SCHAEFER'S THEOREM FOR GRAPHS

MANUEL BODIRSKY AND MICHAEL PINSKER

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

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

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

1. Motivation and the result

In an influential paper in 1978, Schaefer [26] proved a complexity classification for systematic restrictions of the Boolean satisfiability problem. The way in which he restricts the Boolean satisfiability problem turned out to be very fruitful when restricting other computational problems in theoretical computer science, and can be presented as follows.

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

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

The computational complexity of this problem clearly depends on the set Ψ, and is monotone in the sense that if $\Psi \subseteq \Psi'$, then solving Boolean-SAT(Ψ') is at least as hard as solving

---

Date: October 14, 2014.

Key words and phrases. Boolean-SAT, Constraint satisfaction, Graph-SAT, Ramsey theory, random graph.

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

This is the journal version of an extended abstract that appeared at STOC 2011.

Boolean-SAT(Ψ). Schaefer's theorem states that Boolean-SAT(Ψ) can be solved in polynomial time 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 prove a similar classification result, but for the propositional logic of graphs instead of for propositional Boolean logic. More precisely, let E be a relation symbol which denotes an antireflexive and symmetric binary relation and hence stands for the edge relation of a (simple, undirected) graph. We consider formulas that are constructed from atomic formulas of the form $E(x, y)$ and $x = y$ by the usual Boolean connectives (negation, conjunction, disjunction), and call formulas of this form *graph formulas.* A graph formula $\Phi(x_1, \ldots, x_m)$ is *satisfiable* if there exists a graph H and an m-tuple a of elements in H such that $\Phi(a)$ holds in H.

The problem of deciding whether a given graph formula is satisfiable can be very difficult. For example, the question whether or not the Ramsey number $R(5, 5)$ is larger than 43 (which is an open problem, see e.g. [19]) can be easily formulated in terms of satisfiability of a single graph formula. Recall that $R(5, 5)$ is the least number k such that every graph with at least k vertices either contains a clique of size 5 or an independent set of size 5. So the question whether or not $R(5, 5)$ is greater than 43 can be formulated as the question of satisfiability of a graph formula using 43 variables $x_1, \ldots, x_{43}$ on which one imposes the following constraints: all variables denote different vertices in the graph, and for every five-element subset of the variables we add a constraint that forbids that the variables of this subset form a clique or an independent set; this can clearly be stated as a graph formula. If this graph formula is satisfiable, then this implies that $R(5, 5) \leq 43$, and otherwise $R(5, 5) > 43$.

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 91).

Consider now the example where Ψ consists of the formula

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

In this example, 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 either a subgraph with exactly one edge, or a complete triangle. This problem turns out to be tractable.

The class of Graph-SAT problems generalizes the class of problems studied by Schaefer, since to every set Ψ of Boolean formulas we can associate a set Ψ' of graph formulas such that Graph-SAT(Ψ') and Boolean-SAT(Ψ) are essentially the same problem. For every variable x of Ψ there are two variables x_1, x_2 in Ψ'. Then Ψ' contains for every $\psi \in \Psi$ the graph formula obtained from ψ by replacing positive literals x by $E(x_1, x_2)$, and negative literals $\neg x$ by $N(x_1, x_2)$. An instance Φ of Boolean-SAT(Ψ) translates into an instance Φ' of Graph-SAT(Ψ') by modifying Φ in the same way; then Φ is satisfiable if and only if Φ' is satisfiable.

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, the problem of deciding for given Ψ whether* Graph-SAT(Ψ) *is NP-complete or in P is decidable.*

One of the main contributions of this paper is a novel general method combining concepts from universal algebra and model theory with powerful tools of Ramsey theory.

2. Discussion of our strategy

We establish our result by translating Graph-SAT problems into *constraint satisfaction problems* (CSPs) over infinite domains. More specifically, for every set of formulas Ψ we present an infinite 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 with a first-order definition in G, which in turn allows us to find analogies with polymorphisms of structures on a Boolean domain.

We call structures with a first-order definition in G *reducts* of G. While the classical 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 (following [28]) 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 of G 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 [17, 8, 3]) and reductions to the tractable cases of Schaefer's theorem. Reductions of infinite domain CSPs in artificial intelligence (e.g., in temporal and spatial reasoning [18]) 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 [30]. Our results shed some light on the question as to 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 strategy presented in [7] for CSPs of reducts of $(\mathbb{Q};<)$. But while in [7] 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 $\mathcal{C}$ of computational problems 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 structure Δ;
- Δ is homogeneous in a finite language and the class of finite substructures of Δ has the Ramsey property (as in [24]).

The subsequent survey article [10] is devoted to the application of the method of this paper in this more general setting, providing further examples. We remark that in our case, the structure Δ above is the *ordered random graph* (roughly the random graph equipped with the order of the rationals in a random way – confer Section 7) rather than the random graph G itself.

While in [7], 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 [25, 1].

3. Tools from universal algebra and model theory

We now 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 the random graph G.

We write $G = (V;E)$ for the random graph. 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). The random graph G has the property of *quantifier elimination*, that is, every first-order formula is over G equivalent to a quantifier-free first-order formula. Moreover, G has the *extension property*, which often is useful in combinatorial arguments: for all disjoint finite $U, U' \subseteq V$ there exists $v \in V$ such that v is adjacent in G to all members of U and to none in U'. Up to isomorphism, there exists only one unique countably infinite graph which has this extension property, and hence the property can be used as an alternative definition of G. The name of the random graph is due to the fact that if for a countably infinite vertex set, one chooses independently and with probability $\frac{1}{2}$ for each pair of vertices whether to connect the two vertices by an edge, then with probability 1 the resulting graph is isomorphic to the random graph. For the many other remarkable properties of G and its automorphism group $\mathrm{Aut}(G)$, and various connections to many branches of mathematics, see e.g. [14, 15].

Let Γ be a structure with a finite relational signature τ. A first-order τ-formula is called *primitive positive* if it is of the form

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

where the ψ_i are *atomic*, i.e., of the form $y_1 = y_2$ or $R(y_1, \ldots, y_k)$ for a k-ary relation symbol $R \in \tau$ and not necessarily distinct variables y_i. 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 $1 \leq j \leq n$ 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 then follows immediately from the universality of G 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 [23] for finite domain 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 enables the so-called *universal-algebraic approach* to constraint satisfaction, as exposed in the following. We say that a k-ary function (also called *operation*) $f\colon D^k \to D$ *preserves* an m-ary relation $R \subseteq D^m$ if 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, or that f preserves Γ). A unary polymorphism of Γ is also called an *endomorphism* of Γ.

Conversely, for a set F of operations of finite arity defined on a set D and a finitary 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* (see [27], [21]), 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 closed under interpolation (see Proposition 1.6 in [27]): we say that a k-ary operation f on D is *interpolated* by a set of operations F on D if for every finite subset A of D^k there is some k-ary operation $g \in F$ such that g agrees with f on A. We say that F *locally generates* an operation g if g is contained 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 can thus assign to every structure Γ 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 (see for example [22]).

Theorem 4 (from [9])**.** *Let Γ be an ω-categorical structure. Then the relations preserved by the polymorphisms of Γ, i.e., the relations in $\mathrm{Inv}(\mathrm{Pol}(\Gamma))$, are precisely those having a primitive positive definition in Γ.*

Clearly, this theorem together with Lemma 3 imply that if two ω-categorical structures with finite relational signatures have the same clone of polymorphisms, then their CSPs are polynomial-time equivalent. Recall that we have only defined $\mathrm{CSP}(\Gamma)$ for structures Γ with a finite relational signature. But we now see that it makes sense (and here we follow conventions from finite domain constraint satisfaction, see e.g. [13]) to say for arbitrary ω-categorical structures Γ 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 Δ are just the bijective unary polymorphisms of Δ which preserve all relations and their complements; the set of all automorphisms of Δ is denoted by $\mathrm{Aut}(\Delta)$. It follows from the theorem of Ryll-Nardzewski (cf. [22]) that for ω-categorical structures Δ, the closed clones containing $\mathrm{Aut}(\Delta)$ are precisely the polymorphism clones of reducts Γ of Δ. Therefore, in order to determine the computational complexity of the CSP of all reducts Γ of G, it suffices to determine for every closed clone $\mathcal{C}$ containing $\mathrm{Aut}(G)$ the complexity of $\mathrm{CSP}(\Gamma)$ for some reduct Γ of G with $\mathrm{Pol}(\Gamma) = \mathcal{C}$; 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 [27])**.** *Let F be a set of functions on a domain D, and let g be a function on D. Then F locally generates g if and only if g preserves all relations that are preserved by all operations in F, i.e., if and only if $g \in \mathrm{Pol}(\mathrm{Inv}(F))$.*

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

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

We finish this section with a technical general lemma that we will refer to on numerous occasions; 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 [7])**.** *Let Γ be a relational structure with domain D, and suppose that $R \subseteq D^k$ intersects not more than m orbits of k-tuples in Γ. Suppose that an operation f on D violates R. Then $\{f\} \cup \mathrm{Aut}(\Gamma)$ locally generates an at most m-ary operation that violates R.*

4. Overview of the proof

The general method behind proving Theorem 1 can be described as follows; for concreteness, we explain it for our particular situation of the random graph.

The first step is providing hardness proofs for certain relations with a first-order definition over G. More precisely, we define four relations H_1, H_1', H_2, and H_2' which have first order-definitions in G, and show hardness for the CSP defined by each of these relations by reduction of known NP-hard problems. We then know from Lemma 3 that if the CSP for a reduct Γ is not NP-hard, then there is no primitive positive definition of any of these relations in Γ. This implies that there are polymorphisms of Γ which violate the NP-hard relations, by Theorem 4.

We then analyze the polymorphisms of Γ which violate H_1, H_1', H_2, and H_2'. The first, rather basic tool here is Lemma 7, which we use in order to get bounds on the arity of such polymorphisms. The deeper part of our analysis is the simplification of the polymorphisms by means of Ramsey theory. It turns out that the polymorphisms can be assumed to behave regularly in a certain sense with respect to the base structure G (the technical term for functions showing such regular behavior will be *canonical*), making them accessible to case-by-case analysis. In order to be able to use results from Ramsey theory, we have to expand the structure G generically by a linear order $\prec$ on V which is isomorphic to the order of the rational numbers.

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

For reasons of efficiency, we present our proof in a slightly different fashion, albeit the above strategy describes our intuition behind it. We first cite known results on automorphism groups and endomorphism monoids of reducts of G, in particular from [29] and [11]. These older results have been obtained using Ramsey theory, and thus by building on them we outsource the Ramsey-theoretic analysis of unary polymorphisms of reducts. Putting them together, we obtain a statement saying that for any reduct Γ of G, either Γ has a constant endomorphism, and its CSP is tractable, or Γ is homomorphically equivalent to a structure with a first-order definition in $(V; =)$, in which case the complexity of its CSP is known, or its endomorphisms are locally generated by Aut(Γ) (Section 6). The latter case splits into four subcases, corresponding to the precisely four proper subgroups of the full symmetric group on V which are automorphism groups of reducts of G.

In Section 7, we consider each of those four possibilities for Aut(Γ), and analyze the higher arity polymorphisms of Γ to a level of detail not present in the literature (although we do also draw on earlier results on such higher arity polymorphisms from [11]). It is here where we apply Ramsey theory directly in our paper. We show that in all four cases, either one of the hard relations H_1, H_1', H_2, or H_2' has a primitive positive definition in Γ, or Γ has binary or ternary canonical polymorphisms with particular properties. We remark that each of the four hard relations corresponds to one of the possible cases for Aut(Γ).

Finally, Section 8 presents polynomial-time algorithms for reducts having these particular canonical polymorphisms.

The proof of the dichotomy claimed in Theorem 1 is followed by Section 9 in which the classification is stated in more detail and the decidability part of the theorem is derived.

5. Additional conventions

When working with relational structures Γ, we often use the same symbol for a relation of Γ and its relation symbol. In particular, we use the symbol E to denote both the edge relation of G and the corresponding symbol in graph formulas.

