Schaefer’s theorem for graphs

Why to consult the infinite at times

Michael Pinsker

Université Diderot - Paris 7

Tel Aviv University, May 2012
Outline

Part I
Graph-SAT problems

Part II
Making the finite infinite
CSPs of reducts of the random graph

Part III
Making the infinite finite
Ramsey theory and canonical functions

Part IV
The Graph-SAT dichotomy

Part V
The future
CSPs over homogeneous structures

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Outline

Part I
Graph-SAT problems
Outline

- **Part I**
  - Graph-SAT problems

- **Part II**
  - Making the finite infinite
  - CSPs of reducts of the random graph
Outline

- **Part I**
  - Graph-SAT problems

- **Part II**
  - Making the finite infinite
  - CSPs of reducts of the random graph

- **Part III**
  - Making the infinite finite
  - Ramsey theory and canonical functions
Outline

Part I
Graph-SAT problems

Part II
Making the finite infinite
CSPs of reducts of the random graph

Part III
Making the infinite finite
Ramsey theory and canonical functions

Part IV
The Graph-SAT dichotomy
Outline

- **Part I**
  Graph-SAT problems

- **Part II**
  Making the finite infinite
  CSPs of reducts of the random graph

- **Part III**
  Making the infinite finite
  Ramsey theory and canonical functions

- **Part IV**
  The Graph-SAT dichotomy

- **Part V**
  The future
  CSPs over homogeneous structures
Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Part I

Graph-SAT problems
Boolean satisfiability problems

Let $\Psi$ be a finite set of propositional formulas.

Computational problem: Boolean-SAT($\Psi$)

INPUT: A set $W$ of propositional variables, and statements $\phi_1, \ldots, \phi_n$ about the variables in $W$, where each $\phi_i$ is taken from $\Psi$.

QUESTION: Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable?

Computational complexity depends on $\Psi$. Always in NP.

Theorem (Schaefer STOC'78)

1139 citations on google scholar

Boolean-SAT($\Psi$) is either in P or NP-complete, for all $\Psi$.

Schaefer’s theorem for graphs

Michael Pinsker (Paris 7)
Boolean satisfiability problems

Let $\Psi$ be a finite set of propositional formulas.
Boolean satisfiability problems

Let $\Psi$ be a finite set of propositional formulas.

Computational problem: Boolean-SAT($\Psi$)

**INPUT:**
- A set $W$ of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable?
Boolean satisfiability problems

Let $\Psi$ be a finite set of propositional formulas.

**Computational problem: Boolean-SAT($\Psi$)**

**INPUT:**
- A set $W$ of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable?

Computational complexity depends on $\Psi$. Always in NP.

Schaefer’s theorem for graphs

Michael Pinsker  (Paris 7)
Boolean satisfiability problems

Let $\Psi$ be a finite set of propositional formulas.

**Computational problem: Boolean-SAT($\Psi$)**

**INPUT:**
- A set $W$ of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable?

Computational complexity depends on $\Psi$. Always in NP.

**Theorem (Schaefer STOC’78)**

Boolean-SAT($\Psi$) is either in P or NP-complete, for all $\Psi$. 

Schaefer’s theorem for graphs

Michael Pinsker (Paris 7)
Boolean satisfiability problems

Let $\Psi$ be a finite set of propositional formulas.

Computational problem: Boolean-SAT($\Psi$)

**INPUT:**
- A set $W$ of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable?

Computational complexity depends on $\Psi$. Always in NP.

**Theorem (Schaefer STOC’78)** 1139 citations on google scholar

Boolean-SAT($\Psi$) is either in P or NP-complete, for all $\Psi$. 
Let $E$ be a binary relation symbol. (Imagine: edge relation of an undirected graph.) Let $\Psi$ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT($\Psi$)

**INPUT:** A finite set $W$ of variables (vertices), and statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

Computational complexity depends on $\Psi$. Always in NP.

**Question** For which $\Psi$ is Graph-SAT($\Psi$) tractable?

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Let $E$ be a binary relation symbol.
(Imagine: edge relation of an undirected graph.)
Let $\Psi$ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT($\Psi$)

**INPUT:** A finite set $W$ of variables (vertices), and statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

Computational complexity depends on $\Psi$. Always in NP.

**Question** For which $\Psi$ is Graph-SAT($\Psi$) tractable?

Schaefer’s theorem for graphs
Michael Pinsker (Paris 7)
Graph satisfiability problems

Let $E$ be a binary relation symbol.
(Imagine: edge relation of an undirected graph.)
Let $\Psi$ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT($\Psi$)

INPUT:
- A finite set $W$ of variables (vertices), and
- statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\Psi$.

QUESTION: Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Graph satisfiability problems

Let $E$ be a binary relation symbol.  
(Imagine: edge relation of an undirected graph.)
Let $\psi$ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT($\psi$)

**INPUT:**
- A finite set $W$ of variables (vertices), and
- statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

Computational complexity depends on $\psi$. Always in NP.
Graph satisfiability problems

Let $E$ be a binary relation symbol.
(Imagine: edge relation of an undirected graph.)
Let $\Psi$ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT($\Psi$)

INPUT:
- A finite set $W$ of variables (vertices), and
- statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\Psi$.

QUESTION: Is $\wedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

Computational complexity depends on $\Psi$. Always in NP.

Question
For which $\Psi$ is Graph-SAT($\Psi$) tractable?
Graph-SAT: Examples

Example 1

Let $\Psi_1$ only contain $\psi_1(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \lor (E(x, y) \land E(y, z) \land E(x, z))$.

Graph-SAT($\Psi_1$) is NP-complete.

Example 2

Let $\Psi_2$ only contain $\psi_2(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \lor (E(x, y) \land E(y, z) \land E(x, z))$.

Graph-SAT($\Psi_2$) is in P.
Example 1  Let $\Psi_1$ only contain

$$\psi_1(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z))$$
$$\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z))$$
$$\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) .$$

Graph-SAT($\Psi_1$) is NP-complete.

