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 Mathmatique
Université Denis-Diderot Paris 7
UFR de Mathématiques
75205 Paris Cedex 13, France
marula@gmx.at

November 8, 2010

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 *first-order 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 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 then 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 grows with the size of 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 first-order logic of graphs instead for propositional 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?

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 16). 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. 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 [3,7,16]) 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 Γ;
- the class of finite substructures of Γ 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 [1, 21].

The random graph G belongs (together with $(\mathbb{Q}; <)$) 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 [13]). In contrast to CSPs of structures definable in $(\mathbb{Q}; <)$, where there is a lot of dependence between the possible values in a solution, the CSPs of reducts of G illuminate different phenomena in constraint satisfaction (e.g., for all tractable classes the inequality relation $\neq$ is 1-independent from the other constraints, in the terminology of [16]), and the polynomial-time tractable cases are characterized by polymorphisms that behave canonically in a Ramsey-theoretic sense.

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). It is also the up to isomorphism unique countable graph that satisfies the *extension property*, which will be useful throughout the paper: For all disjoint finite $U, U' \subseteq V$ there exists $v \in V$ such that v is in G adjacent to all members of U and to none in U'. For the many other remarkable properties of G and its automorphism group $\mathrm{Aut}(G)$, and various connections to many branches of mathematics, see e.g. [13, 14].

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*) $f : D^k \to D$ *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.

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 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. [12]) 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.*

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 $f : \Gamma \to \Delta$ and $g : \Delta \to \Gamma$. Then $\mathrm{CSP}(\Gamma) = \mathrm{CSP}(\Delta)$.*

The following general lemma allows 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.*

In the following two sections, we outline the main ideas of the proof of our theorem; space does not permit us to give any details, and we refer to the appendix which contains the full version of the paper.

4 Endomorphisms

Throughout the text, Γ denotes a reduct of the random graph $G = (V; E)$. The binary relation $N(x, y)$ on V is defined by $\neg E(x, y) \wedge x \neq y$. This section reduces the classification task to the classification of those structures Γ where the relations E, N,and $\neq$ are primitive positive definable; this section is covered by Section E in the full version.

The proof of the main result of this section (Proposition 10) relies on an analysis of the *endomorphism monoids* of reducts Γ of G. It can be shown [10, 26] that Γ has a constant endomorphism, or an endomorphism whose image induces a clique or an independent set in G, or that all endomorphisms of Γ are locally generated by the automorphisms of Γ. In the last case, we can apply a classification of the closed permutation groups (*closure* for groups is defined in an analogous way as closure was defined earlier for clones) that contain the automorphism group of G due to Thomas [25, 26], see also [10]. Combining these two results, we obtain a good understanding of all endomorphism monoids of reducts Γ, which allows us to conclude that if the relation N or the relation E is not primitive positive definable in Γ, then either the relation $P^{(3)}$ or the relation T are primitive positive definable in Γ (Case (c) in Proposition 10), or the classification can be reduced to structures definable in $(V; =)$ (Cases (a) and (b) in Proposition 10).

Definition 8. *For all $k \geq 3$, let $P^{(k)}$ denote the k-ary relation that holds on $x_1, \ldots, x_k \in V$ if $x_1, \ldots, x_k$ are pairwise distinct, and the graph induced by $\{x_1, \ldots, x_k\}$ in G is neither an independent set nor a clique.*

Definition 9. *Let T be the 4-ary relation that holds on $x_1, x_2, x_3, x_4 \in V$ 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 one of the following holds.*

(a) Γ *has a constant endomorphism, and* $\mathrm{CSP}(\Gamma)$ *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* $\mathrm{CSP}(\Gamma)$ *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 $\mathrm{CSP}(\Gamma)$ *is NP-hard.*

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

5 Higher arity polymorphisms

In the following we assume that $\Gamma = (V; E, N, \neq, \dots)$ is a reduct of G that 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 will require the study of higher arity polymorphisms of such reducts (in the full version, this is covered by Section F). 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*, and that one can find regular patterns in any arbitrary function on the random graph. We make this idea more precise.

Definition 11. *Let Δ, Λ be structures, and let $k \geq 1$. The* type $\mathrm{tp}(a)$ *of an n-tuple $a \in \Delta$ is the set of first-order formulas with free variables $x_1, \dots, x_n$ that hold for a in Δ. A k*-ary type condition *between Δ and Λ is a $k+1$-tuple $(t^1, \dots, t^k, s)$, where each t_i is a type of an n-tuple in Δ, and s is a type of an n-tuple in Λ. A k-ary function $f : \Delta^k \to \Lambda$* satisfies *a type condition $(t^1, \dots, t^k, s)$ 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 the type conditions of the behavior B.*

Definition 12. *Let Δ, Λ be structures. An operation $f : \Delta^k \to \Lambda$ is* canonical *if for all k-tuples $(t^1, \dots, t^k)$ of types of n-tuples in Δ there exists a type s of an n-tuple in Λ such that f satisfies the type condition $(t^1, \dots, t^k, s)$. If $F \subseteq \Delta_1$, then we say that f is* canonical on F *if its restriction to F is canonical.*

We remark that 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. The polymorphisms proving tractability of reducts of G will be canonical.

We now define different behaviors that some of these canonical functions will have. For $Q_1, \dots, Q_k \in \{E, N, =, \neq\}$, 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 $1 \leq i \leq k$. 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 13. *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 types of pairs in V^2, but not all such types; however, being of type max and balanced does mean that a function is canonical. The next definition contains some important behaviors of ternary functions.

Definition 14. *An injective ternary function* $f : V^3 \to V$ *is of type*

- majority *if for all* $u, v \in V^3$ *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* $x, y \in V^3$ *we have* $E(f(x), f(y))$ *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 15. *We define a 6-ary 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 Theorem 10 proves Theorem 1.

Theorem 16. *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.*

We now outline the proof that if $\Gamma = (V; E, N, \neq, \ldots)$ is a reduct of G such that there is no primitive positive definition of H in Γ, then one of the other cases of Theorem 16 applies. By Theorem 4, Γ has a polymorphism that violates H. A function $f : V^n \to V$ is called *essentially unary* if it depends on only one of its variables; otherwise, it is called *essential*. Since E and N are among the relations of Γ, and since any essentially unary polymorphism preserving both E and N preserves all relations with a first-order definition in G, we have that the polymorphism violating H must be essential.

Theorem 17 (of [10])**.** *If* $\Gamma = (V; E, N, \neq, \ldots)$ *is a reduct of* G *that has an essential polymorphism, it must also have one of the following binary injective canonical polymorphisms:*

- *an 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).

Theorem 17 and the following proposition together imply that indeed, if Case (a) of Theorem 16 does not apply, then one of the other cases does.

Proposition 18. *Suppose that* f *is an operation on* V *that preserves the relations* E *and* N *and violates the relation* H*. Then* f *generates a binary injective canonical operation of type* min *or* max*, or a ternary injective canonical operation of type minority or majority.*

In the proof of this proposition, we apply Ramsey theory in the form of the following lemma. The crucial idea here is very general, and worth being pointed out. Suppose that f is an n-ary operation that violates a k-ary relation R. Select tuples of constants $c^1, \ldots, c^k \in V^n$ such that f violates R on those tuples. Lemma 19 below then implies that f generates an operation that still violates R, but is canonical as a function from $(G^n, c^1, \ldots, c^k)$ to G. Since there are finitely many canonical behaviours, this allows for combinatorial analysis (see also [9, 11]).

Lemma 19. *Let $c^1, \ldots, c^k \in V^n$, and let B be a behavior for functions from $(G^n, c^1, \ldots, c^k)$ to G. Let $\mathcal{C}$ be a local clone on V. If for every substructure F of $(G^n, c^1, \ldots, c^k)$ there is a function $f \in \mathcal{C}$ which satisfies B on F, then there is also a function $g \in \mathcal{C}$ which satisfies B on $(G^n, c^1, \ldots, c^k)$. Moreover, g can be chosen to be canonical.*

Proof of Proposition 18. Let f be given. Since the relation H consists of three orbits of 6-tuples, by Lemma 7 f generates an at most ternary function that violates H, and hence we can assume wlog that f itself is at most ternary. The operation f can certainly not be essentially unary, since every essentially unary operation that preserves E and N also preserves H. Applying Theorem 17 to the reduct which has all relations preserved by $\{f\}$ and $\mathrm{Aut}(G)$, and by Proposition 5, we get that f generates a binary injective canonical function of type min, max, or p_1. In the first two cases we are done, so consider the last case and denote the function of type p_1 by g.

By adding a dummy variable, we may assume that f is ternary. Now consider $h(x, y, z) := g(g(g(f(x, y, z), x), y), z)$. Then h is clearly injective, and still violates H – the latter can easily be verified combining the facts that f violates H, g is of type p_1, and all tuples in H have pairwise distinct entries. Because h violates H, one of the following is true. There exist $x, y, u, v, p, q \in V^3$ with $\mathrm{ENN}(x, y)$, $\mathrm{NEN}(u, v)$, $\mathrm{NNE}(p, q)$, and $E(h(x), h(y))$, $E(h(u), h(v))$ and $E(h(p), h(q))$, or there exist $x, y, u, v, p, q \in V^3$ with $\mathrm{ENN}(x, y)$, $\mathrm{NEN}(u, v)$, $\mathrm{NNE}(p, q)$, and $N(h(x), h(y))$, $N(h(u), h(v))$ and $N(h(p), h(q))$. We can now apply Ramsey theory in form of Lemma 19 (for details, see the long version of the paper) to show that in both cases the operation h, and hence also f, generates a binary canonical injection of type min or max, or a ternary canonical injection of type majority or minority. $\square$

To conclude, we want to mention an elegant universal-algebraic formulation of our main result, which lines up with recent conjectures and results on finite domain CSPs [12, 23].

Corollary 20. *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.

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. Submitted for journal publication; 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, Michael Pinsker, and Todor Tsankov. Decidability of definability. Preprint, available from the second author's webpage, 2010.

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

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

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

[15] Hubie Chen. A rendezvous of logic, complexity, and algebra. *SIGACT News*, 2006.

[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] Michael Garey and David Johnson. *A guide to NP-completeness*. CSLI Press, Stanford, 1978.

[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*, page 13311403, 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*. To appear.

[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.

A 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 grows with the size of 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 first-order logic of graphs instead for propositional 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?

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

$$\begin{aligned} &(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z)) \\ \vee\ &(\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z)) \\ \vee\ &(\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z)) \ . \end{aligned} \tag{3}$$

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 at the end of this paper by means of our classification in Theorem 61). There are also many interesting tractable Graph-SAT problems, for instance when Ψ consists of the formulas $x \neq y \vee y = z$ and

$$\begin{aligned} &(E(x,y) \wedge \neg E(y,z) \wedge \neg E(x,z)) \\ \vee\ &(\neg E(x,y) \wedge E(y,z) \wedge \neg E(x,z)) \\ \vee\ &(\neg E(x,y) \wedge \neg E(y,z) \wedge E(x,z)) \\ \vee\ &(E(x,y) \wedge E(y,z) \wedge E(x,z))\ . \end{aligned} \tag{4}$$

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.*

Moreover, in Theorem 61 at the end of this paper, we will see exactly which sets Ψ correspond to tractable problems and which to hard ones.

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 finally obtain the classification result.

B 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. 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 [3,7,16]) 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 Γ;
- the class of finite substructures of Γ 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 [1,21].

The random graph G belongs (together with $(\mathbb{Q}; <)$) 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 [13]). In contrast to CSPs of structures definable in $(\mathbb{Q}; <)$, where there is a lot of dependence between the possible values in a solution, the CSPs of reducts of G illuminate different phenomena in constraint satisfaction (e.g., for all tractable classes the inequality relation $\neq$ is 1-independent from the other constraints, in the terminology of [16]), and the polynomial-time tractable cases are characterized by polymorphisms that behave canonically in a Ramsey-theoretic sense.

C 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). It is also the up to isomorphism unique countable graph that satisfies the *extension property*, which will be useful throughout the paper: For all disjoint finite $U, U' \subseteq V$ there exists $v \in V$ such that v is in G adjacent to all members of U and to none in U'. For the many other remarkable properties of G and its automorphism group $\mathrm{Aut}(G)$, and various connections to many branches of mathematics, see e.g. [13, 14].

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

$$\exists x_1, \dots, x_n.\ \psi_1 \wedge \cdots \wedge \psi_m$$

where the ψ_i are *atomic*, i.e., of the form $y_1 = y_2$ or $R(y_1, \dots, y_k)$, 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 $\Psi = \{\psi_1, \dots, \psi_n\}$ 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 $\psi_j(y_1, \dots, y_m)$ for some $1 \leq j \leq m$ and variables y_k from W, by $R_j(y_1, \dots, y_m)$; 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 $\Gamma = (D; R_1, \dots, R_l)$ be a relational structure, and let R be a relation that has a primitive positive definition in Γ. Then $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(D; R, R_1, \dots, R_l)$ 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*) $f : D^k \to D$ *preserves* an m-ary relation $R \subseteq D^m$ iff for all $t_1, \dots, t_k \in R$ 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 in* $\mathrm{Inv}(\mathrm{Pol}(\Gamma))$ *are precisely those that have 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. [12]) 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.

Note that the *automorphisms* of a structure Γ are bijective unary polymorphisms that preserve all relations and their complements; the set of all automorphisms of Γ is denoted by $\mathrm{Aut}(\Gamma)$. It follows from the theorem of Ryll-Nardzewski that for an ω-categorical structure Γ, the local clones containing $\mathrm{Aut}(\Gamma)$ are precisely the polymorphism clones of reducts of Γ. Therefore, in order to determine the computational complexity of the CSP of all reducts Γ of G, it suffices to determine for every local clone $\mathcal{C}$ containing the automorphism group $\mathrm{Aut}(G)$ of G the complexity of $\mathrm{CSP}(\Gamma)$ for some reduct Γ of G with $\mathrm{Pol}(\Gamma) = \mathcal{C}$, if there exists such a reduct with finitely many relations; then the complexity for all reducts with the same polymorphism clone is polynomial time equivalent to $\mathrm{CSP}(\Gamma)$.

The following proposition is the analog to Theorem 4 on the "operational side", and characterizes the local generating process of functions on a domain D by the operators Inv and Pol.

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 $g \in \mathrm{Pol}(\mathrm{Inv}(F))$.*

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 structures of the same signature which are* homomorphically equivalent*, i.e., there exist homomorphisms $f : \Gamma \to \Delta$ and $g : \Delta \to \Gamma$. Then* $\mathrm{CSP}(\Gamma) = \mathrm{CSP}(\Delta)$.

