Solving Infinite-Domain CSPs Using the Patchwork Property

AAAI 2021

Konrad K. Dabrowski[1] Peter Jonsson[2] Sebastian Ordyniak[3] *George Osipov*[2]

[1]Durham University

[2]Linköping University

[3]University of Leeds

Motivation and Overview

- Infinite-domain CSPs have many applications in AI, e.g. in spatial and temporal reasoning.

Motivation and Overview

- Infinite-domain CSPs have many applications in AI, e.g. in spatial and temporal reasoning.
- Computationally, infinite-domain CSPs can be arbitrarily hard, so we seek restrictions that make them tractable.

Motivation and Overview

- Infinite-domain CSPs have many applications in AI, e.g. in spatial and temporal reasoning.
- Computationally, infinite-domain CSPs can be arbitrarily hard, so we seek restrictions that make them tractable.
- We study infinite-domain CSPs parameterized by the *primal treewidth*.

Motivation and Overview

- Infinite-domain CSPs have many applications in AI, e.g. in spatial and temporal reasoning.
- Computationally, infinite-domain CSPs can be arbitrarily hard, so we seek restrictions that make them tractable.
- We study infinite-domain CSPs parameterized by the *primal treewidth*.
- We identify the *patchwork property* as a sufficient condition for fixed-parameter tractability of infinite-domain CSPs.

Motivation and Overview

- Infinite-domain CSPs have many applications in AI, e.g. in spatial and temporal reasoning.
- Computationally, infinite-domain CSPs can be arbitrarily hard, so we seek restrictions that make them tractable.
- We study infinite-domain CSPs parameterized by the *primal treewidth*.
- We identify the *patchwork property* as a sufficient condition for fixed-parameter tractability of infinite-domain CSPs.
- As a consequence, we obtain algorithms running in $f(w) \cdot O(n)$ time for CSPs over Allen's Interval Algebra, RCC8, etc.

Motivation and Overview

- Infinite-domain CSPs have many applications in AI, e.g. in spatial and temporal reasoning.
- Computationally, infinite-domain CSPs can be arbitrarily hard, so we seek restrictions that make them tractable.
- We study infinite-domain CSPs parameterized by the *primal treewidth*.
- We identify the *patchwork property* as a sufficient condition for fixed-parameter tractability of infinite-domain CSPs.
- As a consequence, we obtain algorithms running in $f(w) \cdot O(n)$ time for CSPs over Allen's Interval Algebra, RCC8, etc.
- Connecting patchwork to *amalgamation*, we obtain results for temporal constraint satisfaction and phylogeny problems.

Constraint Languages

A constraint language consists of:

- a domain D,
- a set of relations $\{R_1, R_2, \ldots, R_m\}$, where $R_i \subseteq D^{arity(R_i)}$.

Constraint Languages

A constraint language consists of:

- a domain D,
- a set of relations $\{R_1, R_2, \dots, R_m\}$, where $R_i \subseteq D^{arity(R_i)}$.

A constraint language is k-ary if all relations have arity k.

Constraint Languages

A constraint language consists of:

- a domain D,
- a set of relations $\{R_1, R_2, \ldots, R_m\}$, where $R_i \subseteq D^{arity(R_i)}$.

A constraint language is k-ary if all relations have arity k.

The relations $\{R_1, R_2, \ldots, R_m\}$ are:

- *jointly exhaustive* (JE) if $\bigcup_{i=1}^{m} R_i = D^k$,
- *pairwise disjoint* (PD) if $R_i \cap R_j = \varnothing$.

Constraint Satisfaction Problem (CSP)

CSP($\mathcal{B}$)

INSTANCE: (V, C), V - variables, C - constraints of form $R(v_1, \dots, v_r)$, where R is a relation from $\mathcal{B}$ and $v_1, \dots, v_r \in V$.

QUESTION: Is there an assignment f of values from to the domain of $\mathcal{B}$ to the variables in V such that $(f(v_1), \dots, f(v_r)) \in R$ for all constraints in C?

Example: CSP over Point Algebra

POINT ALGEBRA is binary constraint language with domain $\mathbb{R}$ and relations $\{<, =, >\}$. We denote it by $(\mathbb{R}; <, =, >)$.

Example: CSP over Point Algebra

Point Algebra is binary constraint language with domain $\mathbb{R}$ and relations $\{<, =, >\}$. We denote it by $(\mathbb{R}; <, =, >)$.

