Schaefer's theorem for graphs

Manuel Bodirsky *
Laboratoire d'Informatique (LIX), CNRS UMR 7161
École Polytechnique
91128 Palaiseau, France
bodirsky@lix.polytechnique.fr

Michael Pinsker †
Équipe de Logique Mathématique
Université Denis-Diderot Paris 7
UFR de Mathématiques
75205 Paris Cedex 13, France
marula@gmx.at

ABSTRACT

Schaefer's theorem is a complexity classification result for so-called *Boolean constraint satisfaction problems*: it states that every Boolean constraint satisfaction problem is either contained in one out of six classes and can be solved in polynomial time, or is NP-complete.

We present an analog of this dichotomy result for the *propositional logic of graphs* instead of Boolean logic. In this generalization of Schaefer's result, the input consists of a set W of variables and a conjunction Φ of statements ("constraints") about these variables in the language of graphs, where each statement is taken from a fixed finite set Ψ of allowed quantifier-free first-order formulas; the question is whether Φ is satisfiable in a graph.

We prove that either Ψ is contained in one out of 17 classes of graph formulas and the corresponding problem can be solved in polynomial time, or the problem is NP-complete. This is achieved by a universal-algebraic approach, which in turn allows us to use structural Ramsey theory. To apply the universal-algebraic approach, we formulate the computational problems under consideration as constraint satisfaction problems (CSPs) whose templates are first-order definable in the countably infinite random graph. Our method to classify the computational complexity of those CSPs produces many statements of independent mathematical interest.

1. MOTIVATION AND THE RESULT

In an influential paper in 1978, Schaefer [22] proved a complexity classification for systematic restrictions of the Boolean satisfiability problem. The way how he restricts the Boolean satisfiability problem turned out to be very fruitful when restricting other computational problems in theoretical computer science, and can be presented as follows.

Let $\Psi = \{\psi_1, \ldots, \psi_n\}$ be a finite set of propositional (Boolean) formulas.

Boolean-SAT(Ψ)
INSTANCE: Given a finite set of variables W and a propositional formula of the form $\Phi = \phi_1 \wedge \cdots \wedge \phi_l$ where each ϕ_i for $1 \leq i \leq l$ is obtained from one of the formulas ψ in Ψ by substituting the variables of ψ by variables from W.
QUESTION: Is there a satisfying Boolean assignment to the variables of W (equivalently, those of Φ)?

The computational complexity of this problem clearly depends on the set Ψ, and is monotone in the sense that if $\Psi \subseteq \Psi'$, then solving Boolean-SAT(Ψ') is at least as hard as solving Boolean-SAT(Ψ). Schaefer's theorem states that Boolean-SAT(Ψ) can be solved in polynomial time if and only if Ψ is a subset of one of six sets of Boolean formulas (called *0-valid, 1-valid, Horn, dual-Horn, affine*, and *bijunctive*), and is NP-complete otherwise. We remark that Schaefer's theorem is usually formulated as a classification result of Boolean constraint satisfaction problems, but the formulation given above is easily seen to be equivalent.

We prove a similar classification result, but for the propositional logic of graphs instead for propositional Boolean logic. More precisely, let E be a relation symbol which denotes an antireflexive and symmetric binary relation and hence stands for the edge relation of a (simple, undirected) graph. We consider formulas that are constructed from atomic formulas of the form $E(x, y)$ and $x = y$ by the usual boolean connectives (negation, conjunction, disjunction), and call formulas of this form *graph formulas*. A graph formula $\Phi(x_1, \ldots, x_m)$ is *satisfiable* if there exists a graph H and an m-tuple a of elements in H such that $\Phi(a)$ holds in H. Let $\Psi = \{\psi_1, \ldots, \psi_n\}$ be a finite set of graph formulas. Then Ψ gives rise to the following computational problem.

Graph-SAT(Ψ)
INSTANCE: Given a set of variables W and a graph formula of the form $\Phi = \phi_1 \wedge \cdots \wedge \phi_l$ where each ϕ_i for $1 \leq i \leq l$ is obtained from one of the formulas ψ in Ψ by substituting the variables from ψ by variables from W.
QUESTION: Is Φ satisfiable?

*The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 257039).
†The second author is grateful for support through an APART-fellowship of the Austrian Academy of Sciences.

STOC'11, June 6–8, 2011, San Jose, California, USA.

As an example, let Ψ be the set that just contains the formula

$$\begin{aligned}&(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z))\\ \vee\ &(\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z))\\ \vee\ &(\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z))\ .\end{aligned} \tag{1}$$

Then Graph-SAT(Ψ) is the problem of deciding whether there exists a graph such that certain prescribed subsets of its vertex set of cardinality at most three induce subgraphs with exactly one edge. This problem is NP-complete (the curious reader can check this by means of our classification in Theorem 17). There are also many interesting tractable Graph-SAT problems, for instance when Ψ consists of the formulas $x \neq y \vee y = z$ and

$$\begin{aligned}&(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z))\\ \vee\ &(\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z))\\ \vee\ &(\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z))\\ \vee\ &(E(x,y) \wedge E(y,z) \wedge E(x,z))\ .\end{aligned} \tag{2}$$

It is obvious that the problem Graph-SAT(Ψ) is for all Ψ contained in NP. The goal of this paper is to prove the following dichotomy result.

THEOREM 1. *For all Ψ, the problem* Graph-SAT(Ψ) *is either NP-complete or in P.*

One of the main contributions of the paper is the general method of combining concepts from universal algebra and model theory, which allows us to use deep results from Ramsey theory to obtain the classification result.

2. DISCUSSION OF OUR STRATEGY

We establish our result by translating Graph-SAT problems into *constraint satisfaction problems* (CSPs) with infinite domains. More specifically, for every set of formulas Ψ we present a relational structure Γ_Ψ such that Graph-SAT(Ψ) is equivalent to CSP(Γ_Ψ) (in a certain sense, Graph-SAT(Ψ) and CSP(Γ_Ψ) are one and the same problem). The relational structure Γ_Ψ has a first-order definition in the *random graph* G, i.e., the (up to isomorphism) unique countably infinite universal homogeneous graph. The random graph belongs to one of the most fundamental ω-categorical structures, and is an important structure in model theory that appears also in many other areas of mathematics (see [14]). This perspective allows us to use the so-called *universal-algebraic approach*, and in particular *polymorphisms* to classify the computational complexity of Graph-SAT problems. In contrast to the universal-algebraic approach for finite domain constraint satisfaction, our proof relies crucially on strong results from structural Ramsey theory; we use such results to find regular patterns in the behavior of polymorphisms of structures on G, which in turn allows us to find analogies with polymorphisms of structures on Boolean domains.

