Schaefer's theorem for graphs

Manuel Bodirsky* and Michael Pinsker°

*Ecole Polytechnique, Palaiseau

°Université Denis Diderot - Paris 7
Technische Universität Wien
Hebrew University of Jerusalem

STOC 2011

Outline

Outline

- The graph satisfiability problem Graph-SAT(Ψ)

Outline

- The graph satisfiability problem Graph-SAT(Ψ)

- Schaefer's theorem for graphs

Outline

- The graph satisfiability problem Graph-SAT(Ψ)
- Graph-SAT $\rightarrow$ CSPs of reducts of the random graph
- Schaefer's theorem for graphs

Outline

- The graph satisfiability problem Graph-SAT(Ψ)
- Graph-SAT $\rightarrow$ CSPs of reducts of the random graph
- The algebraic approach: From reducts to polymorphisms
- Schaefer's theorem for graphs

Outline

- The graph satisfiability problem Graph-SAT(Ψ)

- Graph-SAT $\rightarrow$ CSPs of reducts of the random graph
- The algebraic approach: From reducts to polymorphisms
- Ramsey theory: Patterns in polymorphisms

- Schaefer's theorem for graphs

The graph satisfiability problem

Graph-SAT(Ψ)

The Boolean satisfiability problem

The Boolean satisfiability problem

Let Ψ be a finite set of propositional formulas.

The Boolean satisfiability problem

Let Ψ be a finite set of propositional formulas.

Computational problem: Boolean-SAT(Ψ)

INPUT:

- A set W of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in W, where each ϕ_i is taken from Ψ.

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

The Boolean satisfiability problem

Let Ψ be a finite set of propositional formulas.

Computational problem: Boolean-SAT(Ψ)

INPUT:

- A set W of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in W, where each ϕ_i is taken from Ψ.

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

Computational complexity depends on Ψ. Always in NP.

The Boolean satisfiability problem

Let Ψ be a finite set of propositional formulas.

Computational problem: Boolean-SAT(Ψ)

INPUT:

- A set W of propositional variables, and
- statements $\phi_1, \ldots, \phi_n$ about the variables in W, where each ϕ_i is taken from Ψ.

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

Computational complexity depends on Ψ. Always in NP.

Theorem (Schaefer '78)

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

The Graph Satisfiability Problem

The Graph Satisfiability Problem

Let E be a binary relation symbol.

(Imagine: edge relation of an undirected graph.)

Let Ψ be a finite set of quantifier-free $\{E\}$-formulas.

The Graph Satisfiability Problem

Let E be a binary relation symbol.

(Imagine: edge relation of an undirected graph.)

Let Ψ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT(Ψ)

INPUT:

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

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

The Graph Satisfiability Problem

Let E be a binary relation symbol.

(Imagine: edge relation of an undirected graph.)

Let Ψ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT(Ψ)

INPUT:

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

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

Computational complexity depends on Ψ. Always in NP.

The Graph Satisfiability Problem

Let E be a binary relation symbol.

(Imagine: edge relation of an undirected graph.)

Let Ψ be a finite set of quantifier-free $\{E\}$-formulas.

Computational problem: Graph-SAT(Ψ)

INPUT:

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

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

Computational complexity depends on Ψ. Always in NP.

Question

Is Graph-SAT(Ψ) either in P or NP-complete, for all Ψ?

Graph-SAT: Examples

Graph-SAT: Examples

Example 1 Let Ψ_1 only contain

$$\begin{aligned}\psi_1(x,y,z) :=&(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z)) \\ &\vee (\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z)) \\ &\vee (\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z)) \ .\end{aligned}$$

Graph-SAT: Examples

Example 1 Let Ψ_1 only contain

$$\begin{aligned}\psi_1(x,y,z) :=&(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z))\\ &\vee (\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z))\\ &\vee (\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z))\ .\end{aligned}$$

Graph-SAT(Ψ_1) is NP-complete.

Graph-SAT: Examples

Example 1 Let Ψ_1 only contain

$$\begin{aligned}\psi_1(x,y,z) :=& (E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z)) \\ & \vee (\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z)) \\ & \vee (\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z)) \,.\end{aligned}$$

Graph-SAT(Ψ_1) is NP-complete.