Since all our polymorphism clones contain the automorphism group $\mathrm{Aut}(G)$ of the random graph, we will abuse the notion of *generates* from Section 3, 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 $t_1, \ldots, t_n$. When $f \colon A \to B$ is a function and $C \subseteq A$, we write $f[C]$ for $\{f(a) \mid a \in C\}$.

6. Endomorphisms

The goal of this section is the proof of Proposition 8, which will in particular allow us to reduce the classification task to the classification of those structures whose automorphism generate its endomorphisms. To state the proposition, we first define the following unary functions on V that will play an important role throughout the paper.

If we flip edges and non-edges of G, then the resulting graph is isomorphic to G: it is straightforward to verify the extension property. Let $-$ be such an isomorphism.

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; again, this follows by verifying the extension property. Let sw_S be such an isomorphism for each non-empty finite S. Any two such functions generate one another [28]. We also write sw for $\mathrm{sw}_{\{0\}}$, where $0 \in V$ is any fixed element of V.

There are automorphisms α, β of G such that the mappings $x \mapsto -(\alpha(-(x)))$ and $x \mapsto \mathrm{sw}(\beta(\mathrm{sw}(x)))$ are the identity functions on V; the existence of such automorphisms can be shown with a standard back-and-forth argument, see e.g. [22]. Hence, if $-$ or sw preserve a relation R with a first-order definition in G, they automatically preserve also the complement of R, and thus are automorphisms of the structure $(V; R)$.

The graph G contains all countable graphs as induced subgraphs. In particular, it contains an infinite complete subgraph. The homogeneity of G implies that any two injective unary operations on V whose images induce complete subgraphs in G generate one another (see, e.g., [11]); let e_E be one such operation. Similarly, G contains an infinite independent set. Let e_N be an injective unary operation on V whose image induces an infinite independent set in G.

Proposition 8. *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 in P.*
(b) *Γ has e_E or e_N among its endomorphisms, and 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* [6]*, and is either in P or NP-complete.*
(c) *The endomorphisms of Γ are precisely the functions generated by $\{-\}$.*

(d) *The endomorphisms of* Γ *are precisely the functions generated by* $\{\mathrm{sw}\}$.
(e) *The endomorphisms of* Γ *are precisely the functions generated by* $\{-, \mathrm{sw}\}$.
(f) *The endomorphisms of* Γ *are precisely the functions generated by* $\mathrm{Aut}(G)$, *i.e., all endomorphisms of* Γ *preserve* E *and* N.

Proposition 8 follows from two results about unary functions on G. The first result is from [29]; its reformulation from [11] reads as follows.

Theorem 9. *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 generated by* $\mathrm{Aut}(\Gamma)$.

The second result we use, from [28], states that there exist precisely five permutation groups on V that contain $\mathrm{Aut}(G)$ and which are closed in the sense that they contain all permutations which they interpolate. By the theorem of Ryll-Nardzewski (confer also the discussion in Section 3), these groups correspond precisely to the automorphism groups of reducts of G. Thus, the last case of Theorem 9 splits into five subcases, one for each group of the form $\mathrm{Aut}(\Gamma)$. We will next cite the theorem that lists them.

Definition 10. For $k \geq 1$, let $R^{(k)}$ be the k-ary relation that contains a tuple $(x_1, \ldots, x_k) \in V^k$ if $x_1, \ldots, x_k$ are pairwise distinct, and the number of edges between these k vertices is odd.

Definition 11. We say that two structures Γ, Δ on the same domain are *first-order interdefinable* if all relations of Γ have a first-order definition in Δ and vice-versa.

Theorem 12 (from [28]). *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, $\mathrm{Aut}(\Gamma)$ *contains* $\{-\}$, *but not* $\{\mathrm{sw}\}$.
(3) Γ *is first-order interdefinable with* $(V; R^{(3)})$;
equivalently, $\mathrm{Aut}(\Gamma)$ *contains* $\{\mathrm{sw}\}$, *but not* $\{-\}$.
(4) Γ *is first-order interdefinable with* $(V; R^{(5)})$;
equivalently, $\mathrm{Aut}(\Gamma)$ *contains* $\{-, \mathrm{sw}\}$, *but not all permutations of* V.
(5) Γ *is first-order interdefinable with* $(V; =)$;
equivalently, $\mathrm{Aut}(\Gamma)$ *contains all permutations of* V.

of Proposition 8. If Γ has a constant endomorphism, then $\mathrm{CSP}(\Gamma)$ is trivial, and in P. Otherwise, by Theorem 9, Γ is preserved by e_N, e_E, or the endomorphisms of Γ are generated by $\mathrm{Aut}(\Gamma)$.

We claim that if Γ has the endomorphisms e_E or e_N, then Γ is homomorphically equivalent to an infinite structure that is preserved by all permutations of its domain. But this is clear since $e_E[V]$ and $e_N[V]$ induce structures in G which are invariant under all permutations of their domain.

If the endomorphisms of Γ are generated by $\mathrm{Aut}(\Gamma)$, then the statement follows from Theorem 12: this is clear for the first four cases of the theorem; in the last case, Γ has all unary injections among its endomorphisms, and in particular the functions e_E and e_N. $\square$

7. Higher arity polymorphisms

In the following we will be concerned with reducts Γ of G where the endomorphisms of Γ are either the endomorphisms of $(V; E, N)$, or precisely the functions generated by $\{-\}$, by $\{\mathrm{sw}\}$, or by $\{-, \mathrm{sw}\}$, since for all other reducts Γ of G the complexity of $\mathrm{CSP}(\Gamma)$ has already been determined in Proposition 8. We first introduce the general concepts which allow us to analyze polymorphisms of reducts of G using Ramsey theory (Section 7.1). These concepts will be crucial in all four cases which we shall then approach in Sections 7.2 to 7.5.

7.1. Canonical Behavior.

It will turn out that the relevant polymorphisms 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 13. Let Δ be a structure. The *type* $\mathrm{tp}(a)$ of an n-tuple a of elements in Δ is the set of first-order formulas with free variables $x_1, \ldots, x_n$ that hold for a in Δ. For structures $\Delta_1, \ldots, \Delta_k$ and tuples $a^1, \ldots, a^n \in \Delta_1 \times \cdots \times \Delta_k$, the type of $(a^1, \ldots, a^n)$ in $\Delta_1 \times \cdots \times \Delta_k$, denoted by $\mathrm{tp}(a^1, \ldots, a^n)$, is the k-tuple containing the types of $(a^1_i, \ldots, a^n_i)$ in Δ_i for each $1 \leq i \leq k$.

We bring to the reader's attention the well-known fact that in homogeneous structures in a finite language, in particular in the random graph, two n-tuples have the same type if and only if their orbits coincide.

Definition 14. Let $k \geq 1$ and let $\Delta_1, \ldots, \Delta_k, \Lambda$ be structures. A *type condition* between $\Delta_1 \times \cdots \times \Delta_k$ and Λ is a pair (t, s), where t is a type of an n-tuple in $\Delta_1 \times \cdots \times \Delta_k$, and s is a type of an n-tuple in Λ, for some $n \geq 1$. A function $f\colon \Delta_1 \times \cdots \times \Delta_k \to \Lambda$ *satisfies* a type condition (t, s) between $\Delta_1 \times \cdots \times \Delta_k$ and Λ if for all tuples $a^1, \ldots, a^n \in \Delta_1 \times \cdots \times \Delta_k$ with $\mathrm{tp}(a^1, \ldots, a^n) = t$ the n-tuple $(f(a^1_1, \ldots, a^1_k), \ldots, f(a^n_1, \ldots, a^n_k))$ has type s in Λ. A *behavior* is a set of type conditions between a product of structures $\Delta_1 \times \cdots \times \Delta_k$ and a structure Λ. A function from $\Delta_1 \times \cdots \times \Delta_k$ to Λ *has behavior B* if it satisfies all the type conditions of B.

Definition 15. Let $\Delta_1, \ldots, \Delta_k, \Lambda$ be structures. An operation $f\colon \Delta_1 \times \cdots \times \Delta_k \to \Lambda$ is *canonical* if for all types t of n-tuples in $\Delta_1 \times \cdots \times \Delta_k$ there exists a type s of an n-tuple in Λ such that f satisfies the type condition (t, s). In other words, n-tuples of equal type in $\Delta_1 \times \cdots \times \Delta_k$ are sent to n-tuples of equal type in Λ under f.

We remark that since G is homogeneous and has only binary relations, the type of an n-tuple a in G is determined by its binary subtypes, i.e., the types of the pairs (a_i, a_j), where $1 \leq i, j \leq n$. In other words, the type of a is determined by which of its components are equal, and between which of its components there is an edge. Therefore, a function $f\colon G^k \to G$ 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 m-ary relations $R_1, \ldots, R_k$ over V, we will in the following write $R_1 \cdots R_k$ for the m-ary relation on V^k that holds between k-tuples $x^1, \ldots, x^m \in V^k$ iff $R_i(x^1_i, \ldots, x^m_i)$ holds for all $1 \leq i \leq k$. We start with behaviors of binary functions.

Definition 16. We say that a binary injective operation $f\colon V^2 \to V$ 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 $N(f(u), f(v))$. Note that these are precisely those functions that satisfy a certain behavior with two type conditions. To see this, let t be the type of any $u, v \in V^2$ with $\mathrm{E}{=}(u, v)$, and let s be the type of any $x, y \in V$ with $E(x, y)$. Then the first type condition is (t, s). For the second type condition, let t' be the type of any $u, v \in V^2$ with $\mathrm{N}{=}(u, v)$, and let s' be the type of any $x, y \in V$ with $N(x, y)$. The second type condition is (t', s').
- *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.

It is easy to see that, as explained above for the first item, each of those properties describes the set of all functions of a certain behavior.

Note that a binary injection of type max is reminiscent of the Boolean maximum function on $\{0, 1\}$, where E takes the role of 1 and N the role of 0: for $u, v \in V^2$ with ${\neq}{\neq}(u, v)$, we have $E(f(u), f(v))$ if u, v are connected by an edge in at least one coordinate, and $N(f(u), f(v))$ otherwise. The names "min" and "projection" can be explained similarly.

Also 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 G^2, but not all such types: for example, it does not tell us whether or not $E(f(u), f(v))$ for $u, v \in V^2$ with $u_1 = v_1$. However, being both of type max (or of type min) and balanced does mean that a function is canonical. The next definition contains some important behaviors of ternary functions.

Definition 17. An injective ternary function $f : V^3 \to V$ is of type

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

7.2. **When the endomorphisms of a reduct are generated by** $\mathrm{Aut}(G)$**.** We investigate Case (f) of Proposition 8. In this situation, the following lemma states that we may assume that the reduct contains the relations E and N.

Lemma 18. *Let Γ be a reduct of G. Then the endomorphisms of Γ are generated by* $\mathrm{Aut}(G)$ *if and only if the relations E and N are primitive positive definable in Γ.*

Proof. If these relations are primitive positive definable in Γ, then they are preserved by all endomorphisms of Γ by Theorem 4. Hence, the restriction of any endomorphism to a finite set is a partial isomorphism of G, and thus extends to an automorphism of G by homogeneity. It follows that any endomorphism can be interpolated by an element of $\mathrm{Aut}(G)$ on any finite set, and hence it is generated by $\mathrm{Aut}(G)$.

If the endomorphisms of Γ are generated by $\mathrm{Aut}(G)$, then E and N are primitive positive definable in Γ by Theorem 4 and Lemma 7. $\square$

The following relation characterizes the NP-complete cases in the situation of this section.

Definition 19. We define a 6-ary relation $H_1(x_1, y_1, x_2, y_2, x_3, y_3)$ on V by

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

Our goal for this section is to prove the following proposition, which states that if $\Gamma = (V; E, N, \dots)$ is a reduct of G, then either H_1 has a primitive positive definition in Γ, and $\mathrm{CSP}(\Gamma)$ is NP-complete, or Γ has a canonical polymorphism with a certain behavior. Each of the listed canonical polymorphisms implies tractability for $\mathrm{CSP}(\Gamma)$, and we will present algorithms proving this in Section 8.