Example 2  Let $\Psi_2$ only contain

$$\psi_2(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z))$$
$$\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z))$$
$$\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \lor (E(x, y) \land E(y, z) \land E(x, z)) .$$

Graph-SAT($\Psi_2$) is in P.
**Example 1** Let $\Psi_1$ only contain

\[
\psi_1(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) .
\]

Graph-SAT($\Psi_1$) is NP-complete.
Graph-SAT: Examples

Example 1  Let $\Psi_1$ only contain

\[
\psi_1(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \\
\lor (E(x, y) \land E(y, z) \land E(x, z)) .
\]

Graph-SAT($\Psi_1$) is NP-complete.

Example 2  Let $\Psi_2$ only contain

\[
\psi_2(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \\
\lor (E(x, y) \land E(y, z) \land E(x, z)) .
\]
Graph-SAT: Examples

**Example 1** Let $\Psi_1$ only contain

\[
\psi_1(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \\
\lor (E(x, y) \land E(y, z) \land E(x, z)).
\]

Graph-SAT($\Psi_1$) is NP-complete.

**Example 2** Let $\Psi_2$ only contain

\[
\psi_2(x, y, z) := (E(x, y) \land \neg E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land E(y, z) \land \neg E(x, z)) \\
\lor (\neg E(x, y) \land \neg E(y, z) \land E(x, z)) \\
\lor (E(x, y) \land E(y, z) \land E(x, z)).
\]

Graph-SAT($\Psi_2$) is in P.
Part II

Making the finite infinite

CSPs over the random graph
Graph formulas and reducts of the random graph

Let $G = (V, E)$ denote the random graph, i.e., the unique countably infinite graph which is universal, i.e., all finite graphs are induced subgraphs of $G$; homogeneous, i.e., for all finite $A, B \subseteq G$, for all isomorphisms $i: A \to B$, there exists $\alpha \in \text{Aut}(G)$ extending $i$.

For a graph formula $\psi(x_1, \ldots, x_n)$, define a relation $R_\psi := \{(a_1, \ldots, a_n) \in V^n : \psi(a_1, \ldots, a_n)\}$. For a set $\Psi$ of graph formulas, define a structure $\Gamma_\Psi := (V; (R_\psi : \psi \in \Psi))$. $\Gamma_\Psi$ is a reduct of the random graph, i.e., a structure with a first-order definition in $G$.
Graph formulas and reducts of the random graph

Let $G = (V; E)$ denote the random graph, i.e., the unique countably infinite graph which is

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Let \( G = (V; E) \) denote the random graph, i.e., the unique countably infinite graph which is

- **universal**, i.e., all finite graphs are induced subgraphs of \( G \);
Graph formulas and reducts of the random graph

Let $G = (V; E)$ denote the random graph, i.e., the unique countably infinite graph which is

- universal, i.e., all finite graphs are induced subgraphs of $G$;
- homogeneous, i.e.,
  For all finite $A, B \subseteq G$, for all isomorphisms $i : A \rightarrow B$ there exists $\alpha \in \text{Aut}(G)$ extending $i$. 

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Graph formulas and reducts of the random graph

Let $G = (V; E)$ denote the random graph, i.e., the unique countably infinite graph which is

- **universal**, i.e., all finite graphs are induced subgraphs of $G$;
- **homogeneous**, i.e.,
  
  For all finite $A, B \subseteq G$, for all isomorphisms $i : A \to B$
  there exists $\alpha \in \text{Aut}(G)$ extending $i$.

For a graph formula $\psi(x_1, \ldots, x_n)$, define a relation

$$R_\psi := \{(a_1, \ldots, a_n) \in V^n : \psi(a_1, \ldots, a_n)\}.$$
Let $G = (V; E)$ denote the random graph, i.e., the unique countably infinite graph which is

- universal, i.e., all finite graphs are induced subgraphs of $G$;
- homogeneous, i.e.,
  For all finite $A, B \subseteq G$, for all isomorphisms $i : A \to B$ there exists $\alpha \in \text{Aut}(G)$ extending $i$.

For a graph formula $\psi(x_1, \ldots, x_n)$, define a relation

$$R_\psi := \{(a_1, \ldots, a_n) \in V^n : \psi(a_1, \ldots, a_n)\}.$$

For a set $\Psi$ of graph formulas, define a structure

$$\Gamma_\Psi := (V; (R_\psi : \psi \in \Psi)).$$
Graph formulas and reducts of the random graph

Let $G = (V; E)$ denote the random graph, i.e., the unique countably infinite graph which is

- **universal**, i.e., all finite graphs are induced subgraphs of $G$;
- **homogeneous**, i.e.,

  For all finite $A, B \subseteq G$, for all isomorphisms $i : A \to B$
  there exists $\alpha \in \text{Aut}(G)$ extending $i$.

For a graph formula $\psi(x_1, \ldots, x_n)$, define a relation

$$R_\psi := \{(a_1, \ldots, a_n) \in V^n : \psi(a_1, \ldots, a_n)\}.$$

For a set $\Psi$ of graph formulas, define a structure

$$\Gamma_\Psi := (V; (R_\psi : \psi \in \Psi)).$$

$\Gamma_\Psi$ is a **reduct** of the random graph, i.e., a structure with a first-order definition in $G$. 

**Schaefer’s theorem for graphs**

Michael Pinsker (Paris 7)
Graph-SAT as CSP of a reduct of $G$

An instance $W = \{w_1, \ldots, w_m\}$ of Graph-SAT($\Psi$) has a positive solution $\iff$ the sentence $\exists w_1, \ldots, w_m \bigwedge_i \phi_i$ holds in $\Gamma_{\Psi}$.

The decision problem whether or not a given primitive positive sentence holds in $\Gamma_{\Psi}$ is called the Constraint Satisfaction Problem of $\Gamma_{\Psi}$ (or CSP($\Gamma_{\Psi}$)).

