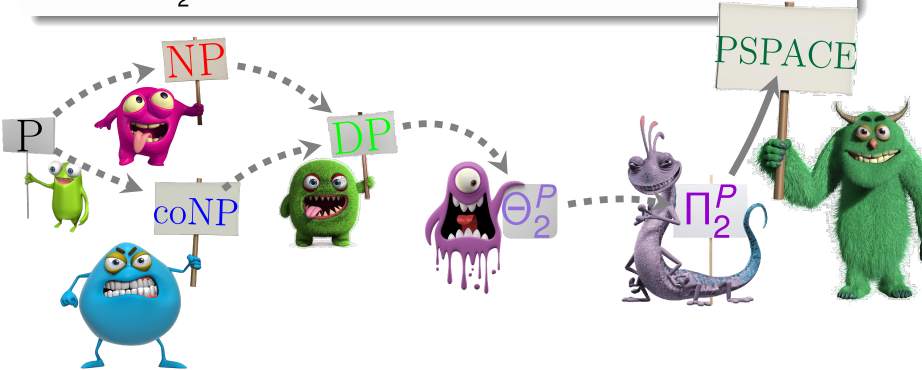


QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.

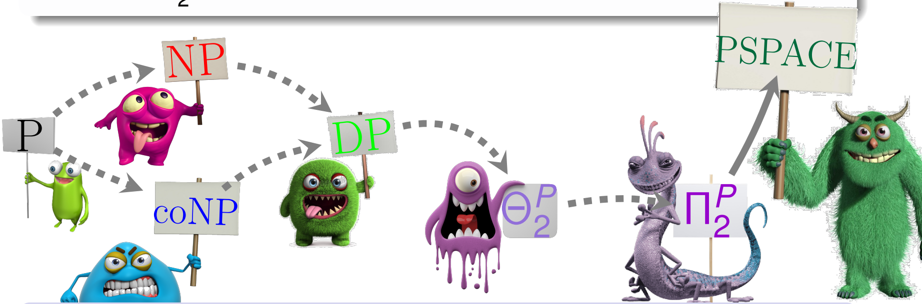


QCSP Complexity Classes

Theorem [Zhuk, 2024]

$\mathrm{QCSP}(\Gamma)$

- is either PSpace-complete,
- or in Π_2^P.



Lemma [Zhuk, 2024]

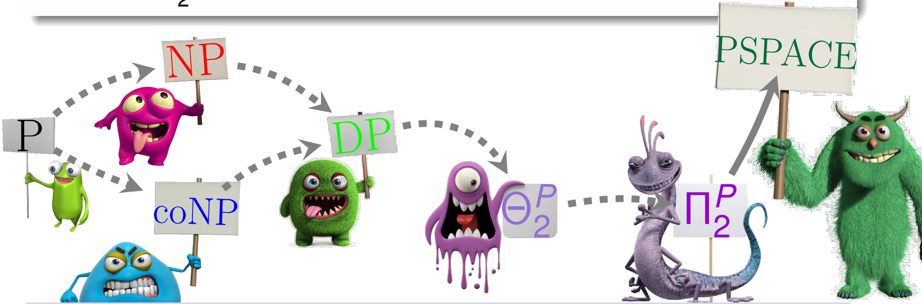
There exists Γ on a 6-element set such that $\mathrm{QCSP}(\Gamma)$ is Π_2^P-complete.

QCSP Complexity Classes

Theorem [Zhuk, 2024]

QCSP(Γ)

- is either PSpace-complete,
- or in Π_2^P.



Lemma [Zhuk, 2024]

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

Are there any other complexity classes?

		CSP	Quantified CSP				
Domain	finite						
	infinite						

Full classification

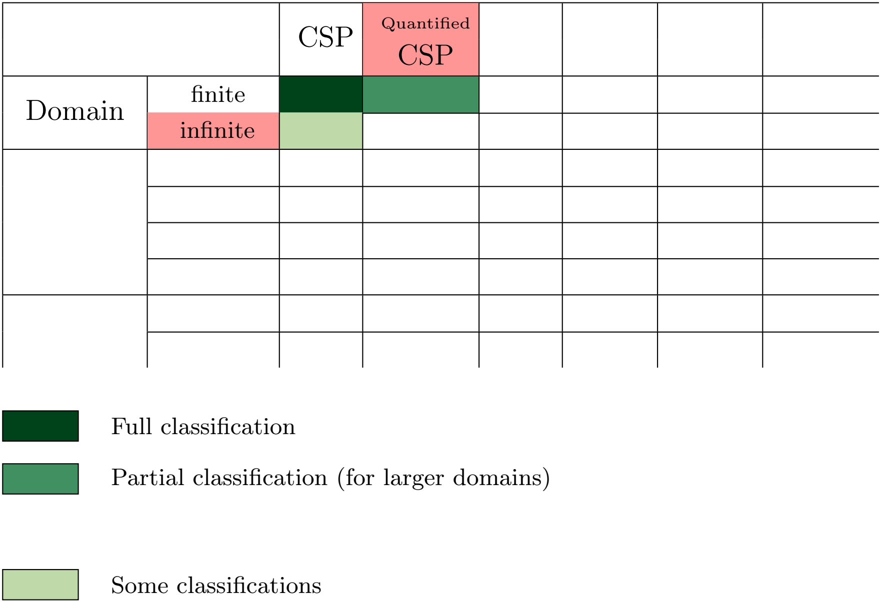
Some classifications

		CSP	Quantified CSP				
Domain	finite						
	infinite						

Full classification

Partial classification (for larger domains)

Some classifications


CSP
Quantified CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications

Infinite Domain QCSP

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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.

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$)

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

QCSP(Γ)

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

Examples

1. QCSP($\{x = y\}$)

QCSP instances:

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$)

QCSP instances:
$\forall x_1 \exists x_2 \exists x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$,

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$)

QCSP instances:
$\forall x_1 \exists x_2 \exists x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, True

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

QCSP(Γ)

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

Examples

1. QCSP($\{x = y\}$)

QCSP instances:
$\forall x_1 \exists x_2 \exists x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, True
$\forall x_1 \forall x_4 \exists x_2 \exists x_3 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$,

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$)

QCSP instances:
$\forall x_1 \exists x_2 \exists x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, True
$\forall x_1 \forall x_4 \exists x_2 \exists x_3 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, False

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$) is in P.

QCSP instances:
$\forall x_1 \exists x_2 \exists x_3 \exists x_4 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, True
$\forall x_1 \forall x_4 \exists x_2 \exists x_3 (x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4)$, False

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

QCSP(Γ)

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

Examples

1. QCSP($\{x = y\}$) is in P.

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$) is in P.
2. QCSP($\{x = y \vee y = z\}$) is NP-complete.

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$) is in P.
2. QCSP($\{x = y \vee y = z\}$) is NP-complete.
3. QCSP($\{x = y \rightarrow z = t\}$) is PSPACE-complete [Bodirsky, Chen, 2010].

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

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

1. QCSP($\{x = y\}$) is in P.
2. QCSP($\{x = y \vee y = z\}$) is NP-complete.
3. QCSP($\{x = y \to z = t\}$) is PSPACE-complete [Bodirsky, Chen, 2010].
4. QCSP($\{x = y \to y = z\}$) is PSPACE-complete [Zhuk, Martin, 2021].

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

QCSP(Γ)

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

Examples

1. QCSP($\{x = y\}$) is in P.
2. QCSP($\{x = y \vee y = z\}$) is NP-complete.
3. QCSP($\{x = y \rightarrow z = t\}$) is PSPACE-complete [Bodirsky, Chen, 2010].
4. QCSP($\{x = y \rightarrow y = z\}$) is PSPACE-complete [Zhuk, Martin, 2021].

Classification for equality constraints [Bodirsky, Chen, 2010 + Zhuk, Martin, 2021]

A full classification of the complexity for constraint languages whose relations are boolean combinations of equalities. (P, NP-complete, PSPACE-complete)

Infinite Domain QCSP

Γ is a set of relations on $\mathbb{Q}$.

QCSP(Γ)

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

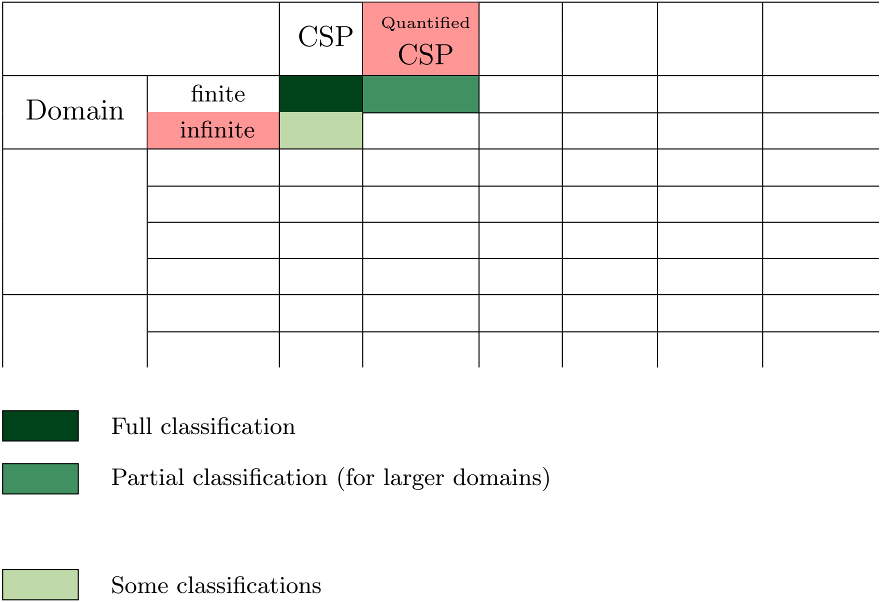
Examples

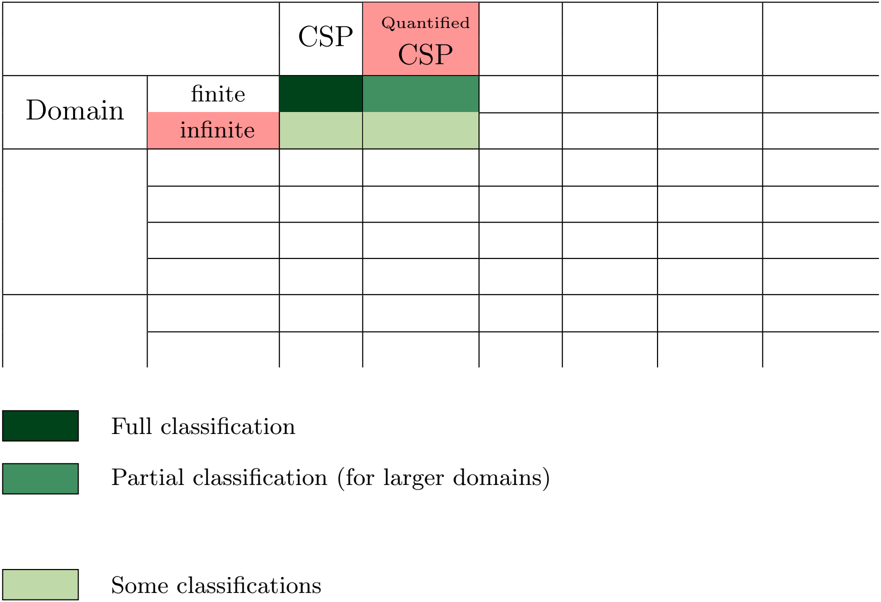
1. QCSP($\{x = y\}$) is in P.
2. QCSP($\{x = y \vee y = z\}$) is NP-complete.
3. QCSP($\{x = y \rightarrow z = t\}$) is PSPACE-complete [Bodirsky, Chen, 2010].
4. QCSP($\{x = y \rightarrow y = z\}$) is PSPACE-complete [Zhuk, Martin, 2021].