Relations of $(\mathbb{R}; <, =, >)$ are JEPD.

Example: CSP over Point Algebra

Point Algebra is binary constraint language with domain $\mathbb{R}$ and relations $\{<,=,>\}$. We denote it by $(\mathbb{R};<,=,>)$.

Relations of $(\mathbb{R};<,=,>)$ are JEPD.

An example of an instance of CSP$(\mathbb{R};<,=,>)$:

$V = \{x, y, z, w\}$
$C = \{x < y, y < z, x < z, y = w\}$

Example: CSP over Point Algebra

Point Algebra is binary constraint language with domain $\mathbb{R}$ and relations $\{<, =, >\}$. We denote it by $(\mathbb{R}; <, =, >)$.

Relations of $(\mathbb{R}; <, =, >)$ are JEPD.

An example of an instance of CSP$(\mathbb{R}; <, =, >)$:

$V = \{x, y, z, w\}$
$C = \{x < y, y < z, x < z, y = w\}$

Solution: $f(x) = 1, f(y) = f(w) = 2, f(z) = 3$.

Constructing Constraint Languages

Let $\mathcal{B}$ be a k-ary constraint language with JEPD relations.

$\mathcal{B}^{\vee=}$ contains unions of all subsets of relations in $\mathcal{B}$.

Example: $(\mathbb{R}; <, =, >)^{\vee=}$ has relations $\{<, >, =, \leq, \neq, \geq, \top, \bot\}$.

Constructing Constraint Languages

Let $\mathcal{B}$ be a k-ary constraint language with JEPD relations.

$\mathcal{B}^{\vee=}$ contains unions of all subsets of relations in $\mathcal{B}$.

Example: $(\mathbb{R}; <, =, >)^{\vee=}$ has relations $\{<, >, =, \leq, \neq, \geq, \top, \bot\}$.

$\langle \mathcal{B} \rangle_{\mathrm{b}}$ contains all relations definable using $\mathcal{B}$-formulas, i.e. logical formulas consisting of the relations in $\mathcal{B}$ and symbols $(,), \wedge, \vee, \neg$.

Example: $\langle (\mathbb{R}; <, =, >) \rangle_{\mathrm{b}}$ contains the relation R_{between} defined as $\{(x, y, z) \in \mathbb{R}^3 \mid (x < y \wedge y < z) \vee (x > y \wedge y > z)\}$.

Constructing Constraint Languages

Let $\mathcal{B}$ be a k-ary constraint language with JEPD relations.

$\mathcal{B}^{\vee=}$ contains unions of all subsets of relations in $\mathcal{B}$.

Example: $(\mathbb{R}; <, =, >)^{\vee=}$ has relations $\{<, >, =, \leq, \neq, \geq, \top, \bot\}$.

$\langle \mathcal{B} \rangle_{\mathrm{b}}$ contains all relations definable using $\mathcal{B}$-formulas, i.e. logical formulas consisting of the relations in $\mathcal{B}$ and symbols $(,), \wedge, \vee, \neg$.

Example: $\langle (\mathbb{R}; <, =, >) \rangle_{\mathrm{b}}$ contains the relation R_{between} defined as $\{(x, y, z) \in \mathbb{R}^3 \mid (x < y \wedge y < z) \vee (x > y \wedge y > z)\}$.

$$\mathcal{B} \subsetneq \mathcal{B}^{\vee=} \subsetneq \langle \mathcal{B} \rangle_{\mathrm{b}}.$$

Certificates: Motivation

Let $\mathcal{B}$ be a constraint language. Consider a finite $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$.

Certificates: Motivation

Let $\mathcal{B}$ be a constraint language. Consider a finite $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$.

If $\mathcal{B}$ has a *finite* domain, then we can enumerate all possible assignments $f : V \to D$ to an instance of CSP(Γ) in $|D|^{|V|}$ time.

Certificates: Motivation

Let $\mathcal{B}$ be a constraint language. Consider a finite $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$.

If $\mathcal{B}$ has a *finite* domain, then we can enumerate all possible assignments $f : V \to D$ to an instance of CSP(Γ) in $|D|^{|V|}$ time.

If $\mathcal{B}$ has an *infinite* domain, then we cannot enumerate all possible assignments in finite time.

Certificates: Motivation

Let $\mathcal{B}$ be a constraint language. Consider a finite $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$.

