Constraint satisfaction problems over infinite domains

Michael Kompatscher, Trung Van Pham

Theory and Logic Group
TU Wien

Research Seminar, 27/04/2016
Let $\Phi$ be a set of propositional formulas.

**Boolean-SAT($\Phi$)**

**Input:**
- A set of propositional variables $V$ and
- statements $\phi_1, \ldots, \phi_n$ about the variables taken from $\Phi$

**Problem:**
Is $\phi_1 \land \ldots \land \phi_n$ satisfiable?
Let $\Phi$ be a set of propositional formulas.

### Boolean-SAT($\Phi$)

**Input:**
- A set of propositional variables $V$ and
- statements $\phi_1, \ldots, \phi_n$ about the variables taken from $\Phi$

**Problem:**
Is $\phi_1 \land \ldots \land \phi_n$ satisfiable?

Computational complexity is in NP and depends on $\Phi$. 

Schaefer’s theorem
Let $\Phi$ be a set of propositional formulas.

### Boolean-SAT($\Phi$)

**Input:**
- A set of propositional variables $V$ and
- statements $\phi_1, \ldots, \phi_n$ about the variables taken from $\Phi$

**Problem:**
- Is $\phi_1 \land \ldots \land \phi_n$ satisfiable?

Computational complexity is in NP and depends on $\Phi$.

**Schaefer ’78 (1661 citations on Google scholar!)**

Boolean-SAT($\Phi$) is either in P or in NP-complete, for all $\Phi$. 
Schaefer’s theorem for partial orders

Let $\Phi$ be a finite set of quantifier-free $\leq$-formulas.

**Poset-SAT($\Phi$)**

**Input:**
- A set of variables $V$ and
- statements $\phi_1, \ldots, \phi_n$ about the variables taken from $\Phi$

**Problem:**
Is there a partial order that satisfies $\phi_1 \land \ldots \land \phi_n$?

Computational complexity is in NP and depends on $\Phi$.

**Theorem (MK, TVP ’16)**

Poset-SAT($\Phi$) is either in P or in NP-complete, for all $\Phi$. 
Outline

1. Constraint satisfaction problems
2. The universal algebraic approach
3. Poset-SAT
4. Summary
Outline

1. Constraint satisfaction problems
2. The universal algebraic approach
3. Poset-SAT
4. Summary
Informally in a constraint satisfaction problem or CSP the aim is to check if there are objects that satisfy a given set of constraints (e.g. Sudoku, Time scheduling, system of linear equations).
Informally in a constraint satisfaction problem or CSP the aim is to check if there are objects that satisfy a given set of constraints (e.g. Sudoku, Time scheduling, system of linear equations).

\[ \Gamma \ldots \text{structure in relational language } \tau \]
Informally in a constraint satisfaction problem or CSP the aim is to check if there are objects that satisfy a given set of constraints (e.g. Sudoku, Time scheduling, system of linear equations).

$\Gamma$... structure in relational language $\tau$

**CSP($\Gamma$)**

*Input:* A sentence $\exists x_1, \ldots, x_n(\phi_1 \land \cdots \land \phi_k)$ where $\phi_i$ are $\tau$-atomic.
Informally in a constraint satisfaction problem or CSP the aim is to check if there are objects that satisfy a given set of constraints (e.g. Sudoku, Time scheduling, system of linear equations).

\[ \exists x_1, \ldots, x_n (\phi_1 \land \cdots \land \phi_k) \] where \( \phi_i \) are \( \tau \)-atomic.

\textbf{CSP}(\Gamma)

\textit{Input:} A sentence \( \exists x_1, \ldots, x_n (\phi_1 \land \cdots \land \phi_k) \) where \( \phi_i \) are \( \tau \)-atomic.

\textit{Problem:} Does the sentence hold in \( \Gamma \)?
**Informally** in a constraint satisfaction problem or CSP the aim is to check if there are objects that satisfy a given set of constraints (e.g. Sudoku, Time scheduling, system of linear equations).

\[ \Gamma \ldots \text{structure in relational language } \tau \]

**CSP(\(\Gamma\))**

*Input:* A sentence \(\exists x_1, \ldots, x_n(\phi_1 \land \cdots \land \phi_k)\) where \(\phi_i\) are \(\tau\)-atomic.

*Problem:* Does the sentence hold in \(\Gamma\)?