We finish this section with a technical general lemma that we will refer to several times in the paper; it allows 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 let $R \subseteq D^k$ be a relation that 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.*

D Additional conventions

Throughout the text, Γ denotes a reduct of the random graph $G = (V; E)$. 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. That is, in this paper we consider the automorphisms of G be present in all sets of functions when speaking about the local generating process.

The binary relation $N(x, y)$ on V is defined by the formula $\neg E(x, y) \wedge x \neq y$. We use $\neq$ both in logical formulas to denote the negation of equality, and to denote the corresponding binary relation on V.

When t is an n-tuple, we refer to its entries by $t_1, \ldots, t_n$.

E Endomorphisms

The goal of this section is the proof of Proposition 10, which will allow us to reduce the classification task to the classification of those structures where the relations E, N and $\neq$ are primitive positive definable.

Definition 8. *For all $k \geq 3$, let $P^{(k)}$ denote the k-ary relation that holds on $x_1, \dots, x_k \in V$ if $x_1, \dots, x_k$ are pairwise distinct, and the graph induced by $\{x_1, \dots, x_k\}$ in G is neither an independent set nor a clique.*

Definition 9. *Let T be the 4-ary relation that holds on $x_1, x_2, x_3, x_4 \in V$ if x_1, x_2, x_3, x_4 are pairwise distinct, and induce in G either*

- *a single edge and two isolated vertices,*
- *a path with two edges and an isolated vertex,*
- *a path with three edges, or*
- *a complement of one of the structures stated above.*

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

(a) Γ *has a constant endomorphism, and* $\mathrm{CSP}(\Gamma)$ *is 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* $\mathrm{CSP}(\Gamma)$ *has been classified in [5], and is either in P or NP-complete.*

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

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

To prove the proposition, we first cite a result about the reducts of the random graph due to Thomas [26] (also see [10] for a formulation of this result as it is used here).

The graph G contains all countable graphs as induced subgraphs. In particular, it contains an infinite complete subgraph, denoted by K_ω. It is clear that any two injective operations from $V \to V$ whose images induce K_ω in G generate one another. Let e_E be one such operation. Similarly, G contains an infinite independent set, denoted by I_ω. Let e_N be an injective operation from $V \to V$ whose image induces I_ω in G.

Theorem 11. *Let Γ be a reduct of G. Then one of the following cases applies.*

1. Γ *has a constant endomorphism.*
2. Γ *has the endomorphism e_E.*
3. Γ *has the endomorphism e_N.*
4. *The endomorphisms of Γ are locally generated by the automorphisms of Γ.*

The last case splits into five sub-cases, corresponding to the five locally closed permutation groups that contain $\mathrm{Aut}(G)$ exhibited by Thomas [25]. Knowledge about these groups will be important for the complexity classification, and we will next cite the theorem that lists them.

For any finite subset S of V, if we flip edges and non-edges between S and $V \setminus S$ in G, then the resulting graph is isomorphic to G (it is straightforward to verify the extension property). Let i_S be such an isomorphism for each non-empty finite S. Any two such functions generate one another. We also write sw for $i_{\{0\}}$, where $0 \in V$ is any fixed element of V.

Let $R^{(k)}$ be the k-ary relation that holds on $x_1, \ldots, x_k \in V$ if $x_1, \ldots, x_k$ are pairwise distinct, and the number of edges between these k vertices is odd. Note that $R^{(4)}$ is preserved by $-$, $R^{(3)}$ is preserved by sw, and that $R^{(5)}$ is preserved by $-$ and by sw, but not by all permutations of V.

Definition 12. *We say that two structures Γ, Δ on the same domain are* first-order interdefinable *iff all relations of Γ have a first-order definition in Δ and vice-versa.*

Theorem 13 (of [25])**.** *Let Γ be a reduct of G. Then exactly one of the following is true.*

1. *Γ is first-order interdefinable with $(V; E)$;*
   equivalently, $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(G)$.

2. *Γ is first-order interdefinable with $(V; R^{(4)})$;*
   equivalently, Γ is preserved by $-$, but not by sw.

3. *Γ is first-order interdefinable with $(V; R^{(3)})$;*
   equivalently, Γ is preserved by sw, but not by $-$.

4. *Γ is first-order interdefinable with $(V; R^{(5)})$;*
   equivalently, Γ is preserved by $-$ and by sw, but not by all permutations of V.

5. *Γ is first-order interdefinable with $(V; =)$;*
   equivalently, Γ is preserved by all permutations of V.

In particular, the reducts $(V; P^{(3)})$ and $(V; T)$ are both first-order interdefinable with one of the five structures of this theorem; we will now show with which one, and prove hardness for both reducts. We start with $(V; P^{(3)})$.

Proposition 14. *The structure $(V; P^{(3)})$ is first-order interdefinable with $(V; R^{(4)})$.*

Proof. The relation $P^{(3)}$ is not preserved by sw: if $x_1, x_2 \in V$ are so that $E(x_1, x_2)$ and $N(0, x_i)$ hold for $i = 1, 2$, then $(0, x_1, x_2) \in P^{(3)}$, but $\{sw(0), sw(x_1), sw(x_2)\}$ is a clique in G, so $(sw(0), sw(x_1), sw(x_2)) \notin P^{(3)}$. On the other hand, $P^{(3)}$ is clearly preserved by $-$. Hence, Theorem 13 implies that $(V; P^{(3)})$ is first-order interdefinable with $(V; R^{(4)})$. $\square$

Proposition 15. $\mathrm{CSP}((V;P^{(3)}))$ *is NP-complete.*

Proof. For $k \geq 3$, let $Q^{(k)}$ be the k-ary relation that holds for a tuple $(x_1,\dots,x_k) \in V^k$ iff $x_1,\dots,x_k$ are pairwise distinct, and $(x_1,\dots,x_k) \notin P^{(k)}$.

We show that the relation $Q^{(3)}$ is primitive positive definable by the relation $P^{(3)}$. Observe first that $\neq$ is primitive positive definable from $P^{(3)}$ by the formula $\exists u.\, P^{(3)}(x,y,u)$. Let μ be the primitive positive formula that states about the tuple $(x_1,\dots,x_6)$ that all its entries are distinct. Let ϕ be a conjunction of atomic formulas with variables from $x_1,\dots,x_6$ that contains for each three-element subset $\{u,v,w\}$ of those variables a conjunct $P^{(3)}(u,v,w)$. It is known that every two-coloring of the edges of the graph K_6 (the clique with six vertices) contains a monochromatic triangle. Therefore, $\phi \wedge \mu$ is unsatisfiable. Let ϕ' be a conjunction over a maximal subset of the conjuncts of ϕ with the property that $\phi' \wedge \mu$ is still satisfiable, and suppose without loss of generality that the conjunct $P^{(3)}(x_1,x_2,x_3)$ of ϕ is missing in ϕ'. We claim that $\psi := \exists x_4,x_5,x_6.\phi' \wedge \mu$ defines $Q^{(3)}$. To see this, suppose that t is a triple that satisfies ψ; that means that there exists a 6-tuple s extending t which satisfies $\phi' \wedge \mu$. From the maximality of ϕ', we infer that s, and hence also t, does not satisfy $P^{(3)}(x_1,x_2,x_3)$, so $t \in Q^{(3)}$. Therefore, ψ defines a subset of $Q^{(3)}$. Observe also that ψ does not define the empty set since $\phi' \wedge \mu$, and hence also ψ, is satisfiable. Let t be any tuple satisfying ψ. Since $P^{(3)}$ is preserved by $-$, one can check that $Q^{(3)}$ consists of only one orbit in $(V;P^{(3)})$. Hence, for any tuple w in $Q^{(3)}$ there exists an automorphism of $(V;P^{(3)})$ that sends t to w. This automorphism clearly preserves ψ, and hence w satisfies ψ, proving the claim.

Next observe that $Q^{(3)}(x,y,u) \wedge Q^{(3)}(y,u,v) \wedge Q^{(3)}(x,y,v)$ defines $Q^{(4)}$, and hence $Q^{(4)}$ has a primitive positive definition from $P^{(3)}$.

We now prove NP-hardness of $\mathrm{CSP}((V;P^{(3)}))$ by reduction from positive not-all-equal 3SAT, which is the variant of 3SAT where all clauses have only positive literals, and the task is to find an assignment of Boolean values to the variables such that in no clause three variables are set to the same value; this problem is hard by Schaefer's theorem. For each Boolean variable x in a given instance Φ to that problem we create two variables u_x and v_x in an instance Ψ to $\mathrm{CSP}((V;P^{(3)},Q^{(4)}))$ (which is sufficient for proving hardness of $\mathrm{CSP}((V;P^{(3)}))$ by Lemma 3). Moreover, we create variables $w_{C,x,y}, w_{C,y,z}, w_{C,z,x}$ for every clause C of Φ with variables x,y,z. The conjuncts of Ψ are as follows: For each clause C of Φ with variables x,y,z we have in Ψ

- the conjunct $P^{(3)}(w_{C,x,y}, w_{C,y,z}, w_{C,z,x})$;
- the three conjuncts $Q^{(4)}(u_x,v_x,w_{C,z,x},w_{C,x,y})$,
  $Q^{(4)}(u_y,v_y,w_{C,x,y},w_{C,y,z})$, and $Q^{(4)}(u_z,v_z,w_{C,y,z},w_{C,z,x})$.

Now suppose that Φ is a satisfiable instance of the positive not-all-3-equal SAT problem, and let s be a solution, i.e., a satisfying Boolean assignment to the variables of Φ. Then define a graph on the variables of Ψ as follows: There is an edge between two variables

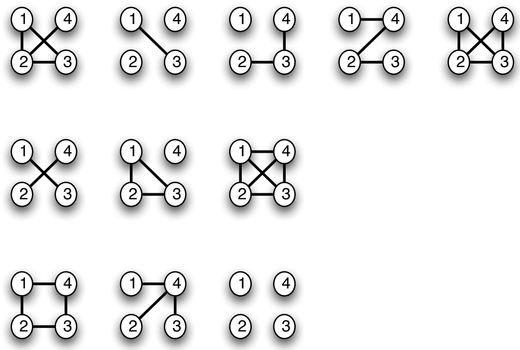

Figure 1: Illustration for the relation T.

$w_{C,i,j}$ and $w_{C,j,k}$ iff the value of j under s is 1. Moreover, there are edges between u_i and all other variables iff the value of i under s is 1, and we impose the same condition for edges with v_i. The resulting graph can be assumed to be a subgraph of G, and it is straightforward to check that it shows that Ψ holds in $(V; P^{(3)}, Q^{(4)})$. The converse that every solution to Ψ yields a solution to Φ is left to the reader. $\square$

We now turn to the reduct $(V;T)$. For an illustration of the relation T, see Figure 1. The first row shows all graphs, up to isomorphism, that are induced by 4-tuples from the relation T (note that T is totally symmetric, i.e., whenever we have a tuple in T and we permute its components, we obtain again a tuple in T). The other two rows show the graphs induced by 4-tuples of distinct elements of V that are not from T, again up to isomorphism. We denote the four vertices of those graphs by 1,2,3 and 4, as shown in the picture.

Proposition 16. *The structure $(V;T)$ is first-order interdefinable with $(V;R^{(5)})$.*

Proof. The relation T is obviously not preserved by all permutations of V, but preserved by $-$. It can be checked that T is preserved by sw. Theorem 13 implies that it is first-order interdefinable with $(V;R^{(5)})$. $\square$

Lemma 17. *Let Γ be a reduct of G with $sw \in \mathrm{Aut}(\Gamma)$. Then T consists of one orbit of 4-tuples with respect to Γ.*

Proof. For a graph on $\{1,2,3,4\}$ and a vertex $v \in \{1,2,3,4\}$, we say in the following that we *flip the graph at v* when we produce a new graph by flipping the edges and non-edges that are adjacent to v. Now consider the graphs of the first row of Figure 1, which are those corresponding to tuples in T. If we flip the first one at 2, we obtain the second one; flipping the second graph at 3 yields the third one; and flipping the third graph at 4 yields the fourth one. Finally, flipping the first graph at 4 yields a graph isomorphic to the fifth graph. Since this flipping operation corresponds to applications of *sw* (more precisely: switches with respect to an appropriately chosen point in V) to tuples in T, and isomorphism of these 4-vertex graphs to applications of appropriately chosen automorphisms of G, we get that indeed any tuple in T can be transformed into any other tuple in T by applications of *sw* and automorphisms of G. □

We will need the following auxiliary lemma before proving hardness for $\mathrm{CSP}((V;T))$.

Lemma 18. *Let T' be the relation consisting of all 4-tuples with distinct entries in V that do not belong to T. Then T' is primitive positive definable in $(V;T)$.*

Proof. Observe first that $\neq$ is primitive positive definable from T by the formula $\exists u, v.\, T(x, y, u, v)$. Let μ be the primitive positive formula that states about the tuple $(x_1, \ldots, x_{18})$ that all its entries are distinct.

It is a well-known fact that every two-coloring of the edges of the graph K_{18} contains a monochromatic clique of size 4. Let ϕ be the conjunction over all atomic formulas of the form $T(a, b, c, d)$ where $\{a, b, c, d\}$ is a 4-element subset of $\{x_1, \ldots, x_{18}\}$. By the above fact, $\phi \wedge \mu$ is unsatisfiable.

Let ϕ_0 denote the conjunction over a maximal subset of the conjuncts of ϕ with the property that $\phi_0 \wedge \mu$ is satisfiable, and let $T(y_1, \ldots, y_4)$ denote a conjunct from ϕ that is not a conjunct of ϕ_0.

We claim that the relation defined by the formula ψ obtained from $\phi_0 \wedge \mu$ by existentially quantifying all variables except for $y_1, \ldots, y_4$ equals T'. To see this, suppose that t is a 4-tuple that satisfies ψ; that means that there exists a 18-tuple s extending t which satisfies $\phi_0 \wedge \mu$. From the maximality of ϕ_0, we infer that s, and hence also t, does not satisfy $T(y_1, \ldots, y_4)$, so $t \in T'$. Therefore, ψ defines a subset of T'. Observe also that ψ does not define the empty set since $\phi_0 \wedge \mu$, and hence also ψ, is satisfiable. Let t be any tuple satisfying ψ. As in Lemma 17, one can check that T' consists of only one orbit in $(V;T)$. Hence, for any tuple w in T' there exists an automorphism of T that sends t to w. This automorphism clearly preserves ψ, and hence w satisfies ψ, proving the claim. □

Proposition 19. $\mathrm{CSP}((V;T))$ *is NP-complete.*

Proof. Our hardness proof is by reduction from an NP-hard Boolean constraint satisfaction problem. In our reduction, the following ('link'-) relation $L \subseteq V^6$ is of central importance. For a finite subset S of V, write $\#S$ for the parity of edges between members of S. Now define

$$L := \big\{x \in V^6 \mid \text{the entries of } x \text{ are pairwise distinct, and} \\ \#\{x_1, x_2, x_3\} = \#\{x_4, x_5, x_6\}\big\} .$$

Let $\phi(x, y, z, u, v, w)$ be the conjunction over all atomic formulas of the form $T'(a, b, c, d)$ for every four-element subset $\{a, b, c, d\}$ of $\{x, y, z, u, v, w\}$. We claim that

$$\psi := \exists y_1, y_2, y_3 . \phi(x_1, x_2, x_3, y_1, y_2, y_3) \wedge \phi(y_1, y_2, y_3, x_4, x_5, x_6)$$

defines $L(x_1, \ldots, x_6)$.