If $\mathcal{B}$ has a *finite* domain, then we can enumerate all possible assignments $f : V \to D$ to an instance of CSP(Γ) in $|D|^{|V|}$ time.

If $\mathcal{B}$ has an *infinite* domain, then we cannot enumerate all possible assignments in finite time.

However, if $\mathcal{B}$ is k-ary and has JEPD relations, then we can enumerate all **complete certificates**, i.e. all satisfiable instances of CSP($\mathcal{B}$) with constraints over all k-tuples of variables in V.

Certificates: Example

Let $\Gamma \subseteq \langle(\mathbb{R}; <, =, >)\rangle_{\mathrm{b}}$ be a constraint language. An instance of CSP(Γ) with 3 variables x, y, z has 13 complete certificates:

$$x = y, \quad x = z, \quad y = z$$

$$x = y, \quad x < z, \quad y < z$$

$$x = y, \quad x > z, \quad y > z$$

$$x < y, \quad x = z, \quad y > z$$

$$x > y, \quad x = z, \quad y < z$$

$$x < y, \quad x < z, \quad y = z$$

$$x > y, \quad x > z, \quad y = z$$

$$x < y, \quad x < z, \quad y < z$$

$$x < y, \quad x < z, \quad y > z$$

$$x > y, \quad x > z, \quad y > z$$

$$x > y, \quad x > z, \quad y < z$$

$$x < y, \quad x > z, \quad y > z$$

$$x > y, \quad x < z, \quad y < z$$

Parameterized Complexity

Parameterized complexity studies the running time of algorithms with respect to a parameter $p \in \mathbb{N}$ and the input size n.

Parameterized Complexity

Parameterized complexity studies the running time of algorithms with respect to a parameter $p \in \mathbb{N}$ and the input size n.

A problem is in XP if it admits an algorithm of form $O(n^{f(p)})$.
A problem is in FPT if it admits an algorithm of form $f(p) \cdot n^{O(1)}$.
$\mathrm{FPT} \subsetneq \mathrm{XP}$.

Parameterized Complexity

Parameterized complexity studies the running time of algorithms with respect to a parameter $p \in \mathbb{N}$ and the input size n.

A problem is in XP if it admits an algorithm of form $O(n^{f(p)})$.
A problem is in FPT if it admits an algorithm of form $f(p) \cdot n^{O(1)}$.
$\mathrm{FPT} \subsetneq \mathrm{XP}$.

We seek fpt algorithms for infinite-domain CSPs parameterized by the *primal treewidth*.

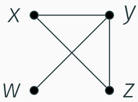
Primal Graph

A *primal graph* associated with an instance (V, C) of CSP has variables V as vertices and an edge for every pair u, v iff u and v appear in the scope of a constraint in C.

Primal Graph

A *primal graph* associated with an instance (V, C) of CSP has variables V as vertices and an edge for every pair u, v iff u and v appear in the scope of a constraint in C.

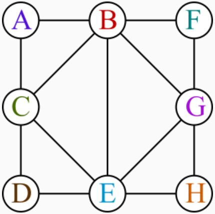
Example: Primal graph of the instance (V, C) of $\mathrm{CSP}(\mathbb{R}; <, =, >)$, where $V = \{x, y, z, w\}$ and $C = \{x < y, y < z, x < z, y = w\}$, is



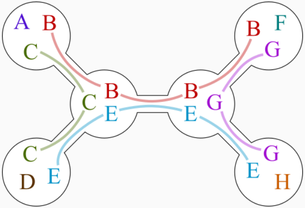
Treewidth

A tree decomposition of a graph G is a tree T and a mapping $X : T \rightarrow 2^V$ such that:

1. If $(u, v) \in E(G)$, then there is $t \in V(T)$ such that $u, v \in X(t)$.
2. For every $v \in V(G)$, the nodes t such that $v \in X(t)$ induce a non-empty connected subtree.



Graph G.

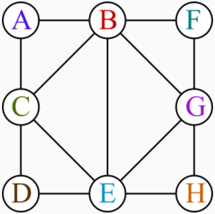


A tree decomposition of G.

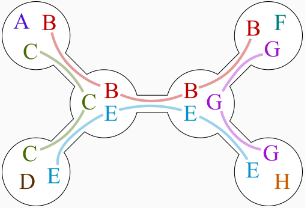
Image Credit: David Eppstein

A tree decomposition of a graph G is a tree T and a mapping $X : T \to 2^V$ that satisfies conditions 1 & 2.