\(\exists x_1, \ldots, x_n(\phi_1 \land \cdots \land \phi_k)\) is called a **primitive positive sentence**.
Informally in a constraint satisfaction problem or CSP the aim is to check if there are objects that satisfy a given set of constraints (e.g. Sudoku, Time scheduling, system of linear equations).

\[ \exists x_1, \ldots, x_n(\phi_1 \land \cdots \land \phi_k) \] where \( \phi_i \) are \( \tau \)-atomic.

\[ \exists x_1, \ldots, x_n(\phi_1 \land \cdots \land \phi_k) \] is called a primitive positive sentence.

**Question**

Given \( \Gamma \), what is the computational complexity of \( \text{CSP}(\Gamma) \)?
Boolean-SAT

2-SAT

*Instance*: A set of 2-clauses \((x, y)\)

*Problem*: Is there a satisfying truth assignment?

\[
\text{CSP}\{\{0, 1\}; 2\text{OR}, \text{NEQ}\} \text{ with } 2\text{OR} = \{(1, 1), (0, 1), (1, 0)\} \text{ and } \text{NEQ} = \{(0, 1), (1, 0)\}.
\]
### Boolean-SAT

#### 2-SAT

*Instance:* A set of 2-clauses \((x, y)\)

*Problem:* Is there a satisfying truth assignment?

\[
\text{CSP}([0, 1]; 2\text{OR}, \text{NEQ}) \text{ with } 2\text{OR} = \{(1, 1), (0, 1), (1, 0)\} \text{ and } \text{NEQ} = \{(0, 1), (1, 0)\}.
\]

#### Positive 1-3-SAT

*Instance:* 3-clauses \((x, y, z)\) with positive literals

*Problem:* Is there a truth assignment such that every clause has exactly one true variable?

\[
\text{CSP}([0, 1]; \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\})
\]
Boolean-SAT

2-SAT

*Instance:* A set of 2-clauses \((x, y)\)

*Problem:* Is there a satisfying truth assignment?

\[
\text{CSP}(\{0, 1\}; 2\text{OR}, \text{NEQ}) \text{ with } 2\text{OR} = \{(1, 1), (0, 1), (1, 0)\} \text{ and } \text{NEQ} = \{(0, 1), (1, 0)\}.
\]

Positive 1-3-SAT

*Instance:* 3-clauses \((x, y, z)\) with positive literals

*Problem:* Is there a truth assignment such that every clause has exactly one true variable?

\[
\text{CSP}(\{0, 1\}; \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\})
\]

CSPs over \(\{0, 1\}\) are exactly the Boolean-SAT(\(\Phi\)) problems.
More examples

CSP(\(\mathbb{Q}, <\))

**Instance:** A pp-sentence in the language \(<\)

**Problem:** Does it hold in \((\mathbb{Q}, <)\)?

**Equivalent to:** Is there a linear order satisfying the pp-sentence?
More examples

**CSP(\(\mathbb{Q}, <\))**

*Instance:* A pp-sentence in the language \(<\)

*Problem:* Does it hold in \((\mathbb{Q}, <)\)?

*Equivalent to:* Is there a linear order satisfying the pp-sentence?

*Instance:* \(\exists x_1, x_2, x_3, x_4 (x_1 < x_2 \land x_1 < x_4 \land x_4 < x_3)\)
More examples

**CSP($\mathbb{Q}, <$)**

*Instance:* A pp-sentence in the language $<$

*Problem:* Does it hold in ($\mathbb{Q}, <$)?

*Equivalent to:* Is there a linear order satisfying the pp-sentence?

*Instance:* $\exists x_1, x_2, x_3, x_4 (x_1 < x_2 \land x_1 < x_4 \land x_4 < x_3)$

![Diagram of CSP($\mathbb{Q}, <$)](attachment://csp_diagram.png)
More examples

\textbf{CSP}(\mathbb{Q}, <)

\textit{Instance:} A pp-sentence in the language $<$

\textit{Problem:} Does it hold in $(\mathbb{Q}, <)$?

\textit{Equivalent to:} Is there a linear order satisfying the pp-sentence?

Instance: $\exists x_1, x_2, x_3, x_4 (x_1 < x_2 \land x_1 < x_4 \land x_4 < x_3)$
**3-COLOR**

*Instance*: A finite graph $(G; E)$

*Problem*: Is it colorable with 3-colors?

*CSP with template* $(K_3, E)$

Instance: $\exists x_1, \ldots, x_5 \ E(x_1, x_2) \land E(x_1, x_4) \land \cdots \land E(x_4, x_5)$
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.