Observe first that L is preserved by sw and $-$. Moreover, since T' is preserved by sw and $-$, so is ψ. We can therefore take the liberty of applying sw and $-$ to tuples when showing the equivalence of ψ and L.

Let $t \in V^6$ be a tuple with pairwise distinct entries. We first show that when $t \in L$ then it also satisfies ψ. Applying sw and $-$ to t repeatedly, we can assume wlog. that $N(t_1, t_2) \wedge N(t_2, t_3) \wedge N(t_1, t_3)$ and $N(t_4, t_5) \wedge N(t_5, t_6) \wedge N(t_4, t_6)$. Choose values $s_1, s_2, s_3 \in V$ for the variables y_1, y_2, y_3 such that between any two distinct vertices x, y from $\{t_1, \ldots, t_6, s_1, s_2, s_3\}$ we have $N(x, y)$. This satisfies all conjuncts in ψ.

For the converse, we suppose that t satisfies ψ. Let $s_1, s_2, s_3 \in V$ be witnesses for y_1, y_2, y_3 that show that ψ holds for t. Assume for contradiction that $t \notin L$, i.e., $\#\{x_1, x_2, x_3\} \neq \#\{x_4, x_5, x_6\}$. Applying sw and $-$, we can assume wlog. that $N(t_1, t_2) \wedge N(t_2, t_3) \wedge N(t_1, t_3)$ and $E(t_4, t_5) \wedge E(t_5, t_6) \wedge E(t_4, t_6)$. Moreover, by applying sw and $-$ again, we can assume that $\{s_1, s_2, s_3\}$ induces a clique (or an independent set, which would be the same for our further argument). Because of $T'(t_1, t_2, t_3, s_1)$, either the relation E holds between all pairs $\{t_1, s_1\}$, $\{t_2, s_1\}$, and $\{t_3, s_1\}$, or the relation N holds between all these pairs. Similarly, we can observe the same for s_2 and s_3, so one of the two possibilities occurs twice. So assume wlog. that E holds for any pair p, q with $p \in \{t_1, t_2, t_3\}$ and $q \in \{s_1, s_2\}$. But then $T'(t_1, t_2, s_1, s_2)$ does not hold, a contradiction.

We can now prove hardness by reduction from

$$\mathrm{CSP}((\{0,1\}; \{t \in \{0,1\}^4 \mid t_1 + t_2 + t_3 + t_4 = 2\})) ,$$

which is NP-hard by Schaefer's theorem [22]. From a given instance Φ of this Boolean CSP with variables $x_1, \ldots, x_n$, we create an instance Ψ of $\mathrm{CSP}((V; T))$ as follows. We first describe the variables of Ψ. There are three variables y^1_x, y^2_x, y^3_x for each $x \in \{x_1, \ldots, x_n\}$. Moreover, Ψ has four variables $z^C_{a,b,c}, z^C_{a,b,d}, z^C_{a,c,d}, z^C_{b,c,d}$ for each constraint C of the form $a + b + c + d = 2$ in Φ. The constraints of Ψ are as follows. For each constraint C of the form $a + b + c + d = 2$ in Φ, we add the constraints

$$\begin{aligned} T(z^C_{a,b,c}, z^C_{a,b,d}, y_{a,c,d}, y_{b,c,d}) \wedge & L(z^C_{a,b,c}, z^C_{a,b,d}, z^C_{a,c,d}, y^1_a, y^2_a, y^3_a) \\ & \wedge L(z^C_{a,b,c}, z^C_{a,b,d}, z^C_{b,c,d}, y^1_b, y^2_b, y^3_b) \\ & \wedge L(z^C_{a,b,c}, z^C_{a,c,d}, z^C_{b,c,d}, y^1_c, y^2_c, y^3_c) \\ & \wedge L(z^C_{a,b,d}, z^C_{a,c,d}, z^C_{b,c,d}, y^1_d, y^2_d, y^3_d) . \end{aligned}$$

Clearly, Ψ can be computed in polynomial time from Φ. We first verify that if Φ has a solution $s : \{x_1, \ldots, x_n\} \to \{0, 1\}$, then Ψ is satisfiable as well. It clearly suffices to specify a graph on the variables of Ψ such that the identity satisfies the constraints of Ψ. For $x \in \{x_1, \dots, x_n\}$, we let $\{y_x^1, y_x^2, y_x^3\}$ form an independent set if $s(x) = 1$, and a clique otherwise. Since s satisfies all constraints $a+b+c+d = 2$ of Φ, s maps exactly two of the variables a, b, c, d to 1; for the sake of notation, suppose that $s(a) = s(b) = 0$ (the other cases are handled analogously). Then the graph on the variables of Ψ has an edge between $z^C_{a,b,c}$ and $z^C_{a,b,d}$. All other pairs of variables of Ψ are not adjacent. It is straightforward to verify that indeed all constraints of Ψ are satisfied.

Then central observation for proving the converse is that for any tuple $t \subseteq V^4$ that satisfies $T(x_1, x_2, x_3, x_4)$, there are exactly two 3-element subsets $\{i, j, k\}$ of $\{1, \dots, 4\}$ such that $R^{(3)}(t_i, t_j, t_k)$ holds. Suppose that there is a solution to Ψ. We define a mapping $s : \{x_1, \dots, x_n\} \to \{0, 1\}$ as follows. For $x \in \{x_1, \dots, x_n\}$, if in the solution to Ψ we have that $R^{(3)}(y_x^1, y_x^2, y_x^3)$ is true, then we set $s(x) = 1$, and otherwise we set $s(x) = 0$. It follows from the observation we just made that s satisfies all constraints of Φ. $\square$

We are now ready to prove Proposition 10, showing that we can in the rest of our classification project focus on those languages that contain the relations N, E, and $\neq$.

Proof of Proposition 10. Suppose E does not have a primitive positive definition in Γ. We have that E consists of just one orbit of pairs in G, and thus, since $\mathrm{Aut}(\Gamma) \supseteq \mathrm{Aut}(G)$, also in Γ. Hence, Lemma 7 shows the existence of an endomorphism e of Γ that violates E.

Theorem 11 states that either all endomorphisms of Γ are generated by its automorphisms, or Γ has a constant endomorphism, or the endomorphism e_E, or the endomorphism e_N. If Γ has a constant endomorphism we are in Case (a) and done. If Γ has the endomorphisms e_E or e_N, then we are in Case (b) since $e_E[V]$ and $e_N[V]$ induce structures in G which are invariant under all permutations of their domain. So assume in the following that Γ has neither e_E, nor e_N, nor a constant as an endomorphism, and that all endomorphisms of Γ are generated by $\mathrm{Aut}(\Gamma)$.

In particular, e is generated by $\mathrm{Aut}(\Gamma)$. This implies that $\mathrm{Aut}(\Gamma)$ does not equal $\mathrm{Aut}(G)$, and hence it contains either $-$ or sw, by Theorem 13.

Suppose that $\mathrm{Aut}(\Gamma)$ contains $-$. If the relation $P^{(3)}$ is primitive positive definable in Γ, then $\mathrm{CSP}(\Gamma)$ is NP-complete by Proposition 15 and Lemma 3, and we are in Case (c). Otherwise, since the relation $P^{(3)}$ consists of only one orbit of triples in Γ and by Lemma 7, there exists an endomorphism e_1 of Γ that violates $P^{(3)}$. Since e_1 is generated by $\mathrm{Aut}(\Gamma)$, there is also an automorphism of Γ that violates $P^{(3)}$, and hence also $R^{(4)}$ by Proposition 14. Thus, Theorem 13 shows that $\mathrm{Aut}(\Gamma)$ contains sw.

We can thus henceforth assume that $\mathrm{Aut}(\Gamma)$ contains sw. Then the relation T consists of just one orbit of 4-tuples with respect to Γ, by Lemma 17. If the relation T is primitive positive definable, then we are in Case (c) and $\mathrm{CSP}(\Gamma)$ is NP-hard by Lemma 3 and Proposition 19. Otherwise, since T consists of one orbit in Γ, Lemma 7 implies that there is an endomorphism e_2 of Γ that violates T. Since e_2 is generated by $\mathrm{Aut}(\Gamma)$, there is an automorphism of Γ violating T, and thus by Proposition 16 there is an automorphism of

Γ violating $R^{(5)}$. It then follows from Theorem 13 that $\mathrm{Aut}(\Gamma)$ contains all permutations, bringing us back into Case (b).

The reasoning for the case when N has no primitive positive definition in Γ is dual. So we may assume that both E and N are primitive positive definable. Then so is $\neq$ since $x \neq y$ iff $\exists z.E(x,z) \wedge N(y,z)$. $\square$

F Higher arity polymorphisms

In the following we assume that $\Gamma = (V; E, N, \neq, \dots)$ is a reduct of G that contains the relations E, N and $\neq$. While in the last section, we only dealt with endomorphisms and automorphisms of reducts of G, the remaining cases will 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*, and that one can find regular patterns in any arbitrary function on the random graph. We make this idea more precise.

Definition 20. *Let Δ be a structure. The* type $\mathrm{tp}(a)$ *of an n-tuple $a \in \Delta$ is the set of first-order formulas with free variables $x_1, \dots, x_n$ that hold for a in Δ.*

Definition 21. *Let Δ, Λ be structures, and let $k \geq 1$. A k-ary* type condition *between Δ and Λ is a $k+1$-tuple $(t^1, \dots, t^k, s)$, where each t_i is a type of an n-tuple in Δ, and s is a type of an n-tuple in Λ. A k-ary function $f : \Delta^k \to \Lambda$* satisfies *a type condition $(t^1, \dots, t^k, s)$ 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 the type conditions of the behavior B.*

Definition 22. *Let Δ, Λ be structures. An operation $f : \Delta^k \to \Lambda$ is* canonical *if for all k-tuples $(t^1, \dots, t^k)$ of types of n-tuples in Δ there exists a type s of an n-tuple in Λ such that f satisfies the type condition $(t^1, \dots, t^k, s)$. If $F \subseteq \Delta_1$, then we say that f is* canonical on *F if its restriction to F is canonical.*

We remark that 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. The polymorphisms proving tractability of reducts of G will be canonical. We now define different behaviors that some of these canonical functions will have. For $Q_1, \dots, Q_k \in \{E, N, =, \neq\}$, 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 $1 \leq i \leq k$.

We start by behaviors of binary functions.

Definition 23. *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 for all* $u, v \in V^2$ *with* $\neq{=}(u,v)$ *we have that* $E(f(u), f(v))$ $(N(f(u), f(v)))$*;*

- 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* $\mathrm{EE}(u,v)$*;*

- of type max *if for all* $u, v \in V^2$ *with* $\neq\neq(u,v)$ *we have* $N(f(u), f(v))$ *if and only if* $\mathrm{NN}(u,v)$*;*

- 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* $E(u_1, v_1)$*;*

- 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 types of pairs in V^2, but not all such types; however, being of type max and balanced does mean that a function is canonical. The next definition contains some important behaviors of ternary functions.

Definition 24. *An injective ternary function* $f : V^3 \to V$ *is of type*

- majority *if for all* $u, v \in V^3$ *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* $x, y \in V^3$ *we have* $E(f(x), f(y))$ *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 will be shown by means of a number of different canonical functions, all hardness cases will be established by the following single relation.

Definition 25. *We define a 6-ary relation $H(x_1, y_1, x_2, y_2, x_3, y_3)$ on V by*

$$\begin{aligned}
&\bigwedge_{i,j\in\{1,2,3\}, i\neq j, u\in\{x_i,y_i\}, v\in\{x_j,y_j\}} N(u,v) \\
&\wedge \big(((E(x_1,y_1) \wedge N(x_2,y_2) \wedge N(x_3,y_3)) \\
&\quad \vee \ (N(x_1,y_1) \wedge E(x_2,y_2) \wedge N(x_3,y_3)) && (5) \\
& && (6) \\
&\quad \vee \ (N(x_1,y_1) \wedge N(x_2,y_2) \wedge E(x_3,y_3))\big) \ .
\end{aligned}$$

Our goal for this section is to prove the following proposition, which together with Proposition 10 proves Theorem 1.

Proposition 26. *Let $\Gamma = (V; E, N, \neq, \dots)$ be a reduct of G. Then at least one of the following holds:*

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

(b) Γ has a canonical polymorphism of type minority, as well as a canonical binary injection which 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 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 remainder of this section contains the proof of Proposition 26, and is organized as follows: In Subsection F.1, we show that the relation H is hard. We then prove in Subsection F.2 that if H does not have a primitive positive definition in a reduct Γ as in Proposition 26, then Γ has one of the polymorphisms listed in Cases (b) to (f) of the proposition. Tractability of Cases (b) and (c) is shown in Subsection F.3, tractability of Case (d) in Subsection F.4, of Case (e) in Subsection F.5, and finally tractability of Case (f) in Subsection F.6.

F.1 Hardness of H

This subsection is devoted to Case (a) of Proposition 26.

Proposition 27. CSP$((V;H))$ *is NP-hard.*

Proof. The proof is reduction from positive 1-in-3-3SAT (one of the hard problems in Schaefer's classification; also see [18]). Let Φ be an instance of positive 1-in-3-3SAT, that is, a set of clauses, each having three positive literals. We create from Φ an instance Ψ of CSP$((V;H))$ as follows. For each variable x in Φ we have a pair u_x, v_x of variables in Ψ. When $\{x,y,z\}$ is a clause in Φ, then we add the conjunct $H(u_x,v_x,u_y,v_y,u_z,v_z)$ to Ψ. Finally, we existentially quantify all variables of the conjunction Ψ in order to obtain a sentence. Clearly, Ψ can be computed from Φ in linear time.