We call structures with a first-order definition in G *reducts* of G. While the standard definition of a reduct of a relational structure Δ is a structure on the same domain obtained by forgetting some relations of Δ, a reduct of Δ in our sense is really a reduct of the expansion of Δ by all first-order definable relations. It turns out that there is one class of reducts Γ for which CSP(Γ) is in P for trivial reasons; further, there are 16 classes of reducts Γ for which CSP(Γ) (and the corresponding Graph-SAT problems) can be solved by non-trivial algorithms in polynomial time.

The presented algorithms are novel combinations of infinite domain constraint satisfaction techniques (such as used in [16, 7, 3]) and reductions to the tractable cases of Schaefer's theorem. Reductions of infinite domain CSPs in artificial intelligence (e.g., in temporal and spatial reasoning [17]) to finite domain CSPs (where typically the domain consists of the elements of a so-called 'relation algebra') have been considered in the more applied artificial intelligence literature [27]. Our results shed some light on the question when such techniques can even lead to *polynomial-time* algorithms for CSPs.

The global classification strategy of the present paper is similar in spirit to the one from a recent result in [6] on CSPs of structures which are first-order definable in $(\mathbb{Q};<)$. But while in [6] the proof might still have appeared to be very specific to constraint satisfaction over linear orders, with the present paper we demonstrate that in principle such a strategy can be used for any class of computational problems $\mathcal{C}$ that satisfies the following:

- All problems in $\mathcal{C}$ can be formulated as a CSP of a structure which is first-order definable in a single ω-categorical structure Δ;
- the class of finite substructures of Δ has the Ramsey property (as in [20]).

While in [6], the classical theorem of Ramsey and its product version were sufficient, the Ramsey theorems used in the present paper are deeper and considerably more difficult to prove [21, 1].

3. TOOLS FROM UNIVERSAL ALGEBRA AND MODEL THEORY

In this section we develop in detail the tools from universal algebra and model theory needed for our approach. We start by translating the problem Graph-SAT(Ψ) into a constraint satisfaction problem for a reduct of G.

We denote the random graph by $G = (V;E)$. The graph G is determined up to isomorphism by the two properties of being *homogeneous* (i.e., any isomorphism between two finite induced subgraphs of G can be extended to an automorphism of G), and *universal* (i.e., G contains all countable graphs as induced subgraphs). Moreover, it satisfies the *extension property*, which often is useful in combinatorial arguments: For all disjoint finite $U, U' \subseteq V$ there exists $v \in V$ such that v is adjacent in G to all members of U and to none in U'. Up to isomorphism, there exists only one unique countably infinite graph which has this extension property, and hence the property can be used as an alternative definition of G. The name of the random graph is due to the fact that if for a countably infinite vertex set, one chooses independently and with probability $\frac{1}{2}$ for each pair of vertices whether to connect the two vertices by an edge, then with probability 1 the resulting graph is isomorphic to the random graph. For the many other remarkable properties of G and its automorphism group Aut(G), and various connections to many branches of mathematics, see e.g. [14, 15].

Let Γ be a structure with a finite relational signature τ. A first-order τ-formula is called *primitive positive* if it is of the form $\exists x_1, \ldots, x_n.\ \psi_1 \wedge \cdots \wedge \psi_m$ where the ψ_i are *atomic*, i.e., of the form $y_1 = y_2$ or $R(y_1, \ldots, y_k)$, where $R \in \tau$ a k-ary relation symbol and the y_i are not necessarily distinct. A τ-formula is called a *sentence* if it contains no free variables.

DEFINITION 2. *The* constraint satisfaction problem for Γ*, denoted by* $\mathrm{CSP}(\Gamma)$*, is the computational problem of deciding for a given primitive positive τ-sentence Φ whether Φ is true in Γ.*

Let $\Psi = \{\psi_1, \ldots, \psi_n\}$ be a set of graph formulas. Then we define Γ_Ψ to be the structure with the same domain V as the random graph G which has for each ψ_i a relation R_i consisting of those tuples in G that satisfy ψ_i (where the arity of R_i is given by the number of variables that occur in ψ_i). Thus by definition, Γ_Ψ is a reduct of G. Now given any instance $\Phi = \phi_1 \wedge \cdots \wedge \phi_l$ with variable set W of Graph-SAT(Ψ), we construct a primitive positive sentence Φ' in the language of Γ_Ψ as follows: In Φ, we replace every ϕ_i, which by definition is of the form $\psi_j(y_1, \ldots, y_m)$ for some $1 \leq j \leq m$ and variables y_k from W, by $R_j(y_1, \ldots, y_m)$; after that, we existentially quantify all variables that occur in Φ'. It is then easy to see that the problem Graph-SAT(Ψ) has a positive answer for Φ if and only if the sentence Φ' holds in Γ_Ψ. Hence, every problem Graph-SAT(Ψ) *is* in fact of the form $\mathrm{CSP}(\Gamma)$, for a reduct Γ of G in a finite signature. We will thus henceforth focus on such constraint satisfaction problems in order to prove our dichotomy.

The following lemma has been first stated in [19] for finite structures Γ only, but the proof there also works for arbitrary infinite structures. It shows us how we can slightly enrich structures without changing the computational complexity of the constraint satisfaction problem they define too much.

LEMMA 3. *Let $\Gamma = (D; R_1, \ldots, R_l)$ be a relational structure, and let R be a relation that has a primitive positive definition in Γ. Then $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(D; R, R_1, \ldots, R_l)$ are polynomial-time equivalent.*

The preceding lemma makes the so-called *universal-algebraic approach* to constraint satisfaction possible, as exposed in the following. We say that a k-ary function (also called *operation*) $f : D^k \rightarrow D$ *preserves* an m-ary relation $R \subseteq D^m$ iff for all $t_1, \ldots, t_k \in R$ the tuple $f(t_1, \ldots, t_k)$ (calculated componentwise) is also contained in R. If an operation f does not preserve a relation R, we say that f *violates* R. If f preserves all relations of a structure Γ, we say that f is a *polymorphism* of Γ (it is also common to say that Γ is *closed under* f). A unary polymorphism of Γ is also called an *endomorphism* of Γ.

Conversely, for a set F of operations defined on a set D and a relation R on D, we say that R is *invariant* under F if R is preserved by all $f \in F$, and we write $\mathrm{Inv}(F)$ for the set of all finitary relations on D that are invariant under F.

The set of all polymorphisms $\mathrm{Pol}(\Gamma)$ of a relational structure Γ forms an algebraic object called a *clone* [24], which is a set of operations defined on a set D that is closed under composition and that contains all projections. Moreover, $\mathrm{Pol}(\Gamma)$ is also closed under interpolation (see Proposition 1.6 in [24]): we say that a k-ary operation f on D is *interpolated* by a set of k-ary operations F on D if for every finite subset A of D^k there is some operation $g \in F$ such that g agrees with f on A. We say that F *locally generates* an operation g if g is in the smallest clone that is closed under interpolation and contains all operations in F. Clones with the property that they contain all functions locally generated by their members are called *locally closed*, *local* or just *closed*.