- $\text{CSP}(\Gamma)$ can be in P,

If $|\Gamma| = 2$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Schaefer '78)

If $|\Gamma| = 3$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Bulatov '06)

If $|\Gamma| \geq 4$: ...?
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.

- $\text{CSP}(\Gamma)$ can be in P,
- $\text{CSP}(\Gamma)$ can be NP-complete,
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.

- $\text{CSP}(\Gamma)$ can be in P,
- $\text{CSP}(\Gamma)$ can be NP-complete,

Dichotomy conjecture (Feder, Vardi '99)
For every finite relational structure $\Gamma$, $\text{CSP}(\Gamma)$ is either in P or NP-complete.
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.

- $\text{CSP}(\Gamma)$ can be in P,
- $\text{CSP}(\Gamma)$ can be NP-complete,

**Dichotomy conjecture (Feder, Vardi ’99)**

For every finite relational structure $\Gamma$, $\text{CSP}(\Gamma)$ is either in P or NP-complete.

- If $|\Gamma| = 2$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Schaefer ’78)
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.

- $\text{CSP}(\Gamma)$ can be in P,
- $\text{CSP}(\Gamma)$ can be NP-complete,

**Dichotomy conjecture (Feder, Vardi '99)**

For every finite relational structure $\Gamma$, $\text{CSP}(\Gamma)$ is either in P or NP-complete.

- If $|\Gamma| = 2$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Schaefer '78)
- If $|\Gamma| = 3$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Bulatov '06)
Finite CSPs

If $\Gamma$ is finite, $\text{CSP}(\Gamma)$ is in NP.

- $\text{CSP}(\Gamma)$ can be in P,
- $\text{CSP}(\Gamma)$ can be NP-complete,

Dichotomy conjecture (Feder, Vardi '99)

For every finite relational structure $\Gamma$, $\text{CSP}(\Gamma)$ is either in P or NP-complete.

- If $|\Gamma| = 2$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Schaefer '78)
- If $|\Gamma| = 3$: $\text{CSP}(\Gamma)$ is in P or NP-complete (Bulatov '06)
- If $|\Gamma| \geq 4$: ...?
Outline

1. Constraint satisfaction problems
2. The universal algebraic approach
3. Poset-SAT
4. Summary
For structures $\Gamma$, $\Delta$ write $\Gamma \leq_{pp} \Delta$ if every relation in $\Gamma$ has a definition with primitive positive formulas in $\Delta$. 
Primitive positive definability

For structures $\Gamma$, $\Delta$ write $\Gamma \leq_{pp} \Delta$ if every relation in $\Gamma$ has a definition with primitive positive formulas in $\Delta$.

**Essential observation**

$$\Gamma \leq_{pp} \Delta \rightarrow \text{CSP}(\Gamma) \leq_{ptime} \text{CSP}(\Delta)$$
For structures $\Gamma$, $\Delta$ write $\Gamma \leq_{pp} \Delta$ if every relation in $\Gamma$ has a definition with primitive positive formulas in $\Delta$.

**Essential observation**

$$\Gamma \leq_{pp} \Delta \rightarrow \text{CSP}(\Gamma) \leq_{ptime} \text{CSP}(\Delta)$$

If $\Gamma \leq_{pp} \Delta$ and $\Delta \leq_{pp} \Gamma$ the problems $\text{CSP}(\Gamma)$ and $\text{CSP}(\Delta)$ are ptime equivalent.
For structures $\Gamma$, $\Delta$ write $\Gamma \leq_{pp} \Delta$ if every relation in $\Gamma$ has a definition with primitive positive formulas in $\Delta$.

\textbf{Essential observation}

$$\Gamma \leq_{pp} \Delta \rightarrow \text{CSP}(\Gamma) \leq_{\text{ptime}} \text{CSP}(\Delta)$$

If $\Gamma \leq_{pp} \Delta$ and $\Delta \leq_{pp} \Gamma$ the problems CSP($\Gamma$) and CSP($\Delta$) are ptime equivalent.

We only need to study structures up to pp-interdefinable.
Polymorphism clones

We say a function $f : D^n \rightarrow D$ preserves a relation $R \subseteq D^k$ if for all $\bar{r}_1, \ldots, \bar{r}_n \in R$ also $f(\bar{r}_1, \ldots, \bar{r}_n) \in R$. 
We say a function $f : D^n \rightarrow D$ preserves a relation $R \subseteq D^k$ if for all $\bar{r}_1, \ldots, \bar{r}_n \in R$ also $f(\bar{r}_1, \ldots, \bar{r}_n) \in R$.