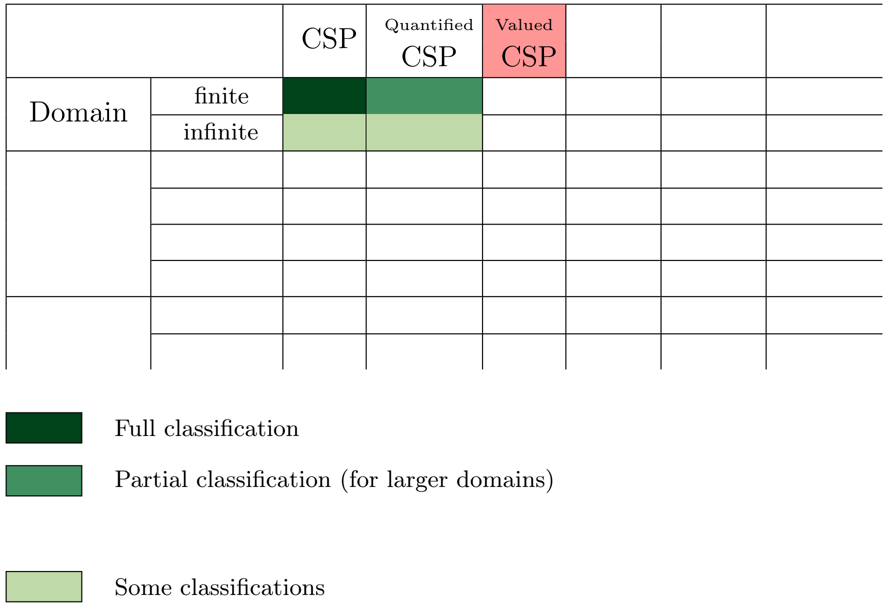
Classification for equality constraints [Bodirsky, Chen, 2010 + Zhuk, Martin, 2021]

A full classification of the complexity for constraint languages whose relations are boolean combinations of equalities. (P, NP-complete, PSPACE-complete)

What is the complexity of QCSP($\{x < y \vee y < z\}$)?


CSP
Quantified CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications


CSP
Quantified CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications


CSP
Quantified CSP
Valued CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications

Valued Constraint Satisfaction Problem

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

VCSP(Γ)

Given: a threshold T and a sum $f_1(\dots) + f_2(\dots) + \cdots + f_s(\dots)$, where $f_1, \dots, f_s \in \Gamma$.
Decide: whether $f_1(\dots) + f_2(\dots) + \cdots + f_s(\dots) < T$ is satisfiable.

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

VCSP(Γ)

Given: a threshold T and a sum $f_1(\dots) + f_2(\dots) + \cdots + f_s(\dots)$, where $f_1, \dots, f_s \in \Gamma$.
Decide: whether $f_1(\dots) + f_2(\dots) + \cdots + f_s(\dots) < T$ is satisfiable.

Example

$$A = \{0, 1\}, f(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}.$$

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

VCSP(Γ)

Given: a threshold T and a sum $f_1(\dots) + f_2(\dots) + \dots + f_s(\dots)$, where $f_1, \dots, f_s \in \Gamma$.
Decide: whether $f_1(\dots) + f_2(\dots) + \dots + f_s(\dots) < T$ is satisfiable.

Example

$$A = \{0,1\}, f(x,y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}.$$

- $f(x_1,x_2) + f(x_1,x_3) + f(x_2,x_3) < 2$ is an instance VCSP($\{f\}$)

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

VCSP(Γ)

Given: a threshold T and a sum $f_1(\dots) + f_2(\dots) + \dots + f_s(\dots)$, where $f_1, \dots, f_s \in \Gamma$.
Decide: whether $f_1(\dots) + f_2(\dots) + \dots + f_s(\dots) < T$ is satisfiable.

Example

$$A = \{0, 1\}, f(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}.$$

- $f(x_1, x_2) + f(x_1, x_3) + f(x_2, x_3) < 2$ is an instance VCSP($\{f\}$)
- VCSP($\{f\}$) is equivalent to MAX-CUT problem.

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

VCSP(Γ)

Given: a threshold T and a sum $f_1(\dots) + f_2(\dots) + \cdots + f_s(\dots)$, where $f_1, \dots, f_s \in \Gamma$.
Decide: whether $f_1(\dots) + f_2(\dots) + \cdots + f_s(\dots) < T$ is satisfiable.

Example

$$A = \{0, 1\}, f(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}.$$

- $f(x_1, x_2) + f(x_1, x_3) + f(x_2, x_3) < 2$ is an instance VCSP($\{f\}$)
- VCSP($\{f\}$) is equivalent to MAX-CUT problem.
- VCSP($\{f\}$) is NP-complete.

Valued Constraint Satisfaction Problem

Γ is a set of cost functions on a finite set A, i.e. mappings $A^n \to \mathbb{Q} \cup \{\infty\}$.

VCSP(Γ)

Given: a threshold T and a sum $f_1(\dots) + f_2(\dots) + \dots + f_s(\dots)$, where $f_1, \dots, f_s \in \Gamma$.
Decide: whether $f_1(\dots) + f_2(\dots) + \dots + f_s(\dots) < T$ is satisfiable.

Example

$$A = \{0, 1\}, f(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}.$$

- $f(x_1, x_2) + f(x_1, x_3) + f(x_2, x_3) < 2$ is an instance VCSP($\{f\}$)
- VCSP($\{f\}$) is equivalent to MAX-CUT problem.
- VCSP($\{f\}$) is NP-complete.

Complexity classification [Kolmogorov, Krokhin, Rolínek, 2015+Bulatov, Zhuk, 2017]

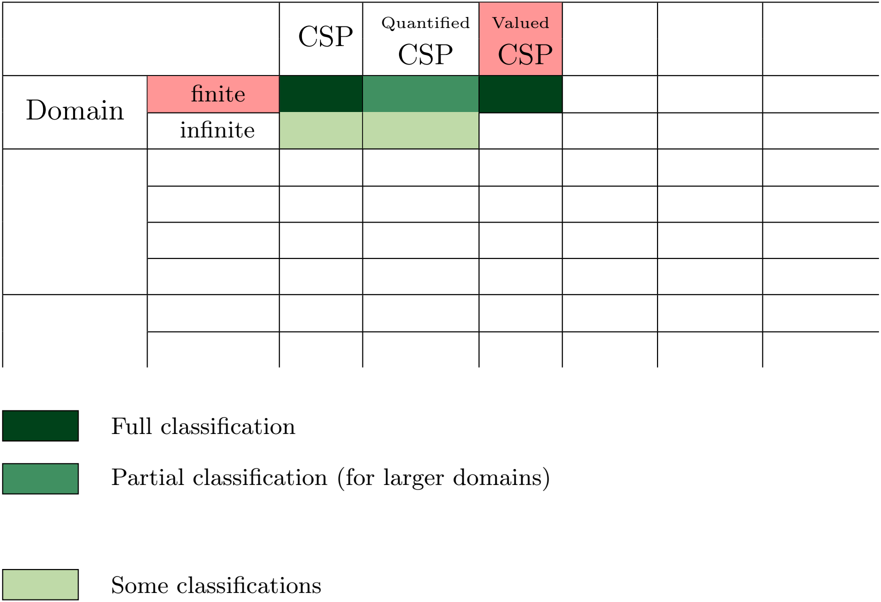
A full classification of the complexity for any finite set of cost functions Γ (P vs NP-complete).

		CSP	Quantified CSP	Valued CSP			
Domain	finite						
	infinite						

Full classification

Partial classification (for larger domains)

Some classifications


CSP
Quantified CSP
Valued CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications

		CSP	Quantified CSP	Valued CSP	Promise CSP		
Domain	finite						
	infinite						

Full classification

Partial classification (for larger domains)

Some classifications

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 1

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 1

- Strong version is 1IN3 $= \{(1,0,0),(0,1,0),(0,0,1)\}$

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 1

- Strong version is $1\mathrm{IN}3 = \{(1,0,0),(0,1,0),(0,0,1)\}$
- Weak version is $\mathrm{NAE}3 = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 1

- Strong version is $1\mathrm{IN}3 = \{(1,0,0),(0,1,0),(0,0,1)\}$
- Weak version is $\mathrm{NAE}3 = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$
- $\mathrm{CSP}(\{\mathrm{NAE}3\})$ and $\mathrm{CSP}(\{1\mathrm{IN}3\})$ are NP-hard, but $\mathrm{PCSP}(\{1\mathrm{IN}3, \mathrm{NAE}3\})$ is in P

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 1

- Strong version is 1IN3 $= \{(1,0,0),(0,1,0),(0,0,1)\}$
- Weak version is NAE3 $= \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$
- CSP($\{$NAE3$\}$) and CSP($\{$1IN3$\}$) are NP-hard, but PCSP($\{$1IN3, NAE3$\}$) is in P (promise helps).

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 1

- Strong version is 1IN3 $= \{(1,0,0),(0,1,0),(0,0,1)\}$
- Weak version is NAE3 $= \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$
- CSP({NAE3}) and CSP({1IN3}) are NP-hard, but PCSP({1IN3, NAE3}) is in P (promise helps).