We thus have that to every structure Γ, we can assign the closed clone $\mathrm{Pol}(\Gamma)$ of its polymorphisms. For certain Γ, this clone captures the computational complexity of $\mathrm{CSP}(\Gamma)$: A countable structure Γ is called *ω-categorical* if every countable model of the first-order theory of Γ is isomorphic to Γ. It is well-known that the random graph G is ω-categorical, and that reducts of ω-categorical structures are ω-categorical as well.

THEOREM 4 (FROM [8]). *Let Γ be an ω-categorical structure. Then the relations preserved by the polymorphisms of Γ (i.e., the relations in $\mathrm{Inv}(\mathrm{Pol}(\Gamma))$) are precisely those having a primitive positive definition in Γ.*

Clearly, the theorem implies that if two ω-categorical structures with finite relational signatures have the same clone of polymorphisms, then their CSPs are polynomial-time equivalent. Recall that we have only defined $\mathrm{CSP}(\Gamma)$ for structures Γ with a finite relational signature. But we now see that it makes sense (and here we follow conventions from finite domain constraint satisfaction, see e.g. [13]) to say that $\mathrm{CSP}(\Gamma)$ is *(polynomial-time) tractable* if the CSP for every finite signature structure Δ with the same polymorphism clone as Γ is in P, and to say that $\mathrm{CSP}(\Gamma)$ is *NP-hard* if there is a finite signature structure Δ with the same polymorphism clone as Γ whose CSP is NP-hard.

The following proposition is the analog to Theorem 4 on the "operational side".

PROPOSITION 5 (COROLLARY 1.9 IN [24]). *Let F be a set of functions on a domain D, and let g be a function on D. Then F locally generates g if and only if g preserves all relations that are preserved by all operations in F (i.e., if and only if $g \in \mathrm{Pol}(\mathrm{Inv}(F))$).*

For some reducts, we will find that their CSP is equivalent to a CSP of a structure that has already been studied, by means of the following basic observation.

PROPOSITION 6. *Let Γ, Δ be* homomorphically equivalent*, i.e., they have the same signature and there are homomorphisms $f : \Gamma \rightarrow \Delta$ and $g : \Delta \rightarrow \Gamma$. Then $\mathrm{CSP}(\Gamma) = \mathrm{CSP}(\Delta)$.*

The following general lemma allows one to restrict the arity of functions violating a relation. For a structure Γ with domain D and a tuple $t \in D^k$, the *orbit of* t in Γ is the set $\{\alpha(t) \mid \alpha \in \mathrm{Aut}(\Gamma)\}$.

LEMMA 7 (FROM [6]). *Let Γ be a relational structure with domain D, and suppose that $R \subseteq D^k$ consists of m orbits of k-tuples in Γ. Suppose that an operation f on D violates R. Then $\{f\} \cup \mathrm{Aut}(\Gamma)$ locally generates an at most m-ary operation that violates R.*

4. OVERVIEW OF THE PROOF

Throughout the text, Γ denotes a reduct of the random graph $G = (V; E)$. The proof of Theorem 1 can be organized in three steps as follows.

The first step is providing hardness proofs for certain relations. More precisely, we define three relations $P^{(3)}$, T, and

H which have first order-definitions in G, and show hardness for the CSP defined by each of these relations by reducing known NP-hard problems to this CSP. We then know from Lemma 3 that if the CSP for a reduct Γ is not NP-hard, then there is no primitive positive definition of any of these relations in Γ. This implies that there are polymorphisms of Γ which violate the NP-hard relations, by Theorem 4.

We then analyze the polymorphisms of Γ which violate $P^{(3)}$, T, and H. The first, rather basic tool here is Lemma 7, which we use in order to get bounds on the arity of such polymorphisms. The deeper part of our analysis is the simplification of the polymorphisms by means of Ramsey theory. It turns out that the polymorphisms can be assumed to behave regularly in a certain sense with respect to the base structure G (the technical term for functions showing such regular behavior will be *canonical*), making them accessible to case-by-case analysis.

Finally, the presence of canonical polymorphisms is used in two ways: in the case of canonical unary polymorphisms, the image under such a polymorphism sometimes is a structure Δ for which the CSP has already been classified, and then one can refer to Proposition 6 to argue that the CSP(Γ) is polynomial-time equivalent to the CSP of this structure Δ. The second, and in our case more important way of employing canonical polymorphisms is to prove tractability of CSP(Γ) by transforming the polymorphism into an algorithm. Here, we adapt known algorithms showing that certain polymorphisms on a Boolean domain imply tractability of Boolean CSPs in order to prove that the same holds for their canonical counterparts on the Random graph.

It turned out convenient to split our proof into two cases, the first one dealing with the case where either E or the relation $N(x,y)$, which is defined by the formula $\neg E(x,y) \wedge x \neq y$, has no primitive positive definition in Γ, and the second one dealing with the case where both E and N are primitive positive definable in Γ. In theory, both cases follow the steps described above. However, the assumption for the first case implies the presence of endomorphisms of Γ which violate either E or N, and in turn allows for the use of known results on such endomorphisms, in particular from [26] and [10]. These older results have been obtained using Ramsey theory, and thus by building on them we "outsource" the Ramsey-theoretic analysis of polymorphisms of Γ in this case. In the second case, all endomorphisms of Γ will be trivial in the sense that they preserve E and N (which means that they locally look like automorphisms of G), and thus the higher arity polymorphisms of Γ must be considered to a level of detail not present in the literature (although we do also draw on earlier results on such higher arity polymorphisms from [10]) . It is here where we apply Ramsey theory directly in our paper.

The following two sections correspond to these two cases, and outline our proof in more but far from full detail. The full proof can be found in the long version of the paper which is available on the arxiv [11], and is equally divided into two sections corresponding to the two cases described above.

Before starting out we add the following convention: since all our polymorphism clones contain the automorphism group $\mathrm{Aut}(G)$ of the random graph, we will abuse the notion of *generates* from the preceding section, and use it as follows: For a set of functions F and a function g on the domain V, we say that F *generates* g when $F \cup \mathrm{Aut}(G)$ locally generates g; also, we say that a function f *generates* g if $\{f\}$ generates g.

5. ENDOMORPHISMS

The goal of this section is Proposition 10, which allows us to reduce the classification task to the classification of those structures Γ where the relations E, N and $\neq$ are primitive positive definable. We first define two hard relations on G.