Suppose now that Φ is satisfiable, i.e., there exists a mapping α from the variables of Φ to $\{0,1\}$ such that in each clause exactly one of the literals is set to 1; we claim that $(V;H)$ satisfies Ψ. To show this, let F be the graph whose vertices are the variables of Ψ, and which has an edge between u_x and v_x if x is set to 1 under the mapping α. By universality of G we may assume that F is a subgraph of G. It is then enough to show that F satisfies the conjunction of Ψ in order to show that $(V;H)$ satisfies Ψ. Indeed, when $H(u_x,v_x,u_y,v_y,u_z,v_z)$ is a clause from Ψ then the conjunction in the first line of the definition of H is clearly satisfied; moreover, from the disjunction in the remaining lines of the definition of H exactly one disjunct will be true, since in the corresponding clause $\{x,y,z\}$ of Φ exactly one of $\alpha(x),\alpha(y),\alpha(z)$ has been set to 1. This argument can easily be inverted to see that every solution to Ψ can be used to define a solution to Φ. $\square$

F.2 Producing canonical functions

We now show that if $\Gamma = (V;E,N,\neq,\ldots)$ is a reduct of G such that there is no primitive positive definition of H in Γ, then one of the other cases of Proposition 26 applies. By Theorem 4, Γ has a polymorphism that violates H.

Definition 28. *A function $f : V^n \to V$ is called* essentially unary *if it depends on only one of its variables; otherwise, it is called* essential*.*

Since E and N are among the relations of Γ, and since any essentially unary polymorphism preserving both E and N preserves all relations with a first-order definition in G, we have that the polymorphism violating H must be essential. Thus the following theorem from [10] applies. Before stating it, it is convenient to define the dual of an operation f on G, which can be imagined as the function obtained from f by exchanging the roles of E and N.

Definition 29. *The* dual *of a function $f(x_1,\ldots,x_n)$ on G is the function $-f(-x_1,\ldots,-x_n)$.*

Theorem 30 (of [10])**.** *If $\Gamma = (V; E, N, \neq, \ldots)$ is a reduct of G that has an essential polymorphism, it must also have at least one of the following binary injective canonical polymorphisms.*

- *an 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).

Theorem 30 and the following proposition together imply that indeed, if Case (a) of Proposition 26 does not apply, then one of the other cases does.

Proposition 31. *Suppose that f is an operation on V that preserves the relations E and N and violates the relation H. Then f generates a binary injective canonical operation of type* min *or* max*, or a ternary injective canonical operation of type minority or majority.*

The remainder of this subsection will be devoted to the proof of Proposition 31. We will need the Ramsey-theoretic machinery which we developed in [10], and start by recalling the following definition from that paper.

Definition 32. *Let τ be any signature and let $\mathcal{C}$ be a class of finite τ-structures closed under substructures and with the property that for any two structures in $\mathcal{C}$ there exists a structure in $\mathcal{C}$ containing both structures. We order $\mathcal{C}$ by the embedding relation $\subseteq$. Let $P(w)$ be any property. We say that P holds*

- for arbitrarily large elements of $\mathcal{C}$ *if for any $F \in \mathcal{C}$ there exists $H \in \mathcal{C}$ such that $F \subseteq H$ and $P(H)$ holds;*
- for all sufficiently large elements of $\mathcal{C}$ *iff there is an element F of $\mathcal{C}$ such that P holds for H whenever F embeds into H.*

Our properties P will in fact always be of the form "$P(H)$ holds if function f has behavior B on H", for a fixed function f and a behavior B. This implies that if $P(H)$ holds, then P also holds for all substructures of H. Definition 32 then says that P holds for arbitrarily large elements of $\mathcal{C}$ iff for any $F \in \mathcal{C}$ there is $F' \in \mathcal{C}$ isomorphic to F such that $P(F')$ holds. The following proposition is the combinatorial core of [10], and will also serve our purposes in this subsection. For a structure Δ and constants $c^1, \ldots, c^k \in \Delta$, we write $(\Delta, c^1, \ldots, c^k)$ for the structure that arises when one adds the constants $c^1, \ldots, c^k$ to the language of Δ.

Proposition 33. *Let $c^1, \ldots, c^k \in V^n$, and let B be a behavior for functions from $(G^n, c^1, \ldots, c^k)$ to G. Let $\mathcal{D}$ be a local clone on V. If for arbitrarily large finite substructures F of $(G^n, c^1, \ldots, c^k)$ there is a function $f \in \mathcal{D}$ which satisfies B on F, then there is also a function $g \in \mathcal{D}$ which satisfies B on $(G^n, c^1, \ldots, c^k)$. Moreover, g can be chosen to be canonical.*

In the sequel, we will also say that a function $f : V^n \to V$ has behavior B *between two points* $x, y \in V^n$ if it has behavior B on the structure $\{x, y\}$.

Lemma 34. *Let $a \in V^2$, and let $h_a : V^2 \to V$ be a binary injection that behaves like p_1 between a and all other points of V^2, and which behaves like p_2 between any two points in $V^2 \setminus \{a\}$. Then h_a generates a binary injection of type* min*.*

Proof. To prove the lemma, we show that for any finite set $F \subseteq V^2$, h_a generates a binary injection that behaves like min on F. The claim then follows from Proposition 33.

We may assume that for all $u, v \in F$ we have $u_1 \neq v_1$ and $u_2 \neq v_2$. This is because if we can generate the desired functions for all F with this property, then given an arbitrary F without this property, we may consider the set $F' := (h_a(x, y), h_a(y, x))[F]$, and find a function t for this set; then the function $t(h_a(x, y), h_a(y, x))$ behaves like min on the original set F.

So let F be given. We use induction over the size of F. The beginning $|F| = 1$ is trivial. So assume that the assertion holds for all F with $|F| = n$, let any F with $|F| = n + 1$ be given, and write $F = B \cup \{b\}$, where $|B| = n$.

We first claim that for all finite $C \subseteq V^2$ we have that h_a generates a binary injection h_C which behaves like p_1 between $u, v \in V^2$ whenever $u \in C$ or $v \in C$, and which behaves like p_2 otherwise. We use induction over the size of C to prove the claim. When C has just one element c, then we can take any automorphisms $\alpha, \beta \in \mathrm{Aut}(G)$ such that $(\alpha(c_1), \beta(c_2)) = a$ and then set $h_C(x, y) := h_a(\alpha(x), \beta(y))$. In the induction step, write $C = D \cup \{d\}$, and let h_D be provided by the induction hypothesis. By using an automorphism of G, we may assume that $h_D(d) = d_2$. Now set $h_C(x, y) := h_{\{d\}}(x, h_D(x, y))$. To check that h_C satisfies the desired condition, one needs to distinguish the cases (1) $u = d$ and $v \in D$, (2) $u = d$ and $v \in V^2 \setminus C$, (3) $u \in D$ and $v \in D$, (4) $u \in D$ and $v \in V^2 \setminus C$, and finally (5) $u \in V^2 \setminus C$ and $v \in V^2 \setminus C$. We leave the details to the reader.

By induction hypothesis, there exists a function t generated by h_a which behaves like min on B. Write $B = B_1 \cup B_2$, where B_1 contains exactly those elements of B which are not connected to b in the first coordinate. Let $s := h_{B_1}$. Then set $d := (s(b), t(b)) \in V^2$, and let $f := h_{\{d\}}$. We claim that $f(s, t)$ behaves like min on F. To see this, let first $u, v \in B$ be given. Then $(s(u), t(u))$ is adjacent to $(s(v), t(v))$ in the second coordinate iff u and v are adjacent in both coordinates, since t behaves like min on B. Since $(s(u), t(u)) \neq d$ and $(s(v), t(v)) \neq d$, we have that f behaves like p_2 between $(s(u), t(u))$ and $(s(v), t(v))$, and thus $f(s, t)(u)$ and $f(s, t)(v)$ are adjacent iff u and v were adjacent in both coordinates, so $f(s, t)$ behaves like min between u and v. It remains to show that $f(s, t)$ also behaves like min between any $u \in B$ and b. Distinguishing the cases $u \in B_1$ and $u \in B_2$, one can verify that $(s(u), t(u))$ and $d = (s(b), t(b))$ are adjacent in the first coordinate iff u and b are adjacent in both coordinates. Since f behaves like p_1 between $(s(u), t(u))$ and d, we get that $f(s,t)(u)$ and $f(s,t)(b)$ are adjacent iff u and b are adjacent in both coordinates, proving that $f(s,t)$ behaves like min between u and b. □

Lemma 35. *Let $a \in V^2$, and let $z_a : V^2 \to V$ be a binary injection that behaves like* min *between a and all other points of V^2, and which behaves like p_1 between any two points of $V^2 \setminus \{a\}$. Then z_a generates a binary injection of type* min.

Proof. This follows from the proof of the preceding lemma, since z_a deletes more edges than h_a. □

Definition 36. *Let Δ be a structure. An* orbit *in Δ is a maximal set of elements of Δ of the same type. An orbit in Δ is called* proper *if it contains more than one element.*

Proposition 37. *Let $f : V^2 \to V$ be a binary injection preserving E and N that is neither of type p_1 nor of type p_2. Then f generates a binary injection of type* min *or of type* max.

Proof. Let f be given.

Case 1. Suppose there exists $c \in V^2$ such that for arbitrarily large finite substructures F of (G^2, c) we have that f behaves like min between c and the other points in F, and like one and the same projection between all other points in F. Then we are done by Proposition 33 and Lemma 35.

Case 2. Suppose there exists $c \in V^2$ such that for arbitrarily large finite substructures F of (G^2, c) we have that f behaves like p_1 between c and the other points in F, and like p_2 between all other points in F. Then we are done by Proposition 33 and Lemma 34. Same if the roles of p_1 and p_2 are interchanged.

Case 3. Suppose there exists $c \in V^2$ such that for arbitrarily large finite substructures F of (G^2, c) we have that f behaves like min on a proper orbit. Then we are obviously done by Proposition 33.

Case 4. Suppose there exists $c \in V^2$ such that for arbitrarily large finite substructures F of (G^2, c) there are proper orbits T_1, T_2 in F such that f behaves like min between any two points $a^1 \in T_1$ and $a^2 \in T_2$, and such that f behaves like one and the same projection on T_1. Then Proposition 33 tells us that f generates a function g which has this property for proper orbits T_1, T_2 in (G^2, c). This function g satisfies the condition of Case 1 (pick any constant in T_2).

Case 5. Suppose there exists $c \in V^2$ such that for arbitrarily large finite substructures F of (G^2, c) there are proper orbits T_1, T_2 in F such that f behaves like p_2 between any two points $a^1 \in T_1$ and $a^2 \in T_2$, and such that f behaves like p_1 on T_1. Then Proposition 33 tells us that f generates a function g which has this property for proper orbits T_1, T_2 in (G^2, c). This function g satisfies the condition of Case 2 (pick any constant in T_2). Same if the roles of p_1 and p_2 are interchanged.

Observe that if one of the cases occurs with max instead of min, then we are done also.

So assume now that we have none of the situations above. This implies that for all $c \in V^2$, if F is a sufficiently large substructure of (G^2, c) and if f is canonical on F, then f behaves like a projection on F.

Next observe the following: For arbitrary $c^1, \ldots, c^n \in V^2$, if F is a substructure of $(G^2, c^1, \ldots, c^n)$ which is large enough and if f is canonical on F, then there exists $i \in \{1,2\}$ such that f behaves like p_i between any $a, b \in F$ with $a \notin \{c^1, \ldots, c^n\}$.

Now fix $c^1, \ldots, c^4 \in V^2$ witnessing that f does not behave like a projection. Let F be a substructure of $(G^2, c^1, \ldots, c^4)$ on which f is canonical, and let $i(F)$ be the i of the observation we just made. This i is the same for arbitrarily large F, say wlog $i = 1$ for arbitrarily large F. Say wlog that $EN(c^1, c^2)$ and $N(f(c^1), f(c^2))$; so f behaves like min between c^1 and c^2. But there are arbitrarily large finite substructures F of (G^2, c^1, c^2) such that f behaves like p_1 between any two elements of F one of which is not equal to c^1 or c^2. Thus, given any finite set F, we can delete edges between two elements of F without adding new edges by applying functions from V^2 to V^2 which have the form $(f(x, y), y)$. Doing this successively we obtain a function from V^2 to V^2 which deletes all possible edges on F, and if we apply a binary function of type projection in the end, the composite of all these operations is a binary term over f that behaves like min on F. The proposition then follows from Proposition 33. $\square$

Lemma 38. *Let $f : V^3 \to V$ be a ternary injection that preserves E and N such that there exist $x, y, u, v, p, q \in V^3$ with* $\mathrm{ENN}(x, y)$, $\mathrm{NEN}(u, v)$, $\mathrm{NNE}(p, q)$, *and $E(f(x), f(y))$, $E(f(u), f(v))$ and $E(f(p), f(q))$. Then f generates a binary injection of type* min *or* max, *or a ternary injection of type minority.*

Proof. We may assume that whenever $F \subseteq V^3$ is large enough, then f is not of type minority on F; otherwise, the lemma follows immediately from Proposition 33. Therefore, there exist $m, n \in V^3$ which witness this assumption and which are connected to all triples appearing in the statement of the lemma by an edge in all components. Without loss of generality, we may assume that $\mathrm{EEN}(m, n)$ and $E(f(m), f(n))$. Let $\alpha \in \mathrm{Aut}(G)$ be so that it sends (p_1, q_1, m_1, n_1) to (p_2, q_2, m_2, n_2). Set $g(s, t) := f(s, \alpha(s), t) : V^2 \to V$. Then $\mathrm{NE}((p_1, p_3), (q_1, q_3))$ and $E(g(p_1, p_3), g(q_1, q_3))$. Moreover, $\mathrm{EN}((m_1, m_3), (n_1, n_3))$ and $E(g(m_1, m_3), g(n_1, n_3))$. Thus, g is a binary injection that preserves E and N and that does not behave like a projection. By Proposition 37, g generates a binary injection of type min or of type max. $\square$

Lemma 39. *Let $f : V^3 \to V$ be a ternary injection that preserves E and N such that there exist $x, y, u, v, p, q \in V^3$ with* $\mathrm{ENN}(x, y)$, $\mathrm{NEN}(u, v)$, $\mathrm{NNE}(p, q)$, *and $N(f(x), f(y))$, $N(f(u), f(v))$ and $N(f(p), f(q))$. Then f generates a binary injection of type* min *or* max, *or a ternary injection of type majority.*

Proof. We may assume that whenever $F \subseteq V^3$ is large enough, then f is not of type majority on F; otherwise, the lemma follows immediately from Proposition 33. Therefore, there exist $m, n \in V^3$ which witness this assumption and which are connected to all triples appearing in the statement of the lemma by an edge in all components. Without loss of generality, we may assume that $\text{EEN}(m, n)$ and $N(f(m), f(n))$. Let $\alpha \in \text{Aut}(G)$ be so that it sends (p_1, q_1, m_1, n_1) to (p_2, q_2, m_2, n_2). Set $g(s, t) := f(s, \alpha(s), t) : V^2 \to V$. Then $\text{NE}((p_1, p_3), (q_1, q_3))$ and $N(g(p_1, p_3), g(q_1, q_3))$. Moreover, $\text{EN}((m_1, m_3), (n_1, n_3))$ and $N(g(m_1, m_3), g(n_1, n_3))$. Thus, g is a binary injection that preserves E and N and that does not behave like a projection. By Proposition 37, g generates a binary injection of type min or of type max. $\square$

We are now ready to finish this subsection and provide the proof of Proposition 31.