Theorem [Ficak, Kozik, Olsák, Stankiewicz, 2019])

A classification of the complexity of PCSP(Γ) for Γ consising of symmetric relations on $\{0,1\}$.

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 2 ((K, L)-colorability)

Given a graph G.

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 2 ((K, L)-colorability)

Given a graph G. Distinguish between two cases

- the graph is K-colorable;
- the graph is not even L-colorable;

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 2 ((K, L)-colorability)

Given a graph G. Distinguish between two cases

- the graph is K-colorable;
- the graph is not even L-colorable;

Open questions

- What is the complexity of $(3, 6)$-colorability?

Promise Constraint Satisfaction Problem

There are two versions of each relation (weak and strong) in Γ

PCSP(Γ)

Given a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$,
Distinguish between two cases:

- Strong version is satisfied
- Weak version is not satisfied

Example 2 ((K, L)-colorability)

Given a graph G. Distinguish between two cases

- the graph is K-colorable;
- the graph is not even L-colorable;

Open questions

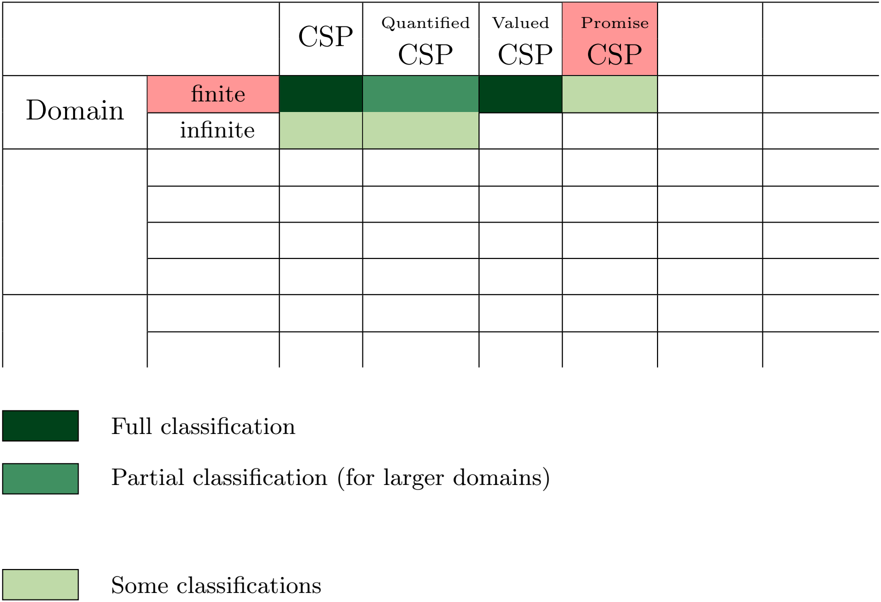
- What is the complexity of $(3, 6)$-colorability?
- What is the complexity of $(3, 1000000000)$-colorability?

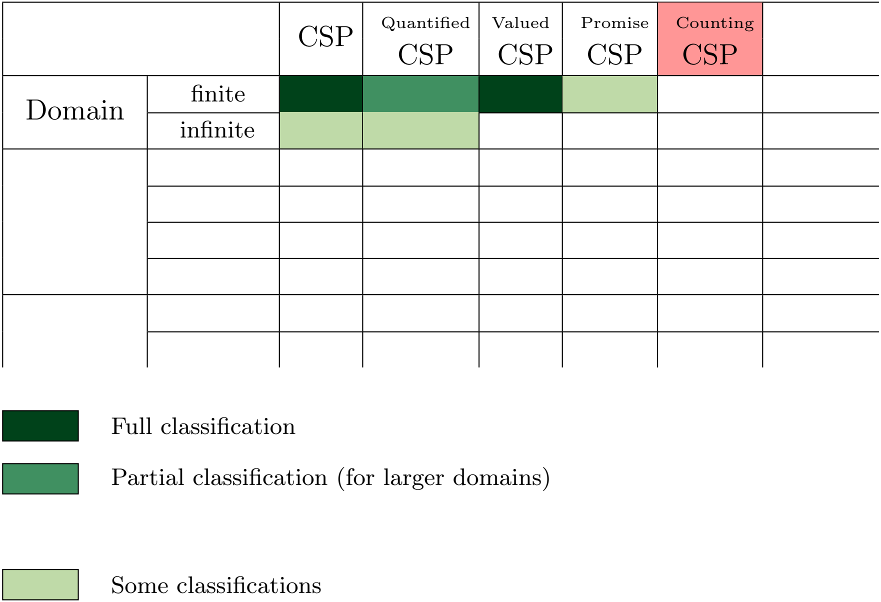
		CSP	Quantified CSP	Valued CSP	Promise CSP		
Domain	finite						
	infinite						

Full classification

Partial classification (for larger domains)

Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications

Counting Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Counting-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Find the number of solutions.

Counting Constraint Satisfaction Problem

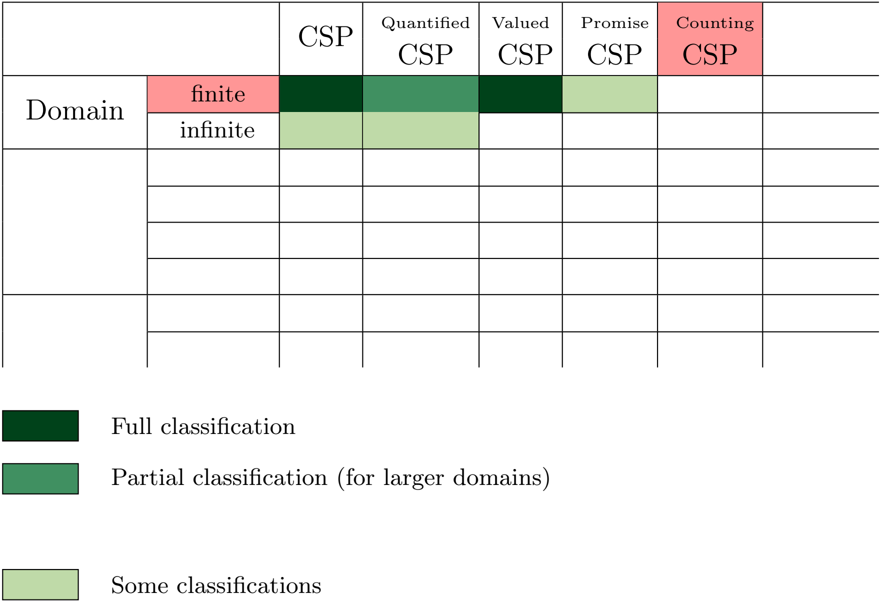
Γ is a set of relations on a finite set A.

Counting-CSP(Γ)

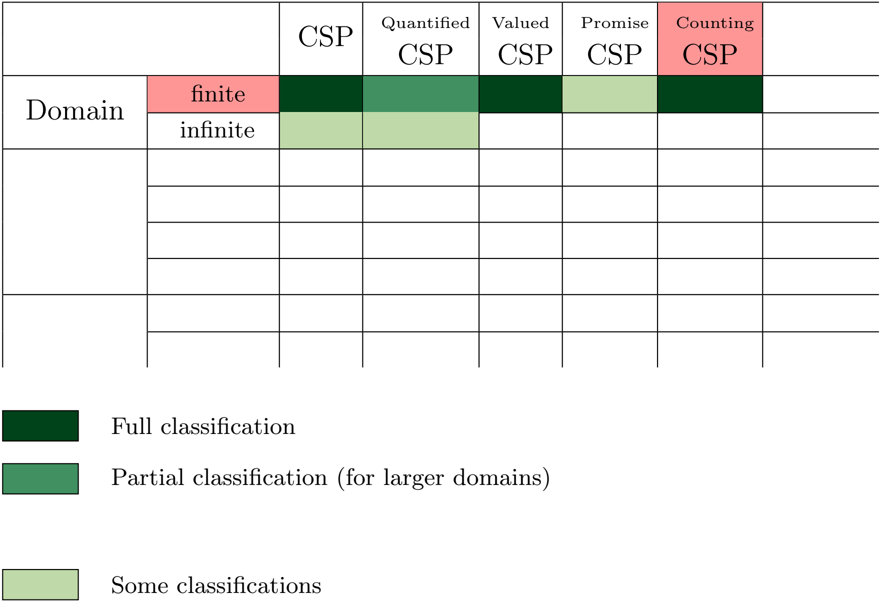
Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$,
where $R_1, \ldots, R_s \in \Gamma$.
Find the number of solutions.

Theorem [Bulatov, 2008]

A classification of the complexity of Counting-CSP(Γ) for every Γ.



CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications

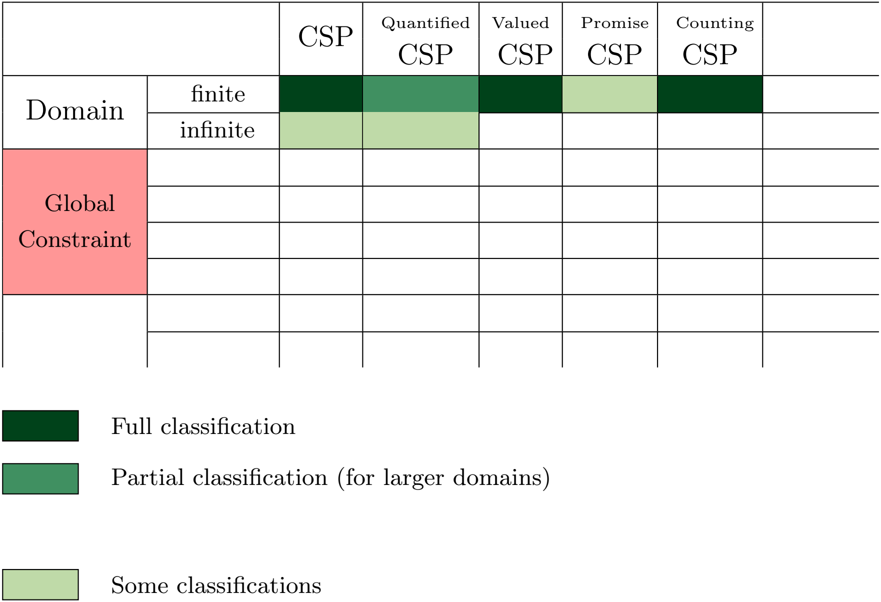

CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Full classification
Partial classification (for larger domains)
Some classifications