So Graph-SAT($\Psi$) and CSP($\Gamma_{\Psi}$) are one and the same problem.
Graph-SAT as CSP of a reduct of $G$

An instance
- $W = \{w_1, \ldots, w_m\}$
- $\phi_1, \ldots, \phi_n$

of Graph-SAT($\Psi$) has a positive solution $\iff$ the sentence $\exists w_1, \ldots, w_m. \bigwedge_i \phi_i$ holds in $\Gamma_\Psi$. 

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Graph-SAT as CSP of a reduct of $G$

An instance

- $W = \{w_1, \ldots, w_m\}$
- $\phi_1, \ldots, \phi_n$

of Graph-SAT($\psi$) has a positive solution $\iff$ the sentence $\exists w_1, \ldots, w_m. \bigwedge_i \phi_i$ holds in $\Gamma_\psi$.

The decision problem
whether or not a given primitive positive sentence holds in $\Gamma_\psi$
is called the **Constraint Satisfaction Problem** of $\Gamma_\psi$ (or CSP($\Gamma_\psi$)).
Graph-SAT as CSP of a reduct of $G$

An instance
- $W = \{w_1, \ldots, w_m\}$
- $\phi_1, \ldots, \phi_n$

of Graph-SAT($\psi$) has a positive solution $\iff$ the sentence $\exists w_1, \ldots, w_m. \land_i \phi_i$ holds in $\Gamma_\psi$.

The decision problem whether or not a given primitive positive sentence holds in $\Gamma_\psi$ is called the Constraint Satisfaction Problem of $\Gamma_\psi$ (or CSP($\Gamma_\psi$)).

So Graph-SAT($\psi$) and CSP($\Gamma_\psi$) are one and the same problem.
Why the random graph?

We have seen:

Classifying the complexity of all Graph-SAT problems is the same as classifying the complexity of CSPs of all reducts of $G$.

Note:

Could have used any universal graph!

But:

$G$ is the nicest universal graph.

Let's study CSP($\Gamma$) for reducts $\Gamma$ of $G$!
Why the random graph?

We have seen:

Classifying the complexity of all Graph-SAT problems is the same as classifying the complexity of CSPs of all reducts of $G$. 

Note: Could have used any universal graph!

But: $G$ is the nicest universal graph.
Why the random graph?

We have seen:

Classifying the complexity of all Graph-SAT problems is the same as classifying the complexity of CSPs of all reducts of $G$.

Note:

Could have used any universal graph!
Why the random graph?

We have seen:
Classifying the complexity of all Graph-SAT problems is the same as classifying the complexity of CSPs of all reducts of \( G \).

Note:
Could have used any universal graph!

But:
\( G \) is the nicest universal graph.
Why the random graph?

We have seen:
Classifying the complexity of all Graph-SAT problems is the same as classifying the complexity of CSPs of all reducts of $G$.

Note:
Could have used any universal graph!

But:
$G$ is the nicest universal graph.

Let’s study CSP($\Gamma$) for reducts $\Gamma$ of $G$!
Primitive positive (pp) definability and polymorphisms

For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff every relation of $\Gamma$ has a pp-definition from $\Delta$.

Easy observation. If $\Gamma \leq_{pp} \Delta$, then CSP($\Gamma$) has a polynomial-time reduction to CSP($\Delta$).

For finite $n \geq 1$, a function $f : \Gamma^n \to \Gamma$ is a polymorphism of $\Gamma$ iff for all relations $R$ of $\Gamma$ and all $r_1, \ldots, r_n \in R$ we have $f(r_1, \ldots, r_n) \in R$.

Generalization of endomorphism, automorphism. We write $\text{Pol}(\Gamma)$ for the set of polymorphisms of $\Gamma$.

"Polymorphism clone of $\Gamma"$

Theorem (Bodirsky, Nešetřil '03). $\Gamma \leq_{pp} \Delta \iff \text{Pol}(\Delta) \subseteq \text{Pol}(\Gamma)$.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff every relation of $\Gamma$ has a pp-definition from $\Delta$. 
For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff
every relation of $\Gamma$ has a pp-definition from $\Delta$.

**Easy observation.**
If $\Gamma \leq_{pp} \Delta$, then CSP($\Gamma$) has a polynomial-time reduction to CSP($\Delta$).
Primitive positive (pp) definability and polymorphisms

For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff every relation of $\Gamma$ has a pp-definition from $\Delta$.

**Easy observation.**

If $\Gamma \leq_{pp} \Delta$, then CSP(\(\Gamma\)) has a polynomial-time reduction to CSP(\(\Delta\)).

For finite $n \geq 1$, a function $f : \Gamma^n \rightarrow \Gamma$ is a *polymorphism* of $\Gamma$ iff for all relations $R$ of $\Gamma$ and all $r_1, \ldots, r_n \in R$ we have $f(r_1, \ldots, r_n) \in R$. 

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Primitive positive (pp) definability and polymorphisms

For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff every relation of $\Gamma$ has a pp-definition from $\Delta$.

**Easy observation.**
If $\Gamma \leq_{pp} \Delta$, then $\text{CSP}(\Gamma)$ has a polynomial-time reduction to $\text{CSP}(\Delta)$.

For finite $n \geq 1$, a function $f : \Gamma^n \to \Gamma$ is a *polymorphism* of $\Gamma$ iff for all relations $R$ of $\Gamma$ and all $r_1, \ldots, r_n \in R$ we have $f(r_1, \ldots, r_n) \in R$.

Generalization of endomorphism, automorphism.
For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff every relation of $\Gamma$ has a pp-definition from $\Delta$.

**Easy observation.**
If $\Gamma \leq_{pp} \Delta$, then CSP($\Gamma$) has a polynomial-time reduction to CSP($\Delta$).

For finite $n \geq 1$, a function $f : \Gamma^n \to \Gamma$ is a *polymorphism* of $\Gamma$ iff for all relations $R$ of $\Gamma$ and all $r_1, \ldots, r_n \in R$ we have $f(r_1, \ldots, r_n) \in R$.

Generalization of endomorphism, automorphism.