Proof of Proposition 31. Let f be given. Since the relation H consists of three orbits of 6-tuples, by Lemma 7 f generates an at most ternary function that violates H, and hence we can assume wlog that f itself is at most ternary. The operation f can certainly not be essentially unary, since every essentially unary operation that preserves E and N also preserves H. Applying Theorem 30 to the reduct which has $\text{Inv}(\{f\} \cup \text{Aut}(G))$ as its relations, and by Proposition 5, we get that f generates a binary injective canonical function of type min, max, or p_1. In the first two cases we are done, so consider the last case and denote the function of type p_1 by g.

By adding a dummy variable, we may assume that f is ternary. Now consider

$$h(x, y, z) := g(g(g(f(x, y, z), x), y), z) .$$

Then h is clearly injective, and still violates H – the latter can easily be verified combining the facts that f violates H, g is of type p_1, and all tuples in H have pairwise distinct entries. Because h violates H, one of the following is true.

- there exist $x, y, u, v, p, q \in V^3$ with $\text{ENN}(x, y)$, $\text{NEN}(u, v)$, $\text{NNE}(p, q)$, and $E(h(x), h(y))$, $E(h(u), h(v))$ and $E(h(p), h(q))$, or
- there exist $x, y, u, v, p, q \in V^3$ with $\text{ENN}(x, y)$, $\text{NEN}(u, v)$, $\text{NNE}(p, q)$, and $N(h(x), h(y))$, $N(h(u), h(v))$ and $N(h(p), h(q))$.

By Lemmas 38 and 39, respectively, we get that h, and hence also f, generates a binary canonical injection of type min or max, or a ternary canonical injection of type majority or minority. $\square$

F.3 Tractability of types minority / majority with unbalanced projections

We now prove tractability of the CSP for reducts Γ as in Cases (b) and (c) of Proposition 26, that is, for reducts Γ which have a ternary polymorphism of type majority or minority, as well as a binary polymorphism of type p_1 which is either E-dominated or N-dominated in the second argument. By duality, we may assume that the polymorphism of type p_1 is E-dominated in the second argument. Throughout this section we assume that Γ has a finite signature.

It turns out that for such Γ, we can reduce CSP(Γ) to the CSP of the *injectivization* of Γ. This implies in turn that the CSP can be reduced to a CSP over a Boolean domain.

Definition 40. *A relation is called* injective *if all its tuples have pairwise distinct entries. A structure is called* injective *if it only has injective relations.*

Definition 41. *We define* injectivizations *for relations, atomic formulas, and structures.*

- *Let R be any relation. Then the* injectivization of R*, denoted by* $\mathrm{inj}(R)$*, is the subrelation of R that consists of all tuples of R that only have pairwise distinct entries.*

- *Let $\phi(x_1, \dots, x_n)$ be an atomic formula in the language of a reduct Γ, where $x_1, \dots, x_n$ is a list of the variables that appear in ϕ. Then the* injectivization of $\phi(x_1, \dots, x_n)$ *is the formula $R^{inj}_\phi(x_1, \dots, x_n)$, where R^{inj}_ϕ is a relation symbol which stands for the injectivization of the relation defined by ϕ.*

- *The* injectivization *of a relational structure Γ, denoted by* $\mathrm{inj}(\Gamma)$*, is the relational structure Δ with the same domain as Γ whose relations are the injectivizations of the atomic formulas over Γ, i.e., the relations R^{inj}_ϕ.*

Note that $\mathrm{inj}(\Gamma)$ also contains the injectivizations of relations that are defined by atomic formulas in which one variable might appear several times. In particular, the injectivization of an atomic formula ϕ might have smaller arity than the relation symbol that appears in ϕ.

To state the reduction to the CSP of an injectivization, we also need the following operations on instances of CSP(Γ).

Definition 42. *Let Δ be the injectivization of Γ, and Φ be an instance of* CSP(Γ)*. Then the* injectivization of Φ*, denoted by* $\mathrm{inj}(\Phi)$*, is the instance Ψ of* CSP(Δ) *obtained from Φ by replacing each conjunct $\phi(x_1, \dots, x_n)$ of Φ by $R^{inj}_\phi(x_1, \dots, x_n)$.*

We say that a constraint (=conjunct) in an instance of CSP(Γ) *is false* if it defines an empty relation in Γ. Note that a constraint $R(x_1, \dots, x_k)$ might be false even if the relation R is non-empty (simply because some of the variables from $x_1, \dots, x_n$ might be equal).

Proposition 43. *Let Γ be preserved by a binary injection f of type p_1 that is E-dominated in the second argument. Then the algorithm shown in Figure 2 is a polynomial-time reduction of* CSP(Γ) *to* CSP(Δ)*, where Δ is the injectivization of Γ.*

Proof. In the main loop, when the algorithm detects a constraint that is false and therefore rejects, then Φ cannot hold in Γ, because the algorithm only contracts variables x and y when $x = y$ in all solutions to Φ – and contractions are the only modifications performed

```
// Input: An instance Φ of CSP(Γ) with variables V
While Φ contains a constraint φ that implies x = y for x, y ∈ V do
    Replace each occurrence of x by y in Φ.
    If Φ contains a false constraint then reject
Loop
Accept if and only if inj(Φ) is satisfiable in Δ.
```

Figure 2: Polynomial-time Turing reduction from the CSP(Γ) for Γ closed under an unbalanced binary injection, to the CSP of its injectivization Δ.

on the input formula Φ. So suppose that the algorithm does not reject, and let Ψ be the instance of CSP(Γ) computed by the algorithm when it reaches the final line of the algorithm.

By the observation we just made it suffices to show that Ψ holds in Γ if and only if inj(Ψ) holds in Δ. It is clear that when inj(Ψ) holds in Δ then Ψ holds in Γ (since the constraints in inj(Ψ) have been made stronger). We now prove that if Ψ has a solution s in Γ, then there is also a solution for inj(Ψ) in Δ.

Let s' be any mapping from the variables to G such that for all distinct variables x, y of Ψ we have that

- if $E(s(x), s(y))$ then $E(s'(x), s'(y))$;
- if $N(s(x), s(y))$ then $N(s'(x), s'(y))$;
- if $s(x) = s(y)$ then $E(s'(x), s'(y))$.

Clearly, such a mapping exists. We claim that s' is a solution to Ψ in Γ. Since s' must be injective, it is then clearly also a solution to inj(Ψ).

To prove the claim, let $R(x_1, \dots, x_n)$ be a constraint in Ψ (where $x_1, \dots, x_n$ is a list of the variables of ϕ – R might have higher arity than n). Since we are at the final stage of the algorithm, we can conclude that ϕ does not imply equality of any of the variables $x_1, \dots, x_n$, and so there is for all $1 \leq i < j \leq n$ a tuple $t^{(i,j)}$ such that $R(t^{(i,j)})$ and $t_i \neq t_j$ hold. Since $R(x_1, \dots, x_n)$ is preserved by a binary injection, it is also preserved by injections of arbitrary arity (it is straightforward to build such terms from a binary injection). Application of an injection of arity $\binom{n}{2}$ to the tuples $t^{(i,j)}$ shows that $R(x_1, \dots, x_n)$ contains an injective tuple $(t_1, \dots, t_n)$.

Consider the mapping $r : \{x_1, \dots, x_n\} \to G$ given by $r(x_l) := f(s(x_l), t_l)$. This assignment has the property that for all $1 \leq i, j \leq n$ if $E(s(x_i), s(x_j))$, then $E(r(x), r(y))$, and if $N(s(x_i), s(x_j))$ then $N(r(x_i), r(x_j))$, because f is of type p_1. Moreover, if $s(x_i) = s(x_j)$ then $E(r(x_i), r(x_j))$ because f is E-dominated in the second argument. Therefore, $(s'(x_1), \dots, s'(x_n))$ and $(r(x_1), \dots, r(x_n))$ have the same type in G. Since f is a polymorphism of Γ, we have that $(r(x_1), \ldots, r(x_n))$ satisfies the constraint $R(x_1, \ldots, x_n)$. Hence, s' satisfies $R(x_1, \ldots, x_n)$ as well.

In this fashion we see that s' satisfies all the constraints of Ψ, proving our claim. $\square$

Definition 44. *Let t be a k-tuple of distinct vertices of G, and let q be $\binom{k}{2}$. Then $B(t)$ is the q-tuple $(a_{1,2}, a_{1,3}, \ldots, a_{1,k},\ a_{2,3}, \ldots, a_{k-1,k}) \in \{0,1\}^q$ such that $a_{i,j} = 0$ if $N(t_i, t_j)$ and $a_{i,j} = 1$ if $E(t_i, t_j)$. If R is a k-ary injective relation, then $B(R)$ is the q-ary Boolean relation $\{B(t) \mid t \in R\}$. If ϕ is a formula that defines a relation R over G, then we also write $B(\phi)$ instead of $B(\text{inj}(R))$. Finally, for an injective reduct Γ, we write $B(\Gamma)$ for the structure over a Boolean domain which has the relations of the form $B(R)$, where R is a relation of Γ.*

Proposition 45. *Let Γ be injective. Then* $\text{CSP}(\Gamma)$ *can be reduced to* $\text{CSP}(B(\Gamma))$ *in polynomial time.*

Proof. Let Φ be an instance of $\text{CSP}(\Gamma)$, with variable set W. We create an instance Ψ of $\text{CSP}(B(\Gamma))$ as follows. The variable set of Ψ is the set of unordered pairs of variables from Φ. When $\phi = R(x_1, \ldots, x_k)$ is a constraint in Φ, then Ψ contains the constraint $B(R)(x_{1,2}, x_{1,3}, \ldots, x_{1,k}, x_{2,1}, \ldots, x_{k-1,k})$. It is straightforward to verify that Ψ can be computed from Φ in polynomial time, and that Φ is a satisfiable instance of $\text{CSP}(\Gamma)$ if and only if Ψ is a satisfiable instance of $\text{CSP}(B(\Gamma))$. $\square$

The *Boolean majority operation* is the unique ternary function f on a Boolean domain satisfying $f(x,x,y) = f(x,y,x) = f(y,x,x) = x$. The *Boolean minority operation* is the unique ternary function f on a Boolean domain satisfying $f(x,x,y) = f(x,y,x) = f(y,x,x) = y$.

The following proposition, together with Propositions 43 and 45 solves the case where $\text{Pol}(\Gamma)$ contains a ternary injection of type minority or majority as well as one of the functions of Theorem 30 which are unbalanced and of type projection. It thus shows tractability of Cases (b) and (c) of Proposition 26 given that none of the other cases applies.

Proposition 46. *Let Γ be injective, and suppose it has an polymorphism of type minority (majority). Then $B(\Gamma)$ has a minority (majority) polymorphism, and hence* $\text{CSP}(B(\Gamma))$ *can be solved in polynomial time.*

Proof. It is straightforward to show that $B(\Gamma)$ has a minority (majority) polymorphism. It is well-known (see [22]) that $\text{CSP}(B(\Gamma))$ can then be solved in polynomial time. $\square$

F.4 Tractability of type minority with balanced projections

We show tractability of reducts Γ which have a polymorphism of type minority as well as a binary canonical injection of type p_1 which is balanced. We start by proving that the relations of such reducts can be defined in G by first-order formulas of a certain restricted syntactic form; this normal form will later we essential for our algorithm.

A Boolean relation is called *affine* if it can be defined by a conjunction of linear equations modulo 2. It is well-known that a Boolean relation is affine if and only if it is preserved by the Boolean minority operation (for a neat proof, see e.g. [15]).

In the following, we denote the Boolean exclusive-or connective (xor) by $\oplus$.

Definition 47. *A graph formula is called* edge affine *if it is a conjunction of formulas of the form*

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

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

Definition 48. *A ternary operation $f : V^3 \to V$ is called* tame *if for every $c \in V$, the binary operations $(x, y) \mapsto f(x, y, c)$, $(x, z) \mapsto f(x, c, z)$, and $(y, z) \mapsto f(c, y, z)$ are balanced injections of type p_1.*

Observe that the existence of tame operations and even tame minority operations follows from the fact that G contains all countable graphs as induced subgraphs.

Proposition 49. *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 every injection of type minority which is tame;*
3. *R is preserved by an injection of type minority, and a balanced binary injection of type p_1.*

Proof. We first show the implication from 1 to 2, that n-ary relations R defined by edge affine formulas $\Psi(x_1, \ldots, x_n)$ are preserved by tame functions f of type minority. By injectivity of f, it is easy to see that we only have to show this for the case that Ψ does not contain disequality disjuncts (i.e., $k = 0$). Now let ϕ be a clause from Ψ, say

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

for $p \in \{0,1\}$ and $u_1,\ldots,u_l,v_1,\ldots,v_l \in \{x_1,\ldots,x_n\}$. In the following, it will sometimes be notationally convenient to consider tuples in G satisfying a formula as mappings from the variable set of the formula to V. Let $t_1,t_2,t_3 : \{x_1,\ldots,x_n\} \to V$ be three mappings that satisfy ϕ. We have to show that the mapping $t_0 : \{x_1,\ldots,x_n\} \to V$ defined by $t_0(x) = f(t_1(x),t_2(x),t_3(x))$ satisfies ϕ.

Suppose first that each of t_1,t_2,t_3 satisfies $u_1 \neq v_1 \wedge \cdots \wedge u_l \neq v_l$. In this case, $t_0(u_1) \neq t_0(v_1) \wedge \cdots \wedge t_0(u_l) \neq t_0(v_l)$, since f preserves $\neq$. Note that $E(t_0(u_i),t_0(v_i))$, for $1 \leq i \leq l$, if and only if $E(t_1(u_i),t_1(v_i)) \oplus E(t_2(u_i),t_2(v_i)) \oplus E(t_3(u_i),t_3(v_i)) = 1$. Therefore, since each t_1,t_2,t_3 satisfies $E(u_1,v_1) \oplus \cdots \oplus E(u_l,v_l) = p$, we find that t_0 also satisfies $E(u_1,v_1) \oplus \cdots \oplus E(u_l,v_l) = p \oplus p \oplus p = p$.