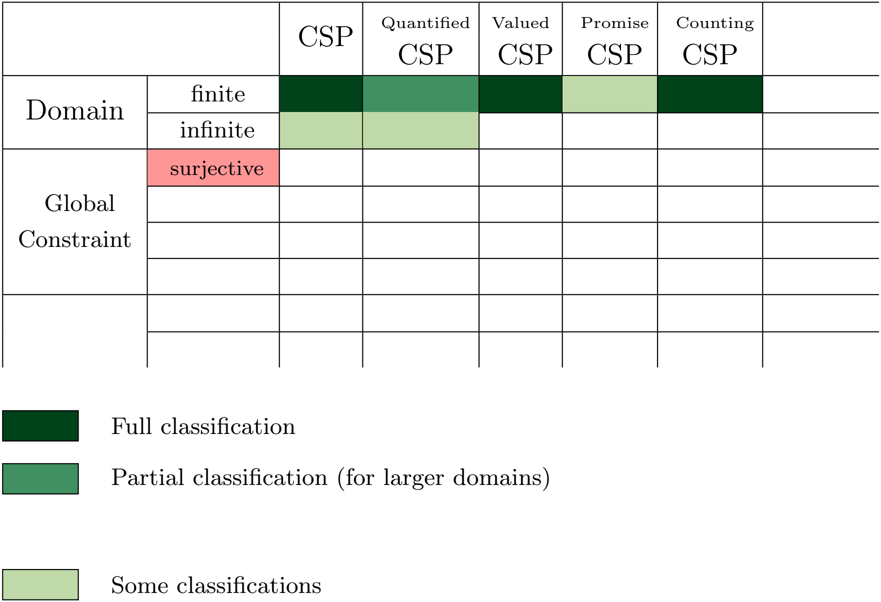
		CSP	Quantified CSP	Valued CSP	Promise CSP	Counting CSP	
Domain	finite						
	infinite						

Full classification

Partial classification (for larger domains)

Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
Full classification
Partial classification (for larger domains)
Some classifications



CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
Full classification
Partial classification (for larger domains)
Some classifications

Surjective Constraint Satisfaction Problem

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Example

$A = \{0, 1, 2\}, \Gamma = \{x \leq y\}$.

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Example

$A = \{0, 1, 2\}, \Gamma = \{x \leq y\}$. Surjective CSP instances:
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_4$,

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.

Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Example

$A = \{0, 1, 2\}, \Gamma = \{x \leq y\}$. Surjective CSP instances:

$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_4$, $x_1 = 0, x_2 = 1, x_3 = x_4 = 2$.

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Example

$A = \{0, 1, 2\}, \Gamma = \{x \leq y\}$. Surjective CSP instances:
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_4$, $x_1 = 0, x_2 = 1, x_3 = x_4 = 2$.
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_1$,

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Example

$A = \{0, 1, 2\}, \Gamma = \{x \leq y\}$. Surjective CSP instances:
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_4$, $x_1 = 0, x_2 = 1, x_3 = x_4 = 2$.
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_1$, No surjective solutions

Surjective Constraint Satisfaction Problem

Γ is a set of relations on A.

SCSP(Γ)

Given: a conjunction of relations, i.e. a formula

$$R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}}),$$

where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether the formula has a surjective solution, that is, a solution such that $\{x_1, \ldots, x_n\} = A$.

Example

$A = \{0, 1, 2\}, \Gamma = \{x \leq y\}$. Surjective CSP instances:
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_4$, $x_1 = 0, x_2 = 1, x_3 = x_4 = 2$.
$x_1 \leq x_2 \wedge x_2 \leq x_3 \wedge x_3 \leq x_1$, No surjective solutions

Question

What is the complexity of the SCSP(Γ)?

Surjective Graph Homomorphism Problem

Surjective Graph Homomorphism Problem

Let H be a finite graph.

Surjective Graph Homomorphism Problem

Let H be a finite graph.

SurjHom(H):

Given: a graph G.

Decide: whether there exists a surjective homomorphism from G to H.

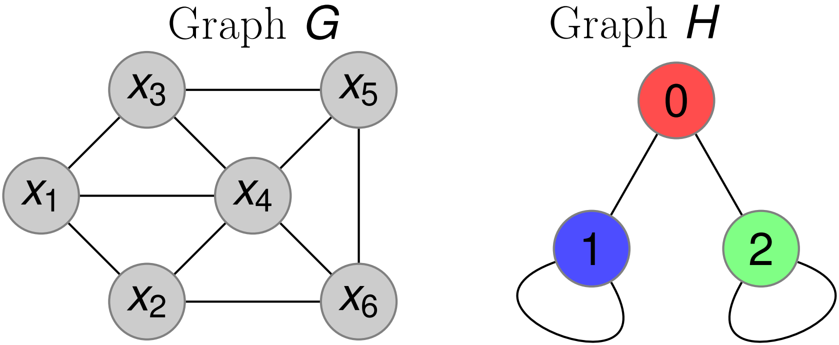
Surjective Graph Homomorphism Problem

Let H be a finite graph.

SurjHom(H):

Given: a graph G.

Decide: whether there exists a surjective homomorphism from G to H.



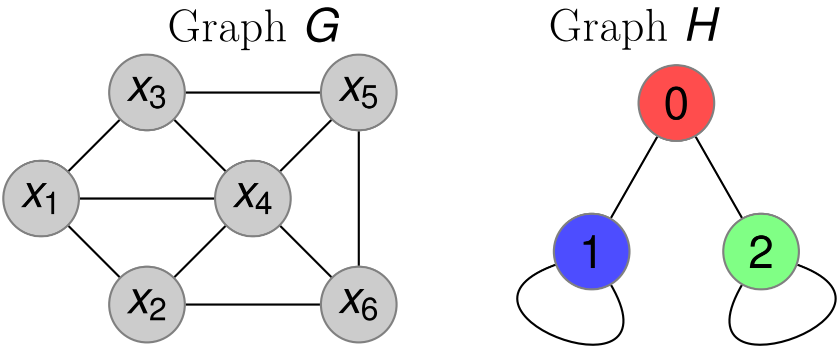
Surjective Graph Homomorphism Problem

Let H be a finite graph.

SurjHom(H):

Given: a graph G.

Decide: whether there exists a surjective homomorphism from G to H.



SurjHom(H) is equivalent to SCSP($\{x + y \neq 0 \mod 3\}$).

History

History

- The complexity of $\mathrm{SCSP}(\Gamma)$ was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].

History

- The complexity of $\mathbf{SCSP}(\Gamma)$ was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of $\mathbf{SurjHom}(H)$ was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]

$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.

History

- The complexity of **SCSP**(Γ) was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of **SurjHom**(H) was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]
- The complexity of **SurjHom**(H) was described for partially reflexive forests [Golovach, Paulusma, Song, 2011]

$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.

History

- The complexity of $\mathsf{SCSP}(\Gamma)$ was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of $\mathsf{SurjHom}(H)$ was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]
- The complexity of $\mathsf{SurjHom}(H)$ was described for partially reflexive forests [Golovach, Paulusma, Song, 2011]
- $\mathsf{SurjHom}(\mathcal{C}_4^{ref})$ is NP-complete [Martin, Paulusma, Vikas, 2011]

$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.

History

- The complexity of $\mathbf{SCSP}(\Gamma)$ was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of $\mathbf{SurjHom}(H)$ was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]
- The complexity of $\mathbf{SurjHom}(H)$ was described for partially reflexive forests [Golovach, Paulusma, Song, 2011]
- $\mathbf{SurjHom}(\mathcal{C}_4^{ref})$ is NP-complete [Martin, Paulusma, Vikas, 2011]
- $\mathbf{SurjHom}(\mathcal{C}_6)$ is NP-complete [Vikas, 2017]

$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.

History

- The complexity of SCSP(Γ) was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of SurjHom(H) was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]
- The complexity of SurjHom(H) was described for partially reflexive forests [Golovach, Paulusma, Song, 2011]
- SurjHom($\mathcal{C}_4^{ref}$) is NP-complete [Martin, Paulusma, Vikas, 2011]
- SurjHom($\mathcal{C}_6$) is NP-complete [Vikas, 2017]
- The No-Rainbow Problem is NP-complete [Zhuk, 2020] SCSP($\{(a,b,c) : |\{a,b,c\}| < 3\}$).

$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.

History

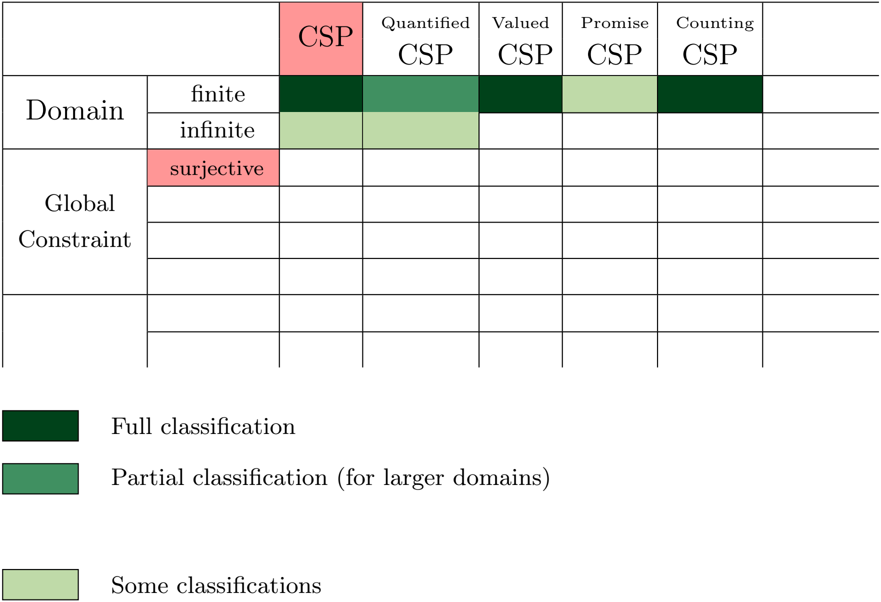
- The complexity of SCSP(Γ) was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of SurjHom(H) was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]
- The complexity of SurjHom(H) was described for partially reflexive forests [Golovach, Paulusma, Song, 2011]
- SurjHom($\mathcal{C}_4^{ref}$) is NP-complete [Martin, Paulusma, Vikas, 2011]
- SurjHom($\mathcal{C}_6$) is NP-complete [Vikas, 2017]
- The No-Rainbow Problem is NP-complete [Zhuk, 2020] SCSP($\{(a, b, c) \colon |\{a, b, c\}| < 3\}$).
- The complexity cannot be described in terms of polymorphisms [Zhuk, 2020]