Width of (T, X) is the size of the largest $X(t)$ minus one.
Treewidth of G is the minimum width of a tree decomposition of G.
Primal treewidth is the treewidth of the primal graph.



Graph G.



A tree decomposition of G.

Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.

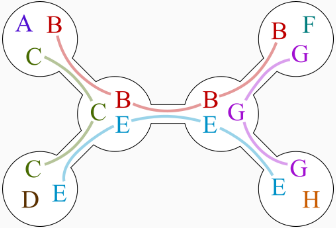
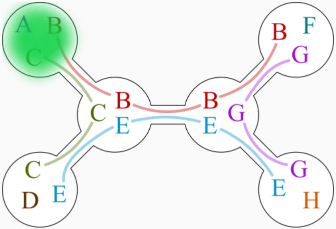


Image Credit: David Eppstein

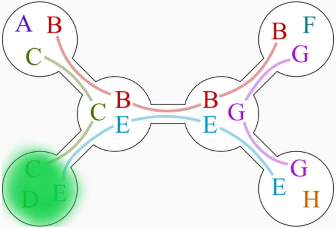
Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.



1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.

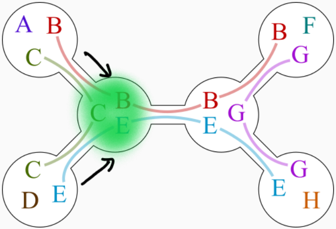


1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.
2. Enumerate $f_2 : \{\mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.

Image Credit: David Eppstein

Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.

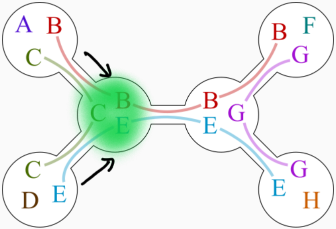


1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.
2. Enumerate $f_2 : \{\mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.
3. Enumerate $f_3 : \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \to D$ that violate no constraints.

Image Credit: David Eppstein

Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.

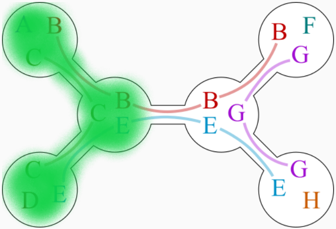


1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.
2. Enumerate $f_2 : \{\mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.
3. Enumerate $f_3 : \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \to D$ that violate no constraints.
4. Remember f_3 only if there are f_1, f_2 such that f_1, f_2, f_3 agree on common variables.

Image Credit: David Eppstein

Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.



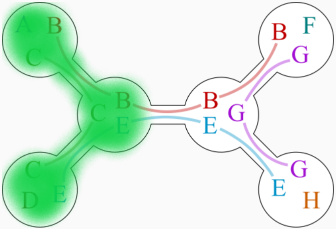
1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.
2. Enumerate $f_2 : \{\mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.
3. Enumerate $f_3 : \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \to D$ that violate no constraints.
4. Remember f_3 only if there are f_1, f_2 such that f_1, f_2, f_3 agree on common variables.

We have enumerated $f : \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.

Image Credit: David Eppstein

Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.



1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.
2. Enumerate $f_2 : \{\mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.
3. Enumerate $f_3 : \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \to D$ that violate no constraints.
4. Remember f_3 only if there are f_1, f_2 such that f_1, f_2, f_3 agree on common variables.

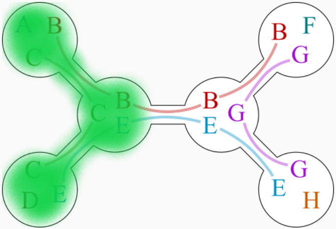
We have enumerated $f : \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.

Continue in the same vein ...

Image Credit: David Eppstein

Patchwork: Motivation

Proposition: Finite-domain CSPs are solvable in $f(w) \cdot n^{O(1)}$ time.



1. Enumerate $f_1 : \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \to D$ that violate no constraints.
2. Enumerate $f_2 : \{\mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.
3. Enumerate $f_3 : \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \to D$ that violate no constraints.
4. Remember f_3 only if there are f_1, f_2 such that f_1, f_2, f_3 agree on common variables.

We have enumerated $f : \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\} \to D$ that violate no constraints.

Continue in the same vein ...