Theorem 20. *Let Γ be a reduct of G whose endomorphisms are generated by* $\mathrm{Aut}(G)$. *Then at least one of the following holds:*

(a) *There is a primitive positive definition of H_1 in Γ.*
(b) $\mathrm{Pol}(\Gamma)$ *contains a canonical ternary injection of type minority, as well as a canonical binary injection which is of type p_1 and either E-dominated or N-dominated in the second argument.*
(c) $\mathrm{Pol}(\Gamma)$ *contains a canonical ternary injection of type majority, as well as a canonical binary injection which is of type p_1 and either E-dominated or N-dominated in the second argument.*
(d) $\mathrm{Pol}(\Gamma)$ *contains a canonical ternary injection of type minority, as well as a canonical binary injection which is balanced and of type projection.*
(e) $\mathrm{Pol}(\Gamma)$ *contains a canonical ternary injection of type majority, as well as a canonical binary injection which is balanced and of type projection.*
(f) $\mathrm{Pol}(\Gamma)$ *contains a canonical binary injection of type* max *or* min.

The remainder of this section contains the proof of Theorem 20, and is organized as follows: we first show that the relation H_1 is hard. We then prove that if H_1 does not have a primitive positive definition in a reduct Γ as in Theorem 20, then Γ has the polymorphisms of one of the Cases (b) to (f) of the theorem.

7.2.1. *Hardness of H_1.* We present the hardness proof of the relation in Case (a) of Theorem 20.

Proposition 21. $\mathrm{CSP}(V; H_1)$ *is NP-hard.*

Proof. The proof is a reduction from positive 1-in-3-3SAT (one of the hard problems in Schaefer's classification; also see [20]). 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($V; H_1$) 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 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 s 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 $(V; H_1)$ satisfies Ψ. To show this, let F be the graph whose vertices are the variables of Ψ, and that has an edge between u_x and v_x if x is set to 1 under the mapping s, and that has no other edges. 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 $(V; H_1)$ satisfies Ψ. Indeed, let $H(u_x, v_x, u_y, v_y, u_z, v_z)$ be a clause from Ψ. By definition of F, the conjunction in the first line of the definition of H_1 is clearly satisfied; moreover, from the disjunction in the remaining lines of the definition of H_1 exactly one disjunct will be true, since in the corresponding clause $\{x, y, z\}$ of Φ exactly one of the values $s(x), s(y), s(z)$ equals 1. This argument can easily be inverted to see that every solution to Ψ can be used to define a solution to Φ (in which for a variable x of Φ one sets $s(x)$ to 1 iff in the solution to Ψ there is an edge between u_x and v_x). $\square$

7.2.2. *Producing canonical functions.* We now show that if $\Gamma = (V; E, N, \ldots)$ is a reduct of G such that there is no primitive positive definition of H_1 in Γ, then one of the other cases of Theorem 20 applies. By Theorem 4, Γ has a polymorphism that violates H_1.

Definition 22. A function $f\colon V^n \to V$ is called *essentially unary* if it depends on only one of its variables; otherwise, it is called *essential.*

Note that any essentially unary function preserving both E and N preserves all relations with a first-order definition in G, and in particular H_1; this is a straightforward consequence of Lemma 18 and the fact that the automorphisms of G have this property (cf. [22]). Therefore we have that if a polymorphism f of Γ violates H_1, then it must be essential. Thus the following theorem from [11] 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 23. The *dual* of a function $f(x_1, \ldots, x_n)$ on G is the function $-f(-x_1, \ldots, -x_n)$.

Theorem 24 (from [11])**.** *Let f be an essential operation on G preserving E and N. Then it generates one of the following binary functions.*

- *a canonical injection of type p_1 which is balanced;*
- *a canonical injection of type* max *which is balanced;*
- *a canonical injection of type p_1 which is E-dominated;*
- *a canonical injection of type* max *which is E-dominated;*
- *a canonical injection of type p_1 which 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).

It follows from Theorem 24 that indeed, if H_1 does not have a primitive positive definition in a reduct $\Gamma = (V; E, N, \ldots)$, then Γ has one of the binary canonical polymorphisms mentioned in Theorem 20. In order to complete the proof of Theorem 20, we have to additionally show that when f does not generate a binary injection of type min or max, it generates ternary canonical injection of type minority or majority. That is, we have to prove the following.

Proposition 25. *Suppose that f is an operation on G that preserves the relations E and N and violates the relation H_1. Then f generates a binary canonical injection of type* min *or* max*, or a ternary canonical injection of type minority or majority.*

The remainder of this section will be devoted to the proof of this proposition. This will be achieved by refining the Ramsey-theoretic methods developed in [11] which are suitable for investigating functions on G in several variables.

In our proof of Proposition 25, we really would like to take one of the "nice" functions g which we know is generated by f of Theorem 24, and then show that g generates one of the functions of Proposition 25. However, the problem with this are the canonical binary injections of type p_1, since functions of type p_1 do not violate H_1 anymore. Hence, when simply passing to a function of the theorem, we lose the information that our f violates H_1, which we must use at some point, since H_1 is a hard relation. We are thus obliged to improve Theorem 24 for functions violating H_1. Before that, let us observe that Theorem 24 implies that we can restrict our attention to binary and ternary injections.

Lemma 26. *Let f be an operation on G which preserves E and N and violates H_1. Then f generates a ternary injection which shares the same properties.*

Proof. Since the relation H_1 consists of three orbits of 6-tuples with respect to G, Lemma 7 implies that f generates an at most ternary function that violates H_1, and hence we can assume that f itself is at most ternary; by adding a dummy variable if necessary, we may assume that f is actually ternary. Moreover, f must certainly be essential, since essentially unary operations that preserve E and N also preserve H_1. Applying Theorem 24, we get that f generates a binary canonical injection g of type min, max, or p_1. In the first two cases we are done, since binary injections of type min and max violate H_1; so consider the last case where g is of type p_1. 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_1 – the latter can easily be verified combining the facts that f violates H_1, g is of type p_1, and all tuples in H_1 have pairwise distinct entries. ☐

It will turn out that just as in the proof of Lemma 26, there are two cases for f in the proof of Proposition 25: Either all binary canonical injections generated by f are of type projection, and f generates a ternary canonical injection of type majority or minority, or f generates a binary canonical injection of type min or max. We start by considering the first case, which is combinatorially less involved.

7.2.3. *Producing majorities and minorities.* A *copy* of a structure F in a structure Δ is an induced substructure of Δ that is isomorphic to F.

Definition 27. Let $\Delta_1, \ldots, \Delta_k$ and Λ be structures, $f \colon \Delta_1 \times \cdots \times \Delta_k \to \Lambda$ be a function, and let (t, s) be a type condition for such functions. If S is a subset of $\Delta_1 \times \cdots \times \Delta_k$, then we say that f *satisfies the type condition* (t, s) *on* S if for all tuples $a^1, \ldots, a^n \in S$ with $\mathrm{tp}(a^1, \ldots, a^n) = t$ in $\Delta_1 \times \cdots \times \Delta_k$ the n-tuple $(f(a^1_1, \ldots, a^1_k), \ldots, f(a^n_1, \ldots, a^n_k))$ has type s in Λ. We say that f *satisfies a behavior* B *on* S if it satisfies all type conditions of B on S.

Finally, we say that f satisfies B *on arbitrarily large (finite) substructures* of $\Delta_1 \times \cdots \times \Delta_k$ if for all finite substructures F_i of Δ_i, where $1 \leq i \leq k$, there exist copies F'_i in Δ_i such that f satisfies B on the product $F'_1 \times \cdots \times F'_k$ of these copies.

In the following general proposition we exceptionally use the notion "locally generates" in its original sense (see Section 3). The proof is a standard compactness argument, which we include nonetheless for the convenience of the reader. Similar proofs can be found, for example, in [12] for arbitrary homogeneous structures in a finite language, or for the random graph in [11].

Proposition 28. *Let $\Delta_1, \ldots, \Delta_k$ and Λ be homogeneous structures on the same countably infinite domain D, and assume that Λ has a finite language. Let moreover B be a behavior for functions from $\Delta_1 \times \cdots \times \Delta_k$ to Λ, and let $f\colon D^k \to D$ be a function which satisfies B on arbitrarily large substructures of $\Delta_1 \times \cdots \times \Delta_k$. Then $\{f\} \cup \mathrm{Aut}(\Lambda) \cup \mathrm{Aut}(\Delta_1) \cup \cdots \cup \mathrm{Aut}(\Delta_k)$ locally generates a function from D^k to D which satisfies B everywhere.*

Proof. Write $D = \{d_0, d_1, \ldots\}$. We construct a sequence $(g_i)_{i\in\omega}$ such that for all $i \in \omega$

(i) g_i is a function from D^k to D locally generated by $\{f\} \cup \mathrm{Aut}(\Lambda) \cup \mathrm{Aut}(\Delta_1) \cup \cdots \cup \mathrm{Aut}(\Delta_k)$;
(ii) g_i behaves like B on $\{d_0, \ldots, d_i\}^k$;
(iii) g_{i+1} agrees with g_i on $\{d_0, \ldots, d_i\}^k$.

The sequence then defines a function $g : D^k \to D$ by setting $g(d_{i_1}, \ldots, d_{i_k}) := g_m(d_{i_1}, \ldots, d_{i_k})$, for any $m \geq i_1, \ldots, i_k$. This function g is clearly locally generated by $\{g_i : i \in \omega\}$ by local closure, and behaves like B everywhere.

To construct the sequence, we first construct a sequence $(h_i)_{i\in\omega}$ which only satisfies (i) and (ii) of the requirements for the sequence $(g_i)_{i\in\omega}$. Let $i \in \omega$ be given. There exist subsets $F_1, \ldots, F_k$ of D such that F_j is isomorphic with $\{d_0, \ldots, d_i\}$ as substructures of Δ_j for all $1 \leq j \leq k$ and such that f behaves like B on $F_1 \times \cdots \times F_k$. Let α_j be an automorphism of Δ_j sending $\{d_0, \ldots, d_i\}$ onto F_j, for all $1 \leq j \leq k$; these automorphisms exist by the homogeneity of the Δ_j. Then we can set $h_i(x_1, \ldots, x_k) := f(\alpha_1(x_1), \ldots, \alpha_k(x_k))$.

Now to obtain the sequence $(g_i)_{i\in\omega}$ from the sequence $(h_i)_{i\in\omega}$, let $a = (a_0, a_1, \ldots)$ be an enumeration of D^k such that the elements of $\{d_0, \ldots, d_i\}^k$ are an initial segment of this enumeration for each $i \in \omega$ (that is, they constitute the first $(i+1)^k$ entries). Denote for all $i, j \in \omega$ by $b^{i,j}$ the $(i+1)^k$-tuple which is obtained by applying h_j to each of the first $(i+1)^k$ entries of the enumeration a. Set $t^{i,j}$ to be the type of $b^{i,j}$ in Λ. For $i, j, r, s \in \omega$ set $t^{i,j} \leq t^{r,s}$ if $i \leq r$ and $t^{i,j}, t^{r,s}$ agree on the variables they have in common, i.e., the restriction of $b^{r,s}$ to its initial segment of length $(i+1)^k$ has the same type as $b^{i,j}$ in Λ. This relation defines a tree on the types $t^{i,j}$. Since Λ is homogeneous in a finite language, for every $i \in \omega$ there are only finitely many different types of $(i+1)^k$-tuples in Λ. Hence, for every $i \in \omega$, there are only finitely many distinct types $t^{i,j}$, and so this tree is finitely branching. Moreover, there exists a $q \in \omega$ such that $t^{i,s} = t^{i,q}$ for infinitely many $s \in \omega$. Deleting all elements of the tree which do not enjoy this latter property, we are thus still left with an infinite tree. Hence by Kőnig's lemma it has an infinite branch $(t^{0,j_0}, t^{1,j_1}, \ldots)$. Since we have reduced the tree to its "infinite" nodes, we may assume that the j_i are strictly increasing, and in particular that $j_i \geq i$ for all $i \in \omega$. Since Λ is homogeneous and by definition of the tree, we can pick for all $i \in \omega$ an automorphism α_i of Λ which sends the initial segment of length $(i+1)^k$ of $b^{i+1,j_{i+1}}$ to b^{i,j_i}. Then setting $g_i := h_{j_i} \circ \alpha_{i-1} \circ \cdots \circ \alpha_0$ for all $i \in \omega$ yields the desired sequence: (i) is obvious. (ii) holds since h_i satisfies (ii), h_{j_i} still satisfies (ii) since $j_i \geq i$, and g_i satisfies (ii) since the property is preserved under applications of automorphisms of Λ. (iii) is by construction. $\square$

Proposition 29. *Let f be an operation on G that preserves E and N and violates H_1. Suppose moreover that all binary injections generated by f are of type projection. Then f generates a ternary canonical injection of type majority or minority.*

Proof. By Lemma 26, we can assume that f is a ternary injection. Because f violates H_1, there are $x^1, x^2, x^3 \in H$ such that $f(x^1, x^2, x^3) \notin H$. In the following, we will write $x_i := (x_i^1, x_i^2, x_i^3)$ for $1 \leq i \leq 6$. So $(f(x_1), \ldots, f(x_6)) \notin H$.