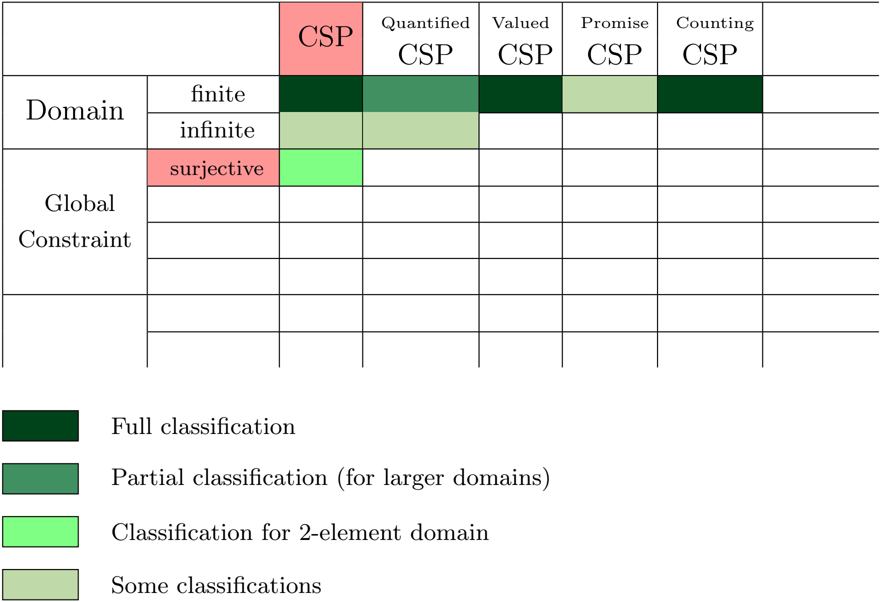
$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.

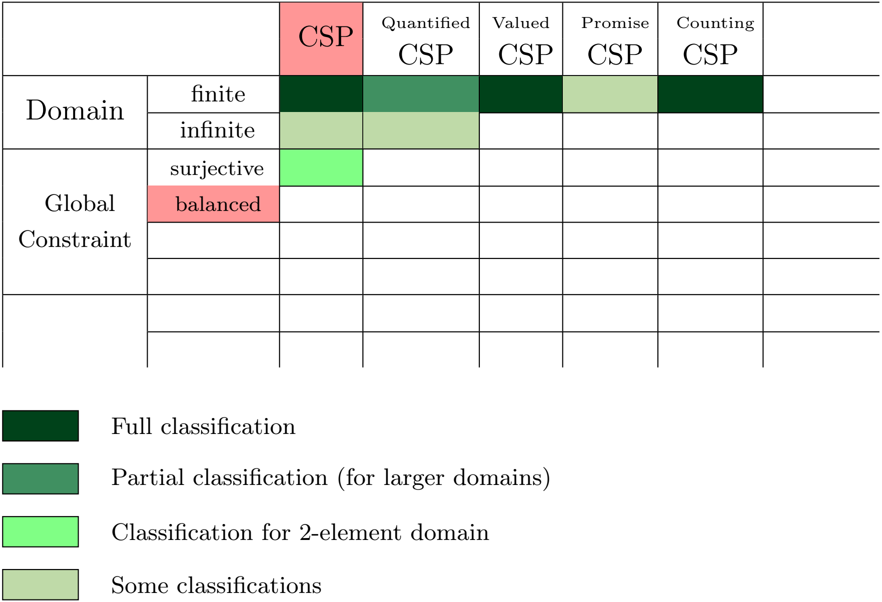
History

- The complexity of $\mathbf{SCSP}(\Gamma)$ was described for every Γ on a two-element domain [Creignou, N., and Hébrard, 1997].
- The complexity of $\mathbf{SurjHom}(H)$ was described for all graphs of size 4 other than $\mathcal{C}_4^{ref}$[S. Dantas, Figueiredo, Gravier, Klein, 2005]
- The complexity of $\mathbf{SurjHom}(H)$ was described for partially reflexive forests [Golovach, Paulusma, Song, 2011]
- $\mathbf{SurjHom}(\mathcal{C}_4^{ref})$ is NP-complete [Martin, Paulusma, Vikas, 2011]
- $\mathbf{SurjHom}(\mathcal{C}_6)$ is NP-complete [Vikas, 2017]
- The No-Rainbow Problem is NP-complete [Zhuk, 2020] $\mathbf{SCSP}(\{(a,b,c)\colon |\{a,b,c\}| < 3\})$.
- The complexity cannot be described in terms of polymorphisms [Zhuk, 2020]
- $\mathbf{SurjHom}(\mathcal{C}_n^{ref})$ is NP-complete for $n \geq 7$ [Korchagin, 2023]

$\mathcal{C}_n$ and $\mathcal{C}_n^{ref}$ are non-reflexive and reflexive cycles with n vertices.


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
Full classification
Partial classification (for larger domains)
Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications



CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a balanced solution, i.e., a solution with equal number of every element.

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a balanced solution, i.e., a solution with equal number of every element.

Balanced-CSP($=$) on $\{0, 1\}$

Given an instance $x_{i_1} = x_{j_1} \wedge \cdots \wedge x_{i_s} = x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$, where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it has a balanced solution, i.e., a solution with equal number of every element.

Balanced-CSP($=$) on $\{0, 1\}$

Given an instance $x_{i_1} = x_{j_1} \wedge \dots \wedge x_{i_s} = x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced-CSP($\leq$) on $\{0, 1\}$

Given an instance $x_{i_1} \leq x_{j_1} \wedge \dots \wedge x_{i_s} \leq x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a balanced solution, i.e., a solution with equal number of every element.

Balanced-CSP($=$) on $\{0, 1\}$ solvable in polynomial time

Given an instance $x_{i_1} = x_{j_1} \wedge \cdots \wedge x_{i_s} = x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced-CSP($\leq$) on $\{0, 1\}$

Given an instance $x_{i_1} \leq x_{j_1} \wedge \cdots \wedge x_{i_s} \leq x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a balanced solution, i.e., a solution with equal number of every element.

Balanced-CSP($=$) on $\{0, 1\}$ solvable in polynomial time

Given an instance $x_{i_1} = x_{j_1} \wedge \cdots \wedge x_{i_s} = x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced-CSP($\leq$) on $\{0, 1\}$ NP-complete

Given an instance $x_{i_1} \leq x_{j_1} \wedge \cdots \wedge x_{i_s} \leq x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Balanced CSP

Γ is a set of relations on a finite set A.

Balanced-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a balanced solution, i.e., a solution with equal number of every element.

Balanced-CSP($=$) on $\{0, 1\}$ — solvable in polynomial time

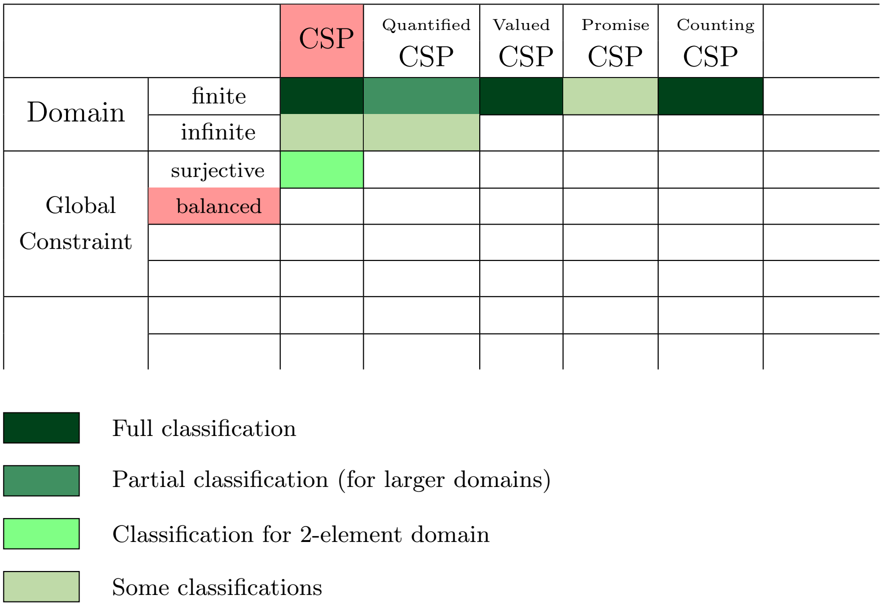
Given an instance $x_{i_1} = x_{j_1} \wedge \cdots \wedge x_{i_s} = x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

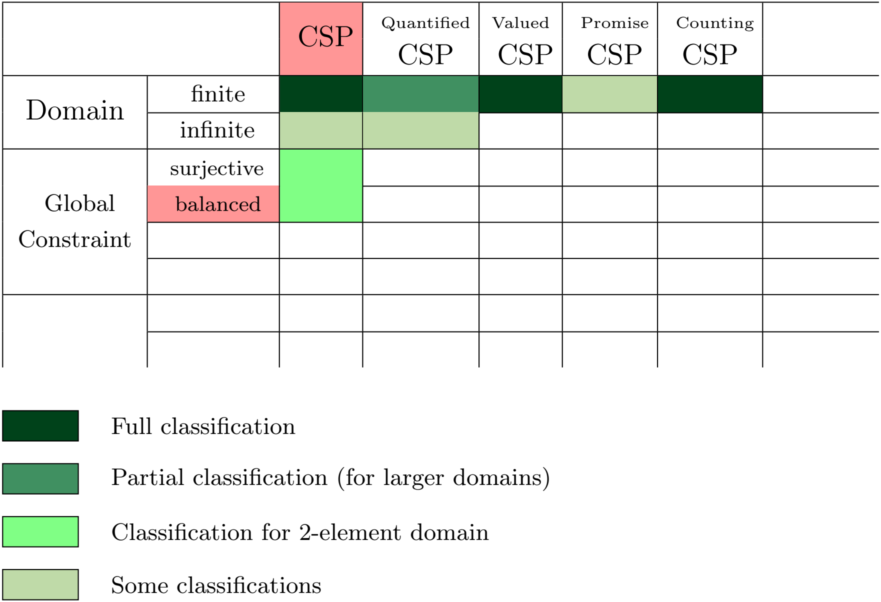
Balanced-CSP($\leq$) on $\{0, 1\}$ — NP-complete