Next, suppose that one of t_1,t_2,t_3 satisfies $u_i = v_i$ for some (and therefore for all) $1 \leq i \leq l$. By permuting arguments of f, we can assume that $t_1(u_i) = t_1(v_i)$ for all $i \in \{1,\ldots,l\}$. Since the function f is tame, the operation $g : (y,z) \mapsto f(t_1(u_i),y,z)$ is a balanced injection of type p_1. Suppose that $t_2(u_i) = t_2(v_i)$. Then $E(t_0(u_i),t_0(v_i))$ if and only if $E(t_3(u_i),t_3(v_i))$, since g is balanced. Hence, t_0 satisfies ϕ. Now suppose that $t_2(u_i) \neq t_2(v_i)$. Then $E(t_0(u_i),t_0(v_i))$ if and only if $E(t_2(u_i),t_2(v_i))$, since g is of type p_1. Again, t_0 satisfies ϕ. This shows that f preserves ϕ, and hence also Ψ.

The implication from 2 to 3 is trivial, since every tame function of type minority generates a balanced binary injection of type p_1 by identification of two of its variables. It is also here that we have to check the existence of tame injections of type minority; as mentioned above, this follows easily from the universality of G.

We show the implication from 3 to 1 by induction on the arity n of the relation R. Let g be the balanced binary injection of type p_1, and let h be the operation of type minority. For $n = 2$ the statement of the theorem holds, because all binary relations with a first-order definition in G can be defined over G by expressions as in Definition 47:

- For $x \neq y$ we set $k = 1$ and $l = 0$.
- For $\neg E(x,y)$ we can set $k = 0, l = 1, p = 0$.
- For $\neg N(x,y)$ we can set $k = 0, l = 1, p = 1$.
- Then, $E(x,y)$ can be expressed as $(x \neq y) \wedge \neg N(x,y)$.
- $N(x,y)$ can be expressed as $(x \neq y) \wedge \neg E(x,y)$.
- $x = y$ can be expressed as $\neg E(x,y) \wedge \neg N(x,y)$.
- The empty relation can be expressed as $E(x,y) \wedge N(x,y)$.
- Finally, V^2 can be defined by the empty conjunction.

For $n > 2$, we construct the formula Ψ that defines the relation $R(x_1,\ldots,x_n)$ as follows. If there are distinct $i,j \in \{1,\ldots,n\}$ such that for all tuples t in R we have $t_i = t_j$, consider the relation defined by $\exists x_i.R(x_1,\dots,x_n)$. This relation is also preserved by g and h, and by inductive assumption has a definition Φ as required. Then the formula $\Psi := (x_i = x_j \wedge \Phi)$ proves the claim. So let us assume that for all distinct i,j there is a tuple $t \in R$ where $t_i \neq t_j$. Note that since R is preserved by the binary injective operation g, this implies that R also contains an injective tuple.

Since R is preserved by an operation of type minority, the relation $B(\text{inj}(R))$ is preserved by the Boolean minority operation, and hence has a definition by a conjunction of linear equations modulo 2. From this definition it is straightforward to obtain a definition $\Phi(x_1,\dots,x_n)$ of $\text{inj}(R)$ which is the conjunction of $\bigwedge_{i<j\leq n} x_i \neq x_j$ and of formulas of the form

$$E(u_1,v_1) \oplus \cdots \oplus E(u_l,v_l) = p\ ,$$

for $u_1,\dots,u_l,v_1,\dots,v_l \in \{x_1,\dots,x_n\}$. It is clear that we can assume that none of the formulas of the form $E(u_1,v_1) \oplus \cdots \oplus E(u_l,v_l) = p$ in Φ can be equivalently replaced by a conjunction of shorter formulas of this form.

For all $i,j \in \{1,\dots,n\}$ with $i < j$, let $R_{i,j}$ be the relation that holds for the tuple $(x_1,\dots,x_{i-1},x_{i+1},\dots,x_n)$ iff $R(x_1,\dots,x_{i-1},x_j,x_{i+1},\dots,x_n)$ holds. Because $R_{i,j}$ is preserved by g and h, but has arity $n-1$, it has a definition $\Phi_{i,j}$ as in the statement by inductive assumption. We call the conjuncts of $\Phi_{i,j}$ also the *clauses* of $\Phi_{i,j}$. We add to each clause of $\Phi_{i,j}$ a disjunct $x_i \neq x_j$.

Let Ψ be the conjunction composed of conjuncts from the following two groups:

1. all the modified clauses from all formulas $\Phi_{i,j}$;
2. when $\phi = (E(u_1,v_1) \oplus \cdots \oplus E(u_l,v_l) = p)$ is a conjunct of Φ, then Ψ contains the formula

$$\begin{aligned} &(u_1 \neq v_1 \wedge \cdots \wedge u_l \neq v_l \ \wedge\ \phi) \\ \vee &(u_1 = v_1 \wedge \cdots \wedge u_l = v_l)\ . \end{aligned}$$

Obviously, Ψ is a formula in the required form. We have to verify that Ψ defines R.

Let t be an n-tuple such that $t \notin R$. If t is injective, then t violates a formula of the form

$$E(u_1,v_1) \oplus \cdots \oplus E(u_l,v_l) = p$$

from the formula Φ defining $\text{inj}(R)$, and hence it violates a conjunct of Ψ of the second group. If there are i,j such that $t_i = t_j$ then the tuple $t^i := (t_1,\dots,t_{i-1},t_{i+1},\dots,t_n) \notin R_{i,j}$. Therefore some conjunct ϕ of $\Phi_{i,j}$ is not satisfied by t^i, and $\phi \vee x_i \neq x_j$ is not satisfied by t. Thus, in this case t does not satisfy Ψ either.

It remains to verify that all $t \in R$ satisfy Ψ. Let ψ be a conjunct of Ψ created from some clause in $\Phi_{i,j}$. If $t_i \neq t_j$, then ψ is satisfied by t because ϕ contains $x_i \neq x_j$. If $t_i = t_j$, then $(t_1,\dots,t_{i-1},t_{i+1},\dots,t_n) \in R_{i,j}$ and thus this tuple satisfies $\Phi_{i,j}$. This also implies that t satisfies ψ. Now, let ψ be a conjunct of Ψ from the second group. We distinguish three cases.

1. For all $1 \leq i \leq l$ we have that t satisfies $u_i = v_i$. In this case we are clearly done since t satisfies the second disjunct of ψ.

2. For all $1 \leq i \leq l$ we have that t satisfies $u_i \neq v_i$. Suppose for contradiction that t does not satisfy $E(u_1, v_1) \oplus \cdots \oplus E(u_l, v_l) = p$. Let $r \in R$ be injective, and consider the tuple $s := g(t, r)$. Then $s \in R$, and s is injective since the tuple r and the function g are injective. However, since g is of type p_1, we have $E(s(u_i), s(v_i))$ if and only if $E(t(u_i), t(v_i))$, for all $1 \leq i \leq l$. Hence, s violates the conjunct $E(u_1, v_1) \oplus \cdots \oplus E(u_l, v_l) = p$ from Φ, a contradiction since $s \in \text{inj}(R)$.

3. The remaining case is that there is a proper non-empty subset S of $\{1, \ldots, l\}$ such that t satisfies $u_i = v_i$ for all $i \in S$ and t satisfies $u_i \neq v_i$ for all $i \in \{1, \ldots, n\} \setminus S$. We claim that this case cannot occur. Suppose that all tuples t' from $\text{inj}(R)$ satisfy that $\bigoplus_{i \in S} E(u_i, v_i) = d$ for some $d \in \{0, 1\}$. In this case we could have replaced $E(u_1, v_1) \oplus \cdots \oplus E(u_l, v_l) = p$ by the two shorter formulas $\bigoplus_{i \in S} E(u_i, v_i) = d$ and $\bigoplus_{i \in \{1, \ldots, n\} \setminus S} E(u_i, v_i) = p \oplus d$, in contradiction to our assumption on Φ. So, for each $d \in \{0, 1\}$ there is a tuple $s_d \in \text{inj}(R)$ where $\bigoplus_{i \in S} E(u_i, v_i) = d$ (and thus $\bigoplus_{i \in \{1, \ldots, n\} \setminus S} E(u_i, v_i) = p \oplus d$). Now, for the tuple $g(t, s_{1-p})$ we have

$$\begin{aligned} \bigoplus_{i \in [n]} E(u_i, v_i) &= \bigoplus_{i \in S} E(u_i, v_i) \oplus \bigoplus_{i \in [n] \setminus S} E(u_i, v_i) \\ &= p \oplus (p \oplus (1 - p)) \\ &= 1 - p \neq p \end{aligned}$$

which is a contradiction since $g(t, s_{1-p}) \in \text{inj}(R)$.

Hence, all $t \in R$ satisfy all conjuncts ψ of Ψ. We conclude that Ψ defines R. $\square$

We now present a polynomial-time algorithm for $\text{CSP}(\Gamma)$ for the case that Γ has finitely many edge affine relations.

Definition 50. *Let Γ only have edge affine relations, and let Φ be an instance of* $\text{CSP}(\Gamma)$*. Then the* graph of Φ *is the (undirected) graph whose vertices are unordered pairs of distinct variables of Φ, and which has an edge between distinct sets $\{a, b\}$ and $\{c, d\}$ if Φ contains a constraint whose definition as in Definition 47 has a conjunct of the form*

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

such that $\{a, b\} = \{u_i, v_i\}$ and $\{c, d\} = \{u_j, v_j\}$ for some $i, j \in \{1, \ldots, l\}$.

It is clear that for Γ with finite signature, the graph of an instance Φ of $\text{CSP}(\Gamma)$ can be computed in linear time from Φ.

```
// Input: An instance Φ of CSP(Γ) with variables V
Repeat
  For each connected component C of the graph of Φ do
    Let Ψ be the affine Boolean formula inj(Φ, C).
    If Ψ is unsatisfiable then
      For each {x, y} ∈ C do
        Replace each occurrence of x by y in Φ.
      If Φ contains a false constraint then reject
  Loop
Until inj(Φ, C) is satisfiable for all components C
Accept
```

Figure 3: A polynomial-time algorithm for CSP(Γ) when Γ is preserved by a tame operation of type minority.

Definition 51. *Let* Γ *only have edge affine relations, and let* Φ *be an instance of* $\mathrm{CSP}(\Gamma)$. *For a set* C *of 2-element subsets of variables of* Φ*, we define* $\mathrm{inj}(\Phi, C)$ *to be the following affine Boolean formula. The set of variables of* $\mathrm{inj}(\Phi, C)$ *is* C. *The constraints of* $\mathrm{inj}(\Phi, C)$ *are obtained from the constraints* ϕ *of* Φ *as follows. If* ϕ *has a definition as in Definition 47 with a clause of the form*

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

where all pairs $\{u_i, v_i\}$ *are in* C*, then* $\mathrm{inj}(\Phi, C)$ *contains the conjunct* $\{u_1, v_1\} \oplus \cdots \oplus \{u_l, v_l\} = p$.

Tractability of Case (d) of Proposition 26 now follows from the following proposition and Proposition 49.

Proposition 52. *Let* $\Gamma = (V; E, N, \neq, \ldots)$ *be a reduct of* G *with a finite signature and which is preserved by a tame function of type minority. Then the algorithm shown in Figure 3 solves* $\mathrm{CSP}(\Gamma)$ *in polynomial time.*

Proof. We first show that when the algorithm detects a constraint that is false and therefore rejects in the innermost loop, then Φ must be unsatisfiable. Since variable contractions are the only modifications performed on the input formula Φ, it suffices to show that the algorithm only equates variables x and y when $x = y$ in all solutions. To see that this is true, assume that $\Psi := \mathrm{inj}(\Phi, C)$ is an unsatisfiable Boolean formula for some connected component C. Hence, in any solution s to Φ there must be a $\{x, y\}$ in C such that $s(x) = s(y)$. It follows immediately from the definition of the graph of Φ that then $s(u) = s(v)$ for all $\{u, v\}$ adjacent to $\{x, y\}$ in the graph of Φ. By connectivity of C, we have that $s(u) = s(v)$ for all $\{u, v\} \in C$. Since this holds for any solution to Φ, the contractions in the innermost loop of the algorithm preserve satisfiability.

So we only have to show that when the algorithm accepts, there is indeed a solution to Φ. When the algorithm accepts, we must have that $\mathrm{inj}(\Phi, C)$ has a solution s_C for all components C of the graph of Φ, and no equality is forced by an individual constraint. Let s be a mapping from the variables of Φ to the V such that $E(x_i, x_j)$ if $\{x_i, x_j\}$ is in component C of the graph of Φ and $s_C(\{x_i, x_j\}) = 1$, and $N(x_i, x_j)$ otherwise. It is straightforward to verify that this assignment satisfies all of the constraints. □

F.5 Tractability of type majority with balanced projections

We turn to Case (e) of Proposition 26, i.e., the case where Γ has ternary injection of type majority and a binary canonical injection of type p_1 which is balanced.

A Boolean relation is called *bijunctive* if it can be defined by a conjunction of clauses of size at most two (i.e., it is the solution set to a 2SAT instance). It is well-known that a Boolean relation is bijunctive if and only if it is preserved by the Boolean majority operation (see e.g. [15]).

Definition 53. *A relation R on G is called* graph bijunctive *if it can be defined in G by a conjunction of disjunctions of disequalities, and of formulas of the form*

$$\begin{aligned} x_1 \neq y_1 \vee \ldots \vee x_k \neq y_k \\ &\vee \big(u_1 \neq v_1 \wedge u_2 \neq v_2 \wedge (X(u_1, v_1) \vee Y(u_2, v_2))\big) \\ &\vee (u_1 = v_1 \wedge u_2 = v_2)\ , \end{aligned}$$

where $X, Y \in \{E, N\}$, variables need not be distinct, and k can be 0.

Proposition 54. *Let R be a relation with a first-order definition in G. Then the following are equivalent.*

1. *R is graph bijunctive;*
2. *R is preserved by every tame function of type majority;*
3. *R is preserved by a function of type majority and a balanced injection of type p_1.*

Proof. The proof is very similar to the proof of Proposition 49. We first show the implication from 1 to 2, that relations that are graph bijunctive are preserved by tame functions f of type majority. By injectivity of f, it suffices to show this for the case that the formulas do not contain disequality disjuncts (i.e., $k = 0$). Since the clauses ϕ of such a formula are such that $B(\phi)$ is bijunctive, the claim follows from the fact that bijunctive Boolean relations are preserved by the Boolean majority operation in very much the same way as in Proposition 49.

For the implication from 2 to 3, observe that tame functions of type majority exist since G is universal, and that identifying two variables of such an operation yields a balanced injection of type p_1.