If there were an automorphism α of G such that $\alpha(x^i) = x^j$ for $1 \leq i \neq j \leq 3$, then f would generate a binary injection that still violates H_2, which contradicts the assumption that all binary injections generated by f are of type projection. By permuting arguments of f if necessary, we can therefore assume without loss of generality that

$$\mathrm{ENN}(x_1, x_2), \ \mathrm{NEN}(x_3, x_4), \text{ and } \ \mathrm{NNE}(x_5, x_6).$$

We set

$$S := \{y \in V^3 \mid \mathrm{NNN}(x_i, y) \text{ for all } 1 \leq i \leq 6\} .$$

Consider the binary relations $Q_1Q_2Q_3$ on V^3, where $Q_i \in \{E, N\}$ for $1 \leq i \leq 3$. We claim that for each such relation $Q_1Q_2Q_3$, whether $E(f(u), f(v))$ or $N(f(u), f(v))$ holds for $u, v \in S$ with $Q_1Q_2Q_3(u, v)$ does not depend on u, v; that is, whenever $u, v, u', v' \in S$ satisfy $Q_1Q_2Q_3(u, v)$ and $Q_1Q_2Q_3(u', v')$, then $E(f(u), f(v))$ if and only if $E(f(u'), f(v'))$. We go through all possibilities of $Q_1Q_2Q_3$.

(1) $Q_1Q_2Q_3 = \mathrm{ENN}$. Let $\alpha \in \mathrm{Aut}(G)$ be such that (x_1^2, x_2^2, u_2, v_2) is mapped to (x_1^3, x_2^3, u_3, v_3); such an automorphism exists since $\mathrm{NNN}(x_1, u), \mathrm{NNN}(x_1, v), \mathrm{NNN}(x_2, u), \mathrm{NNN}(x_2, v)$, and since (x_1^2, x_2^2) has the same type as (x_1^3, x_2^3), and (u_2, v_2) has the same type as (u_3, v_3). By assumption, the operation g defined by $g(x, y) := f(x, y, \alpha(y))$ must be of type projection. Hence, $E(g(u_1, u_2), g(v_1, v_2))$ iff $E(g(x_1^1, x_1^2), g(x_2^1, x_2^2))$. Combining this with the equations $(f(u), f(v)) = (g(u_1, u_2), g(v_1, v_2))$ and $(g(x_1^1, x_1^2), g(x_2^1, x_2^2)) = (f(x_1), f(x_2))$, we get that $E(f(u), f(v))$ iff $E(f(x_1), f(x_2))$, and so we are done.
(2) $Q_1Q_2Q_3 = \mathrm{NEN}$ or $Q_1Q_2Q_3 = \mathrm{NNE}$. These cases are analogous to the previous case.
(3) $Q_1Q_2Q_3 = \mathrm{NEE}$. Let α be defined as in the first case. By assumption, the operation defined by $f(x, y, \alpha(y))$ must be of type projection. Reasoning as above, one gets that $E(f(u), f(v))$ iff $N(f(x_1), f(x_2))$.
(4) $Q_1Q_2Q_3 = \mathrm{ENE}$ or $Q_1Q_2Q_3 = \mathrm{EEN}$. These cases are analogous to the previous case.
(5) $Q_1Q_2Q_3 = \mathrm{EEE}$ or $Q_1Q_2Q_3 = \mathrm{NNN}$. These cases are trivial since f preserves E and N.

To show that f generates an operation of type majority or minority, by Proposition 28 it suffices to prove that f generates a function of type majority or minority on S, since S contains copies of products of arbitrary finite substructures of G. We show this by another case distinction, based on the fact that $(f(x_1), \ldots, f(x_6)) \notin H$.

(1) Suppose that $E(f(x_1), f(x_2)), E(f(x_3), f(x_4)), E(f(x_5), f(x_6))$. Then by the above, f itself is of type minority on S.
(2) Suppose that $N(f(x_1), f(x_2)), N(f(x_3), f(x_4)), N(f(x_5), f(x_6))$. Then f behaves like a majority on S.
(3) Suppose that $E(f(x_1), f(x_2)), E(f(x_3), f(x_4)), N(f(x_5), f(x_6))$. Let e be a self-embedding of G such that for all $w \in V$, all $1 \leq j \leq 3$, and all $1 \leq i \leq 6$ we have that $N(x_i^j, e(w))$. Then $(u_1, u_2, e(f(u_1, u_2, u_3))) \in S$ for all $(u_1, u_2, u_3) \in S$. Hence, by the above, the ternary operation defined by $f(x, y, e(f(x, y, z)))$ is of type majority on S.

(4) Suppose that $E(f(x_1), f(x_2)), N(f(x_3), f(x_4)), E(f(x_5), f(x_6))$, or $N(f(x_1), f(x_2))$, $E(f(x_3), f(x_4))$, $E(f(x_5), f(x_6))$. These cases are analogous to the previous case.

Let $h(x, y, z)$ be a ternary injection of type majority or minority generated by f; it remains to make h canonical. By Theorem 24, f generates a binary canonical injection $g(x, y)$, which is of type projection by our assumption on f. Set $t(x, y, z) := g(x, g(y, z))$. Then the function $h(t(x, y, z), t(y, z, x), t(z, x, y))$ is still of type majority or minority and canonical; we leave the straightforward verification to the reader. □

7.2.4. *Producing* max *and* min. Having proven Proposition 29, it is enough to show the following proposition in order to obtain a full proof of Proposition 25.

Proposition 30. *Let $f\colon V^2 \to V$ be a binary injection preserving E and N that is not of type projection. Then f generates a binary canonical injection of type* min *or of type* max.

In the remainder of this section we will prove this proposition by Ramsey theoretic analysis of f, which requires the following definitions and facts from [11].

Equip V with a total order $\prec$ in such a way that $(V; E, \prec)$ is the random ordered graph, i.e., the unique countably infinite homogeneous totally ordered graph containing all finite totally ordered graphs (for existence and uniqueness of this structure, see e.g. [22]). The order $(V; \prec)$ is then isomorphic to the order $(\mathbb{Q}; <)$ of the rationals. The ordered random graph has the advantage of being a so-called *Ramsey structure*, i.e., it enjoys a certain combinatorial property (which the random graph without the order does not) – see for example [10]. Using this Ramsey property, starting from a function on $(V; E, \prec)$ one can generate a canonical function whilst keeping such information as violation of a relation. Our combinatorial tool will be the following proposition, which has first been used in [11] in a slightly simpler form, and which has been stated in full generality for ordered homogeneous Ramsey structures in [12].

Proposition 31. *Let $f\colon V^k \to V$ be a function, and let $c^1, \ldots, c^m \in V^k$. Then f generates a function which is canonical as a function from $(V; E, \prec, c^1_1, \ldots, c^m_1) \times \cdots \times (V; E, \prec, c^1_k, \ldots, c^m_k)$ to $(V; E, \prec)$, and which is identical to f on $\{c^1_1, \ldots, c^m_1\} \times \cdots \times \{c^1_k, \ldots, c^m_k\}$. Moreover, if f is injective, then the generated canonical function can be chosen to be injective as well.*

The global strategy behind what follows now is to take a binary injection f and fix a finite number of constants $c^i \in V^2$ which witness that f is not of type projection. Then, using Proposition 31, we generate a binary canonical function which is identical to f on all c^i; this canonical function then still is not of type projection, and can be handled more easily as it is canonical. However, we do not present the proof like that for the reason that there would be too many possibilities of canonical functions for primitive case-by-case analysis. What we do instead is rule out behaviors of canonical functions more systematically, for example before even adding constants to the language. As in [11], let us define the following behaviors for functions from $(V; E, \prec)^2$ to $(V; E)$. We write $\succ$ for the relation $\{(a, b) \mid b \prec a\}$.

Definition 32. Let $f\colon V^2 \to V$ be injective. If for all $u, v \in V^2$ with $u_1 \prec v_1$ and $u_2 \prec v_2$ we have

- $E(f(u), f(v))$ if and only if $\mathrm{EE}(u, v)$, then we say that f *behaves like* min *on input* $(\prec, \prec)$.
- $N(f(u), f(v))$ if and only if $\mathrm{NN}(u, v)$, then we say that f *behaves like* max *on input* $(\prec, \prec)$.

- $E(f(u), f(v))$ if and only if $E(u_1, v_1)$, then we say that f *behaves like* p_1 *on input* $(\prec, \prec)$.
- $E(f(u), f(v))$ if and only if $E(u_2, v_2)$, then we say that f *behaves like* p_2 *on input* $(\prec, \prec)$.

Analogously, we define behavior on input $(\prec, \succ)$ using pairs $u, v \in V^2$ with $u_1 \prec v_1$ and $u_2 \succ v_2$.

Of course, we could also have defined "behavior on input $(\succ, \succ)$" and "behavior on input $(\succ, \prec)$"; however, behavior on input $(\succ, \succ)$ equals behavior on input $(\prec, \prec)$, and behavior on input $(\succ, \prec)$ equals behavior on input $(\prec, \succ)$ since graphs are symmetric. Thus, there are only two kinds of inputs to be considered, namely the "straight input" $(\prec, \prec)$ and the "twisted input" $(\prec, \succ)$.

Proposition 33. *Let $f \colon V^2 \to V$ be injective and canonical as a function from $(V; E, \prec)^2$ to $(V; E, \prec)$, and suppose it preserves E and N. Then it behaves like* min*,* max*, p_1 or p_2 on input $(\prec, \prec)$ (and similarly on input $(\prec, \succ)$).*

Proof. By definition of the term canonical; one only needs to enumerate all possible types of pairs (u, v), where $u, v \in V^2$. □

Definition 34. If an injection $f \colon V^2 \to V$ behaves like X on input $(\prec, \prec)$ and like Y on input $(\prec, \succ)$, where $X, Y \in \{\max, \min, p_1, p_2\}$, then we say that f is of *type* X/Y.

We would like to emphasize that the term "canonical" depends on the structures under consideration; that is, a function $f \colon V^2 \to V$ might be canonical as a function from $(V; E, \prec)^2$ to $(V; E, \prec)$, but not as a function from $(V; E)^2$ to $(V; E)$, and vice-versa. In the following, we will for this reason carefully specify the structures we have in mind when using this term.

Observe that canonical functions from $(V; E, \prec)^2$ to $(V; E, \prec)$ also behave regularly with respect to the order $\prec$: this implies, for example, that f is either strictly increasing or decreasing with respect to the pointwise order.

The structures $(V; E, \prec)$ and $(V; E, \succ)$ are isomorphic by the theory of homogeneous structures (see, e.g., [22]), since they are both homogeneous and embed the same finite structures. Fix an isomorphism α. Then α is an automorphism of G which reverses the order $\prec$. By applying α to a canonical function if necessary, we may (in the presence of $\mathrm{Aut}(G)$) always assume that all canonical functions f we use are strictly increasing. Having that, one easily checks that one of the implications

$$u_1 \prec v_1 \wedge u_2 \neq v_2 \to f(u) \prec f(v)$$

and

$$u_1 \neq v_1 \wedge u_2 \prec v_2 \to f(u) \prec f(v).$$

hold. In the first case, we say that f *obeys p_1 for the order*, in the second case f *obeys p_2 for the order*. By switching the variables of f, we can always achieve that f obeys p_1 for the order.