A function $f : \Gamma^n \rightarrow \Gamma$ is called a polymorphism if it preserves all relations in $\Gamma$. 
Polymorphism clones

We say a function $f : D^n \rightarrow D$ preserves a relation $R \subseteq D^k$ if for all $\bar{r}_1, \ldots, \bar{r}_n \in R$ also $f(\bar{r}_1, \ldots, \bar{r}_n) \in R$.

A function $f : \Gamma^n \rightarrow \Gamma$ is called a polymorphism if it preserves all relations in $\Gamma$. The set of all polymorphisms is called polymorphism clone $\text{Pol}(\Gamma)$. 
Polymorphism clones

We say a function \( f : D^n \rightarrow D \) preserves a relation \( R \subseteq D^k \) if for all \( \bar{r}_1, \ldots, \bar{r}_n \in R \) also \( f(\bar{r}_1, \ldots, \bar{r}_n) \in R \).

A function \( f : \Gamma^n \rightarrow \Gamma \) is called a polymorphism if it preserves all relations in \( \Gamma \). The set of all polymorphisms is called polymorphism clone \( \text{Pol}(\Gamma) \).

Theorem (Geiger '68)

For finite structures \( \Delta \) and \( \Gamma \):

\[
\Delta \leq_{pp} \Gamma \iff \text{Pol}(\Delta) \supseteq \text{Pol}(\Gamma)
\]
Polymorphism clones

We say a function \( f : D^n \to D \) preserves a relation \( R \subseteq D^k \) if for all \( \bar{r}_1, \ldots, \bar{r}_n \in R \) also \( f(\bar{r}_1, \ldots, \bar{r}_n) \in R \).

A function \( f : \Gamma^n \to \Gamma \) is called a polymorphism if it preserves all relations in \( \Gamma \). The set of all polymorphisms is called polymorphism clone \( \text{Pol}(\Gamma) \).

**Theorem (Geiger ’68)**

For finite structures \( \Delta \) and \( \Gamma \):

\[
\Delta \preceq_{pp} \Gamma \iff \text{Pol}(\Delta) \supseteq \text{Pol}(\Gamma)
\]

\( \rightarrow \) the complexity of \( \text{CSP}(\Gamma) \) is determined by \( \text{Pol}(\Gamma) \)!
Schaefer’s theorem revisited

The Boolean CSP(Γ) is in P if and only if

<table>
<thead>
<tr>
<th>All relations in Γ</th>
<th>Pol(Γ) contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>contain (0, ..., 0)</td>
<td>constant 0</td>
</tr>
<tr>
<td>contain (1, ..., 1)</td>
<td>constant 1</td>
</tr>
<tr>
<td>are Horn</td>
<td>(x, y) → x ∧ y</td>
</tr>
<tr>
<td>are dual Horn</td>
<td>(x, y) → x ∨ y</td>
</tr>
<tr>
<td>are affine</td>
<td>(x, y, z) → x − y + z</td>
</tr>
<tr>
<td>are 2-clauses</td>
<td>(x, y, z) → (x ∨ y) ∧ (x ∨ z) ∧ (y ∨ z)</td>
</tr>
</tbody>
</table>
Schaefer’s theorem revisited

The Boolean CSP($\Gamma$) is in P if and only if

<table>
<thead>
<tr>
<th>All relations in $\Gamma$</th>
<th>$\text{Pol}(\Gamma)$ contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>contain $(0, \ldots, 0)$</td>
<td>constant 0</td>
</tr>
<tr>
<td>contain $(1, \ldots, 1)$</td>
<td>constant 1</td>
</tr>
<tr>
<td>are Horn</td>
<td>$(x, y) \rightarrow x \land y$</td>
</tr>
<tr>
<td>are dual Horn</td>
<td>$(x, y) \rightarrow x \lor y$</td>
</tr>
<tr>
<td>are affine</td>
<td>$(x, y, z) \rightarrow x - y + z$</td>
</tr>
<tr>
<td>are 2-clauses</td>
<td>$(x, y, z) \rightarrow (x \lor y) \land (x \lor z) \land (y \lor z)$</td>
</tr>
</tbody>
</table>

Tractability conjecture (Bulatov, Jeavons, Krokhin,...)

Let $\Gamma$ be finite (+ mc core, contains all constants). Then either

- $\exists f \in \text{Pol}(\Gamma): f(x_1, x_2, \ldots, x_n) = f(x_2, x_3, \ldots, x_n, x_1)$
  and CSP($\Gamma$) is in P
- or CSP($\Gamma$) is NP-complete.
The lattice of all clones on \( \{0, 1\} \)
Infinite CSPs

If $\Gamma$ is infinite, CSP($\Gamma$) can be undecidable:

**Diophant**

*Instance:* Equations using 0, 1, +, ·

*Problem:* Is there an integer solution?