We write Pol($\Gamma$) for the set of polymorphisms of $\Gamma$.

“Polymorphism clone of $\Gamma$”
Primitive positive (pp) definability and polymorphisms

For reducts $\Gamma, \Delta$, set $\Gamma \leq_{pp} \Delta$ iff every relation of $\Gamma$ has a pp-definition from $\Delta$.

**Easy observation.**

If $\Gamma \leq_{pp} \Delta$, then CSP($\Gamma$) has a polynomial-time reduction to CSP($\Delta$).

For finite $n \geq 1$, a function $f : \Gamma^n \rightarrow \Gamma$ is a *polymorphism* of $\Gamma$ iff for all relations $R$ of $\Gamma$ and all $r_1, \ldots, r_n \in R$ we have $f(r_1, \ldots, r_n) \in R$.

Generalization of endomorphism, automorphism.

We write $\text{Pol}(\Gamma)$ for the set of polymorphisms of $\Gamma$.

"*Polymorphism clone of $\Gamma$"

**Theorem** (Bodirsky, Nešetřil ’03). $\Gamma \leq_{pp} \Delta$ $\iff$ $\text{Pol}(\Delta) \subseteq \text{Pol}(\Gamma)$.
The polymorphism strategy

Larger reducts \( \Gamma \leq \Delta \rightarrow \text{CSP(\Gamma)} \leq \text{Poltime} \text{CSP(\Delta)} \)

Larger polymorphism clones \( \text{Pol}(\Gamma) \subseteq \text{Pol}(\Delta) \rightarrow \text{CSP(\Delta)} \leq \text{Poltime} \text{CSP(\Gamma)} \)

Strategy:
(i) Prove hardness for certain reducts;
(ii) Prove that all reducts which do not pp-define any of these hard reducts are tractable.

Reducts of (ii) have polymorphisms violating the relations of (i).

Polymorphisms provide algorithms.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP

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

Strategy:
(i) Prove hardness for certain reducts;
(ii) Prove that all reducts which do not pp-define any of these hard reducts are tractable.

Polymorphisms provide algorithms.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP

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

Larger polymorphism clones $\rightarrow$ easier CSP

$\text{Pol}(\Gamma) \subseteq \text{Pol}(\Delta) \rightarrow \text{CSP}(\Delta) \leq_{\text{Poltime}} \text{CSP}(\Gamma)$
The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP
\[ \Gamma \leq_{pp} \Delta \quad \rightarrow \quad \text{CSP}(\Gamma) \leq_{\text{Poltime}} \text{CSP}(\Delta) \]

Larger polymorphism clones $\rightarrow$ easier CSP
\[ \text{Pol}(\Gamma) \subseteq \text{Pol}(\Delta) \quad \rightarrow \quad \text{CSP}(\Delta) \leq_{\text{Poltime}} \text{CSP}(\Gamma) \]

Strategy:

(i) Prove hardness for certain reducts;
(ii) Prove that all reducts which do not pp-define any of these hard reducts are tractable.
The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP
\[ \Gamma \leq_{pp} \Delta \quad \rightarrow \quad \text{CSP}(\Gamma) \leq_{\text{Poltime}} \text{CSP}(\Delta) \]

Larger polymorphism clones $\rightarrow$ easier CSP
\[ \text{Pol}(\Gamma) \subseteq \text{Pol}(\Delta) \quad \rightarrow \quad \text{CSP}(\Delta) \leq_{\text{Poltime}} \text{CSP}(\Gamma) \]

Strategy:

(i) Prove hardness for certain reducts;
(ii) Prove that all reducts which do not pp-define any of these hard reducts are tractable.

Reducts of (ii) have polymorphisms violating the relations of (i).
Polymorphisms provide algorithms.
Part III

Making the infinite finite

Canonical polymorphisms
Canonical functions

We have seen: Polymorphisms should prove tractability.
Canonical functions

**We have seen:** Polymorphisms should prove tractability.
True for CSP of finite structures, e.g. max on \{0, 1\} (Schaefer).
**Canonical functions**

We have seen: Polymorphisms should prove tractability. True for CSP of finite structures, e.g. max on \( \{0, 1\} \) (Schaefer).

How can we use an *infinite* polymorphism \( f : \Gamma^n \rightarrow \Gamma \) in an algorithm?

---

**Definition.** A function \( f : G \rightarrow G \) is canonical \( \iff \) whenever two pairs \( (x, y), (u, v) \in G^2 \) have the same type, then \( (f(x), f(y)) \) and \( (f(u), f(v)) \) have the same type as well.

**Examples**
- Function which switches edges and non-edges.
- Injection onto complete subgraph of \( G \).
- Constant function.

Generalization of canonical to functions \( f : G^n \rightarrow G \) possible.

**Example.** edge-max: \( G^2 \rightarrow G \).

Schaefer's theorem for graphs
We have seen: Polymorphisms should prove tractability. True for CSP of finite structures, e.g. max on \{0, 1\} (Schaefer).

How can we use an infinite polymorphism \( f : \Gamma^n \rightarrow \Gamma \) in an algorithm?

**Definition.** A function \( f : G \rightarrow G \) is canonical \( \iff \) whenever two pairs \((x, y), (u, v) \in G^2\) have the the same type, then \((f(x), f(y))\) and \((f(u), f(v))\) have the same type as well.
We have seen: Polymorphisms should prove tractability.
True for CSP of finite structures, e.g. max on \{0, 1\} (Schaefer).
How can we use an \textit{infinite} polymorphism \( f : \Gamma^n \rightarrow \Gamma \) in an algorithm?

\textbf{Definition.} A function \( f : G \rightarrow G \) is \textit{canonical} \iff whenever two pairs \((x, y), (u, v) \in G^2\) have the the same \textit{type}, then \((f(x), f(y))\) and \((f(u), f(v))\) have the same type as well.

\textbf{Examples}

- Function which switches edges and non-edges.
We have seen: Polymorphisms should prove tractability. True for CSP of finite structures, e.g. max on \{0, 1\} (Schaefer).