7.2.5. *Eliminating mixed behavior.* In the following lemmas, we show that when we have an injective canonical binary function which behaves differently on input $(\prec, \prec)$ and on input $(\prec, \succ)$, then it generates a function which behaves the same on both inputs.

Lemma 35. *Suppose that $f \colon V^2 \to V$ is injective and canonical as a function from $(V; E, \prec)^2$ to $(V; E, \prec)$, and suppose that it is of type* $\max/p_i$ *or of type* $p_i/\max$*, where $i \in \{1, 2\}$. Then f generates a binary injection of type* max.

Proof. Assume without loss of generality that f is of type $\max/p_i$, and note that we may assume that f obeys p_1 for the order. Set $h(x,y) := f(x,\alpha(y))$. Then h behaves like p_i on input $(\prec,\prec)$ and like max on input $(\prec,\succ)$; moreover, $h(u) \prec h(v)$ iff $f(u) \prec f(v)$, for all $u,v \in V^2$ with $u_1 \neq v_1$ and $u_2 \neq v_2$. We then have that $g(x,y) := f(f(x,y),h(x,y))$ is of type $\max/\max$, which means that it is of type max when viewed as a function from G^2 to G. □

Lemma 36. *Suppose that $f\colon V^2 \to V$ is injective and canonical as a function from $(V;E,\prec)^2$ to $(V;E,\prec)$, and suppose that it is of type* $\min/p_i$ *or of type* $p_i/\min$, *where $i \in \{1,2\}$. Then f generates a binary injection of type* min.

Proof. The dual proof works. □

Lemma 37. *Suppose that $f\colon V^2 \to V$ is injective and canonical as a function from $(V;E,\prec)^2$ to $(V;E,\prec)$, and suppose that it is of type* $\max/\min$ *or of type* $\min/\max$. *Then f generates a binary injection of type* max *(and by duality, a binary injection of type* min*).*

Proof. Assume without loss of generality that f is of type $\max/\min$, and remember that we may assume that f obeys p_1 for the order. Then $g(x,y) := f(x,f(x,y))$ is of type $\max/p_1$ and generates a binary injection of type max by Lemma 35. □

We next deal with the last remaining mixed behavior, p_1/p_2, by combining operational with relational arguments.

Lemma 38. *Let $\Gamma = (V;E,N,\ldots)$ be a reduct of G which is preserved by a binary injection of type p_1. Then the following are equivalent.*

(1) *Γ has a binary injective polymorphism of behavior* min.
(2) *For every primitive positive formula ϕ over Γ, if $\phi \wedge N(x_1,x_2) \wedge \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$ and $\phi \wedge N(x_3,x_4) \wedge \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$ are satisfiable over Γ, then $\phi \wedge N(x_1,x_2) \wedge N(x_3,x_4)$ is satisfiable over Γ as well.*
(3) *For every finite $F \subseteq V^2$ there exists a binary injective polymorphism of Γ which behaves like* min *on F.*

Proof. The implication from (1) to (2) follows directly by applying a binary injective polymorphism of behavior min to tuples r,s satisfying $\phi \wedge N(x_1,x_2) \wedge \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$ and $\phi \wedge N(x_3,x_4) \wedge \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$ respectively.

To prove that (2) implies (3), assume (2) and let $F \subseteq V^2$ be finite. Without loss of generality we can assume that F is of the form $\{e_1,\ldots,e_n\}^2$, for sufficiently large n. Let Δ be the structure induced by F in Γ^2. We construct an injective homomorphism h from Δ to Γ with the property that for all $u,v \in F$ with $\mathrm{EN}(u,v)$ or $\mathrm{NE}(u,v)$ we have $N(h(u),h(v))$. Any homomorphism from Δ to Γ, in particular h, can clearly be extended to a binary polymorphism of Γ, for example inductively by using the universality of G. Such an extension of h then behaves like min on F.

To construct h, consider the formula ϕ_0 with variables $x_{i,j}$ for $1 \leq i,j \leq n$ which is the conjunction over all literals $R(x_{i_1,j_1},\ldots,x_{i_k,j_k})$ such that R is a relation in Γ and $R(e_{i_1},\ldots,e_{i_k})$ and $R(e_{j_1},\ldots,e_{j_k})$ hold in Γ. So ϕ_0 states precisely which relations hold in Γ^2 on elements from F. Since Γ is preserved by a binary injection, we have that $\phi_1 := \phi_0 \wedge \bigwedge_{1\leq i,j,k,l\leq n,(i,j)\neq(k,l)} x_{i,j} \neq x_{k,l}$ is satisfiable in Γ.

Let P be the set of pairs of the form $((i_1,i_2),(j_1,j_2))$ with $i_1,i_2,j_1,j_2 \in \{1,\ldots,n\}$, $i_1 \neq j_1$, $i_2 \neq j_2$, and where $N(e_{i_1},e_{j_1})$ or $N(e_{i_2},e_{j_2})$. We show by induction on the size of $I \subseteq P$ that the formula $\phi_1 \wedge \bigwedge_{((i_1,i_2),(j_1,j_2))\in I} N(x_{i_1,i_2}, x_{j_1,j_2})$ is satisfiable over Γ. Note that this statement applied to the set $I = P$ gives us the a homomorphism h from Δ to Γ such that for all $a, b \in F$ we have $N(h(a), h(b))$ whenever $\mathrm{EN}(a, b)$ or $\mathrm{NE}(a, b)$ by setting $h(e_i, e_j) := s(x_{i,j})$, where s is the satisfying assignment for $\phi_1 \wedge \bigwedge_{((i_1,i_2),(j_1,j_2))\in P} N(x_{i_1,i_2}, x_{j_1,j_2})$.

For the induction beginning, let $p = ((i_1, i_2), (j_1, j_2))$ be any element of P. Let r, s be the n^2-tuples defined as follows.

$$r := (e_1, \ldots, e_1, e_2, \ldots, e_2, \ldots, e_n, \ldots, e_n)$$
$$s := (e_1, e_2, \ldots, e_n, e_1, e_2, \ldots, e_n, \ldots, e_1, e_2, \ldots, e_n)$$

In the following we use double indices for the entries of n^2-tuples; for example, $r = (r_{1,1}, \ldots, r_{1,n}, r_{2,1}, \ldots, r_{n,n})$. The two tuples r and s satisfy ϕ_0. To see this observe that by definition of ϕ_0 the tuple

$$((e_1, e_1), (e_1, e_2), \ldots, (e_1, e_n), (e_2, e_1), \ldots, (e_n, e_n))$$

satisfies ϕ_0 in Γ^2; since r and s are obtained by applying projections to that tuple onto the first and second coordinate, respectively, and projections are homomorphisms, r and s satisfy ϕ_0 as well. Let g be a binary injective polymorphism of Γ which is of type p_1, and set $r' := g(r, s)$ and $s' := g(s, r)$. Then r' and s' satisfy ϕ_1 since g is injective. Since $p \in P$, we have that $N(e_{i_1}, e_{j_1})$ or $N(e_{i_2}, e_{j_2})$. Assume that $N(e_{i_1}, e_{j_1})$; the other case is analogous. Since $r_{i_1,i_2} = e_{i_1}, r_{j_1,j_2} = e_{j_1}$, $r' := g(r, s)$, and g is of type p_1, we have that $N(r'_{i_1,i_2}, r'_{j_1,j_2})$, proving that $\phi_1 \wedge N(x_{i_1,i_2}, x_{j_1,j_2})$ is satisfiable in Γ.

In the induction step, let $I \subseteq P$ be a set of cardinality $n \geq 2$, and assume that the statement has been shown for subsets of P of cardinality $n - 1$. Pick any distinct $q_1, q_2 \in I$. Set

$$\psi := \phi_1 \wedge \bigwedge_{((i_1,i_2),(j_1,j_2))\in I\setminus\{q_1,q_2\}} N(x_{i_1,i_2}, x_{j_1,j_2})$$

and observe that ψ is a primitive positive formula over Γ since Γ contains E and N and since the binary relation $x \neq y$ can be defined in Γ by the primitive positive formula $\exists z.\ (E(x, z) \wedge N(y, z))$. Write $q_1 = ((u_1, u_2), (v_1, v_2))$ and $q_2 = ((u'_1, u'_2), (v'_1, v'_2))$. Then the inductive assumption shows that each of $\psi \wedge N(x_{u_1,u_2}, x_{v_1,v_2})$ and $\psi \wedge N(x_{u'_1,u'_2}, x_{v'_1,v'_2})$ is satisfiable in Γ. Note that ψ contains in particular conjuncts that state that the four variables $x_{u_1,u_2}, x_{v_1,v_2}, x_{u'_1,u'_2}, x_{v'_1,v'_2}$ denote distinct elements. Hence, by (2), the formula $\psi \wedge N(x_{u_1,u_2}, x_{v_1,v_2}) \wedge N(x_{u'_1,u'_2}, x_{v'_1,v'_2})$ is satisfiable over Γ as well, which is what we had to show.

The implication from (3) to (1) follows from Proposition 28. $\square$

Lemma 39. *Let $f : V^2 \to V$ be a binary injection of behavior p_1/p_2 which preserves E and N. Then f generates a binary injection of type* min *and a binary injection of type* max*.*

Proof. By Theorem 24, f generates a binary injection of type max, min, or p_1.

Suppose first that it does not generate a binary injection of type max or min; we will lead this to a contradiction. Let Γ be the reduct of G which has all relations that are first-order definable in G and preserved by f. Since f generates a binary injection of type p_1, we may apply implication (2) $\to$ (1) from Lemma 38. Let ϕ be a primitive positive formula with variable set S, $\{x_1, \ldots, x_4\} \subseteq S$, such that the formulas $\phi \wedge N(x_1, x_2) \wedge \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$ and $\phi \wedge N(x_3, x_4) \wedge \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$ have in Γ the satisfying assignments r and s from $S \to V$, respectively.

We can assume without loss of generality that $r(x_1) \prec r(x_2)$ and $r(x_3) \prec r(x_4)$; otherwise, since $r(x_1), \ldots, r(x_4)$ must be pairwise distinct, we can apply an automorphism of G to r such that the resulting map has the required property. Similarly, by applying an automorphism of G to s, we can assume without loss of generality that $s(x_1) \prec s(x_2)$ and $s(x_3) \succ s(x_4)$. Then the mapping $t \colon S \to V$ defined by $t(x) = f(r(x), s(x))$ shows that $\phi \wedge N(x_1, x_2) \wedge N(x_3, x_4)$ is satisfiable in Γ:

- The assignment t satisfies ϕ since f is a polymorphism of Γ.
- We have that $N(t(x_1), t(x_2))$ since $r(x_1) \prec r(x_2)$, $s(x_1) \prec s(x_2)$, f is of type p_1 on input $(\prec, \prec)$, and $N(r(x_1), r(x_2))$.
- We have that $N(t(x_3), t(x_4))$ since $r(x_3) \prec r(x_4)$, $s(x_3) \succ s(x_4)$, f is of type p_2 on input $(\prec, \succ)$, and $N(s(x_3), s(x_4))$.

By Lemma 38, we conclude that Γ is preserved by a binary injection of type min, and consequently f generates a binary injection of type min – a contradiction.

Therefore, f generates a binary injection of type max or min. Since the assumptions of the lemma are symmetric in E and N, we infer *a posteriori* that f generates both a binary injection of type max and a binary injection of type min. $\square$

7.2.6. *Behaviors relative to vertices.* Having ruled out some behaviors without constants, we now examine behaviors when we add constants to the language. In the sequel, we will also say that a function $f \colon V^2 \to V$ has behavior B *between two points* $u, v \in V^2$ if it has behavior B on the structure $\{u, v\}$.

Lemma 40. *Let $u \in V^2$, and set $U := (V \setminus \{u_1\}) \times (V \setminus \{u_2\})$. Let $f \colon V^2 \to V$ be a binary injection which preserves E and N, behaves like p_1 on U, and which behaves like p_2 between u and all points in U. Then f generates a binary injection of type* min *as well as a binary injection of type* max.