We show the implication from 3 to 1 by induction on the arity n of the relation R. Let g be the balanced binary injection of type p_1, and let h be the operation of type majority. For $n = 2$ the statement of the proposition holds because all binary relations with a first-order definition over G can be defined as in Definition 53.

- for $\neg E(x, y)$ we can set $k = 0$, $X = Y := N$, $u_1 = v_1 := x$, $u_2 = v_2 := y$; dually, $\neg N(x, y)$ can be defined;
- For $x \neq y$, this is trivial;
- $E(x, y)$ can be defined as the conjunct of $x \neq y$ and $\neg N(x, y)$; dually, we can define $N(x, y)$;
- The relation $x = y$ can be obtained as the conjunction of $\neg E(x, y)$ and $\neg N(x, y)$;
- The empty relation is obtained as the conjunction of $E(x, y)$ and $N(x, y)$;
- Finally, V^2 can be defined by the empty conjunction.

For $n > 2$, we construct the formula Ψ that defines the relation $R(x_1, \dots, x_n)$ as follows. If there are distinct $i, j \in \{1, \dots, n\}$ such that for all tuples t in R we have $t_i = t_j$, consider the relation defined by $\exists x_i . R(x_1, \dots, x_n)$. This relation is also preserved by g and h, and by inductive assumption has a definition Φ as required. Then the formula $\Psi := (x_i = x_j \wedge \Phi)$ proves the claim. So let us assume that for all distinct i, j there is a tuple $t \in R$ where $t_i \neq t_j$. Note that since R is preserved by the binary injective operation g, this implies that R also contains an injective tuple.

Since R is preserved by a function of type majority, the relation $B(\mathrm{inj}(R))$ is preserved by the Boolean majority operation, and hence is bijunctive. From this definition it is straightforward to obtain a definition $\Phi(x_1, \dots, x_n)$ of $\mathrm{inj}(R)$ which is the conjunction of $\bigwedge_{1 \leq i < j \leq n} x_i \neq x_j$ and of formulas of the form $E(u, v)$, $N(u, v)$, or

$$X(u_1, v_1) \vee Y(u_1, v_1) \ ,$$

for $u_1, u_2, v_1, v_2 \in \{x_1, \dots, x_n\}$, and $X, Y \in \{E, N\}$. We can assume (by removing successively literals from clauses) that this formula is *reduced*, i.e., that each of the conjuncts is such that removing any of its literals results in an inequivalent formula.

For all $i, j \in \{1, \dots, n\}$ with $i < j$, let $R_{i,j}$ be the relation that holds for the tuple $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ iff $R(x_1, \dots, x_{i-1}, x_j, x_{i+1}, \dots, x_n)$ holds. Because also $R_{i,j}$ is preserved by g and h, but has arity $n - 1$, it has a definition $\Phi_{i,j}$ as in the statement by inductive assumption. We call the conjuncts of $\Phi_{i,j}$ also the *clauses* of $\Phi_{i,j}$. We add to each clause of $\Phi_{i,j}$ a disjunct $x_i \neq x_j$.

Let Ψ be the conjunction composed of conjuncts from the following two groups:

1. all the modified clauses from all formulas $\Phi_{i,j}$;

2. when $\phi = (X(u_1, v_1) \vee Y(u_2, v_2))$ is a conjunct of Φ, then Ψ contains the formula

$$(\phi \wedge u_1 \neq v_1 \wedge u_2 \neq v_2) \vee (u_1 = v_1 \wedge u_2 = v_2)\ .$$

Obviously, Ψ is a formula in the required form. We have to verify that Ψ defines R.

Let t be an n-tuple such that $t \notin R$. If t is injective, then since $t \notin \mathrm{inj}(R)$, it violates a clause of the form $X(u_1, v_1) \vee Y(u_1, v_1)$ of Φ, and hence the corresponding clause in Ψ. If there are i, j such that $t_i = t_j$ then the tuple $t^i := (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) \notin R_{i,j}$. Therefore some conjunct ϕ of $\Phi_{i,j}$ is not satisfied by t^i, and $\phi \vee x_i \neq x_j$ is not satisfied by t. Thus, in this case t does not satisfy Ψ either.

It remains to verify that all $t \in R$ satisfy Ψ. Let ψ be a conjunct of Ψ created from some clause in $\Phi_{i,j}$. If $t_i \neq t_j$, then ψ is satisfied by t because ψ contains $x_i \neq x_j$. If $t_i = t_j$, then $(t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) \in R_{i,j}$ and thus this tuple satisfies $\Phi_{i,j}$. This also implies that t satisfies ψ. Now, let ψ be a conjunct of Ψ from the second group, so it is of the form

$$\begin{aligned}\psi = &\big(u_1 \neq v_1 \wedge u_2 \neq v_2 \wedge (X(u_1, v_1) \vee Y(u_2, v_2))\big)\\ &\vee\ (u_1 = v_1 \wedge u_2 = v_2)\ .\end{aligned}$$

We distinguish three cases.

1. The tuple t satisfies both $u_1 = v_1$ and $u_2 = v_2$. In this case we are clearly done since t satisfies the second disjunct of ψ.

2. The tuple t satisfies $u_1 \neq v_2$ and $u_2 \neq v_2$. Then the argument is exactly the same as the argument in the proof of Proposition 49.

3. The remaining case is that t satisfies $u_1 = v_1$ and $u_2 \neq v_2$ (or $u_1 \neq v_1$ and $u_2 = v_2$, but the proof there is symmetric). We claim that this case cannot occur. If t satisfies $Y(u_2, v_2)$, we are done; so let us assume that t satisfies $\neg Y(u_2, v_2)$. Since we assumed that Φ is *reduced*, it follows that there exists a tuple $s \in \mathrm{inj}(R)$ (and hence in R) where $\neg X(u_1, u_1)$ and $Y(u_1, v_1)$; otherwise, we could have replaced the clause $X(u_1, v_1) \vee Y(u_2, v_2)$ by $X(u_1, v_1)$. Then the tuple $r := g(t, s)$ is also injective, and satisfies $\neg Y(u_2, u_2)$ (since g is of type p_1) and it also satisfies $\neg X(u_1, v_1)$ (since g is balanced). Since g is injective, we have found a tuple $r \in \mathrm{inj}(R)$ that does not satisfy $X(u_1, v_1) \vee Y(u_1, v_1)$, a contradiction.

□

Combining the following proposition with Proposition 54 finishes the proof of tractability of Case (e) of Proposition 26.

Proposition 55. *Let $\Gamma = (V; E, N, \neq, \ldots)$ be a reduct of G with a finite signature and which is preserved by a tame function of type majority. Then* $\text{CSP}(\Gamma)$ *can be solved in polynomial time.*

Proof. The algorithm for $\text{CSP}(\Gamma)$ is a straightforward adaptation of the procedure given in Figure 3, with the difference that instead of affine Boolean equation systems we have to solve 2-SAT instances in the inner loop. □

F.6 Tractability of types max and min

We are left with the case where Γ has a canonical binary injective polymorphism of type max or min, which corresponds to Case (f) of Proposition 26.

We claim that we can assume that this polymorphism is either balanced, or of type max and E-dominated, or of type min and N-dominated.

Proposition 56. *If $\Gamma = (V; E, N, \neq, \ldots)$ is a reduct of the random graph that has a canonical binary injective polymorphism of type* max *(*min*), then it also has a canonical binary injective polymorphism of type* max *which is balanced or E-dominated (N-dominated).*

Proof. We prove the statement for type max (the situation for min is dual). Let p be the polymorphism of type max. Then $h(x, y) := p(x, q(x, y))$ is not N-dominated in the first argument; this is easy to see. But then $p(h(x, y), h(y, x))$ is either balanced or E-dominated, and still of type max. □

We will need the following result which was shown in [3, Proposition 14]. For a relational structure Γ, we denote by $\hat{\Gamma}$ the expansion of Γ that also contains the complement for each relation in Γ. We call a homomorphism between two structures Γ and Δ *strong* if it is also a homomorphism between $\hat{\Gamma}$ and $\hat{\Delta}$.

Proposition 57. *Let Γ be an ω-categorical homogeneous structure such that* $\text{CSP}(\hat{\Gamma})$ *is tractable, and let Δ be a reduct of Γ. If Δ has a polymorphism which is a strong homomorphism from Γ^2 to Γ, then* $\text{CSP}(\Delta)$ *is tractable as well.*

In the following, a strong homomorphism from a power of Γ to Γ will be called *strong polymorphism*. We apply Proposition 57 to our setting as follows.

Proposition 58. *Let $\Gamma = (V; E, N, \neq, \ldots)$ be a reduct of G with a finite signature, and which is preserved by a binary canonical injection which is of type* max *and balanced or E-dominated, or of type* min *and balanced or N-dominated. Then* $\text{CSP}(\Gamma)$ *can be solved in polynomial time.*

Proof. We have the following.

- A canonical binary injection which is of type min and N-dominated is a strong polymorphism of $(V; E, =)$.

- A canonical binary injection which is of type max and E-dominated is a strong polymorphism of $(V; N, =)$.
- A canonical binary injection which is of type max and balanced is a strong polymorphism of $(V; \neg E, =)$.
- A canonical binary injection which is of type min and balanced is a strong polymorphism of $(V; \neg N, =)$.

The tractability result follows from Proposition 57, because

$$\mathrm{CSP}(V; E, \neg E, N, \neq N, =, \neq)$$

can be solved in polynomial time. One way to see this is to verify that all relations are preserved by a tame polymorphism of type majority, and to use the algorithm presented in Section F.5. $\square$

This completes the proof of Proposition 26!

G Classification

By Theorem 1, all reducts of the random graph with finitely many relations define 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 Theorem 13, 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.

Recall the notion of a *clone* from Section C. By Theorem 4 and Proposition 5, the lattice of primitive positive closed reducts of G and the lattice of locally closed clones containing $\mathrm{Aut}(G)$ 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 59. *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 have already met a special case of this concept in Definition 48 of Section F.4: A ternary function is tame if and only if it is hyperplanely balanced and of type p_1.

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

Definition 60. *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.*

Before stating the theorem that lists the minimal tractable polymorphism clones of reducts of G, we observe that if two canonical functions $f, g : V^n \to V$ have the same behavior, then they generate the same clone. This follows easily from the homogeneity of G and by local closure.

Theorem 61. *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.*

Proof. We briefly discuss the tractability of these clones: Clone 1 is tractable by Proposition 10, and Clones 2 to 5 are tractable by Case (f) of Proposition 26. The clones generated by a function of type majority or minority (Clones 6 to 15) are tractable by Cases (b) to (e) of Proposition 26: in those cases, certain binary canonical injections of type projection are required – these are obtained by identifying any two variables of the function of type majority / minority; Figure 4 shows which function of type majority / minority yields which type of binary injection. We leave the verification to the reader. Finally, let $f(x, y)$ be an E-constant binary injection generating Clone 16, and denote the reduct corresponding to this clone by Γ. Then $g(x) := f(x, x)$ is a homomorphism from Γ to the structure Δ induced by the image $g[V]$ in Γ. This structure Δ is invariant under all permutations of its domain, and hence is definable in $(g[V]; =)$; such structures definable by equality only have been called *equality constraint languages* in [5], and their computational complexity has been classified. The structure Δ has a binary injection among its polymorphisms

Binary injection type p_1	Type majority	Type minority
Balanced	Hp. balanced, type p_1	Hp. balanced, type p_1
E-dominated	Hp. E-constant	Hp. type p_1, E-dominated
N-dominated	Hp. N-constant	Hp. type p_1, N-dominated
Balanced in 1st, E-dom. in 2nd arg.	Hp. type max, E-dom.	Hp. type xnor, E-dom.
Balanced in 1st, N-dom. in 2nd arg.	Hp. type min, N-dom.	Hp. type xor, N-dom.

Figure 4: Minimal tractable canonical functions of type majority / minority and their corresponding canonical binary injections of type projection.

(namely, the restriction of f to Δ). It then follows from the results in [5] that CSP(Δ) is tractable. Hence, CSP(Γ) tractable as well, since Γ and Δ are homomorphically equivalent (cf. Proposition 6). The argument for Clone 17 is identical.

It is not difficult to see and even automatically verifiable that the clones are all distinct – a task we leave to the reader or his computer.

We now show minimality for each clone, i.e., we show for each clone $\mathcal{C}$ of the theorem that any subclone of $\mathcal{C}$ containing $\mathrm{Aut}(G)$ is not tractable. By the results of [10], Clones 1 to 5 only have the clone generated by $\mathrm{Aut}(G)$ as proper subclone; this latter clone corresponds to an NP-complete problem since it only contains essentially unary functions (see [3]), and hence minimality follows. The largest proper subclone of Clone 16 which contains $\mathrm{Aut}(G)$ is the clone generated by e_E – this clone contains only essentially unary functions, and hence is hard. The same argument proves minimality for Clone 17. In order to show minimality for the rest of the clones (Clones 6 to 15), we need the following notation: For a ternary function t and a binary function p, we define a ternary function s by $s(x,y,z) := t(p(x,y),p(y,z),p(z,x))$, and a ternary function w by $w(x,y,z) := s(p(x,y),p(y,z),p(z,x))$. Let $\mathcal{C}$ be one of the Clones 6 to 15, and suppose that $\mathcal{D}$ is a tractable subclone. Then Proposition 26 applies to $\mathcal{D}$, and since $\mathcal{C}$ contains no binary injection of type max or min, we see that $\mathcal{D}$ contains a function of type majority or minority; denote this operation by t. Moreover, $\mathcal{D}$ contains a binary canonical injection p of type projection by Theorem 30. But then it contains also the function w as defined above, which is one of the functions generating Clones 6 to 15 (which of the functions depends on the precise behavior of p, and is shown in Figure 4 – we leave the verification to the reader). Hence $\mathcal{D} = \mathcal{C}$.