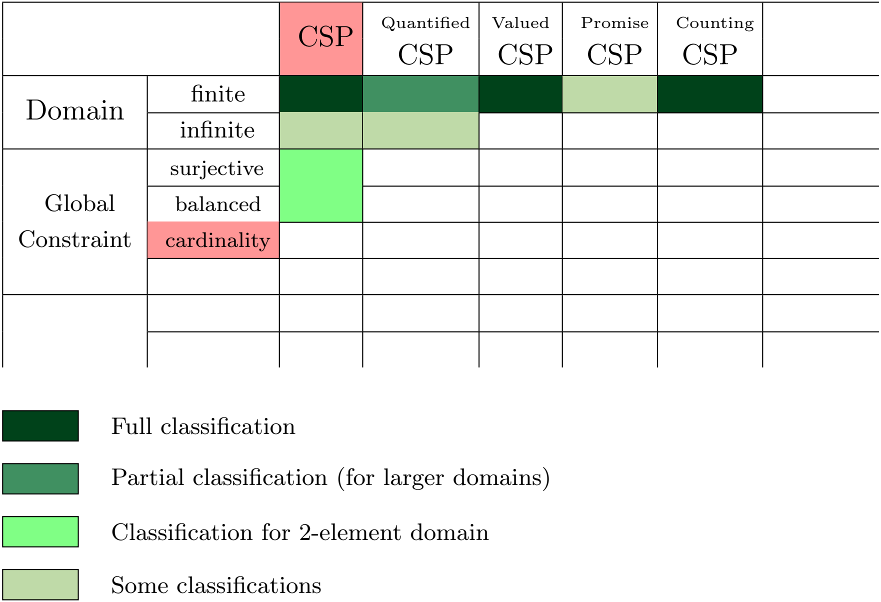
Given an instance $x_{i_1} \leq x_{j_1} \wedge \cdots \wedge x_{i_s} \leq x_{j_s}$. Decide whether it has a solution with equal number of 0 and 1.

Theorem [Creignou, H. Schnoor, I. Schnoor, 2008]

A classification of the complexity of Balanced-CSP(Γ) and Cardinality-CSP(Γ) for each Γ on $\{0, 1\}$.


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications

Global cardinality constraint

Γ is a set of relations on a finite set A.

Cardinality-CSP(Γ)

Given: a mapping $\pi\colon A \to \mathbb{N}$ and a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a solution containing each element $a \in A$ exactly $\pi(a)$ times.

Global cardinality constraint

Γ is a set of relations on a finite set A.

Cardinality-CSP(Γ)

Given: a mapping $\pi \colon A \to \mathbb{N}$ and a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a solution containing each element $a \in A$ exactly $\pi(a)$ times.

Cardinality-CSP(Linear Equations in $\mathbb{Z}_2$)

Given a system of linear equations in $\mathbb{Z}_2$ and $k \in \mathbb{N}$.
Decide whether there exists a solution with exactly k 1s.

Global cardinality constraint

Γ is a set of relations on a finite set A.

Cardinality-CSP(Γ)

Given: a mapping $\pi\colon A \to \mathbb{N}$ and a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a solution containing each element $a \in A$ exactly $\pi(a)$ times.

Cardinality-CSP(Linear Equations in $\mathbb{Z}_2$) NP-complete

Given a system of linear equations in $\mathbb{Z}_2$ and $k \in \mathbb{N}$.
Decide whether there exists a solution with exactly k 1s.

Global cardinality constraint

Γ is a set of relations on a finite set A.

Cardinality-CSP(Γ)

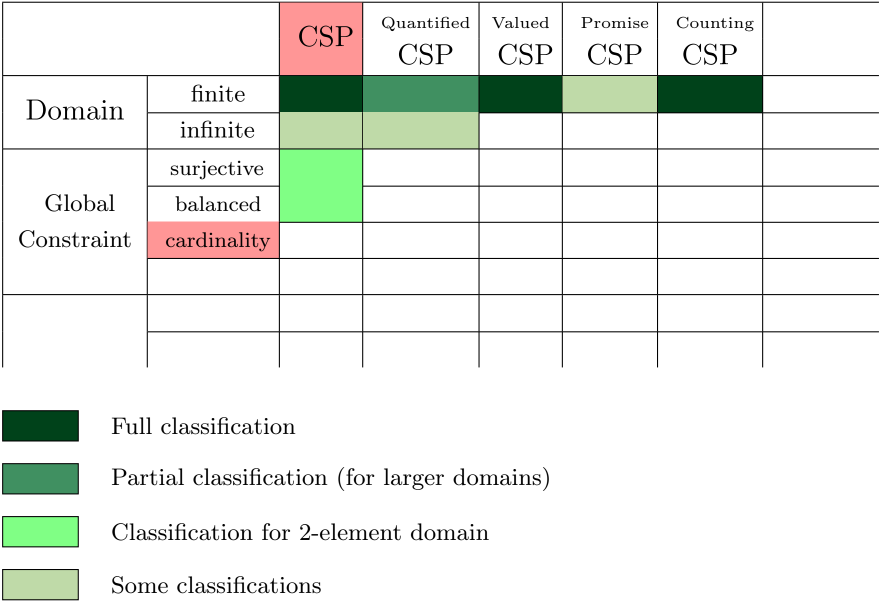
Given: a mapping $\pi \colon A \to \mathbb{N}$ and a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a solution containing each element $a \in A$ exactly $\pi(a)$ times.

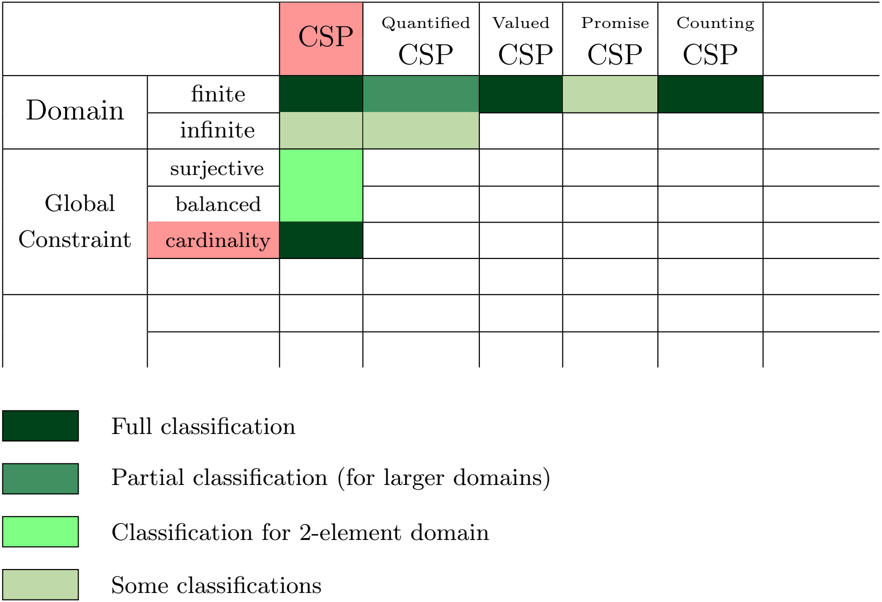
Cardinality-CSP(Linear Equations in $\mathbb{Z}_2$) NP-complete

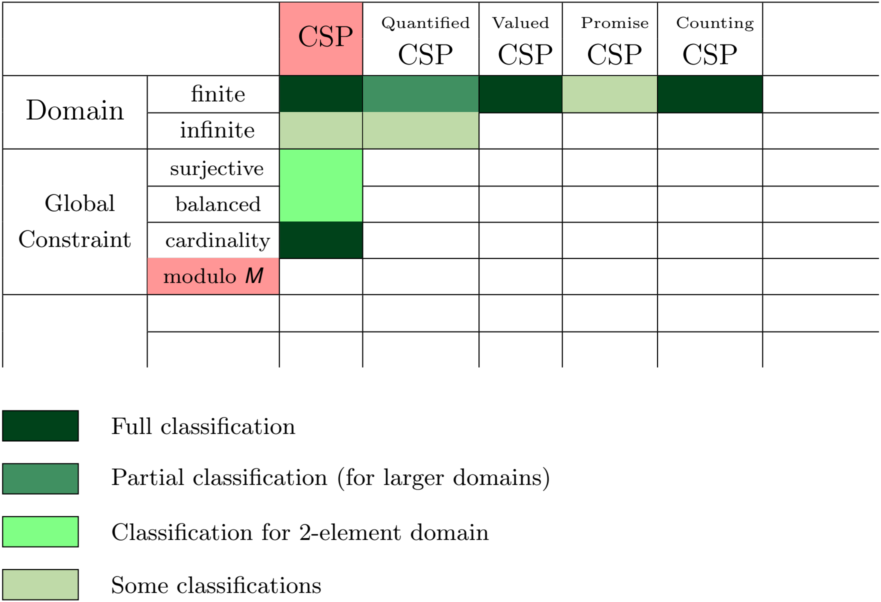
Given a system of linear equations in $\mathbb{Z}_2$ and $k \in \mathbb{N}$.
Decide whether there exists a solution with exactly k 1s.

Theorem [Bulatov, Marx, 2009]

A classification of the complexity of Cardinality-CSP(Γ) for each Γ.


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications

Global modular constraint

Mod_M-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$, where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it has a solution satisfying $x_1 + \dots + x_n = 0 \mod M$.

Global modular constraint

Mod_M-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a solution satisfying $x_1 + \cdots + x_n = 0 \mod M$.

- If Γ consists of linear equations on $\{0, 1\}$ and $M = 25$ then Mod_M-CSP(Γ) is tractable

Global modular constraint

Mod_M-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$.
Decide: whether it has a solution satisfying $x_1 + \cdots + x_n = 0 \mod M$.