Proof. Let Γ be the reduct of G having all relations that are first-order definable in G and preserved by f. Since U contains copies of products of arbitrary finite graphs, f behaves like p_1 on arbitrarily large finite substructures of G^2, and hence generates a binary injection of type p_1 by Proposition 28. Hence Γ is also preserved by such a function, and we may apply the implication from (2) to (1) in Lemma 38 to Γ.

Let ϕ be a primitive positive formula with variable set S, $\{x_1, \ldots, x_4\} \subseteq S$, such that $\phi \wedge N(x_1, x_2) \wedge \bigwedge_{1 \leq i < j \leq 4} x_i \neq x_j$ and $\phi \wedge N(x_3, x_4) \wedge \bigwedge_{1 \leq i < j \leq 4} x_i \neq x_j$ are satisfiable over Γ, witnessed by satisfying assignments $r, s \colon S \to V$, respectively.

Let α be an automorphism of G that maps $r(x_3)$ to u_1, and let β be an automorphism of G that maps $s(x_3)$ to u_2. Then $(\alpha(r(x_3)), \beta(s(x_3))) = u$, and $v := (\alpha(r(x_4)), \beta(s(x_4))) \in U$ since $\alpha(r(x_4)) \neq \alpha(r(x_3)) = u_1$ and $\beta(s(x_4)) \neq \beta(s(x_3)) = u_2$. Thus, f behaves like p_2 between u and v, and since s satisfies $N(x_3, x_4)$, we have that $t \colon S \to V$ defined by

$$t(x) = f(\alpha(x), \beta(x))$$

satisfies $N(x_3, x_4)$, too. Since α, β, f are polymorphisms of Γ, the assignment t also satisfies ϕ. To see that t also satisfies $N(x_1, x_2)$, observe that $\alpha(r(x_1)) \neq \alpha(r(x_3))$ and $\beta(s(x_1)) \neq \beta(s(x_3))$, and hence $p := (\alpha(r(x_1)), \beta(s(x_1))) \notin U$. Similarly, $q := (\alpha(r(x_2)), \beta(s(x_2))) \notin U$. Hence, f behaves as p_1 between p and q, and since $N(r(x_1), r(x_2))$, so does t.

By Lemma 38 we conclude that Γ is preserved by a binary injection of type min, and consequently f generates a binary injection of type min.

Since our assumptions on f were symmetric in E and N, it follows that f also generates a binary injection of type max. $\square$

Lemma 41. *Let $u \in V^2$, and set $U := (V \setminus \{u_1\}) \times (V \setminus \{u_2\})$. Let $f \colon V^2 \to V$ be a binary injection which preserves E and N, behaves like p_1 on U, and which behaves like* min *between u and all points in U. Then f generates a binary injection of type* min.

Proof. The proof is identical with the proof in the preceding lemma; note that our assumptions on f here imply more deletions of edges than the assumptions in that lemma, so it can only be easier to generate a binary injection of type min. □

Lemma 42. *Let $u, v \in V^2$ such that $\neq\neq(u, v)$ and set $W := (V \setminus \{u_1, v_1\}) \times (V \setminus \{u_2, v_2\})$. Let $f \colon V^2 \to V$ be a binary injection that*

- *behaves like p_1 on W*
- *behaves like p_1 between any point in $\{u, v\}$ and any point in W*
- *does not behave like p_1 between u and v.*

Then f generates e_E, e_N, or a binary injection of type min *as well as binary injection of type* max.

Proof. Consider first the case where $\mathrm{EE}(u, v)$ and $N(f(u), f(v))$. Let $\alpha \in \mathrm{Aut}(G)$ send u_1 to u_2 and v_1 to v_2, and consider the function $h(x) := f(x, \alpha(x))$. Then $N(h(u_1), h(v_1))$, and h preserves E and N between any point in $\{u_1, v_1\}$ and all points in $V \setminus \{u_1, v_1\}$, and so it generates e_N by a standard iterative argument. Similarly, if $\mathrm{NN}(u, v)$ and $E(f(u), f(v))$ then f generates e_E.

It remains to consider the case where $\mathrm{EN}(u, v)$ and $N(f(u), f(v))$, and the case where $\mathrm{NE}(u, v)$ and $E(f(u), f(v))$. In the first case we prove that f generates a binary injection of type min; it then follows by duality that in the second case, f generates a binary injection of type max.

As in Lemma 40, we apply the implication (2) $\to$ (1) from Lemma 38. Let Γ, ϕ, S, $x_1, \ldots, x_4$, r, and s be as in the proof of Lemma 40; by the same argument as before, Γ is preserved by a binary injection of type p_1.

If $N(r(x_3), r(x_4))$, then the assignment r shows that $\phi \wedge N(x_1, x_2) \wedge N(x_3, x_4)$ is satisfiable and we are done. Otherwise, since $r(x_3) \neq r(x_4)$, we have $E(r(x_3), r(x_4))$. Therefore, there is an $\alpha \in \mathrm{Aut}(G)$ such that $(\alpha(r(x_3)), \alpha(r(x_4))) = (u_1, v_1)$. Similarly, since $N(s(x_3), s(x_4))$ and $N(u_2, v_2)$, there is a $\beta \in \mathrm{Aut}(G)$ such that $(\beta(s(x_3)), \beta(s(x_4))) = (u_2, v_2)$. We claim that the map $t \colon S \to V$ defined by

$$t(x) = f(\alpha(x), \beta(x))$$

is a satisfying assignment for $\phi \wedge N(x_1, x_2) \wedge N(x_3, x_4)$. The assignment t satisfies ϕ since α, β and f are polymorphisms of Γ. Then $N(t(x_3), t(x_4))$ holds because $(\alpha(r(x_3)), \beta(s(x_3))) = u$ and $(\alpha(r(x_4)), \beta(s(x_4))) = v$, and $N(f(u), f(v))$. To prove that $N(t(x_1), t(x_2))$ holds, observe that $r(x_1) \neq r(x_3)$ and $r(x_1) \neq r(x_4)$, and hence $\alpha(r(x_1)) \notin \{\alpha(r(x_3)), \alpha(r(x_4))\} = \{u_1, v_1\}$. Similarly, $\beta(s(x_1)) \notin \{\beta(s(x_3)), \beta(s(x_4))\} = \{u_2, v_2\}$. Hence, $(\alpha(r(x_1)), \beta(s(x_1)) \in W$. A similar argument for x_2 in place of x_1 shows that $(\alpha(r(x_2)), \beta(s(x_2)) \in W$. Since f behaves like p_1 on W, and since r satisfies $N(x_1, x_2)$, we have proved the claim. This shows that Γ is preserved by a binary injection of type min, and hence f generates such a function.

By symmetry of our assumptions on f in E and N, it follows that f generates a binary injection of type min if and only if it generates a binary injection of type max. □

We are now set up to prove Proposition 30. This completes the proof of Proposition 25, and in turn the proof of Theorem 20.

Proof of Proposition 30. Let f be given. By Theorem 24, f generates a binary canonical injection g of type projection, min, or max. In the last two cases we are done, so consider the first case. We claim that f also generates a (not necessarily canonical) binary injection h of type min or max. Then $h(g(x,y),g(y,x))$ is still of type min or max and in addition canonical, and the proposition follows.

To prove our claim, fix a finite set $C := \{c_1,\ldots,c_m\} \subseteq V$ such that the fact that f does not behave like a projection is witnessed on C. Invoking Proposition 31, we may henceforth assume that f is canonical as a function from $(V;E,\prec,c_1,\ldots,c_m)^2$ to $(V;E,\prec)$ (and hence also to $(V;E)$ since tuples of equal type in $(V;E,\prec)$ have equal type in $(V;E)$). It is clear that this new f must be injective.

In the following we will consider orbits of elements in the structure $(V;E,\prec,c_1,\ldots,c_m)$. The infinite orbits are precisely the sets of the form

$$\{v \in V \mid Q_i(v,c_i) \text{ and } R_i(v,c_i) \text{ for all } 1 \leq i \leq m\},$$

for $Q_1,\ldots,Q_m \in \{E,N\}$, and $R_1,\ldots,R_m \in \{\prec,\succ\}$. The finite orbits are of the form $\{c_i\}$ for some $1 \leq i \leq m$. It is well-known that each infinite orbit of $(V;E,\prec,c_1,\ldots,c_m)$ contains copies of arbitrary linearly ordered finite graphs, and in particular, forgetting about the order, of all finite graphs. Therefore, if f behaves like min or max on an infinite orbit of $(V;E,\prec,c_1,\ldots,c_m)$, then by Proposition 28 it generates a function which behaves like min or max everywhere, and we are done.

Moreover, if f is of mixed type on an infinite orbit, then, by Proposition 28, f generates a canonical function which has the same mixed behavior everywhere. But then we are done by Lemmas 35, 36, 37, and 39. Hence, we may assume that f behaves like a projection on every infinite orbit. Fix in the following an infinite orbit O and assume without loss of generality that f behaves like p_1 on O.

Let W be any infinite orbit. Then since f is canonical, it behaves like p_1, p_2, min, or max between all u,v with $u \in O^2$, $v \in W^2$ and $u_1 \prec v_1$ and $u_2 \prec v_2$. Consider the case where there exists an infinite orbit W such that f behaves like p_2 between all points $u \in O^2$ and $v \in W^2$ for which $u_1 \prec v_1$ and $u_2 \prec v_2$. Then fix any $v \in W^2$, and set $O_1 := \{o \in O \mid o \prec v_1\}$ and $O_2 := \{o \in O \mid o \prec v_2\}$. Set $O_1' := O_1 \cup \{v_1\}$ and $O_2' := O_2 \cup \{v_2\}$. We then have that f behaves like p_2 between v and any point u of $(O_1' \setminus \{v_1\}) \times (O_2' \setminus \{v_2\})$, and like p_1 between any two points of $(O_1' \setminus \{v_1\}) \times (O_2' \setminus \{v_2\})$. Since $(O_i';E,v_i)$ contains copies of all finite substructures of $(V;E,v_i)$, for $i \in \{1,2\}$, by Proposition 28 we get that f generates a function which behaves like p_2 between v and any point u of $(V \setminus \{v_1\}) \times (V \setminus \{v_2\})$, and which behaves like p_1 between any two points of $(V \setminus \{v_1\}) \times (V \setminus \{v_2\})$. Then Lemma 40 implies that f generates a binary injection of type min and we are done.

This argument is easily adapted to any situation where there exists an infinite orbit W such that f behaves like p_2 between all points $u \in O^2$ and $v \in W^2$ with $R_1(u_1,v_1)$ and $R_2(u_2,v_2)$, for $R_1,R_2 \in \{\prec,\succ\}$.

When there exists an infinite orbit W such that f behaves like min between all points $u \in O^2$ and $v \in W^2$ with $R_1(u_1,v_1)$ and $R_2(u_2,v_2)$, then we can argue similarly, invoking Lemma 41 at the end. Replacing min by max we can use the dual argument, with the notable difference that f generates a binary injection of type max rather than min.

Since f is canonical, one of the situations described so far must occur. Putting this together, we conclude that for every infinite orbit W and all points $u \in O^2$ and $v \in W^2$, f behaves like p_1 between u and v. Having that, suppose that for an infinite orbit W, f behaves like p_2 on W. Then exchanging the roles of O and W and of p_1 and p_2 above, we can again conclude that f generates a binary injection of type min. We may thus henceforth assume that f behaves like p_1 on $(V \setminus C)^2$.

Pick any $u \in C^2$. Suppose that there exists $v \in (V \setminus C)^2$ such that f does not behave like p_1 between u and v; say without loss of generality that $\mathrm{EN}(u, v)$ and $N(f(u), f(v))$. Let O_i be the (infinite) orbit of v_i, for $i \in \{1, 2\}$. Then for all $v \in O_1 \times O_2$ we have $\mathrm{EN}(u, v)$ and $N(f(u), f(v))$ since f is canonical. Now let $w \in O_2 \times O_1$. We distinguish the two cases $E(f(u), f(w))$ and $N(f(u), f(w))$. In the first case, f behaves like p_2 between u and all $v \in (O_1 \cup O_2)^2$. We can then argue as above and are done. In the second case, f behaves like min between u and all $v \in (O_1 \cup O_2)^2$, and we are again done by the corresponding argument above. We conclude that we may assume that for all $u \in C^2$ and all $v \in (V \setminus C)^2$, f behaves like p_1 between u and v as well.