How can we use an infinite polymorphism \(f : \Gamma^n \rightarrow \Gamma\) in an algorithm?

**Definition.** A function \(f : G \rightarrow G\) is canonical \(\leftrightarrow\) whenever two pairs \((x, y), (u, v) \in G^2\) have the the same type, then \((f(x), f(y))\) and \((f(u), f(v))\) have the same type as well.

**Examples**
- Function which switches edges and non-edges.
- Injection onto complete subgraph of \(G\).
Canonical functions

We have seen: Polymorphisms should prove tractability. True for CSP of finite structures, e.g. max on \{0, 1\} (Schaefer). How can we use an \textit{infinite} polymorphism \(f : \Gamma^n \rightarrow \Gamma\) in an algorithm?

Definition. A function \(f : G \rightarrow G\) is \textit{canonical} \iff whenever two pairs \((x, y), (u, v) \in G^2\) have the the same type, then \((f(x), f(y))\) and \((f(u), f(v))\) have the same type as well.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of \(G\).
- Constant function.
Canonical functions

We have seen: Polymorphisms should prove tractability.
True for CSP of finite structures, e.g. max on \{0, 1\} (Schaefer).
How can we use an infinite polymorphism \( f : \Gamma^n \rightarrow \Gamma \) in an algorithm?

Definition. A function \( f : G \rightarrow G \) is canonical if whenever two pairs \((x, y), (u, v) \in G^2\) have the same type, then \((f(x), f(y))\) and \((f(u), f(v))\) have the same type as well.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of \( G \).
- Constant function.

Generalization of canonical to functions \( f : G^n \rightarrow G \) possible.

Example. edge-max: \( G^2 \rightarrow G \).
Canonical functions theorem

We wish to work with canonical polymorphisms.
Canonical functions theorem

We wish to work with canonical polymorphisms.

**Fact.** $G$ has the following **Ramsey**-type property:

Every function $f: G \to G$ induces a coloring of the edges of $G$.

Exploiting this further, one obtains:

**Theorem (roughly).** If a polymorphism of $\Gamma$ violates a relation $R$, then there exists a canonical polymorphism of $\Gamma$ which violates $R$.

General modern proof uses topological dynamics, i.e., continuous group actions on compact topological spaces.
Canonical functions theorem

We wish to work with canonical polymorphisms.

Fact. $G$ has the following Ramsey-type property:

For all finite graphs $H$
there exists a finite graph $S$ such that
whenever the edges of $S$ are colored with two colors
then there exists a copy of $H$ in $S$ on which the coloring is constant.
We wish to work with canonical polymorphisms.

**Fact.** $G$ has the following *Ramsey*-type property:

For all finite graphs $H$ there exists a finite graph $S$ such that whenever the edges of $S$ are colored with two colors then there exists a copy of $H$ in $S$ on which the coloring is constant.

Every function $f : G \to G$ induces a coloring of the edges of $G$. Exploiting this further, one obtains:
We wish to work with canonical polymorphisms.

**Fact.** $G$ has the following *Ramsey*-type property:

For all finite graphs $H$
there exists a finite graph $S$ such that
whenever the edges of $S$ are colored with two colors
then there exists a copy of $H$ in $S$ on which the coloring is constant.

Every function $f : G \rightarrow G$ induces a coloring of the edges of $G$.
Exploiting this further, one obtains:

**Theorem** (roughly). If a polymorphism of $\Gamma$ violates a relation $R$, then there exists a canonical polymorphism of $\Gamma$ which violates $R$. 
Canonical functions theorem

We wish to work with canonical polymorphisms.

Fact. $G$ has the following Ramsey-type property:
For all finite graphs $H$
there exists a finite graph $S$ such that
whenever the edges of $S$ are colored with two colors
then there exists a copy of $H$ in $S$ on which the coloring is constant.

Every function $f : G \to G$ induces a coloring of the edges of $G$.
Exploiting this further, one obtains:

Theorem (roughly). If a polymorphism of $\Gamma$ violates a relation $R$,
then there exists a canonical polymorphism of $\Gamma$ which violates $R$.

General modern proof uses topological dynamics, i.e.,
continuous group actions on compact topological spaces.
We wish to work with canonical polymorphisms.

**Fact.** $G$ has the following **Ramsey-type** property:

For all finite graphs $H$ there exists a finite graph $S$ such that whenever the edges of $S$ are colored with two colors then there exists a copy of $H$ in $S$ on which the coloring is constant.

Every function $f : G \to G$ induces a coloring of the edges of $G$.

Exploiting this further, one obtains:

**Theorem** (roughly). If a polymorphism of $\Gamma$ violates a relation $R$, then there exists a canonical polymorphism of $\Gamma$ which violates $R$.

General modern proof uses topological dynamics, i.e., continuous group actions on compact topological spaces.

Canonical functions are finite objects: functions on types!
Part IV

The Graph-SAT dichotomy
Complexity of CSP for reducts of $G$

Theorem (Bodirsky, MP '10)

Let $\Gamma$ be a reduct of the random graph. Then:

Either $\Gamma$ has one out of 17 canonical polymorphisms, and $\text{CSP}(\Gamma)$ is tractable,

or $\text{CSP}(\Gamma)$ is NP-complete.

Theorem (Bodirsky, MP '10)

Let $\Gamma$ be a reduct of the random graph. Then:

Either $\Gamma$ pp-defines one out of 4 hard relations, and $\text{CSP}(\Gamma)$ is NP-complete,

or $\text{CSP}(\Gamma)$ is tractable.
Theorem (Bodirsky, MP ’10)

Let $\Gamma$ be a reduct of the random graph. Then:

- Either $\Gamma$ has one out of 17 canonical polymorphisms, and CSP($\Gamma$) is tractable,
- or CSP($\Gamma$) is NP-complete.
Complexity of CSP for reducts of $G$

**Theorem (Bodirsky, MP ’10)**

Let $\Gamma$ be a reduct of the random graph. Then:

- Either $\Gamma$ has one out of 17 canonical polymorphisms, and $\text{CSP}(\Gamma)$ is tractable,
- or $\text{CSP}(\Gamma)$ is NP-complete.