Example 2 Let Ψ_2 only contain

$$\begin{aligned}\psi_2(x,y,z) :=& (E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z)) \\ & \vee (\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z)) \\ & \vee (\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z)) \\ & \vee (E(x,y) \wedge E(y,z) \wedge E(x,z)) \,.\end{aligned}$$

Graph-SAT: Examples

Example 1 Let Ψ_1 only contain

$$\begin{aligned}\psi_1(x,y,z) :=&(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z))\\ &\vee (\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z))\\ &\vee (\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z))\ .\end{aligned}$$

Graph-SAT(Ψ_1) is NP-complete.

Example 2 Let Ψ_2 only contain

$$\begin{aligned}\psi_2(x,y,z) :=&(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z))\\ &\vee (\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z))\\ &\vee (\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z))\\ &\vee (E(x,y) \wedge E(y,z) \wedge E(x,z))\ .\end{aligned}$$

Graph-SAT(Ψ_2) is in P.

Graph-SAT

$\rightarrow$

Constraint Satisfaction Problems

of reducts of the random graph

Graph formulas and reducts of 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

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 (ultra-)homogeneous

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 (ultra-)homogeneous
- contains all finite (even countable) graphs.

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 (ultra-)homogeneous
- contains all finite (even countable) graphs.

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

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

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 (ultra-)homogeneous
- contains all finite (even countable) graphs.

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 Ψ 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 (ultra-)homogeneous
- contains all finite (even countable) graphs.

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 Ψ of graph formulas, define a structure

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

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

Graph-SAT as Constraint Satisfaction Problem

Graph-SAT as Constraint Satisfaction Problem

An instance

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

of Graph-SAT(Ψ) has a positive solution $\leftrightarrow$

Graph-SAT as Constraint Satisfaction Problem

An instance

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

of Graph-SAT(Ψ) has a positive solution $\leftrightarrow$

the sentence $\exists w_1, \ldots, w_m. \bigwedge_i \phi_i$ holds in Γ_Ψ.

Graph-SAT as Constraint Satisfaction Problem

An instance

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

of Graph-SAT(Ψ) has a positive solution $\leftrightarrow$

the sentence $\exists w_1, \ldots, w_m. \bigwedge_i \phi_i$ holds in Γ_Ψ.

The decision problem

whether or not a given *primitive positive sentence* holds in Γ_Ψ

is called the *Constraint Satisfaction Problem* of Γ_Ψ (or CSP(Γ_Ψ)).

Graph-SAT as Constraint Satisfaction Problem

An instance

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

of Graph-SAT(Ψ) has a positive solution $\leftrightarrow$

the sentence $\exists w_1, \ldots, w_m. \bigwedge_i \phi_i$ holds in Γ_Ψ.

The decision problem

whether or not a given *primitive positive sentence* holds in Γ_Ψ

is called the *Constraint Satisfaction Problem* of Γ_Ψ (or CSP(Γ_Ψ)).

So Graph-SAT(Ψ) and CSP(Γ_Ψ) are one and the same problem.

The algebraic approach:

From reducts to polymorphisms

Primitive positive (pp) definability and polymorphisms

Primitive positive (pp) definability and polymorphisms

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

Primitive positive (pp) definability and polymorphisms

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

Easy observation.

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

Primitive positive (pp) definability and polymorphisms

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

Easy observation.

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

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

Primitive positive (pp) definability and polymorphisms

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

Easy observation.

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

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

Generalization of endomorphism, automorphism.

Primitive positive (pp) definability and polymorphisms

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

Easy observation.

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

A function $f : \Gamma^n \to \Gamma$ is a *polymorphism* of Γ iff
for all relations R of Γ 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(Γ) for the set of polymorphisms of Γ.
"Polymorphism clone of Γ"

Primitive positive (pp) definability and polymorphisms

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

Easy observation.

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

A function $f : \Gamma^n \to \Gamma$ is a *polymorphism* of Γ iff
for all relations R of Γ 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(Γ) for the set of polymorphisms of Γ.
"Polymorphism clone of Γ"

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

The polymorphism strategy

The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP

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

The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP

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

Larger clones $\rightarrow$ easier CSP

$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \quad \rightarrow \quad \mathrm{CSP}(\Delta) \leq_{Poltime} \mathrm{CSP}(\Gamma)$

The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP

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

Larger clones $\rightarrow$ easier CSP

$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \quad \rightarrow \quad \mathrm{CSP}(\Delta) \leq_{Poltime} \mathrm{CSP}(\Gamma)$

Strategy:

(i) Prove hardness for certain relations

(ii) Prove that all reducts which do not pp-define any of these relations are tractable.