Now pick $u, v \in C^2$ such that f does not behave like p_1 between u and v, say without loss of generality $\mathrm{EN}(u, v)$ and $N(f(u), f(v))$; this is possible since the fact that f does not behave like p_1 everywhere is witnessed on C. Set $W_i := (V \setminus C) \cup \{u_i, v_i\}$ for $i = \{1, 2\}$. Since W_1 and W_2 induce a structure isomorphic to the random graph in G, and f behaves like p_2 between u and v, and like p_1 between all points in $\{u, v\}$ and all points $(W_1 \setminus \{u_1, v_1\}) \times (W_2 \setminus \{u_2, v_2\})$, we are done by Lemma 42. $\square$

7.3. **When the endomorphisms of a reduct are generated by $\{-\}$.** We next consider Case (c) of Proposition 8. That is, we will assume that the endomorphisms of Γ are exactly the functions generated by $\{-\}$. In particular, $\mathrm{Aut}(\Gamma)$ contains $-$ but not sw, and the automorphisms of Γ generate its endomorphisms.

Definition 43. Let H_1' be the smallest 6-ary relation that is preserved by $\{-\}$ and contains H_1.

The following is an analog of Theorem 20 for the situation of this section.

Theorem 44. *Let Γ be a reduct of G whose endomorphisms are precisely the unary functions generated by $\{-\}$. Then either H_1' is primitive positive definable in Γ, or one of the cases (b)-(e) of Theorem 20 applies.*

Proof. Note that H_1' consists of three orbits of 6-tuples in $\mathrm{Aut}(\Gamma)$, and hence, if H_1' is not primitive positive definable in Γ, then there exists by Theorem 4 and Lemma 7 a ternary polymorphism f of Γ that violates H_1'. That is, there are $t^1, t^2, t^3 \in H_1'$ such that $f(t^1, t^2, t^3) \notin H_1'$. Note that for each t^j, either t^j or $-t^j \in H_1$. In the first case we set g_j to be the identity function on V, in the second case we let g_j be the operation $-$. Now consider the function f' defined by $f'(x_1, x_2, x_3) := f(g_1(x_1), g_2(x_2), g_3(x_3))$. We have that $s^j := g_j^{-1}(t^j) \in H_1$, but $f'(s^1, s^2, s^3) = f(t^1, t^2, t^3)$ is not in H_1'. Consider the function $h(x) := f'(x, x, x)$; since the endomorphisms of Γ are generated by $\{-\}$, h either preserves E and N, or it flips them. By replacing f' by $-(f')$ in the latter case we may assume that h preserves E and N. Note that we still have that $f'(s^1, s^2, s^3)$ is not in H_1', and therefore not in H_1 either. Hence, f' violates H_1.

Now suppose that f' violates E or N; we will derive a contradiction. Say without loss of generality that there are $u, v \in V^3$ with $\mathrm{EEE}(u, v)$ such that $E(f'(u), f'(v))$ does not hold. Pick distinct $a, b, c, d \in V$ such that $\{a, b, c, d\}$ induces a clique in G, and such that each element is connected to all entries on u, v by an edge. Pick then $\alpha_1, \alpha_2, \alpha_3 \in \mathrm{Aut}(G)$ such that $\alpha_i(a) = u_i$ and $\alpha_i(b) = v_i$ for all $i \in \{1, 2, 3\}$, and such that $\alpha_1(c) = \alpha_2(c) = \alpha_3(c) = c$ and $\alpha_1(d) = \alpha_2(d) = \alpha_3(d) = d$. We then have that the function $x \mapsto f'(\alpha_1(x), \alpha_2(x), \alpha_3(x))$

maps (c,d) to an edge since $h(x)$ preserves E, but it does not map (a,b) to an edge, by our assumption on u and v. This is, however, impossible, since the function must be generated by $\{-\}$.

Therefore, f' preserves E and N. Then Theorem 20 implies that f' generates functions with the desired properties, or a binary function of type max or min. A function of type max together with $\{-\}$ generates a function of type min, and vice versa. Then $\max(\min(x,y),\min(y,z),\min(x,z))$ is a ternary function of type majority with the desired properties, and we are also done in this case. □

Proposition 45. $\mathrm{CSP}(V; H_1')$ *is NP-hard.*

Proof. One can show NP-hardness similarly as in the proof of Proposition 21, by reduction from positive Not-all-three-equal-3SAT instead of positive 1-in-3-3SAT. □

7.4. **When the endomorphisms of a reduct are generated by** $\{\mathrm{sw}\}$**.** In this section we will prove Theorem 48 below, which treats Case (d) in Proposition 8.

Definition 46. For $k \geq 1$, let $S^{(k)}$ be 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 number of edges between these k vertices is even.

Recall also the definition of $R^{(k)}$ from Section 6. The structure of this section will be similar to the one of Section 7.2, but $R^{(3)}$ will take the role of E, and $S^{(3)}$ will take the role of N. The relation H_1 will be replaced by the following relation.

Definition 47. Let H_2 be the smallest 9-ary relation that is preserved by $\{\mathrm{sw}\}$ and contains all tuples $(x_1,y_1,z_1,x_2,y_2,z_2,x_3,y_3,z_3) \in V^9$ such that

$$\begin{aligned} &\bigwedge_{i,j\in\{1,2,3\},i\neq j,u\in\{x_i,y_i,z_i\},v\in\{x_j,y_j,z_j\}} N(u,v) \\ &\wedge \big((R^{(3)}(x_1,y_1,z_1) \wedge S^{(3)}(x_2,y_2,z_2) \wedge S^{(3)}(x_3,y_3,z_3)) \\ &\quad \vee\ (S^{(3)}(x_1,y_1,z_1) \wedge R^{(3)}(x_2,y_2,z_2) \wedge S^{(3)}(x_3,y_3,z_3)) \\ &\quad \vee\ (S^{(3)}(x_1,y_1,z_1) \wedge S^{(3)}(x_2,y_2,z_2) \wedge R^{(3)}(x_3,y_3,z_3))\big)\ . \end{aligned}$$

Theorem 48. *Let* Γ *be a reduct of* G *whose endomorphisms are precisely the unary functions generated by* $\{\mathrm{sw}\}$*. Then either* H_2 *is primitive positive definable in* Γ*, or* Γ *satisfies item (b) or (d) of Theorem 20.*

Proposition 49. $\mathrm{CSP}(V; H_2)$ *is NP-hard.*

Proof. This can be shown analogously to Proposition 21 by reduction from 1-in-3-3SAT, but this time we represent 1 by triples from $R^{(3)}$ instead of pairs that satisfy E, and 0 by triples from $S^{(3)}$, and then use H_2 analogously as we have used H_1 in the proof of Proposition 21. □

7.4.1. *Producing canonical functions of type projection.* We use a combination of Lemma 5.3 in [5] with Lemma 42 in [11]. Those lemmas are stated below for the convenience of the reader.

Lemma 50 (Lemma 5.3 in [5])**.** *Let* Γ *be a relational structure over an infinite domain* D *such that the set of primitive positive definable binary relations in* Γ *is exactly* $\{D^2, \neq, =, \emptyset\}$*. Suppose that* Γ *contains an* n*-ary relation* Q *such that there are pairwise distinct* $1 \leq i,j,k,l \leq n$ *for which the following conditions hold:*

(1) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_j$ *is satisfiable;*
(2) $Q(x_1, \ldots, x_n) \wedge x_k \neq x_l$ *is satisfiable;*
(3) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_j \wedge x_k \neq x_l$ *is unsatisfiable.*

Then the relation $S_D := \{(x, y, z) \in D^3 \mid y \neq z \wedge (x = y \vee x = z)\}$ *has a primitive positive definition in* Γ.

Lemma 51 (Lemma 42 in [11])**.** *Let* Γ *be a countable* ω*-categorical structure in which* $\neq$ *is primitive positive definable. Then the following are equivalent.*

(1) *If* ϕ *is a primitive positive formula such that both* $\phi \wedge x \neq y$ *and* $\phi \wedge u \neq v$ *are satisfiable over* Γ*, then* $\phi \wedge x \neq y \wedge u \neq v$ *is satisfiable over* Γ *as well.*
(2) Γ *is preserved by a binary injective operation.*

We use the following combination of these two lemmata.

Proposition 52. *Let* Γ *be an* ω*-categorical structure with a 2-transitive automorphism group (i.e., for which the relation* $\neq$ *equals one orbit of pairs). Then one of the following applies.*

(1) *All polymorphisms of* Γ *are essentially unary.*
(2) Γ *has a constant endomorphism.*
(3) Γ *has a binary injective endomorphism.*

Proof. Write D for the domain of Γ. If Γ has a non-injective endomorphism, then a straightforward iterative argument using the 2-transitivity of $\mathrm{Aut}(\Gamma)$ and local closure shows that Γ also has a constant endomorphism and there is nothing to show. Otherwise, since $\neq$ only consists of one orbit of pairs, it is preserved by all polymorphisms and hence primitive positive definable by Theorem 4. By the 2-transitivity of $\mathrm{Aut}(\Gamma)$ it is now clear that the set of primitive positive definable binary relations in Γ is exactly $\{D^2, \neq, =, \emptyset\}$. Hence, by Lemma 50 one of the following holds:

- the first item of Lemma 51 applies, and hence by Lemma 51 the structure Γ has a binary injective polymorphism;
- there is a formula which is a counterexample to first item of Lemma 51. In that case, the expansion of Γ by the relation defined by this formula satisfies the hypotheses of Lemma 50, and hence the relation S_D is primitive positive definable in Γ. It then follows that all polymorphisms of Γ are essentially unary (this can be shown as in Proposition 5.3.2 in [2]).

□

Proposition 53. *Let* Γ *be a reduct of* G *with an essential polymorphism. Then* Γ *is preserved by a constant function,* e_E*,* e_N*, or by a canonical binary injection of type* min, max*, or* p_1*.*

Proof. If there is a primitive positive definition of E and N, then the statement follows from Theorem 24. So suppose that this is not that case; also suppose that Γ is not preserved by e_E, e_N, or a constant function. Then the automorphisms of Γ generate its endomorphisms by Theorem 9, and so they must violate E and N as otherwise these relations would have a primitive positive definition. By Theorem 12, we then see that $\mathrm{Aut}(\Gamma)$ is 2-transitive. By Proposition 52, Γ has a binary injective polymorphism g. By Proposition 31, g generates a binary function h which is canonical as a function from $(V; E, \prec)^2$ to $(V; E, \prec)$; this function is again injective. The function $x \mapsto h(x, x)$ either preserves E and N, or behaves like $-$, e_E or e_N. We can assume that it does not behave like e_E or e_N, and if it behaves like $-$, we can replace h by $-h$ and assume that $x \mapsto h(x, x)$ preserves E and N. Now consider the function

$x \mapsto h(x, \alpha(x))$, where $\alpha \in \mathrm{Aut}(G)$ reverses $\prec$. Again, we may exclude the possibility that it behaves like e_E or e_N. But then the function $(x, y) \mapsto h(h(x, y), h(y, x))$ preserves E and N and we can apply Theorem 24 to conclude that it generates a binary injection which is canonical as a function from G^2 to G and of type min, max, or p_1. □

Corollary 54. *Let $\Gamma = (V; R^{(3)}, S^{(3)}, \dots)$ be a reduct of G with an essential polymorphism. Then Γ is preserved by a binary canonical injection of type p_1.*

Proof. Since e_N and functions of type min do not preserve $R^{(3)}$ and e_E and functions of type max do not preserve $S^{(3)}$, Proposition 53 implies that Γ is preserved by a binary canonical injection of type p_1. □

7.4.2. *Eliminating mixed behavior.*

Lemma 55. *Let $f: V^2 \to V$ be a binary injection that preserves $R^{(3)}$ and $S^{(3)}$. Then f is not of type p_1/p_2.*