**Theorem (Bodirsky, MP ’10)**

Let $\Gamma$ be a reduct of the random graph. Then:

- Either $\Gamma$ pp-defines one out of 4 hard relations, and $\text{CSP}(\Gamma)$ is NP-complete,
- or $\text{CSP}(\Gamma)$ is tractable.
The Graph-SAT dichotomy visualized

Schaefer’s theorem for graphs

Michael Pinsker (Paris 7)
Theorem

The following 17 distinct clones are precisely the minimal tractable closed clones containing $\text{Aut}(G)$:

1. The clone generated by a constant operation.
2. The clone generated by a balanced binary injection of type max.
3. The clone generated by a balanced binary injection of type min.
4. The clone generated by an $E$-dominated binary injection of type max.
5. The clone generated by an $N$-dominated binary injection of type min.
6. The clone generated by a function of type majority which is hyperplanely balanced and of type projection.
7. The clone generated by a function of type majority which is hyperplanely $E$-constant.
8. The clone generated by a function of type majority which is hyperplanely $N$-constant.
9. The clone generated by a function of type majority which is hyperplanely of type max and $E$-dominated.
10. The clone generated by a function of type majority which is hyperplanely of type min and $N$-dominated.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
The Meta Problem

Meta-Problem of Graph-SAT($\Psi$)

INPUT: A finite set $\Psi$ of graph formulas.

QUESTION: Is Graph-SAT($\Psi$) in P?

Theorem (Bodirsky, MP '10)

The Meta-Problem of Graph-SAT($\Psi$) is decidable.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
The Meta Problem

Meta-Problem of Graph-SAT(ψ)

INPUT: A finite set ψ of graph formulas.

QUESTION: Is Graph-SAT(ψ) in P?
## The Meta Problem

### Meta-Problem of Graph-SAT(ψ)

**INPUT:** A finite set \( ψ \) of graph formulas.

**QUESTION:** Is Graph-SAT(ψ) in P?

### Theorem (Bodirsky, MP ’10)

The Meta-Problem of Graph-SAT(ψ) is decidable.
Graph satisfiability problems

Let $\Psi$ be a finite set of graph formulas. Computational problem: Graph-SAT($\Psi$)

**INPUT:** A set $W$ of variables (vertices), and statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

**Theorem (Bodirsky, MP '10)**

Graph-SAT($\Psi$) is either in P or NP-complete, for all $\Psi$.

Schaefer's theorem for graphs
Michael Pinsker (Paris 7)
Graph satisfiability problems

Let $\psi$ be a finite set of graph formulas.

Computational problem: Graph-SAT($\psi$)

INPUT:
- A set $W$ of variables (vertices), and
- statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\psi$.

QUESTION: Is $\wedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

Theorem (Bodirsky, MP '10)

Graph-SAT($\psi$) is either in P or NP-complete, for all $\psi$. 

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Graph satisfiability problems

Let $\Psi$ be a finite set of graph formulas.

Computational problem: Graph-SAT($\Psi$)

**INPUT:**
- A set $W$ of variables (vertices), and
- statements $\phi_1, \ldots, \phi_n$ about the elements of $W$, where each $\phi_i$ is taken from $\Psi$.

**QUESTION:** Is $\bigwedge_{1 \leq i \leq n} \phi_i$ satisfiable in a graph?

**Theorem (Bodirsky, MP ’10)**

Graph-SAT($\Psi$) is either in P or NP-complete, for all $\Psi$. 

Schaefer’s theorem for graphs  
Michael Pinsker  (Paris 7)
Part V

The future

CSPs over homogeneous structures
Amalgamation classes

Graph-SAT(\(\Psi\)): Is there a finite graph such that...

Linorder-SAT(\(\Psi\)): Is there a linear order such that...

The classes of finite graphs and linear orders are amalgamation classes.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Amalgamation classes

Graph-SAT(ψ): Is there a finite graph such that... (graph constraints)
Amalgamation classes

Graph-SAT(\(\psi\)): Is there a finite graph such that... (graph constraints)

Linorder-SAT(\(\psi\)): Is there a linear order such that... (order constraints, “temporal constraints”)
Amalgamation classes

**Graph-SAT**\(\psi\): Is there a finite graph such that... (graph constraints)

**Linorder-SAT**\(\psi\): Is there a linear order such that... (order constraints, “temporal constraints”)

The classes of finite graphs and linear orders are *amalgamation classes*.

Schaefer’s theorem for graphs

Michael Pinsker (Paris 7)
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}$ with age $\mathcal{C}$ which is homogeneous.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}$ with age $\mathcal{C}$ which is homogeneous.

- The age of a homogeneous structure is an amalgamation class.
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}'$ with age $\mathcal{C}$ which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

$\mathcal{C}'$ is called the Fraïssé limit of $\mathcal{C}$. Example $(\mathbb{Q}, \lt)$.
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If \( \mathcal{C} \) is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure \( \mathcal{C} \) with age \( \mathcal{C} \) which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

\( \mathcal{C} \) is called the Fraïssé limit of \( \mathcal{C} \). Example \((\mathbb{Q}, <)\).

Further amalgamation classes.

Schaefer’s theorem for graphs

Michael Pinsker (Paris 7)
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}$ with age $\mathcal{C}$ which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

$\mathcal{C}$ is called the Fraïssé limit of $\mathcal{C}$. Example $(\mathbb{Q}, \prec)$.

Further amalgamation classes.

- Partial orders
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}$ with age $\mathcal{C}$ which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

$\mathcal{C}$ is called the Fraïssé limit of $\mathcal{C}$. Example $(\mathbb{Q}, <)$.

Further amalgamation classes.