- If Γ consists of linear equations on $\{0, 1\}$ and $M = 25$ then Mod_M-CSP(Γ) is tractable
- If Γ consists of linear equations on $\{0, 1\}$ and $M = 15$ then Mod_M-CSP(Γ) is not tractable

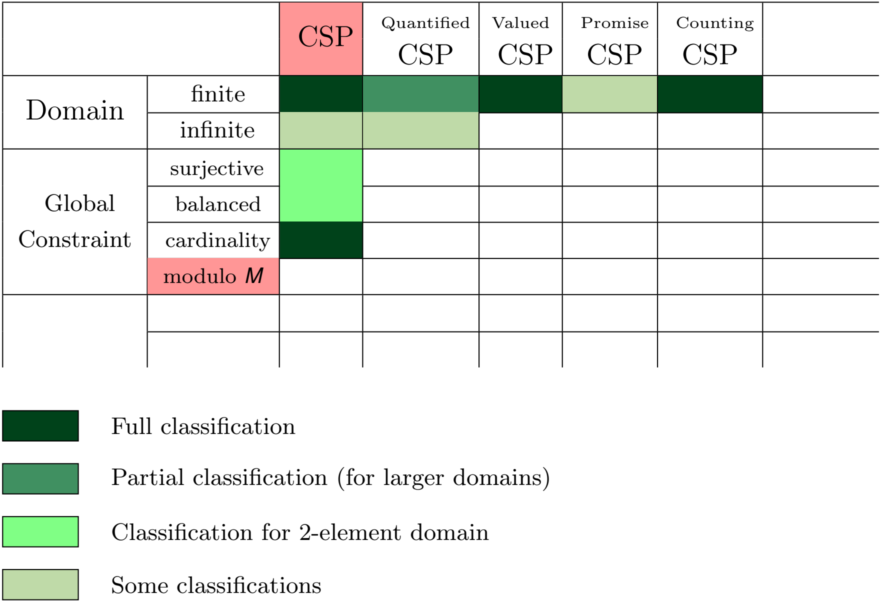
Global modular constraint

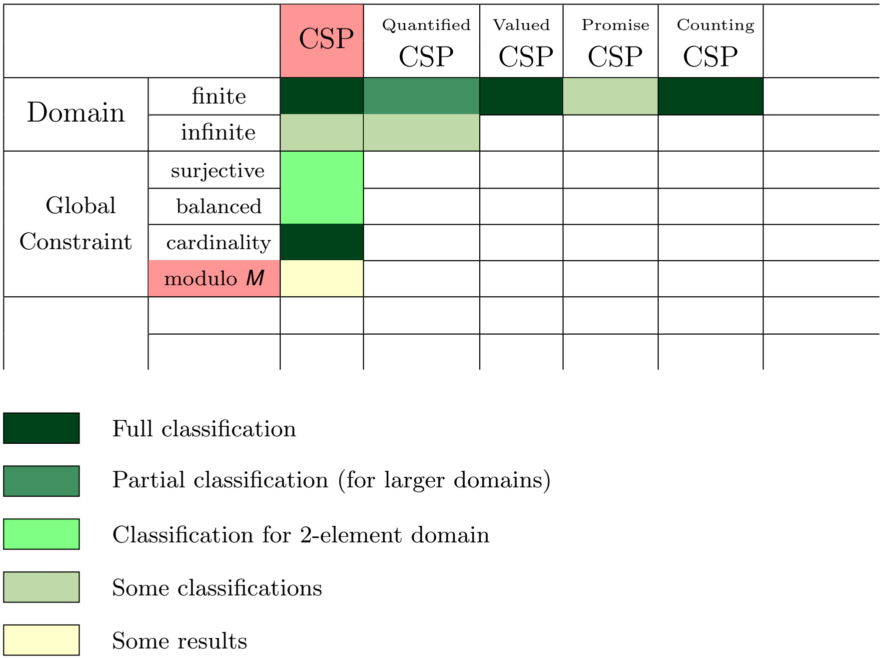
Mod_M-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$, where $R_1, \dots, R_s \in \Gamma$.
Decide: whether it has a solution satisfying $x_1 + \cdots + x_n = 0 \mod M$.

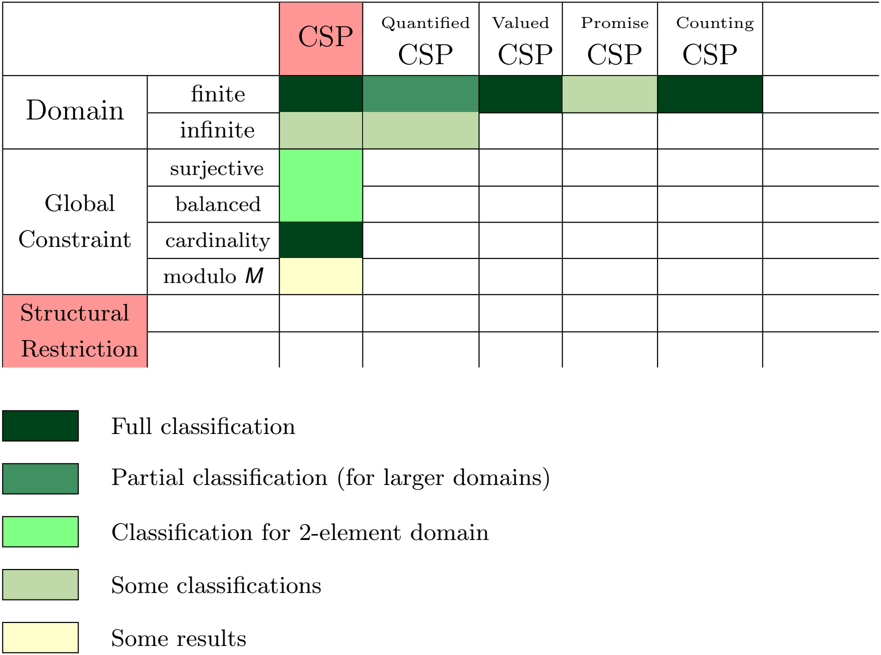
- If Γ consists of linear equations on $\{0,1\}$ and $M = 25$ then Mod_M-CSP(Γ) is tractable
- If Γ consists of linear equations on $\{0,1\}$ and $M = 15$ then Mod_M-CSP(Γ) is not tractable
- If Γ consists of linear equations on $\{0,1\}$ and $M = 24$ then the complexity of Mod_M-CSP(Γ) is not known.

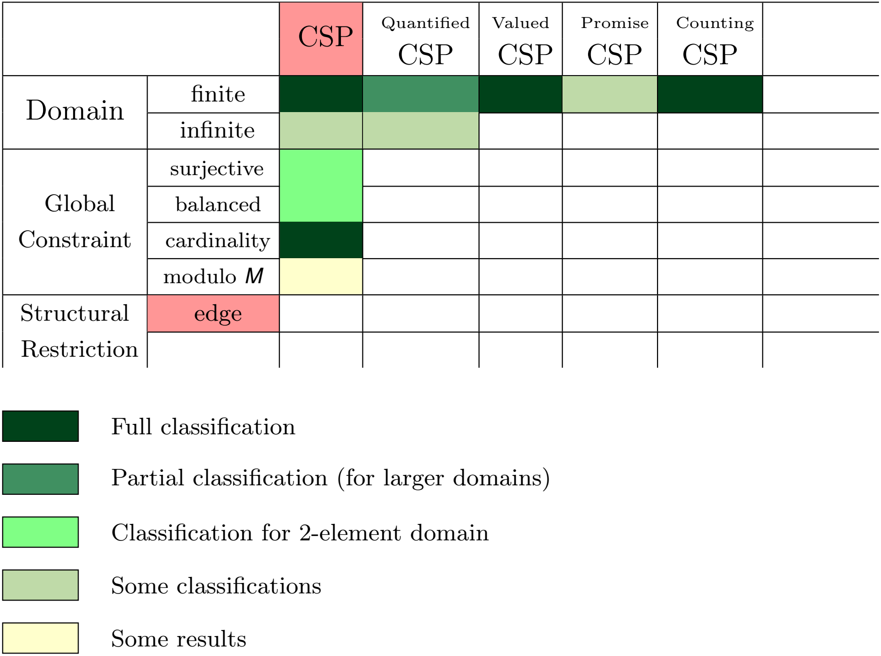
$$\begin{cases} x_1 + x_2 + x_3 = 0 & \mod 2 \\ x_1 + x_3 + x_5 = 0 & \mod 2 \\ x_2 + x_4 + x_5 = 1 & \mod 2 \\ x_2 + x_3 + x_5 = 0 & \mod 24 \end{cases}$$


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications



CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results



CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
edge
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results

Edge Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Edge Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Edge-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$ and every variable appears exactly twice.

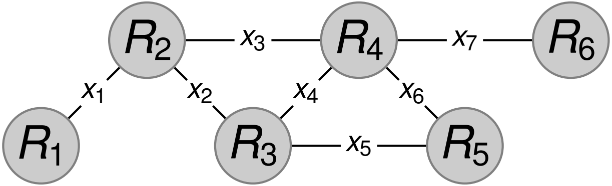
Decide: whether it has a solution.

Edge Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Edge-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$ and every variable appears exactly twice.
Decide: whether it has a solution.

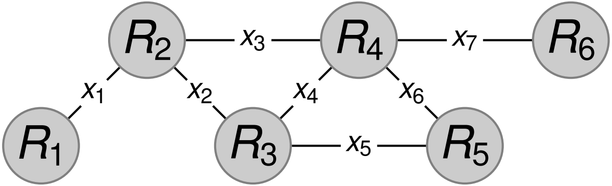


Edge Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Edge-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \dots, x_{i_{1,n_1}}) \wedge \dots \wedge R_s(x_{i_{s,1}}, \dots, x_{i_{s,n_s}})$, where $R_1, \dots, R_s \in \Gamma$ and every variable appears exactly twice.
Decide: whether it has a solution.



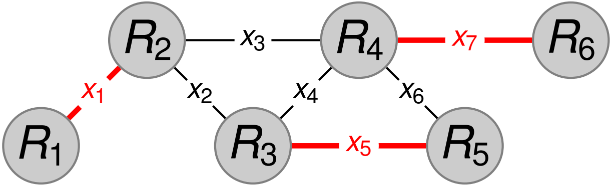
- Edge-CSP($\{1\text{IN}2, 1\text{IN}3, 1\text{IN}4, \dots\}$) is equivalent to the Perfect Matching Problem.

Edge Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Edge-CSP(Γ)

Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$ and every variable appears exactly twice.
Decide: whether it has a solution.



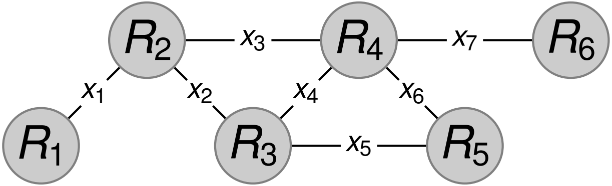
- Edge-CSP($\{1\text{IN}2, 1\text{IN}3, 1\text{IN}4, \ldots\}$) is equivalent to the Perfect Matching Problem.

Edge Constraint Satisfaction Problem

Γ is a set of relations on a finite set A.