DEFINITION 8. *Let $P^{(3)}$ denote the ternary relation that holds for $(x_1, x_2, x_3) \in V^3$ if x_1, x_2, x_3 are pairwise distinct, and the graph induced by $\{x_1, x_2, x_3\}$ in G is neither an independent set nor a clique.*

DEFINITION 9. *Let T be the 4-ary relation that holds for a tuple $(x_1, x_2, x_3, x_4) \in V^4$ if x_1, x_2, x_3, x_4 are pairwise distinct, and induce in G one of the following*

a single edge and two isolated vertices

a path with two edges and an isolated vertex

a path with three edges

the complement of one of those structures.

PROPOSITION 10. *Let Γ be a reduct of G. Then at least one of the following holds.*

(a) Γ has a constant endomorphism, and CSP(Γ) *is tractable (it is in fact trivial).*

(b) Γ is homomorphically equivalent to a countably infinite structure that is preserved by all permutations of its domain; in this case the complexity of CSP(Γ) *has been classified in [5], and is either tractable or NP-hard.*

(c) There is a primitive positive definition of $P^{(3)}$ or T in Γ, and CSP(Γ) *is NP-hard.*

(d) The relations N, E, and $\neq$ have primitive positive definitions in Γ.

The idea of the proof of this proposition is to first show hardness of the relations $P^{(3)}$ and T, and then work under the assumption that neither (c) nor (d) hold. This assumption implies the presence of polymorphisms of Γ violating $P^{(3)}$, T, and either E or N (the relation $\neq$ has a primitive positive definition from E and N, and is irrelevant in this section). We then use a result due to Thomas [26] (also see [10]) about endomorphism monoids of reducts of G. The result states that if Γ is any reduct of G, then either Γ has a constant endomorphism or an endomorphism whose image induces a clique or an independent set in G, or all endomorphisms of Γ are locally generated by the automorphisms of Γ. In the first case, i.e., if Γ has one of the mentioned endomorphisms, then either (a) or (b) of Proposition 10 hold, and we are done. Otherwise, we know that the endomorphisms of Γ locally look like automorphisms, and apply a classification of all automorphism groups of reducts of G from [25] (see also [10]) – there are only 5 such groups (observe that distinct reducts can have identical automorphism groups; in fact, this is the case if and only if the reducts first-order define one another). By our assumption above, Γ has a polymorphism violating either E or N; since E and N consist of just one orbit of pairs over G and since $\mathrm{Aut}(\Gamma) \supseteq \mathrm{Aut}(G)$, they consist of at most one orbit of pairs over Γ. Hence, by Lemma 7 there is an endomorphism of Γ violating E or N, and since this endomorphism is locally generated by $\mathrm{Aut}(\Gamma)$, there is even an automorphism of Γ that violates E or N. In particular, $\mathrm{Aut}(\Gamma) \supsetneq \mathrm{Aut}(G)$. We then argue similarly using the existence of polymorphisms that violate $P^{(3)}$ and T together with the above-mentioned classification of automorphism groups of reducts of G to conclude that all permutations of V are automorphisms of Γ, putting us back into Case (b) of Proposition 10.

We mention that the proof of the two results from [25, 26] (our own proof of both results can be found in [10]) naturally involves Ramsey-theoretic methods; these methods are applied in order to find patterns of regular behavior in *unary* functions on G. In the next section, we consider the case where it will be necessary to refine these methods in order to understand *higher arity* functions on G.

6. HIGHER ARITY POLYMORPHISMS

In this section we assume that $\Gamma = (V; E, N, \neq, \dots)$ is a reduct of G for which (d) in Proposition 10 applies, and which therefore contains the relations E, N and $\neq$. While the result of the last section was based on an analysis of the endomorphisms and automorphisms of reducts of G, the remaining cases require the study of higher arity polymorphisms of such reducts. It turns out that the relevant polymorphisms proving tractability have, in a certain sense, regular behavior with respect to the structure of G. Combinatorially, this is due to the fact that the set of finite ordered graphs is a *Ramsey class*: for structures Δ which can be expanded by a linear order in such a way that the set of finite induced substructures of this expansion is a Ramsey class, one can find "regular patterns" in any arbitrary function on Δ. A survey of this general method is [9] (see also [12]). We now make this notion of regularity more precise.

Definition 11. *Let Δ be a structure. The* type $\mathrm{tp}(a)$ *of an n-tuple a of elements of Δ is the set of first-order formulas with free variables $x_1, \dots, x_n$ that hold for a in Δ. Types of n-tuples in Δ are also called* n-types.

If a is an n-tuple, we write a_i for the i-th component of a.

Definition 12. *Let Δ, Λ be structures, and let $k \geq 1$. A* k-ary type condition *between Δ and Λ is a pair (t, s), where t is a k-tuple $(t^1, \dots, t^k)$ of n-types t^i in Δ, and s is an n-type in Λ. A k-ary function $f : \Delta^k \to \Lambda$* satisfies *a type condition (t, s) if for all n-tuples a^i of type t^i in Δ the n-tuple $(f(a^1_1, \dots, a^k_1), \dots, f(a^1_n, \dots, a^k_n))$ is of type s in Λ. A* behavior *is a set of k-ary type conditions between two structures Δ and Λ, where $k \geq 1$ is fixed. A k-ary function* has behavior B *if it satisfies all type conditions of the behavior B.*

Definition 13. *Let Δ, Λ be structures. An operation $f : \Delta^k \to \Lambda$ is* canonical *if for all $n \geq 1$ and all k-tuples t of n-types in Δ there exists an n-type s in Λ such that f satisfies the type condition (t, s).*

Observe that if $f : \Delta^k \to \Lambda$ is canonical, then for each n it defines a function from the set of k-tuples of n-types in Δ into the set of n-types in Λ. In the case where Δ and Λ have only finitely many n-types (in particular if $\Delta = \Lambda = G$, since G is homogeneous in a finite language – see [18]), these functions are finite objects, and there are only finitely many such functions for each n. Moreover, since G has only binary relations, a function $f : G^k \to G$ is canonical iff it satisfies the condition of the definition for types of 2-tuples (i.e., for $n = 2$). Thus, its behavior is determined by a function from the set of k-tuples of 2-types in G to the set of 2-types in G – a finite object.

It follows easily from the homogeneity of G and by local closure that if two canonical functions $f, g : V^k \to V$ have the same behavior, then they generate one another. Thus, for our purposes canonical functions on G can really be considered as finite objects (namely, as the functions on 2-types they define).

The polymorphisms proving tractability of reducts of G will be canonical (with respect to the structure G, not the reduct). We now define different behaviors that some of these canonical functions will have. For relatons $Q_1, \dots, Q_k \in \{E, N, =, \neq\}$, we will in the following write $Q_1 \cdots Q_k$ for the binary relation on V^k that holds between two k-tuples $x, y \in V^k$ iff $Q_i(x_i, y_i)$ holds for all $1 \leq i \leq k$ (here x_i and y_i denote the i-th component of the k-tuples x and y, respectively).