We now show that there are no other minimal tractable clones except for those of the theorem. Suppose that $\mathcal{C}$ is any minimal tractable clone, and denote by Γ the corresponding reduct. We apply Theorem 7. If $\mathcal{C}$ contains a constant operation, then it contains Clone 1, and hence is equal to this clone by minimality. So we assume that this is not the case. Assume next that $\mathcal{C}$ contains e_E. Then consider the structure Δ induced in Γ on the image $e_E[V]$. Since Γ and Δ are homomorphically equivalent, CSP(Δ) is tractable. Since Δ is definable in the structure $(e_E[V]; =)$ it follows from the results in [5] that it has a polymorphism which is either a constant or a binary injection. The former case is impossible as otherwise also Γ has a constant polymorphism by composing the constant of Δ with e_E. Let thus $f(x,y)$ be the binary injection on $e[V]$ which is a polymorphism of Δ. Then $g(x,y) := f(e_E(x), e_E(y))$ is a polymorphism of Γ. But g is a binary canonical injection which is E-constant, and hence $\mathcal{C}$ contains Clone 16 of our list. By minimality, $\mathcal{C}$ is equal to this clone. The argument when $\mathcal{C}$ contains e_N is identical. Hence by Theorem 11 it remains to consider the case where the endomorphisms of Γ are generated by its automorphisms. Let $\mathcal{G} := \mathrm{Aut}(\Gamma)$. By Theorem 13 there are five possibilities for $\mathcal{G}$. Suppose first that $\mathcal{G}$ is the group of all permutations on V. Then the assumption that $\mathcal{C}$ contains no constant operation and the tractability of $\mathcal{C}$ imply that $\mathcal{G}$ contains all binary injections, by the results of [5]. In particular, $\mathcal{C}$ contains, say, a balanced binary canonical injection operation of type max. But then the clone generated only by this operation and $\mathrm{Aut}(G)$ is a tractable proper subclone of $\mathcal{C}$, contradicting the minimality of $\mathcal{C}$. Hence, $\mathcal{G}$ cannot contain all permutations. Next consider the case where $\mathcal{G}$ is the group generated by the function $- : V \to V$ together with $\mathrm{Aut}(G)$. Then the hard relation $P^{(3)}$ consists of only one orbit in Γ, and hence is violated by an automorphism of Γ. This is a contradiction since the function $-$ preserves $P^{(3)}$. Now suppose that $\mathcal{G}$ contains sw and is a proper subgroup of the full symmetric group. Then the hard relation T has just one orbit in Γ, and since T must be violated, it is also violated by an endomorphism, and hence also an automorphism of Γ. This is again a contradiction since the functions in $\mathcal{G}$ preserve T. It remains to consider the case $\mathcal{G} = \mathrm{Aut}(G)$; this implies that all polymorphisms of Γ preserve E and N, so Γ contains the relations E and N. Thus Proposition 26 applies, and $\mathcal{C}$ contains a binary canonical injection of type max or min, or a function of type minority or majority. If it contains a canonical injection of type max or min, then it contains one of the Clones 2 to 5 by Proposition 56. Otherwise, it contains an a function of type minority or majority, and one of the binary canonical injections of type projection listed in Theorem 30. As in the proof of the minimality of Clones 6 to 15 above, building the terms s and w one can now show that $\mathcal{C}$ contains one of the clones of our list. □

By inspection of all hardness proofs in this paper and the hardness proofs in [5], we also obtain the following. The relation E_6 refers to a relation that forces the clone $\mathrm{Pol}(E_6)$ to contain precisely all unary injective operations, and no other operations, and is defined by

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

Corollary 62. *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 5 showing 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 which are not shown in the picture. Clones are symbolized with a double border when they have a dual clone (generated by the dual function in the sense of Definition 29, 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 61. "E-semidominated" refers to "balanced in the first and E-dominated in the second argument".

Interestingly, our classification result can be put into the form of the so-called *tractability conjecture* [12] for finite domain CSPs. This conjecture says that a finite structure Γ where all endomorphisms are automorphisms either admits a primitive positive interpretation of $(\{0,1\};\{(0,0,1),(0,1,0),(1,0,0)\})$, and CSP($\Gamma$) is NP-hard, or else Γ has a four-ary polymorphism f that satisfies $f(y,y,x,x) = f(x,x,x,y) = f(y,x,y,x)$ for all elements x, y of Γ, and CSP(Γ) can be solved in polynomial time. The only part that is open in the tractability conjecture is whether the existence of a polymorphism with the given properties implies polynomial-time tractability. We show that an analogous version of the tractability conjecture is true for constraint languages Γ that are reducts of the random graph.

Theorem 63. *Let Γ be a reduct of $(V;E)$. Then exactly one of the following holds.*

- *Γ has a canonical 4-ary polymorphism f and automorphisms α_1 and α_2 such that for all $x, y \in V$ we have*

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

  In this case, CSP(Γ) *is tractable.*

- *Γ admits a primitive positive interpretation of* $(\{0,1\};\{(0,0,1),(0,1,0),(1,0,0)\})$. *In this case,* CSP($\Gamma$) *is NP-complete.*

Before we derive this statement from what was shown earlier, we prove a general lemma that will certainly be useful also in other contexts. As usual in universal algebra, when $\mathbf{A}$ is an algebra[1] we denote by $\mathbf{A}^k$ the k-th direct power $\mathbf{A} \times \cdots \times \mathbf{A}$ of $\mathbf{A}$, and by $H(\mathbf{A})$ the set of all algebras with the same signature as $\mathbf{A}$ that are homomorphic images of $\mathbf{A}$. A factor of an algebra is the homomorphic image of a subalgebra of $\mathbf{A}$.

Lemma 64. *Let $(D; f)$ be an algebra with domain D. Suppose f is a k-canonical polymorphism of a structure Γ with domain D, and suppose that Γ has a finite number m of k-types. Then there is an algebra $(\{1,\ldots,m\}; f)$ in $H((D;f)^k)$. Moreover, when Γ is homogeneous and the maximal arity of the relations in Γ is less than or equal to k, then*

[1] In universal algebra, an algebra is simply a structure with a purely functional signature.

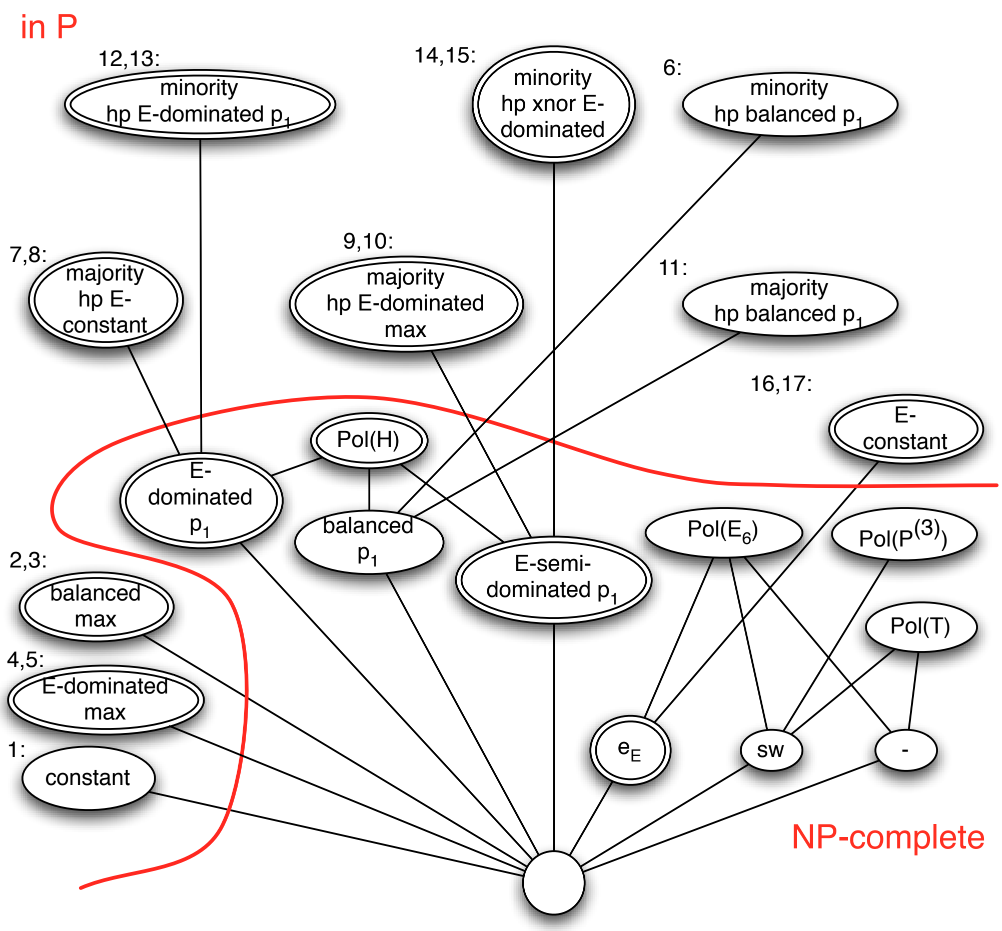

Figure 5: The border: Minimal tractable and maximal hard clones containing $\mathrm{Aut}(G)$.

$(\{1,\ldots,m\};f)$ *satisfies a universally quantified atomic formula* $t_1(\bar{x}) = t_2(\bar{x})$ *if and only if there is an automorphism* α *of* Γ *such that* $(D;f,\alpha)$ *satisfies*

$$\forall \bar{x}.\ t_1(\bar{x}) = \alpha(t_2(\bar{x}))$$

Proof. Let $T_1,\ldots,T_m$ be the k-types of Γ. Define $\mu : D^k \to \{1,\ldots,m\}$ by $\mu(t_1,\ldots,t_k) = i$ if $(t_1,\ldots,t_k)$ has type T_i. Since f is k-canonical, the kernel of μ is a congruence of $(D;f)^k$, and we obtain $(\{1,\ldots,m\};f)$ as a factor algebra.

Now suppose that Γ is homogeneous and the maximal arity of Γ is less than or equal to k. To prove the two implications from the statement, let α be an automorphism of Γ such that $(D;f,\alpha)$ satisfies $\forall \bar{x}.\ t_1(\bar{x}) = \alpha(t_2(\bar{x}))$. Then the kernel of μ is certainly also a congruence $\sim$ of $(D;f,\alpha)$. All universally quantified atomic formulas satisfied by an algebra also hold in products and homomorphic images, and therefore the sentence $\forall \bar{x}.\ t_1(\bar{x}) = \alpha(t_2(\bar{x}))$ also holds in the quotient-algebra $\mathbf{A} := (D;f,\alpha)/\sim$. Since α denotes the identity in this quotient algebra, we can assume wlog that $\mathbf{A} = (\{1,\ldots,m\};f,\mathrm{id})$. It follows that $(\{1,\ldots,m\};f)$ satisfies $\forall \bar{x}.\ t_1(\bar{x}) = t_2(\bar{x})$.

For the opposite direction, we show the existence of the required automorphism α of Γ as follows. Let p be the length of the vector $\bar{x}$. By local closure, it suffices to show that for every finite subset $S = \{d_1,\ldots,d_q\}$ of D^p there is an automorphism β of Γ such that $t_1(d) = \beta(t_2(d))$ for all $d \in S$. By our assumptions on Γ, the type of $(t_1(d_1),\ldots,t_1(d_q))$ in Γ is determined by the k-types that hold on all tuples $(t_1(e_1,\ldots,t_1(e_k)))$ for $e_1,\ldots,e_k$ from S. By assumption, $(t_1(e_1),\ldots,t_1(e_k))$ and $(t_2(e_1),\ldots,t_2(e_k))$ have the same k-type. Hence, $(t_1(d_1),\ldots,t_1(d_q))$ and $(t_2(d_1),\ldots,t_2(d_q))$ have the same type in Γ as well, and so there is an automorphism β of Γ that maps $t_2(d)$ to $t_1(d)$ for all $d \in S$. $\square$

Proof of Corollary 20. Let f be one of the 17 at most ternary canonical polymorphisms from Theorem 61. The number of 2-types in $(V;E)$ is 3, so by Lemma 64 there is a homomorphism μ from $(D;f)^2$ to an algebra $\mathbf{A} = (\{=,E,N\},f)$ (where $=$, E, and N are the image of μ for all pairs (x,y) such that $x = y$, $E(x,y)$, or $N(x,y)$, respectively). A recent unpublished but already famous result in universal algebra, which improves the main result of [23] (see concluding remarks in [23]), implies that every finite idempotent algebra either has a factor all of whose operations are generated by permutations, or the operations of the algebra generate an operation g satisfying

$$g(y,y,x,x) = g(x,x,x,y) = g(y,x,y,x)\ .$$

Hence, by Lemma 64, it suffices to show that all factors of $\mathbf{A}$ contain operations that are not generated by permutations. Three out of the 17 operations are such that the operation f denotes a constant operation in $\mathbf{A}$; in this case, we are clearly done. So suppose we are not in one of those cases.

In all the remaining 14 cases, $\{E,N\}$ induces a subalgebra in which f either acts as a majority (that is, $f(x,x,y) = f(x,y,x) = f(y,x,x) = x$), as a minority (that is,

$f(x,x,y) = f(x,y,x) = f(y,x,x) = y$), or they are binary and satisfy $f(x,y) = f(y,x)$. It is then clear that f cannot be generated by permutations. Four out of the 14 remaining operations are balanced, which is equivalent to saying that both $\{E,=\}$ and $\{N,=\}$ induce a subalgebra $\mathbf{B}$ in $\mathbf{A}$. In this case it is easy to check from the description of the balanced operations in Theorem 61 that

$$f(x,y) \text{ satisfies } f(x,y) = f(y,x) \text{ if } f \text{ is binary, and} \tag{7}$$

$$g(x,y) := f(x,x,y) \text{ satisfies } g(x,y) = g(y,x) \text{ if } f \text{ is ternary.} \tag{8}$$

So f is not generated by permutations in $\mathbf{B}$ as well. For five of the remaining non-balanced operations we have that $\{E,=\}$ induces a subalgebra of $\mathbf{A}$. Again, f satisfies the condition in (7). For the other five remaining operations, the set $\{N,=\}$ induces a subalgebra, and the argument that the operation f is not generated by permutations in those algebras is analogous.

Finally, we have to argue that in none of the 2-element homomorphic images of $\mathbf{A}$ the function f is generated by permutations. All of the 14 remaining operations admit a factoring by the equivalence relation with the classes $\{E,N\}$ and $\{=\}$. Then the function f satisfies (7) in the corresponding factor. It can be verified that from all 14 operations, only

- the balanced operation of type max,
- the N-dominated operation of type min,
- and the edge majority that is hyperplanely of type min and N-dominated

preserve the relation $E(x,y) \leftrightarrow E(u,v)$. In those cases, the algebra $\mathbf{A}$ has a factor $\mathbf{B}$ with kernel classes $\{E\}$ and $\{N,=\}$. For the balanced operation of type max, and the N-dominated operation of type min, the operation f of $\mathbf{B}$ satisfies the condition in (7). For the edge majority that is hyperplanely of type min and N-dominated, the condition in (8) applies. Factors of $\mathbf{A}$ with the classes $\{N\}$ and $\{E,=\}$ can be checked analogously.

These are the only non-trivial homomorphic images of $\mathbf{A}$. This follows from the fact that none of the 14 operations preserves the relation $\{(x,y,u,v) \mid x = y \leftrightarrow u = v\}$ (since all the operations are essential). This completes the proof that f is not generated by permutations in all factors of $\mathbf{A}$. $\square$