- Partial orders
- Metric spaces with finite set of distances
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If \( \mathcal{C} \) is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure \( \mathcal{C} \) with age \( \mathcal{C} \) which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

\( \mathcal{C} \) is called the Fraïssé limit of \( \mathcal{C} \). Example \((\mathbb{Q}, <)\).

Further amalgamation classes.

- Partial orders
- Metric spaces with finite set of distances
- Tournaments
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)
- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}$ with age $\mathcal{C}$ which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

$\mathcal{C}$ is called the **Fraïssé limit** of $\mathcal{C}$. Example $(\mathbb{Q}, <)$.

Further amalgamation classes.
- Partial orders
- Metric spaces with finite set of distances
- Tournaments
- $K_n$-free graphs
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathcal{C}$ with age $\mathcal{C}$ which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

$\mathcal{C}$ is called the Fraïssé limit of $\mathcal{C}$. Example $(\mathbb{Q}, <)$.

Further amalgamation classes.

- Partial orders
- Metric spaces with finite set of distances
- Tournaments
- $K_n$-free graphs
- Ordered graphs

Schaefer’s theorem for graphs

Michael Pinsker (Paris 7)
Amalgamation classes have homogeneous limit

Theorem (Fraïssé)

- If $\mathcal{C}$ is a countable class of structures closed under substructures which has amalgamation, then there exists a unique structure $\mathfrak{C}$ with age $\mathfrak{C}$ which is homogeneous.
- The age of a homogeneous structure is an amalgamation class.

$\mathfrak{C}$ is called the Fraïssé limit of $\mathcal{C}$. Example $(\mathbb{Q}, <)$.

Further amalgamation classes.
- Partial orders
- Metric spaces with finite set of distances
- Tournaments
- $K_n$-free graphs
- Ordered graphs
- Permutations
General method for amalgamation classes

Given amalgamation class $C$, consider all $C$-SAT problems.

Every problem $C$-SAT($\Psi$) translates into CSP($\Gamma \Psi$), where $\Gamma \Psi$ is a reduct of the (homogeneous infinite) Fraïssé limit $C$ of $C$.

For each reduct $\Gamma$ of this limit $C$, the complexity of CSP($\Gamma \Psi$) is captured by the polymorphism clone $\text{Pol}(\Gamma)$.

Tractability is implied by presence of polymorphisms in $\text{Pol}(\Gamma)$.

If $C$ is Ramsey, then even implied by canonical polymorphisms. These are essentially functions on finite sets.

Adaptations of the algorithms for these finite functions.

Hardness proofs: by reduction of known finite CSPs. Modern method: exposing a continuous homomorphism from $\text{Pol}(\Gamma)$ to the projection clone on $\{0, 1\}$.

Topological Birkhoff.

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Given amalgamation class $\mathcal{C}$, consider all $\mathcal{C}$-SAT problems.

Every problem $\mathcal{C}$-SAT($\Psi$) translates into CSP($\Gamma\Psi$), where $\Gamma\Psi$ is a reduct of the (homogeneous infinite) Fraïssé limit $\mathcal{C}$ of $\mathcal{C}$.

For each reduct $\Gamma$ of this limit $\mathcal{C}$, the complexity of CSP($\Gamma\Psi$) is captured by the polymorphism clone $\text{Pol}(\Gamma)$.

Tractability is implied by presence of polymorphisms in $\text{Pol}(\Gamma)$.

If $\mathcal{C}$ is Ramsey, then even implied by canonical polymorphisms. These are essentially functions on finite sets.

Adaptations of the algorithms for these finite functions.

Hardness proofs: by reduction of known finite CSPs.

Modern method: exposing a continuous homomorphism from $\text{Pol}(\Gamma)$ to the projection clone on $\{0, 1\}$.

Topological Birkhoff.
General method for amalgamation classes

1 Given amalgamation class \( \mathcal{C} \), consider all \( \mathcal{C} \)-SAT problems.

2 Every problem \( \mathcal{C} \)-SAT(\( \Psi \)) translates into CSP(\( \Gamma_\Psi \)), where \( \Gamma_\Psi \) is a reduct of the (homogeneous infinite) Fraïssé limit \( \mathcal{C} \) of \( \mathcal{C} \).
Given amalgamation class $\mathcal{C}$, consider all $\mathcal{C}$-SAT problems.

Every problem $\mathcal{C}$-SAT($\Psi$) translates into CSP($\Gamma_\Psi$), where $\Gamma_\Psi$ is a reduct of the (homogeneous infinite) Fraïssé limit $\mathcal{C}$ of $\mathcal{C}$.

For each reduct $\Gamma$ of this limit $\mathcal{C}$, the complexity of CSP($\Gamma$) is captured by the polymorphism clone Pol($\Gamma$).
General method for amalgamation classes

1. Given amalgamation class $\mathcal{C}$, consider all $\mathcal{C}$-SAT problems.

2. Every problem $\mathcal{C}$-SAT($\Psi$) translates into CSP($\Gamma_\Psi$), where $\Gamma_\Psi$ is a reduct of the (homogeneous infinite) Fraïssé limit $\mathcal{C}$ of $\mathcal{C}$.

3. For each reduct $\Gamma$ of this limit $\mathcal{C}$, the complexity of CSP($\Gamma$) is captured by the polymorphism clone $\text{Pol}(\Gamma)$.

4. **Tractability** is implied by presence of polymorphisms in $\text{Pol}(\Gamma)$.

---

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
1. Given amalgamation class $\mathcal{C}$, consider all $\mathcal{C}$-SAT problems.

2. Every problem $\mathcal{C}$-SAT($\Psi$) translates into CSP($\Gamma_\Psi$), where $\Gamma_\Psi$ is a reduct of the (homogeneous infinite) Fraïssé limit $\mathcal{C}$ of $\mathcal{C}$.

3. For each reduct $\Gamma$ of this limit $\mathcal{C}$, the complexity of CSP($\Gamma$) is captured by the polymorphism clone $\text{Pol}(\Gamma)$.

4. **Tractability** is implied by presence of polymorphisms in $\text{Pol}(\Gamma)$.