If we flip edges and non-edges of G, then the resulting graph is isomorphic to G (it is straightforward to verify the extension property). Let $-$ be such an isomorphism. The *dual* of an operation f on G is the operation

$$(x_1, \dots, x_n) \mapsto -f(-x_1, \dots, -x_n) \,,$$

and can be imagined as the function obtained from f by exchanging the roles of E and N.

We start by behaviors of binary functions.

Definition 14. *We say that a binary injective operation $f : V^2 \to V$ is*

- balanced in the first argument *if for all $u, v \in V^2$ we have that* $\mathrm{E{=}}(u, v)$ *implies $E(f(u), f(v))$ and* $\mathrm{N{=}}(u, v)$ *implies $N(f(u), f(v))$;*
- balanced in the second argument *if $(x, y) \mapsto f(y, x)$ is balanced in the first argument;*
- balanced *if f is balanced in both arguments, and* unbalanced *otherwise;*
- E-dominated (N-dominated) in the first argument *if $E(f(u), f(v))$ $(N(f(u), f(v)))$ for all $u, v \in V^2$ with ${\neq}{=}(u, v)$; and* E-dominated (N-dominated) in the second argument *if $(x, y) \mapsto f(y, x)$ is E-dominated (N-dominated) in the first argument;*
- E-dominated (N-dominated) *if it is E-dominated (N-dominated) in both arguments;*
- of type min *if for all $u, v \in V^2$ with ${\neq}{\neq}(u, v)$ we have $E(f(u), f(v))$ if and only if* $\mathrm{EE}(u, v)$*; and* of type max *if the dual of f is of type* min.
- of type p_1 *if for all $u, v \in V^2$ with ${\neq}{\neq}(u, v)$ we have $E(f(u), f(v))$ if and only if $E(u_1, v_1)$, and* of type p_2 *if $(x, y) \mapsto f(y, x)$ is of type p_1;*
- of type projection *if it is of type p_1 or p_2.*

Note that, for example, being of type max is a behavior of binary functions that does not force a function to be canonical, since the condition only talks about certain 2-tuples of

2-types, but not all such 2-tuples; however, being both of type max and balanced does imply for a function that it is canonical.

The next definition contains some important behaviors of ternary functions.

Definition 15. *An injective ternary function $f : V^3 \to V$ is of type*

- majority *if for all $u, v \in V^3$ with $\neq\neq\neq(u, v)$ we have that $E(f(u), f(v))$ if and only if* $\mathrm{EEE}(u, v)$, $\mathrm{EEN}(u, v)$, $\mathrm{ENE}(u, v)$, *or* $\mathrm{NEE}(u, v)$*;*
- minority *if for all $u, v \in V^3$ with $\neq\neq\neq(u, v)$ we have that $E(f(u), f(v))$ if and only if* $\mathrm{EEE}(u, v)$, $\mathrm{NNE}(u, v)$, $\mathrm{NEN}(u, v)$, *or* $\mathrm{ENN}(u, v)$.

While the tractability results of this section are shown by means of a number of different canonical functions, all hardness cases are established by the following single relation.

Definition 16. *We define a relation $H(x_1, y_1, x_2, y_2, x_3, y_3)$ on V by*

$$\begin{aligned} &\bigwedge_{i,j\in\{1,2,3\}, i\neq j, u\in\{x_i,y_i\}, v\in\{x_j,y_j\}} N(u, v) \\ &\wedge \big(((E(x_1, y_1) \wedge N(x_2, y_2) \wedge N(x_3, y_3)) \\ &\quad \vee\ (N(x_1, y_1) \wedge E(x_2, y_2) \wedge N(x_3, y_3)) \\ &\quad \vee\ (N(x_1, y_1) \wedge N(x_2, y_2) \wedge E(x_3, y_3))\big)\ . \end{aligned}$$

The following theorem together with Proposition 10 proves Theorem 1.

Theorem 17. *Let $\Gamma = (V; E, N, \neq, \dots)$ be a reduct of G. Then one of the following holds:*

(a) There is a primitive positive definition of H in Γ, and $\mathrm{CSP}(\Gamma)$ is NP-hard.

(b) Γ has a canonical polymorphism of type minority, as well as a canonical binary injection which is of type p_1 and E-dominated or N-dominated in the second argument, and $\mathrm{CSP}(\Gamma)$ is tractable.

(c) Γ has a canonical polymorphism of type majority, as well as a canonical binary injection which is of type p_1 and E-dominated or N-dominated in the second argument, and $\mathrm{CSP}(\Gamma)$ is tractable.

(d) Γ has a canonical polymorphism of type minority, as well as a canonical binary injection which is balanced and of type projection, and $\mathrm{CSP}(\Gamma)$ is tractable.

(e) Γ has a canonical polymorphism of type majority, as well as a canonical binary injection which is balanced and of type projection, and $\mathrm{CSP}(\Gamma)$ is tractable.

(f) Γ has a canonical polymorphism of type max *or* min*, and $\mathrm{CSP}(\Gamma)$ is tractable.*

The proof of Theorem 17 splits into two tasks: we have to show that if (a) does not hold, then Γ has the polymorphisms of one of the other cases; the second task is to prove that these polymorphisms imply tractability of the CSP defined by Γ. The former is achieved by applying structural Ramsey theory to polymorphisms to find canonical behavior; the latter by adapting algorithms for polymorphisms on Boolean domains.

We now outline our method for the first task in more detail. If (a) does not hold, then by Theorem 4 there exists a polymorphism of Γ which violates H. Since H consists of 3 orbits of 6-tuples with respect to G, there exists a ternary polymorphism f violating H, by Lemma 7.

In spirit, the idea then is to analyze f by means of Ramsey theory. (We remark that in the full proof of the long version of this paper [11], we employ a more sophisticated strategy in order to simplify f before this analysis.) In order to be able to apply Ramsey theory, we expand G by a linear order $<$ on V in such a way that $(V; E, <)$ is the *ordered random graph*, i.e., the unique countably infinite homogeneous graph which contains all countable linearly ordered graphs as induced subgraphs. Since the set of finite linearly ordered graphs is a Ramsey class, we can then apply the following proposition (see [9], [10], [12]).