CSP($\mathbb{Z}; 0, 1, +, \cdot$).
**Infinite CSPs**

If $\Gamma$ is infinite, $\text{CSP}(\Gamma)$ can be undecidable:

<table>
<thead>
<tr>
<th>Diophant</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instance:</strong> Equations using $0, 1, +, \cdot$</td>
</tr>
<tr>
<td><strong>Problem:</strong> Is there an integer solution?</td>
</tr>
</tbody>
</table>

$\text{CSP}(\mathbb{Z}; 0, 1, +, \cdot)$.

For $|\Gamma| = \omega$ all complexities can appear!
Infinite CSPs

If $\Gamma$ is infinite, $\text{CSP}(\Gamma)$ can be undecidable:

**Diophant**

*Instance:* Equations using 0, 1, +, $\cdot$  
*Problem:* Is there an integer solution?  

$\text{CSP}(\mathbb{Z}; 0, 1, +, \cdot)$.  

For $|\Gamma| = \omega$ all complexities can appear!  

Clone lattice on $\omega$ is complicated...
Infinite CSPs

If $\Gamma$ is infinite, $\text{CSP}(\Gamma)$ can be undecidable:

**Diophant**

*Instance:* Equations using $0, 1, +, \cdot$

*Problem:* Is there an integer solution?

$\text{CSP}(\mathbb{Z}; 0, 1, +, \cdot)$.

For $|\Gamma| = \omega$ all complexities can appear!

Clone lattice on $\omega$ is complicated...

$\text{Pol}(\Gamma) \supseteq \text{Pol}(\Delta)$ does not imply $\Gamma \leq_{pp} \Delta$ in general.
If $\Gamma$ is infinite, $\text{CSP}(\Gamma)$ can be undecidable:

**Diophant**

*Instance:* Equations using $0, 1, +, \cdot$
*Problem:* Is there an integer solution?

$\text{CSP}(\mathbb{Z}; 0, 1, +, \cdot)$.

For $|\Gamma| = \omega$ all complexities can appear!

Clone lattice on $\omega$ is complicated...
$\text{Pol}(\Gamma) \supseteq \text{Pol}(\Delta)$ does not imply $\Gamma \leq_{pp} \Delta$ in general.

Hope: Algebraic approach still works for “nice” structures.
Outline

1. Constraint satisfaction problems
2. The universal algebraic approach
3. Poset-SAT
4. Summary
The random partial order \( \mathbb{P} := (P; \leq) \) is the unique countable partial order that:

---

\textbf{Poset-SAT as CSP}

---

The random partial order \( \mathbb{P} := (P; \leq) \) is the unique countable partial order that:
The random partial order $\mathbb{P} := (P; \leq)$ is the unique countable partial order that:

- is *universal*, i.e., contains all finite partial orders
The random partial order $\mathbb{P} := (P; \leq)$ is the unique countable partial order that:

- is *universal*, i.e., contains all finite partial orders
- is *homogeneous*, i.e. for finite $A, B \subseteq P$, every isomorphism $I : A \rightarrow B$ extends to an automorphism $\alpha \in \text{Aut}(\mathbb{P})$. 

For every $\{\leq\}$-formula $\phi(x_1, \ldots, x_n)$ we define the relation $R_\phi := \{(a_1, \ldots, a_n) \in \mathbb{P}^n : \phi(a_1, \ldots, a_n)\}$. 

$\text{Poset-SAT}(\Phi) = \text{CSP}(\mathbb{P}; R_\phi \mid \phi \in \Phi)$. 

$(\mathbb{P}; R_\phi \mid \phi \in \Phi)$ is a reduct of $\mathbb{P}$, i.e. a structure that is first-order definable in $\mathbb{P}$. 
Poset-SAT as CSP

The random partial order $\mathbb{P} := (P; \leq)$ is the unique countable partial order that:

- is *universal*, i.e., contains all finite partial orders
- is *homogeneous*, i.e., for finite $A, B \subseteq P$, every isomorphism $I : A \rightarrow B$ extends to an automorphism $\alpha \in \text{Aut}(P)$.