5. If $\mathcal{C}$ is Ramsey, then even implied by canonical polymorphisms. These are essentially functions on finite sets.
General method for amalgamation classes

1. Given amalgamation class $\mathcal{C}$, consider all $\mathcal{C}$-SAT problems.

2. Every problem $\mathcal{C}$-SAT($\Psi$) translates into CSP($\Gamma_{\Psi}$), where $\Gamma_{\Psi}$ is a reduct of the (homogeneous infinite) Fraïssé limit $\mathcal{C}$ of $\mathcal{C}$.

3. For each reduct $\Gamma$ of this limit $\mathcal{C}$, the complexity of CSP($\Gamma$) is captured by the polymorphism clone $\text{Pol}(\Gamma)$.

4. Tractability is implied by presence of polymorphisms in $\text{Pol}(\Gamma)$.

5. If $\mathcal{C}$ is Ramsey, then even implied by canonical polymorphisms. These are essentially functions on finite sets.

6. Adaptations of the algorithms for these finite functions.
General method for amalgamation classes

1. Given amalgamation class \( \mathcal{C} \), consider all \( \mathcal{C} \)-SAT problems.

2. Every problem \( \mathcal{C} \)-SAT(\( \Psi \)) translates into CSP(\( \Gamma_\Psi \)), where \( \Gamma_\Psi \) is a reduct of the (homogeneous infinite) Fraïssé limit \( \mathcal{C} \) of \( \mathcal{C} \).

3. For each reduct \( \Gamma \) of this limit \( \mathcal{C} \), the complexity of CSP(\( \Gamma \)) is captured by the polymorphism clone \( \text{Pol}(\Gamma) \).

4. Tractability is implied by presence of polymorphisms in \( \text{Pol}(\Gamma) \).

5. If \( \mathcal{C} \) is Ramsey, then even implied by canonical polymorphisms. These are essentially functions on finite sets.

6. Adaptations of the algorithms for these finite functions.

7. Hardness proofs: by reduction of known finite CSPs.
General method for amalgamation classes

1. Given amalgamation class $\mathcal{C}$, consider all $\mathcal{C}$-SAT problems.

2. Every problem $\mathcal{C}$-SAT($\Psi$) translates into CSP($\Gamma_{\Psi}$), where $\Gamma_{\Psi}$ is a reduct of the (homogeneous infinite) Fraïssé limit $\mathcal{C}$ of $\mathcal{C}$.

3. For each reduct $\Gamma$ of this limit $\mathcal{C}$, the complexity of CSP($\Gamma$) is captured by the polymorphism clone Pol($\Gamma$).

4. **Tractability** is implied by presence of polymorphisms in Pol($\Gamma$).

5. If $\mathcal{C}$ is Ramsey, then even implied by canonical polymorphisms. These are essentially functions on finite sets.

6. Adaptations of the algorithms for these finite functions.

7. **Hardness proofs:** by reduction of known finite CSPs.
   - Modern method: exposing a continuous homomorphism from Pol($\Gamma$) to the projection clone on $\{0, 1\}$. *Topological Birkhoff.*

Schaefer's theorem for graphs

Michael Pinsker (Paris 7)
Future research

(a) Find (improve “making finite”): Meta-method for translating tractability of the type function of a canonical function into tractability of the canonical function.

(b) Prove (complete “making finite”): If the dichotomy / tractability conjecture for finite structures holds, then it holds for all reducts of homogeneous Ramsey structures.

(c) Answer (improve “making infinite”): Can all homogeneous structures be made Ramsey by adding finitely many relations?

(d) Apply method to:
- finite partial orders – Poset-SAT(Ψ)
- finite Boolean algebras – “set constraints” etc.

Schaefer’s theorem for graphs
Michael Pinsker (Paris 7)
Future research

(a) Find (improve “making finite”):
Meta-method for translating *tractability of the type function* of a canonical function into *tractability of the canonical function*.

(b) Prove (complete “making finite”):
If the dichotomy / tractability conjecture for finite structures holds, then it holds for all reducts of homogeneous Ramsey structures.

(c) Answer (improve “making infinite”):
Can all homogeneous structures be made Ramsey by adding finitely many relations?

(d) Apply method to:
- finite partial orders – Poset-SAT($\Psi$)
- finite Boolean algebras – ”set constraints” etc.
Future research

(a) Find (improve “making finite”):
Meta-method for translating \textit{tractability of the type function} of a canonical function into \textit{tractability of the canonical function}.

(b) Prove (complete “making finite”):
If the dichotomy / tractability conjecture for finite structures holds, then it holds for all reducts of homogeneous Ramsey structures.
(a) Find (improve “making finite”):
   Meta-method for translating \textit{tractability of the type function} of a canonical function into \textit{tractability of the canonical function}.

(b) Prove (complete “making finite”):
   If the dichotomy / tractability conjecture for finite structures holds, then it holds for all reducts of homogeneous Ramsey structures.

(c) Answer (improve “making infinite”):
   Can all homogeneous structures be made Ramsey by adding finitely many relations?
Future research

(a) Find (improve “making finite”):
Meta-method for translating tractability of the type function of a canonical function into tractability of the canonical function.

(b) Prove (complete “making finite”):
If the dichotomy / tractability conjecture for finite structures holds, then it holds for all reducts of homogeneous Ramsey structures.

(c) Answer (improve “making infinite”):
Can all homogeneous structures be made Ramsey by adding finitely many relations?

(d) Apply method to:
- finite partial orders – Poset-SAT($\Psi$)
- finite Boolean algebras – “set constraints” etc.
Graph-SAT dichotomy:

Schaefer’s theorem for graphs
by Manuel Bodirsky and Michael Pinsker,

Canonical functions method:

Reducts of Ramsey structures
by Manuel Bodirsky and Michael Pinsker,

Modern hardness proofs:

Topological Birkhoff
by Manuel Bodirsky and Michael Pinsker,
Schaefer’s theorem for graphs
Michael Pinsker (Paris 7)