Proposition 18 ([10], [12]). *Let Δ be a homogeneous structure with a linear order which has the property that its set of finite induced substructures is a Ramsey class, and let $g : \Delta^k \to \Delta$. Let moreover constants $c_1, \dots, c_j$ in Δ be given, and denote by $(\Delta, c_1, \dots, c_j)$ the expansion of Δ by these constants. Then g generates together with $\mathrm{Aut}(\Delta)$ a k-ary operation h on Δ which is canonical as a function from $(\Delta, c_1, \dots, c_j)^k$ to Δ and which agrees with g on $(c_1, \dots, c_j)^k$.*

This implies that we can fix finitely many constants $c_1, \dots, c_j$ in G such that f violates H on $\{c_1, \dots, c_j\}$, and then by the preceding proposition assume that f is canonical as a function from $(V; E, <, c_1, \dots, c_j)^3$ to $(V; E, <)$, making it accessible to case-by-case analysis. This analysis finally gives us the canonical functions of the cases of Theorem 17.

We remark that the description just given only shows the rough idea of the proof, and that in fact the proof deviates quite a bit from this description. The main reason for this is that there are too many possible behaviors of canonical functions from $(V; E, <)^3$ to $(V, E, <)$ for simple case-by-case analysis. In our proof, we avoid cases in particular by composing f with functions from the following theorem, which helps us to reduce the task significantly; this theorem has, however, been proven by the same Ramsey-theoretic methods.

Theorem 19 ([10]). *If $\Gamma = (V; E, N, \neq, \dots)$ is a reduct of G that has an essential polymorphism, it must also have one of the following binary injective canonical polymorphisms:*

- *a balanced operation of type p_1;*
- *a balanced operation of type* max*;*
- *an E-dominated operation of type* max*;*
- *an E-dominated operation of type p_1;*
- *a binary operation of type p_1 that is balanced in the first and E-dominated in the second argument;*

or one of the duals of the last four operations (the first operation is self-dual).

We now come to the second task mentioned for the proof of Theorem 17: we have to present a polynomial-time algorithm for each of the Cases (b), (c), (d), (e), and (f) in Theorem 17. The algorithms for Case (b) and (c) will be simple reductions to Cases (d) and (e), and will not be discussed further in the extended abstract. For the remaining cases, our algorithms use the fact that a relation R with a first-order definition in G is preserved by the operations specified in (d), (e), or (f) if and only if R can be defined by a first-order formula over G that satisfies certain syntactic restrictions. This syntactic form can then be exploited by an algorithm.

For operations of type max or min, such a syntactic description and a corresponding algorithm is already known [3]; for example, a relation is preserved by a binary N-dominated injection of type min if and only if it has a quantifier-free Horn definition over the structure $(V; E, =)$, that is, a definition over $(V; E, =)$ in conjunctive normal form where every clause contains at most one negative literal (of the form $\neg E(x, y)$ or of the form $x \neq y$).

When a relation R with a first-order definition over G has the additional property that all its tuples contain only pairwise distinct entries, we have an analogy to the known tractable classes of Schaefer's classification:

- when R as above is preserved by a function of type majority, then it can be defined by a conjunction of formulas of the form $u \neq v$ or of the form $X(u_1, v_1) \vee Y(u_2, v_2)$ where $X, Y \in \{E, N\}$ (resembling the class of 2-SAT formulas in the classification of Schaefer).
- when R as above is preserved by a function of type minority, then it can be defined by a conjunction of formulas of the form $u \neq v$ or of the form $E(u_1, v_1) \oplus \cdots \oplus E(u_l, v_l) = p$ where $p \in \{0, 1\}$ and $\oplus$ denotes the Boolean exclusive-or connective (resembling the class of affine formulas in the classification of Schaefer).

In the general case, when the relation might also contain tuples with equal entries, our syntactic form is more intricate; we only present it here for balanced operations of type minority (Case (d)); the situation in case of operations of type majorities (Case (e)) is similar (but the corresponding proofs are different). A graph formula is called *edge affine* if it is a conjunction of formulas of the form

$$\begin{aligned} x_1 \neq y_1 \vee \ldots \vee x_k \neq y_k \\ \vee (u_1 \neq v_1 \wedge \cdots \wedge u_l \neq v_l \\ \wedge E(u_1, v_1) \oplus \cdots \oplus E(u_l, v_l) = p) \\ \vee (u_1 = v_1 \wedge \cdots \wedge u_l = v_l), \end{aligned}$$

where $p \in \{0, 1\}$, variables need not be distinct, and each of k and l can be 0.

PROPOSITION 20. *Let R be a relation with a first-order definition over G. Then the following are equivalent:*

1. *R can be defined by an edge affine formula;*
2. *R is preserved by an injection of type minority, and a balanced binary injection of type p_1.*

When all constraints in the input have a definition by an edge affine formula, this can be exploited by an algorithm as follows. The central idea is to compute an additional graph on *unorderd pairs* of vertices of the given CSP instance. Two such pairs $\{a, b\}$ and $\{c, d\}$ are connected in this graph if there is a constraint whose definition has a conjunct

$$\begin{aligned} (u_1 \neq v_1 \wedge \cdots \wedge u_l \neq v_l \wedge (E(u_1, v_1) \oplus \cdots \oplus E(u_l, v_l) = p)) \\ \vee (u_1 = v_1 \wedge \cdots \wedge u_l = v_l) \end{aligned}$$

such that $\{a, b\} = \{u_i, v_i\}$ and $\{c, d\} = \{u_j, v_j\}$ for some $i, j \in \{1, \ldots, l\}$. Conjuncts of this form (that is, conjuncts as in the definition of edge affine formulas where the leading disjunction of inequalities is empty, $k = 0$) are called *basic*. For each connected component C of the graph, the algorithm tries to find a solution for the subset of basic conjuncts where each pair $\{u_i, v_i\}$ is contained in C. If the algorithm finds such a solution for each component, it is straightforward to patch together those solutions to obtain a solution to all constraints. Otherwise, if for component C we do not find such a solution, then the syntactic form tells us that in all solutions, for all pairs $\{a, b\}$ in C, the variables a and b must have the same value, and so we contract all pairs in C, and restart the algorithm. For details of this sketch we have to refer to the full version.

7. CLASSIFICATION

By Theorem 1, each reduct of the random graph with finitely many relations defines a CSP which is either tractable or NP-complete. We now give a list of 17 reducts Γ with the following properties (assuming that P$\neq$ NP): (1) For any reduct Δ with finitely many relations, CSP(Δ) is in P if and only if the relations of Δ are a subset of one of the reducts of our list, and (2) the list is minimal, i.e., if one reduct Γ is removed from our list, then the list loses property (1).