For every $\{\leq\}$-formula $\phi(x_1, \ldots, x_n)$ we define the relation

$$R_\phi := \{(a_1, \ldots, a_n) \in P^n : \phi(a_1, \ldots, a_n)\}.$$
The random partial order $\mathbb{P} := (P; \leq)$ is the unique countable partial order that:

- is *universal*, i.e., contains all finite partial orders
- is *homogeneous*, i.e. for finite $A, B \subseteq P$, every isomorphism $I : A \to B$ extends to an automorphism $\alpha \in \text{Aut}(\mathbb{P})$.

For every $\{\leq\}$-formula $\phi(x_1, \ldots, x_n)$ we define the relation

$$R_\phi := \{(a_1, \ldots, a_n) \in P^n : \phi(a_1, \ldots, a_n)\}.$$
The random partial order \( \mathbb{P} := (P; \leq) \) is the unique countable partial order that:

- is *universal*, i.e., contains all finite partial orders
- is *homogeneous*, i.e. for finite \( A, B \subseteq P \), every isomorphism \( I : A \to B \) extends to an automorphism \( \alpha \in \text{Aut}(\mathbb{P}) \).

For every \( \{\leq\} \)-formula \( \phi(x_1, \ldots, x_n) \) we define the relation

\[
R_\phi := \{(a_1, \ldots, a_n) \in P^n : \phi(a_1, \ldots, a_n) \}.
\]

\( \text{Poset-SAT}(\Phi) = \text{CSP}((P; R_\phi)_{\phi \in \Phi}) \).

\( (P; R_\phi)_{\phi \in \Phi} \) is a *reduct* of \( \mathbb{P} \), i.e. a structure that is first-order definable in \( \mathbb{P} \).
CSPs over random partial order

**ω-categorical structure**

A structure $\Gamma$ is called $\omega$-categorical, if its theory has, up to isomorphism, exactly one countable model.

**Engeler, Ryll-Nardzewski, Svenonius**

An countably infinite structure $\Gamma$ with countable signature is $\omega$-categorical if and only if for every $k \in \mathbb{N}$, there are finitely many $k$-orbits of $\text{Aut}(\Gamma)$. 
CSPs over random partial order

**ω-categorical structure**

A structure $\Gamma$ is called $\omega$-categorical, if its theory has, up to isomorphism, exactly one countable model.

**Engeler, Ryll-Nardzewski, Svenonius**

An countably infinite structure $\Gamma$ with countable signature is $\omega$-categorical if and only if for every $k \in \mathbb{N}$, there are finitely many $k$-orbits of $\text{Aut}(\Gamma)$.

**Why is $\mathbb{P}$ $\omega$-categorical?**

For every $k \in \mathbb{N}$, there are finitely many posets on $k$ elements.
CSPs over random partial order

\(\omega\)-categorical structure

A structure \(\Gamma\) is called \(\omega\)-categorical, if its theory has, up to isomorphism, exactly one countable model.

Engeler, Ryll-Nardzewski, Svenonius

An countably infinite structure \(\Gamma\) with countable signature is \(\omega\)-categorical if and only if for every \(k \in \mathbb{N}\), there are finitely many \(k\)-orbits of \(\text{Aut}(\Gamma)\).

Bodirsky, Nešetřil '03

For \(\omega\)-categorical structures \(\Gamma, \Delta\) we have

\[
\Gamma \leq_{pp} \Delta \iff \text{Pol}(\Gamma) \supseteq \text{Pol}(\Delta)
\]
Outline

1. Constraint satisfaction problems
2. The universal algebraic approach
3. Poset-SAT
4. Summary
Strategy for Poset-SAT(\(\Phi\))

\[
\begin{align*}
\text{Boolean-SAT}(\Phi) & \quad \Downarrow \\
\text{CSPs of Boolean structures } (\{0, 1\}; R_1, \ldots R_n) & \quad \Downarrow \\
\text{are reducts of } (\{0, 1\}, 0, 1) & \\
\Downarrow \\
\text{Clones over } \{0, 1\} &
\end{align*}
\]

\[
\begin{align*}
\text{Poset-SAT}(\Phi) & \quad \Downarrow \\
\text{CSPs of reducts of random partial order } \mathbb{P} & \\
\Downarrow \\
\text{Closed clones containing } \text{Aut}(\mathbb{P}) &
\end{align*}
\]
### Important NP-complete relations