Edge-CSP(Γ)

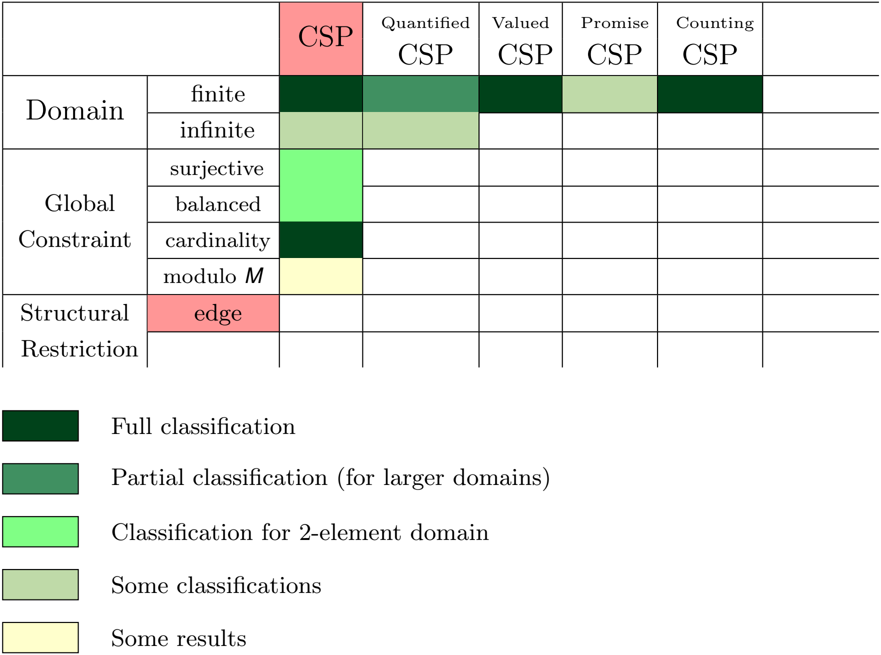
Given: a formula $R_1(x_{i_{1,1}}, \ldots, x_{i_{1,n_1}}) \wedge \cdots \wedge R_s(x_{i_{s,1}}, \ldots, x_{i_{s,n_s}})$, where $R_1, \ldots, R_s \in \Gamma$ and every variable appears exactly twice.
Decide: whether it has a solution.



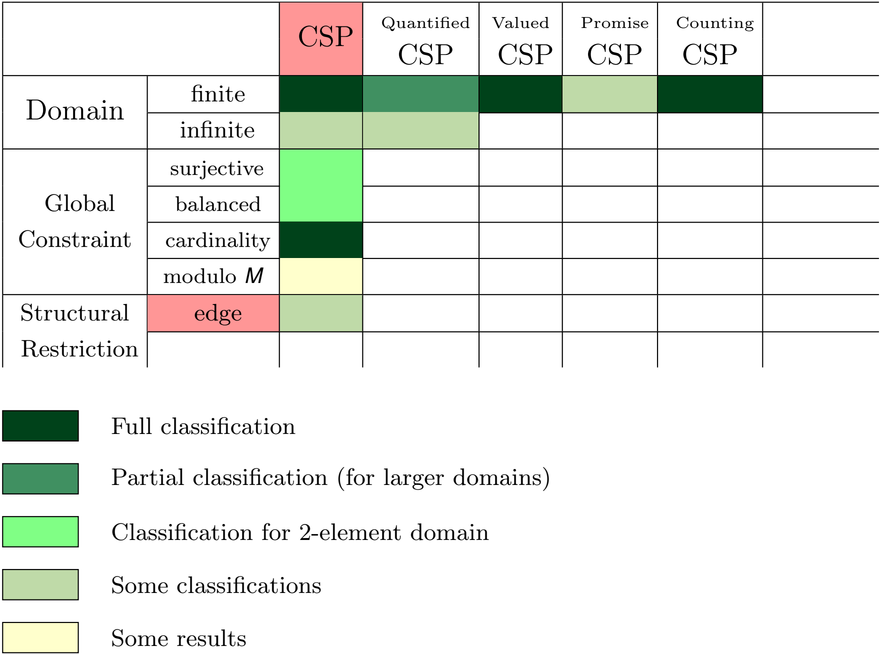
- Edge-CSP($\{1IN2, 1IN3, 1IN4, \ldots\}$) is equivalent to the Perfect Matching Problem.

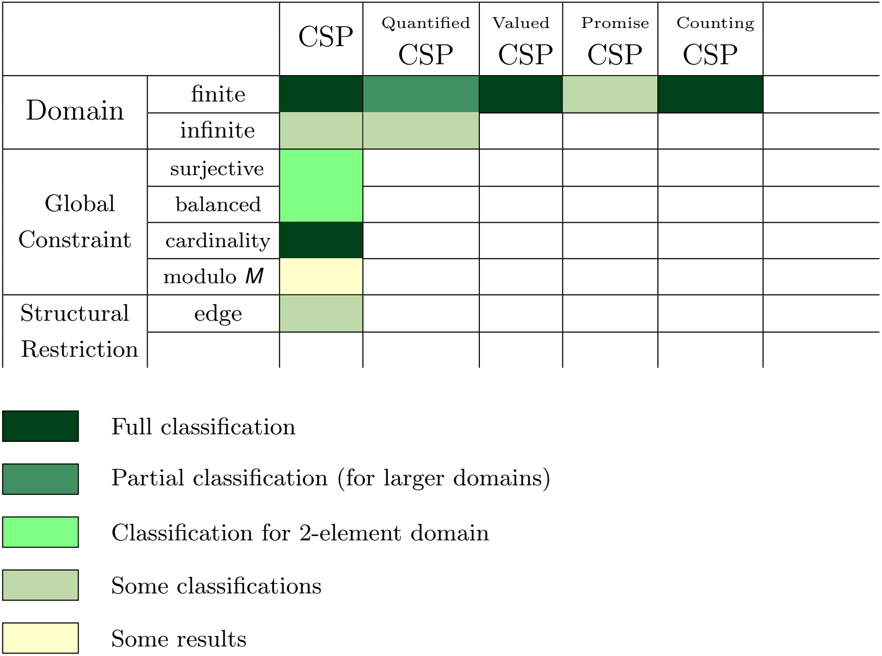
Theorem [Kazda, Kolmogorov, Rolinek, 2018]

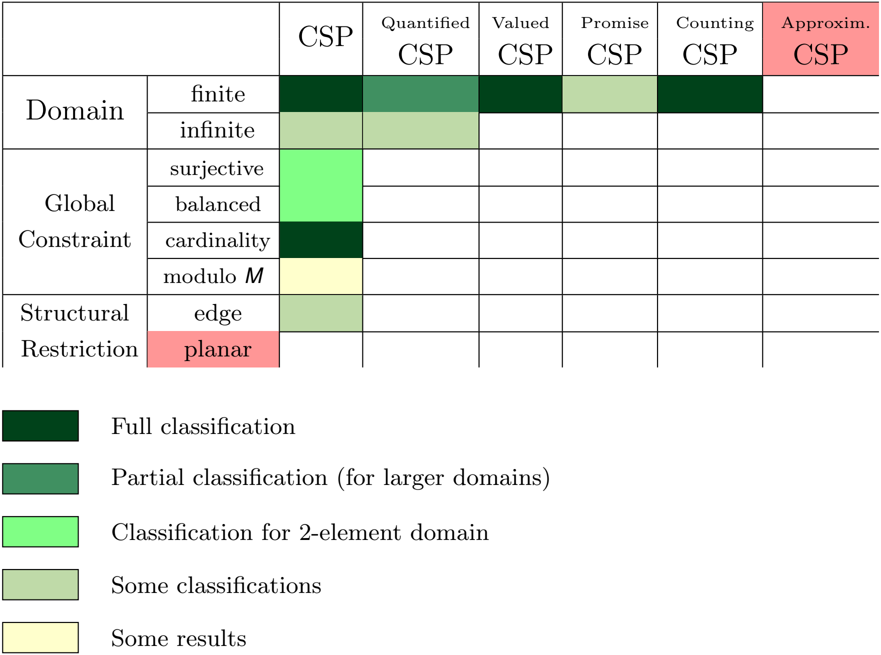
A classification of the complexity for planar Edge-CSP(Γ) for every Γ on $\{0, 1\}$.



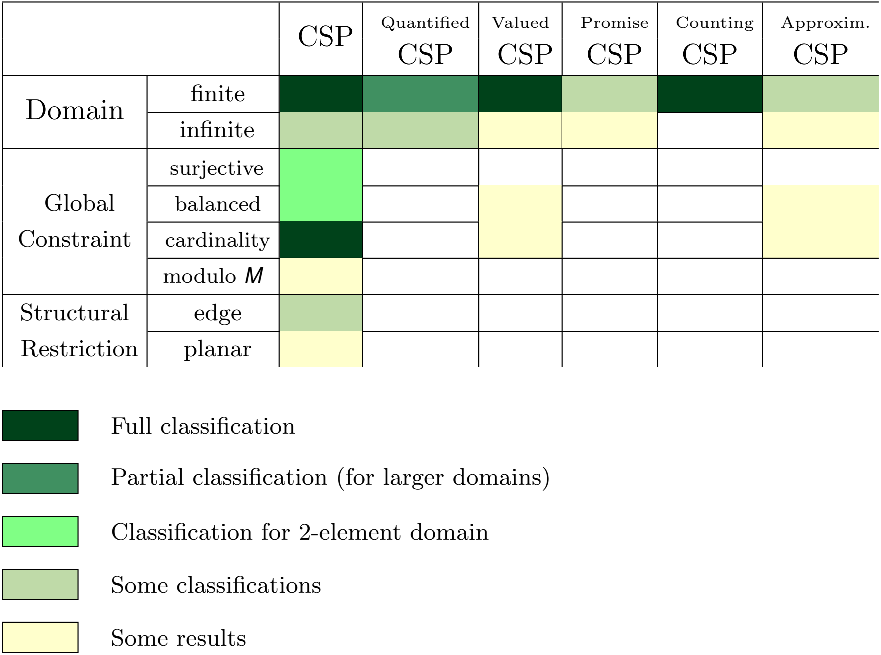
CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
edge
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
edge
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
edge
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results



CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Approxim. CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
edge
planar
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results


CSP
Quantified CSP
Valued CSP
Promise CSP
Counting CSP
Approxim. CSP
Domain
finite
infinite
Global Constraint
surjective
balanced
cardinality
modulo M
Structural Restriction
edge
planar
Full classification
Partial classification (for larger domains)
Classification for 2-element domain
Some classifications
Some results