Clearly, if we add relations to a reduct Γ, then the CSP of the structure thus obtained is computationally at least as complex as the CSP of Γ. On the other hand, by Lemma 3, adding relations with a primitive positive definition to a reduct does not increase the computational complexity of the corresponding CSP more than polynomially. In this section, we consider the lattice of reducts of G which are closed under primitive positive definitions (i.e., which contain all relations that are primitive positive definable from the reduct), and describe the border between tractability and NP-completeness in this lattice. We remark that the reducts will, since we expand them by all primitive positive definable relations, have infinitely many relations, and hence do not define a CSP; however, as already stated earlier, consider a reduct Γ tractable if and only if all structures with domain V which have finitely many relations, all taken from Γ, have a tractable CSP. Similarly, we consider a reduct Γ to be hard if it has at least one hard relation. With this convention, it is interesting to determine the *maximal tractable* reducts, i.e., those reducts closed under primitive positive definitions which do not contain any hard relation and which cannot be further extended without losing this property.

By Theorem 4 and Proposition 5, the lattice of primitive positive closed reducts of G and the lattice of locally closed clones containing Aut(G) are antiisomorphic via the mappings $\Gamma \mapsto \mathrm{Pol}(\Gamma)$ (for reducts Γ) and $\mathcal{C} \mapsto \mathrm{Inv}(\mathcal{C})$ (for clones $\mathcal{C}$). We refer to the introduction of [4] for a detailed exposition of this well-known connection. Therefore, the maximal tractable reducts correspond to *minimal tractable* clones, which are precisely the clones of the form $\mathrm{Pol}(\Gamma)$

for a maximal tractable reduct. Determining these minimal tractable clones is the goal of this section.

Definition 21. *Let B be a behavior for binary functions on G. A ternary injection $f : V^3 \to V$ is* hyperplanely of type B *if the binary functions $(x, y) \mapsto f(x, y, c)$, $(x, z) \mapsto f(x, c, z)$, and $(y, z) \mapsto f(c, y, z)$ have behavior B for all $c \in V$.*

We now define some more behaviors of binary functions which will appear "hyperplanely" in ternary functions in our classification.

Definition 22. *A binary injection $f : V^2 \to V$ is of type*

- E-constant *if the image of f is a clique;*
- N-constant *if the image of f is an independent set;*
- xnor *if for all $u, v \in V^2$ with $\neq\neq(u, v)$ the relation $E(f(u), f(v))$ holds if and only if* $\mathrm{EE}(u, v)$ *or* $\mathrm{NN}(u, v)$ *holds;*
- xor *if for all $u, v \in V^2$ with $\neq\neq(u, v)$ the relation $E(f(u), f(v))$ holds if and only if neither* $\mathrm{EE}(u, v)$ *nor* $\mathrm{NN}(u, v)$ *hold.*

Theorem 23. *The following 17 distinct clones are precisely the minimal tractable local clones containing* $\mathrm{Aut}(G)$*:*

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.*
11. *The clone generated by a function of type minority which is hyperplanely balanced and of type projection.*
12. *The clone generated by a function of type minority which is hyperplanely of type projection and E-dominated.*
13. *The clone generated by a function of type minority which is hyperplanely of type projection and N-dominated.*
14. *The clone generated by a function of type minority which is hyperplanely of type* xnor *and E-dominated.*
15. *The clone generated by a function of type minority which is hyperplanely of type* xor *and N-dominated.*
16. *The clone generated by a binary injection which is E-constant.*
17. *The clone generated by a binary injection which is N-constant.*

It follows from Theorem 23 that the so-called *meta-problem* of deciding for a given finite set of graph formulas Ψ whether Graph-SAT(Ψ) is in P or NP-hard, is decidable. Under the assumption that a formula in Ψ with k variables is given as a disjunction of formulas each of which defines a distinct orbit of k-tuples over the random graph, this problem is in fact even in P. It is clear that every graph formula can be transformed into an equivalent formula satisfying this assumption (sometimes this transformation might lead to an exponential blow-up).

The algorithm for deciding whether Graph-SAT(Ψ) is in P then works as follows. We test whether all formulas ψ in Ψ are preserved by one of the canonical operations implying tractability of Graph-SAT(Ψ) from the statement of the main result. To do so, we apply the canonical operation to orbit representatives from tuples satisfying ψ in all possible ways; since the operations we have to consider are at most ternary, the number of possibilities is at most cubic in the number of orbits satisfying ψ (which equals the representation size of ψ, by the assumption made above).

We also obtain the following. Define a relation E_6 by

$$\begin{aligned}\{(x_1, x_2, y_1, y_2, z_1, z_2) \in V^6 \mid (x_1 = x_2 \wedge y_1 \neq y_2 \wedge z_1 \neq z_2)\\ \vee\ (x_1 \neq x_2 \wedge y_1 = y_2 \wedge z_1 \neq z_2)\\ \vee\ (x_1 \neq x_2 \wedge y_1 \neq y_2 \wedge z_1 = z_2)\}\ .\end{aligned}$$

This relation has the property that the clone $\mathrm{Pol}(E_6)$ contains precisely all unary injective operations, and no other operations.

Corollary 24. *For all reducts Γ of G, $\mathrm{CSP}(\Gamma)$ is tractable, or one of the following relations has a primitive positive definition in Γ: the relation E_6, or the relation T, H, or $P^{(3)}$.*

Figure 1 shows the border between the hard and the tractable clones. The picture contains all minimal tractable clones as well as all maximal hard clones, plus some other clones that are of interest in this context. When two clones are connected by a line, we do not mean to imply that there are no other clones between them. Clones are symbolized with a double border when they have a dual clone (generated by the dual function as defined before, whose behavior is obtained by exchanging E with N, max with min, and xnor with xor). Of two dual clones, only one representative (the one which has E and max in its definition) is included in the picture. The numbers of the minimal tractable clones refer to the numbers in Theorem 23. "E-semidominated" refers to "balanced in the first and E-dominated in the second argument".

To conclude, we would like to mention an elegant universal-algebraic formulation of our main result, which lines up with recent conjectures and results on finite domain CSPs [23, 13].

Corollary 25. *Let Γ be a reduct of G. Then exactly one of the following applies.*

- *The structure* $(\{0,1\}; \{(0,0,1),(0,1,0),(1,0,0)\})$ *has a primitive positive interpretation (see e.g. [2, 9]) in* Γ*. In this case,* $\mathrm{CSP}(\Gamma)$ *is NP-hard.*
- Γ *has a canonical 4-ary polymorphism* f *and* $\alpha_1, \alpha_2 \in \mathrm{Aut}(\Gamma)$ *so that for all* $x, y \in V$

$$f(y,y,x,x) = \alpha_1(f(x,x,x,y)) = \alpha_2(f(y,x,y,x)) .$$

In this case, $\mathrm{CSP}(\Gamma)$ *is is tractable.*

8. REFERENCES

[1] Fred G. Abramson and Leo Harrington. Models without indiscernibles. *Journal of Symbolic Logic*, 43(3):572–600, 1978.