Can we extend this result to **infinte-domain** CSPs?

Patchwork: Motivation

Can we extend this result to infinte-domain CSPs?

We should be able to:

1. enumerate assignments.
2. combine partial assignments.

Patchwork: Motivation

Can we extend this result to infinte-domain CSPs?

We should be able to:

1. enumerate assignments.
2. combine partial assignments.

Q: How to avoid enumerating assignments?

Patchwork: Motivation

Can we extend this result to infinte-domain CSPs?

We should be able to:

1. enumerate assignments.
2. combine partial assignments.

Q: How to avoid enumerating assignments?
A: Enumerate certificates instead.

Patchwork: Motivation

Can we extend this result to infinte-domain CSPs?

We should be able to:

1. enumerate assignments.
2. combine partial assignments.

Q: How to avoid enumerating assignments?
A: Enumerate certificates instead.

Q: How to make step 2 sound?

Patchwork: Motivation

Can we extend this result to infinte-domain CSPs?

We should be able to:

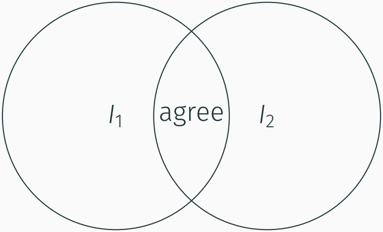
1. enumerate assignments.
2. combine partial assignments.

Q: How to avoid enumerating assignments?
A: Enumerate certificates instead.

Q: How to make step 2 sound?
A: Require the **patchwork property**.

Patchwork: Definition

Definition: A JEPD constraint language $\mathcal{B}$ has *patchwork property* if for every pair of complete satisfiable instances $I_1 = (V_1, C_1)$ and $I_2 = (V_2, C_2)$ of CSP($\mathcal{B}$) such that $I_1[V_1 \cap V_2] = I_2[V_1 \cap V_2]$, the instance $(V_1 \cup V_2, C_1 \cup C_2)$ is also satisfiable.



Fixed-Parameter Tractability

Theorem: Let $\mathcal{B}$ be a finite k-ary constraint language with JEPD relations and the patchwork property. Assume CSP($\mathcal{B}$) is decidable. For any finite constraint language $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$, CSP($\Gamma$) is fpt.

Fixed-Parameter Tractability

Theorem: Let $\mathcal{B}$ be a finite k-ary constraint language with JEPD relations and the patchwork property. Assume CSP($\mathcal{B}$) is decidable. For any finite constraint language $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$, CSP($\Gamma$) is fpt.

More specifically, an instance of CSP(Γ) is solvable in

$$\tau_{\mathcal{B}}(w+1)^2 \cdot w^k \cdot O(n)$$

time, where $\tau_{\mathcal{B}}$ is the time complexity of enumerating complete instances of CSP($\mathcal{B}$).

Consequences

Point Algebra $(\mathbb{R}; <, =, >)$ has patchwork property.
$\tau(v) = 2^{O(v \log v)}$ (# ordered partitions of v elements).

Consequences

Point Algebra $(\mathbb{R}; <, =, >)$ has patchwork property.
$\tau(v) = 2^{O(v \log v)}$ (# ordered partitions of v elements).

Corollary: For every finite $\Gamma \subseteq \langle(\mathbb{R}; <, =, >)\rangle_{\mathrm{b}}$, CSP($\Gamma$) is solvable in $2^{O(w \log w)} \cdot O(n)$ time.

Consequences

Point Algebra $(\mathbb{R}; <, =, >)$ has patchwork property.
$\tau(v) = 2^{O(v \log v)}$ (# ordered partitions of v elements).

Corollary: For every finite $\Gamma \subseteq \langle(\mathbb{R}; <, =, >)\rangle_{\mathrm{b}}$, CSP$(\Gamma)$ is solvable in $2^{O(w \log w)} \cdot O(n)$ time.