- **Betw** \((x, y, z)\) := \(x < y < z \lor z < y < x\).
- **Cycl** \((x, y, z)\) := \((x < y \land y < z) \lor (z < x \land x < y) \lor (y < z \land z < x) \lor (x < y \land z \perp x \land z \perp y) \lor (y < z \land x \perp y \land x \perp z) \lor (z < x \land y \perp z \land y \perp x)\).
- **Sep** \((x, y, z, t)\) :=
  
  \[ ((\text{Cycl}(x, y, z) \land \text{Cycl}(y, z, t) \land \text{Cycl}(x, y, t) \land \text{Cycl}(x, z, t)) \lor (\text{Cycl}(z, y, x) \land \text{Cycl}(t, z, y) \land \text{Cycl}(t, y, x) \land \text{Cycl}(t, z, x))). \]
- **Low** \((x, y, z)\) := \((x < y \land x \perp z \land y \perp z) \lor (x < z \land x \perp y \land z \perp y)\).
Theorem (MK, TVP ’16)

Let $\Gamma$ be reduct of $\mathbb{P}$. Then one of the following cases holds:

1. $\text{CSP}(\Gamma)$ can be reduced to a CSP of a reduct of $(\mathbb{Q}; \leq)$. Thus $\text{CSP}(\Gamma)$ is in P or NP-complete (M. Bodirsky and J. K´ara).

2. Low, Betw, Cycl or Sep is pp-definable in $\Gamma$ and $\text{CSP}(\Gamma)$ is NP-complete.

3. $\text{Pol}(\Gamma)$ contains functions $f, g_1, g_2$ such that $g_1(f(x, y)) = g_2(f(y, x))$ and $\text{CSP}(\Gamma)$ can be solved in polynomial time.

Consequence: Poset-SAT($\Phi$) is in P or NP-complete.

Given $\Phi$, it is decidable to tell if Poset-SAT($\Phi$) is in P.
Complexity dichotomy

Theorem (MK, TVP ’16)

Let \( \Gamma \) be reduct of \( \mathbb{P} \). Then one of the following cases holds:

- CSP(\( \Gamma \)) can be reduced to a CSP of a reduct of \((\mathbb{Q}; \leq)\). Thus CSP(\( \Gamma \)) is in \( \mathbb{P} \) or NP-complete (M. Bodirsky and J. Kára).

Consequence:
Poset-SAT(\( \Phi \)) is in \( \mathbb{P} \) or NP-complete.
Constraint satisfaction

The universal algebraic approach

Poset-SAT

Summary

Complexity dichotomy

Theorem (MK, TVP ’16)

Let \( \Gamma \) be reduct of \( \mathbb{P} \). Then one of the following cases holds:

- CSP(\( \Gamma \)) can be reduced to a CSP of a reduct of \((\mathbb{Q}; \leq)\). Thus CSP(\( \Gamma \)) is in P or NP-complete (M. Bodirsky and J. Kára).
- Low, Betw, Cycl or Sep is pp-definable in \( \Gamma \) and CSP(\( \Gamma \)) is NP-complete.
Theorem (MK, TVP ’16)

Let $\Gamma$ be reduct of $\mathbb{P}$. Then one of the following cases holds:

- CSP($\Gamma$) can be reduced to a CSP of a reduct of $(\mathbb{Q}; \leq)$. Thus CSP($\Gamma$) is in P or NP-complete (M. Bodirsky and J. Kára).
- Low, Betw, Cycl or Sep is pp-definable in $\Gamma$ and CSP($\Gamma$) is NP-complete.
- Pol($\Gamma$) contains functions $f$, $g_1$, $g_2$ such that

$$g_1(f(x, y)) = g_2(f(y, x))$$

and CSP($\Gamma$) can be solved in polynomial time.

Consequence:

Poset-SAT($\Phi$) is in P or NP-complete.
Complexity dichotomy

**Theorem (MK, TVP ’16)**

Let $\Gamma$ be reduct of $\mathbb{P}$. Then one of the following cases holds:

- CSP($\Gamma$) can be reduced to a CSP of a reduct of $(\mathbb{Q}; \leq)$. Thus CSP($\Gamma$) is in $\mathbb{P}$ or NP-complete (M. Bodirsky and J. Kára).
- Low, Betw, Cycl or Sep is pp-definable in $\Gamma$ and CSP($\Gamma$) is NP-complete.
- Pol($\Gamma$) contains functions $f$, $g_1$, $g_2$ such that
  \[ g_1(f(x, y)) = g_2(f(y, x)) \]

and CSP($\Gamma$) can be solved in polynomial time.