The polymorphism strategy

Larger reducts $\rightarrow$ harder CSP

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

Larger clones $\rightarrow$ easier CSP

$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \quad \rightarrow \quad \mathrm{CSP}(\Delta) \leq_{Poltime} \mathrm{CSP}(\Gamma)$

Strategy:

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

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

Ramsey theory:

Patterns in polymorphisms

Canonical functions

A function $f : G \to G$ 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.

Canonical functions

A function $f : G \to G$ 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.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of G.

Canonical functions

A function $f : G \to G$ 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.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of G.

Ramsey theory implies:
Every finite graph has a copy in G on which f is canonical.

Canonical functions

A function $f : G \to G$ 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.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of G.

Ramsey theory implies:
Every finite graph has a copy in G on which f is canonical.

Generalization of canonical to higher arity functions.

Canonical functions

A function $f : G \to G$ 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.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of G.

Ramsey theory implies:
Every finite graph has a copy in G on which f is canonical.

Generalization of canonical to higher arity functions.

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

Canonical functions

A function $f : G \to G$ 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.

Examples

- Function which switches edges and non-edges.
- Injection onto complete subgraph of G.

Ramsey theory implies:
Every finite graph has a copy in G on which f is canonical.

Generalization of canonical to higher arity functions.

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

Canonical functions are finite objects!

The Graph Satisfiability Problem

The Graph Satisfiability Problem

Let Ψ be a finite set of graph formulas.

Computational problem: Graph-SAT(Ψ)

INPUT:

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

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

The Graph Satisfiability Problem

Let Ψ be a finite set of graph formulas.

Computational problem: Graph-SAT(Ψ)

INPUT:

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

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

Theorem

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

The theorem in more detail

The theorem in more detail

Theorem

Let Γ be a reduct of the random graph. Then:

- Either Γ has one out of 17 canonical polymorphisms, and CSP(Γ) is tractable,
- or CSP(Γ) is NP-complete.

The theorem in more detail

Theorem

Let Γ be a reduct of the random graph. Then:

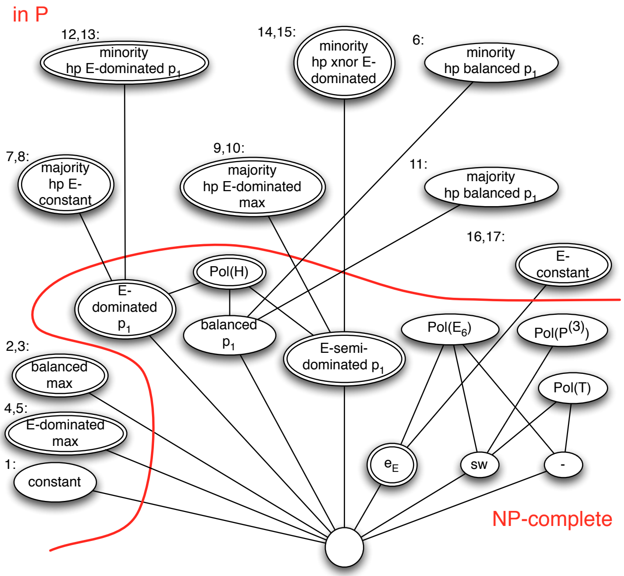
- Either Γ has one out of 17 canonical polymorphisms, and CSP(Γ) is tractable,
- or CSP(Γ) is NP-complete.

Theorem

Let Γ be a reduct of the random graph. Then:

- Either Γ pp-defines one out of 4 hard relations, and CSP(Γ) is NP-complete,
- or CSP(Γ) is tractable.

The border


in P
12,13:
minority hp E-dominated p_1
14,15:
minority hp xnor E-dominated
6:
minority hp balanced p_1
7,8:
majority hp E-constant
9,10:
majority hp E-dominated max
11:
majority hp balanced p_1
16,17:
E-constant
Pol(H)
E-dominated p_1
balanced p_1
E-semi-dominated p_1
Pol(E_6)
Pol($P^{(3)}$)
Pol(T)
2,3:
balanced max
4,5:
E-dominated max
1:
constant
e_E
sw
-
NP-complete

The Meta Problem

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

The Meta-Problem of Graph-SAT(Ψ) is decidable.