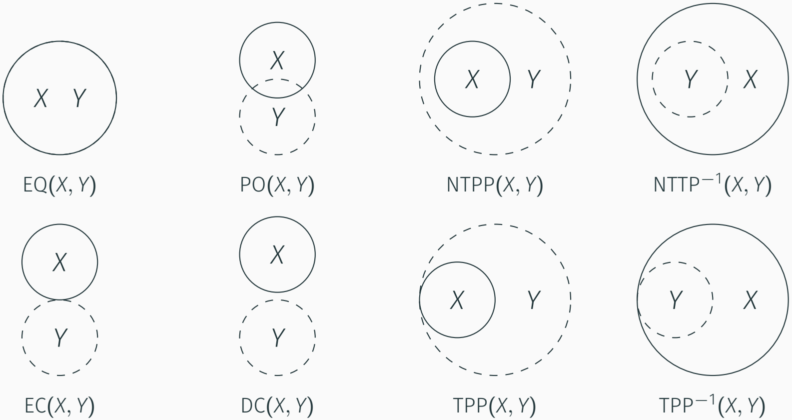
Corollary: For every finite $\Gamma \subseteq \langle(\mathbb{R}; <)\rangle_{\mathrm{fo}}$, CSP$(\Gamma)$ is solvable in $2^{O(w \log w)} \cdot O(n)$ time.

Consequences: Allen's Interval Algebra

Basic relation		Example	Endpoints
I precedes J	p	iii	$I^+ < J^-$
J preceded by I	p^{-1}	jjj	
I meets J	m	$iiii$	$I^+ = J^-$
J met-by I	m^{-1}	$jjjj$	
I overlaps J	o	$iiii$	$I^- < J^- < I^+$,
J overl.-by I	o^{-1}	$jjjj$	$I^+ < J^+$
I during J	d	iii	$I^- > J^-$,
J includes I	d^{-1}	$jjjjjjj$	$I^+ < J^+$
I starts J	s	iii	$I^- = J^-$,
J started by I	s^{-1}	$jjjjjjj$	$I^+ < J^+$
I finishes J	f	iii	$I^+ = J^+$,
J finished by I	f^-1	$jjjjjjj$	$I^- > J^-$
I equals J	e	$iiii$	$I^- = J^-$,
		$jjjj$	$I^+ = J^+$

$\mathrm{AIA} \subseteq \langle(\mathbb{R}; <)\rangle_{\mathrm{fo}} \implies$ CSP($\langle\mathrm{AIA}\rangle_{\mathrm{b}}$) is solvable in $2^{O(w \log w)} \cdot O(n)$.

Consequences: RCC8



$\mathrm{RCC8}$ has patchwork $\implies$ CSP($\langle \mathrm{RCC8} \rangle_{\mathrm{b}}$) is solvable in $2^{O(w^2)} \cdot O(n)$.

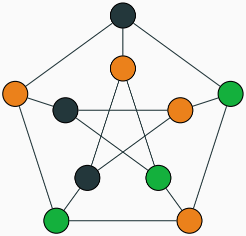
Model-Theoretic Point of View

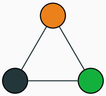
CSP can be thought of a *homomorphism* problem between *relational structures*. A homomorphism is a mapping $h : \mathcal{A} \to \mathcal{B}$ that preserves relations, i.e. if $(a_1, \ldots, a_r) \in R^{\mathcal{A}}$, then $(h(a_1), \ldots, h(a_r)) \in R^{\mathcal{B}}$.

Model-Theoretic Point of View

CSP can be thought of a *homomorphism* problem between *relational structures*. A homomorphism is a mapping $h : \mathcal{A} \to \mathcal{B}$ that preserves relations, i.e. if $(a_1, \ldots, a_r) \in R^{\mathcal{A}}$, then $(h(a_1), \ldots, h(a_r)) \in R^{\mathcal{B}}$.

For example, $\mathrm{CSP}(\{0, 1, 2\}; \{\neq\})$ (aka Graph 3-Coloring) asks whether there is a homomorphism from an input graph G to K_3.





Model-Theoretic Point of View

From the model-theoretic point of view, $\mathcal{B}$ has patchwork if it has the *amalgamation property* (AP).

Theorem: If $\mathcal{B}$ is *homogeneous*, then it has AP.

Homogeneity has been verified for many relational structures.

Model-Theoretic Point of View

ω-categoricity is a more general property than homogeneity.

Bodisky & Dalmau have shown that CSP(Γ) is in XP if $\Gamma \subseteq \langle \mathcal{B} \rangle_{\mathrm{b}}$ and $\mathcal{B}$ is ω-categorical, i.e. solvable in $n^{f(w)}$ time for some computable f.

Question: Is there an ω-categorical relational structure $\mathcal{B}$ such that CSP($\mathcal{B}$) is in XP but not in FPT (under plausible complexity-theoretic assumptions)?

Thank you!