[2] Manuel Bodirsky. Constraint satisfaction problems with infinite templates. In Heribert Vollmer, editor, *Complexity of Constraints (a collection of survey articles)*, pages 196–228. Springer, LNCS 5250, 2008.

[3] Manuel Bodirsky, Hubie Chen, Jan Kára, and Timo von Oertzen. Maximal infinite-valued constraint languages. *Theoretical Computer Science (TCS)*, 410:1684–1693, 2009. A preliminary version appeared at ICALP'07.

[4] Manuel Bodirsky, Hubie Chen, and Michael Pinsker. The reducts of equality up to primitive positive interdefinability. *Journal of Symbolic Logic*, 75(4):1249–1292, 2010.

[5] Manuel Bodirsky and Jan Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 3(2):136–158, 2008. A conference version appeared in the proceedings of CSR'06.

[6] Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2), 2009. An extended abstract appeared in the proceedings of STOC'08.

[7] Manuel Bodirsky and Jan Kára. A fast algorithm and Datalog inexpressibility for temporal reasoning. *ACM Transactions on Computational Logic*, 11(3), 2010.

[8] Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.

[9] Manuel Bodirsky and Michael Pinsker. Reducts of Ramsey structures. In *Model Theoretic Methods in Finite Combinatorics*, Contemporary mathematics. American Mathematical Society. To appear; preprint available from the authors' websites.

[10] Manuel Bodirsky and Michael Pinsker. Minimal functions on the random graph. Preprint, arXiv:1003.4030, 2010.

[11] Manuel Bodirsky and Michael Pinsker. Schaefer's theorem for graphs (full version). Preprint available from http://www.dmg.tuwien.ac.at/pinsker/, 2010.

[12] Manuel Bodirsky, Michael Pinsker, and Todor Tsankov. Decidability of definability. In *LICS'11*, 2011. To appear. Preprint arXiv:1012.2381.

[13] Andrei Bulatov, Andrei Krokhin, and Peter G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.

[14] Peter J. Cameron. The random graph. *Algorithms and Combinatorics*, 14:333–351, 1997.

[15] Peter J. Cameron. The random graph revisited. In *Proceedings of the European Congress of Mathematics*, volume 201, pages 267–274. Birkhäuser, 2001.

[16] David A. Cohen, Peter Jeavons, Peter Jonsson, and Manolis Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.

[17] Ivo Duentsch. Relation algebras and their application in temporal and spatial reasoning. *Artificial Intelligence Review*, 23:315–357, 2005.

[18] Wilfrid Hodges. *Model theory*. Cambridge University Press, 1993.

[19] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.

[20] Jaroslav Nešetřil. Ramsey theory. *Handbook of Combinatorics*, pages 1331–1403, 1995.

[21] Jaroslav Nešetřil and Vojtech Rödl. Ramsey classes of set systems. *Journal of Combinatorial Theory, Series A*, 34(2):183–201, 1983.

[22] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.

[23] Mark H. Siggers. A strong Mal'cev condition for varieties omitting the unary type. *Algebra universalis*, 64(1):15–20, 2010.

[24] Ágnes Szendrei. *Clones in universal algebra*. Séminaire de Mathématiques Supérieures. Les Presses de L'Université de Montréal, 1986.

[25] Simon Thomas. Reducts of the random graph. *Journal of Symbolic Logic*, 56(1):176–181, 1991.

[26] Simon Thomas. Reducts of random hypergraphs. *Annals of Pure and Applied Logic*, 80(2):165–193, 1996.

[27] Matthias Westphal and Stefan Wölfl. Qualitative CSP, finite CSP, and SAT: Comparing methods for Qualitative Constraint-based Reasoning. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 628–633, 2009.

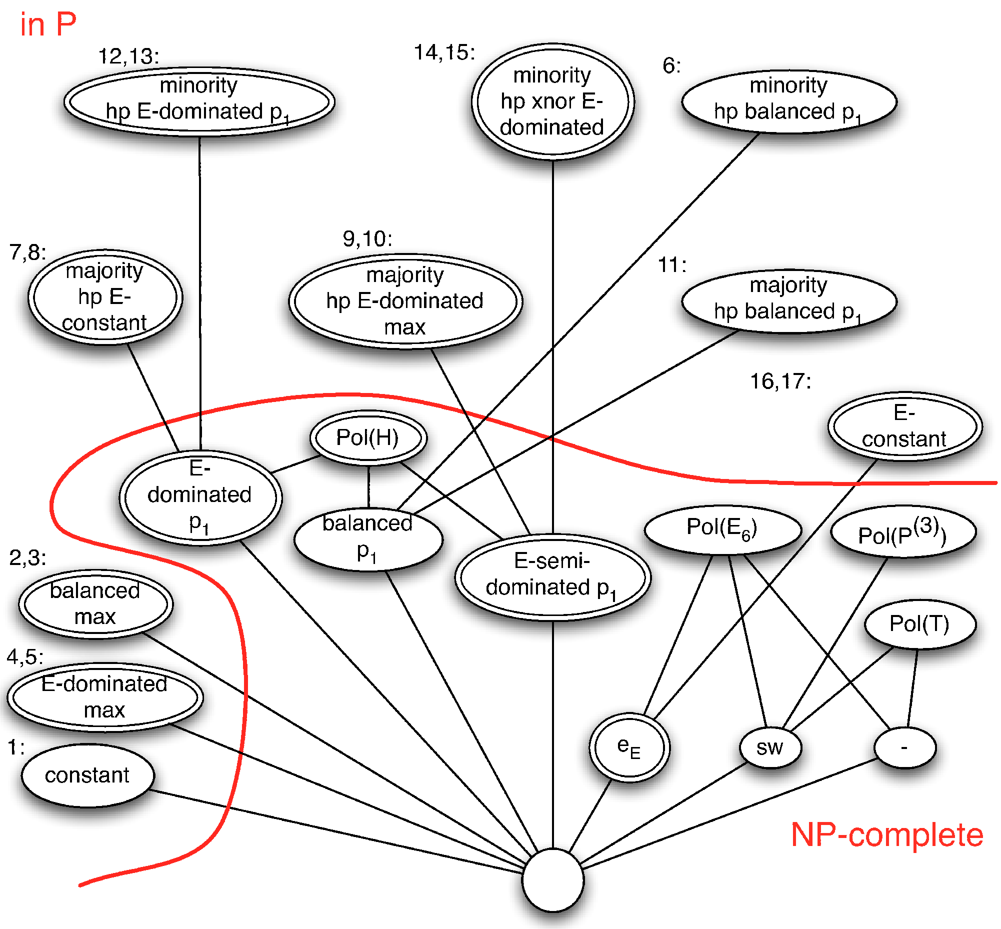

Figure 1: The border: Minimal tractable and maximal hard clones containing $\mathrm{Aut}(G)$**.**