**Consequence:**

Poset-SAT($\Phi$) is in $\mathbb{P}$ or NP-complete.
Given $\Phi$, it is decidible to tell if Poset-SAT($\Phi$) is in $\mathbb{P}$.
The method for the classification

Canonicalization theorem (Bodirsky, Pinsker and Tsankov, 2012)

Let $\Delta$ be ordered homogeneous Ramsey with finite relational signature, $f : \Delta \rightarrow \Delta$, and let $c_1, c_2, \ldots, c_n \in \Delta$. Then $f$ generates over $\Delta$ a function which agrees with $f$ on $\{c_1, c_2, \ldots, c_n\}$ and which is canonical as a function from $(\Delta, c_1, c_2, \ldots, c_n)$. 
The method for the classification

**Canonical functions**

A function \( f : P^2 \rightarrow P \) is called **canonical** if the type of image depends only on the types of arguments of the function in the domain.

**Example**

<table>
<thead>
<tr>
<th>( e_\leq )</th>
<th>( = )</th>
<th>( &lt; )</th>
<th>( &gt; )</th>
<th>( \bot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( = )</td>
<td>( = )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( &lt; )</td>
<td>( \bot )</td>
<td>( &lt; )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( &gt; )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( &gt; )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
</tbody>
</table>

Embedding from \((P; <)^2\) to \((P; <)\).

<table>
<thead>
<tr>
<th>( e_\leq )</th>
<th>( = )</th>
<th>( &lt; )</th>
<th>( &gt; )</th>
<th>( \bot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( = )</td>
<td>( = )</td>
<td>( &lt; )</td>
<td>( &gt; )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( &lt; )</td>
<td>( &lt; )</td>
<td>( &lt; )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( &gt; )</td>
<td>( &gt; )</td>
<td>( \bot )</td>
<td>( &gt; )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
</tbody>
</table>

Embedding from \((P; \leq)^2\) to \((P; \leq)\).
The method for the classification

Canonical functions

A function $f : P^2 \rightarrow P$ is called canonical if the type of image depends only on the types of arguments of the function in the domain.

Example

<table>
<thead>
<tr>
<th>$e_&lt;$</th>
<th>$=$</th>
<th>$&lt;$</th>
<th>$&gt;$</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$=$</td>
<td>$=$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$&lt;$</td>
<td>$\bot$</td>
<td>$&lt;$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$&gt;$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$&gt;$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
</tbody>
</table>

Embedding from $(P; <)^2$ to $(P; <)$.

<table>
<thead>
<tr>
<th>$e_\leq$</th>
<th>$=$</th>
<th>$&lt;$</th>
<th>$&gt;$</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$=$</td>
<td>$=$</td>
<td>$&lt;$</td>
<td>$&gt;$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$&lt;$</td>
<td>$&lt;$</td>
<td>$&lt;$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$&gt;$</td>
<td>$&gt;$</td>
<td>$\bot$</td>
<td>$&gt;$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
</tbody>
</table>

Embedding from $(P; \leq)^2$ to $(P; \leq)$.

for every $x < y$ and $x' > y'$, we have $e_<(x, x') \bot e_<(y, y')$. 
The method for the classification

Lemma

Let \( \Gamma \) be a reduct of \((P; \leq)\). If \( <, \perp \in \langle \Gamma \rangle_{pp}, \text{Low} \notin \langle \Gamma \rangle_{pp} \), then \( e_\prec \) or \( e_\leq \) is a polymorphism of \( \Gamma \).

Proof

1. Since \text{Low} is not primitive positive definable in \( \Gamma \), there is a binary polymorphism \( f \) of \( \Gamma \) that violates \text{Low}.

2. We can find three elements \( a, b, c \in P \) such that \( a < b \land ab \perp c \), and \((f(a,a), f(b,c), f(c,b)) \notin \text{Low} \).

3. We can assume that \( f \) is canonical as a function from \((P; \leq, \preceq, a, b, c)^2 \) to \((P; \leq, \preceq, a, b, c) \).

4. Use an extensive combinatorial analysis on \( f \). . .
The method for classification

Using the same method one could successfully classify the complexity of a number of CSPs on infinite domains.

1. Graph-SAT (M. Bodirsky and M. Pinsker, 2015).
2. Phylogeny CSPs (M. Bodirsky, P. Jonsson and T. V. Pham, 2015).
3. Henson graphs (M. Bodirsky, B. Martin, M. Pinsker and A. Pongrács, 2016).
4. Semilinear order-SAT (M. Bodirsky and T. V. Pham, in preparation).
Lattice of polymorphism clones containing $\text{Aut}(\mathcal{P})$
Thank you!