Proof. Suppose for contradiction that f does have the behavior p_1/p_2. Let $u_1, u_2, u_3 \in V$ with $u_1 \prec u_2 \prec u_3$, $E(u_1, u_2)$, $N(u_2, u_3)$, and $N(u_1, u_3)$. Let $v_1, v_2, v_3 \in V$ with $v_1 \prec v_2 \prec v_3$ and $N(v_1, v_2), E(v_2, v_3), N(v_1, v_3)$. Then $E(f(u_1, v_1), f(u_2, v_3))$ and $N(f(u_1, v_1), f(u_3, v_2))$ since f behaves like p_1 on input $(\prec, \prec)$. Moreover, $E(f(u_2, v_3), f(u_3, v_2))$ since f behaves like p_2 on input $(\prec, \succ)$. Then $(u_1, u_2, u_3) \in R^{(3)}$ and $(u_1, u_2, u_3) \in R^{(3)}$, but $(f(u_1, v_2), f(u_2, v_3), f(u_3, v_2)) \notin R^{(3)}$, in contradiction to our assumptions. □

7.4.3. *Behaviors relative to vertices.*

Lemma 56. *Let $u \in V^2$, and set $U := (V \setminus \{u_1\}) \times (V \setminus \{u_2\})$. Let $f: V^2 \to V$ be a binary injection which behaves like p_1 on U, and which behaves like p_2 or* max *between u and all points in U. Then f does not preserve $R^{(3)}$.*

Proof. Let $v, w \in U$ be such that $\mathrm{NE}(u, v)$, $\mathrm{EN}(v, w)$, and $\mathrm{NN}(u, w)$. Then we have $E(f(u), f(v))$, $E(f(v), f(w))$, and $N(f(u), f(w))$. Hence, $R^{(3)}(u_i, v_i, w)$ for $i \in \{1, 2\}$, but $S^{(3)}(f(u), f(v), f(w))$. □

Definition 57. We say that a binary injective function $f: V^2 \to V$ is

- of type $R^{(3)}$-p_i, for $i \in \{1, 2\}$, iff for all $u, v, w \in V^2$ with $\neq\neq(u, v)$, $\neq\neq(v, w)$, and $\neq\neq(u, w)$ we have $R^{(3)}(f(u), f(v), f(w))$ if and only if $R^{(3)}(u_i, v_i, w_i)$.
- of type $R^{(3)}$-projection iff it is of type $R^{(3)}$-p_1 or of type $R^{(3)}$-p_2.

Proposition 58. *Let $f: V^2 \to V$ be a binary injective polymorphism of $(V; R^{(3)}, S^{(3)})$. Then f is of type $R^{(3)}$-projection.*

Proof. The proof is similar to the proof of Proposition 30. Fix a finite set $C := \{c_1, \dots, c_m\} \subseteq V$ such that the fact that f is not of type $R^{(3)}$-projection is witnessed on C. Invoking Proposition 31, we may henceforth assume that f is canonical as a function from $(V; E, \prec, c_1, \dots, c_m)^2$ to $(V; E, \prec)$.

In the following we will consider orbits of elements in the structure $(V; E, \prec, c_1, \dots, c_m)$. The infinite orbits are precisely the sets of the form

$$\{v \in V \mid Q_i(v, c_i) \text{ and } R_i(v, c_i) \text{ for all } 1 \leq i \leq m\},$$

for $Q_1, \dots, Q_m \in \{E, N\}$, and $R_1, \dots, R_m \in \{\prec, \succ\}$. The finite orbits are of the form $\{c_i\}$ for some $1 \leq i \leq m$. Each infinite orbit of $(V; E, \prec, c_1, \dots, c_m)$ is isomorphic to $(V; E, \prec)$.

Therefore Proposition 28 implies that if f has a certain behavior on such an infinite orbit, then it generates a canonical function which has the same behavior everywhere. Therefore we have for all infinite orbits O that f

- cannot be of type min or max on O since it preserves $R^{(3)}$ and $S^{(3)}$;
- cannot have behavior $\max/p_i$ or $p_i/\max$ for $i \in \{1,2\}$ on O, by Lemma 35;
- cannot have behavior $\min/p_i$ or $p_i/\min$ for $i \in \{1,2\}$ on O, by 36;
- it cannot have behavior $\max/\min$ or $\min/\max$ on O, by Lemma 37;
- it cannot have behavior p_1/p_2 or p_2/p_1 on O, by Lemma 55.

Hence, we may assume that f behaves like a projection on every infinite orbit. Fix in the following an infinite orbit O and assume without loss of generality that f behaves like p_1 on O.

Let W be any infinite orbit. Then since f is canonical, it behaves like p_1, p_2, min, or max between all u, v with $u \in O^2$, $v \in W^2$ and $u_1 \prec v_1$ and $u_2 \prec v_2$. Consider the case where there exists an infinite orbit W such that f behaves like p_2 or max between all points $u \in O^2$ and $v \in W^2$ for which $u_1 \prec v_1$ and $u_2 \prec v_2$. Then fix any $v \in W^2$, and set $O_1 := \{o \in O \mid o \prec v_1\}$ and $O_2 := \{o \in O \mid o \prec v_2\}$. Set $O_1' := O_1 \cup \{v_1\}$ and $O_2' := O_2 \cup \{v_2\}$. We then have that f behaves like p_2 or max between v and any point u of $(O_1' \setminus \{v_1\}) \times (O_2' \setminus \{v_2\})$, and like p_1 between any two points of $(O_1' \setminus \{v_1\}) \times (O_2' \setminus \{v_2\})$. Since $(O_i'; E, v_i)$ is isomorphic to $(V; E, v_i)$, for $i \in \{1,2\}$, by Proposition 28 we get that f generates a function which behaves like p_2 or max between v and any point u of $(V \setminus \{v_1\}) \times (V \setminus \{v_2\})$, and which behaves like p_1 between any two points of $(V \setminus \{v_1\}) \times (V \setminus \{v_2\})$. This is impossible by Lemma 56. This argument is easily adapted to any situation where there exists an infinite orbit W such that f behaves like p_2 between all points $u \in O^2$ and $v \in W^2$ with $R_1(u_1, v_1)$ and $R_2(u_2, v_2)$, for $R_1, R_2 \in \{\prec, \succ\}$. When there exists an infinite orbit W such that f behaves like min between all points $u \in O^2$ and $v \in W^2$ with $R_1(u_1, v_1)$ and $R_2(u_2, v_2)$, then we can argue similarly.

Since f is canonical, one of the situations described so far must occur. Putting this together, we conclude that for every infinite orbit W and all points $u \in O^2$ and $v \in W^2$, f behaves like p_1 between u and v. Having that, suppose that for an infinite orbit W, f behaves like p_2 on W. Then exchanging the roles of O and W and of p_1 and p_2 above, we again arrive at a contradiction. We may thus henceforth assume that f behaves like p_1 on $(V \setminus C)^2$.

Pick any $u \in C^2$. Suppose that there exists $v \in (V \setminus C)^2$ such that f does not behave like p_1 between u and v. Assume first that $\mathrm{EN}(u, v)$ and $N(f(u), f(v))$. Let O_i be the (infinite) orbit of v_i, for $i \in \{1,2\}$. Then for all $v \in O_1 \times O_2$ we have $\mathrm{EN}(u, v)$ and $N(f(u), f(v))$ since f is canonical. Now let $w \in O_2 \times O_1$. We distinguish the two cases $E(f(u), f(w))$ and $N(f(u), f(w))$. In the first case, f behaves like p_2 between u and all $v \in (O_1 \cup O_2)^2$. We can then argue as above and are done. In the second case, f behaves like min between u and all $v \in (O_1 \cup O_2)^2$, and we are again done by the corresponding argument above. The dual argument works when $\mathrm{NE}(u, v)$ and $E(f(u), f(v))$. Now assume that $\mathrm{EE}(u, v)$ and $N(f(u), f(v))$. We claim that $\mathrm{EE}(u, v')$ implies $N(f(u), f(v'))$ and $\mathrm{NN}(u, v')$ implies $E(f(u), f(v'))$ for all $v' \in (V \setminus C)^2$. Suppose that $v' \in (V \setminus C)^2$ is a counterexample. We can find $v'' \in (V \setminus C)^2$ such that v_1', v_1'' and v_2', v_2'' belong to the same orbit and such that $R^{(3)}(u_i, v_i, v_i'')$ for $i \in \{1,2\}$. But then $S^{(3)}(f(u), f(v), f(v''))$, a contradiction. By applying a version of sw which switches edges and non-edges with respect to $f[C^2]$ to f from the left, we may assume that f behaves like p_1 between all $u \in C^2$ and all $v \in (V \setminus C)^2$

Since f does not behave like $R^{(3)}$-p_1 on C^2, in particular it does not behave like p_1 on C^2. Pick $u, v \in C^2$ witnessing this. Then f behaves like p_1 between any point in $\{u, v\}$ and any point in $(V \setminus C)^2$. Since $(V \setminus C) \cup \{u_i, v_i\}$ induces an isomorphic copy of the random graph for $i \in \{1,2\}$, we can refer to Lemma 42 to arrive at a contradiction: f generates e_E, e_N, or a binary injection of type min or max, all of which violate either $R^{(3)}$ or $S^{(3)}$. □

Definition 59. We say that a ternary injective function $f \colon V^3 \to V$ is

- *of type $R^{(3)}$-majority* iff for all $u, v, w \in V^3$ with $\neq\neq\neq(u,v)$, $\neq\neq\neq(u,w)$, $\neq\neq\neq(v,w)$ we have $R^{(3)}(f(u), f(v), f(w))$ if and only if $R^{(3)}R^{(3)}R^{(3)}(u,v,w)$, $R^{(3)}R^{(3)}S^{(3)}(u,v,w)$, $R^{(3)}S^{(3)}R^{(3)}(u,v,w)$, or $S^{(3)}R^{(3)}R^{(3)}(u,v,w)$.
- *of type $R^{(3)}$-minority* iff for all $u, v, w \in V^3$ with $\neq\neq\neq(u,v)$, $\neq\neq\neq(u,w)$, $\neq\neq\neq(v,w)$ we have $R^{(3)}(f(u), f(v), f(w))$ if and only if $R^{(3)}R^{(3)}R^{(3)}(u,v,w)$, $R^{(3)}S^{(3)}S^{(3)}(u,v,w)$, $S^{(3)}R^{(3)}S^{(3)}(u,v,w)$, or $S^{(3)}S^{(3)}R^{(3)}(u,v,w)$.

Lemma 60. *A function $f \colon V^3 \to V$ of type $R^{(3)}$-majority does not preserve $R^{(3)}$.*

Proof. Let $u^1, u^2, u^3 \in V^4$ be such that

- $E(u^1_1, u^1_2)$ and $N(u^1_i, u^1_j)$ for all pairs (i,j) of distinct elements from $\{1, \ldots, 4\}$ that are distinct from $(1,2)$.
- $E(u^2_2, u^2_3)$ and $N(u^1_i, u^1_j)$ for all pairs (i,j) of distinct elements from $\{1, \ldots, 4\}$ that are distinct from $(2,3)$.
- $E(u^3_1, u^3_3)$ and $N(u^3_i, u^3_j)$ for all pairs (i,j) of distinct elements from $\{1, \ldots, 4\}$ that are distinct from $(1,3)$.

Since f is of type $R^{(3)}$-majority, we have $S^{(3)}(f(u_1), f(u_2), f(u_4))$, $S^{(3)}(f(u_1), f(u_3), f(u_4))$, and $S^{(3)}(f(u_2), f(u_3), f(u_4))$. Since for all four-element subsets of V there must always be an even number of three-element subsets in $R^{(3)}$, we then have $S^{(3)}(f(x_1), f(x_2), f(x_3))$, and hence f does not preserve $R^{(3)}$. □

Lemma 61. *Let $f \colon V^3 \to V$ be of type $R^{(3)}$-minority. Then $\{f, \mathrm{sw}\}$ generates a function of type minority.*

Proof. Let g be any ternary injection of type minority, and let $u, v, w \in V^3$ with $\neq\neq\neq(u,v), \neq\neq\neq(u,w), \neq\neq\neq($ be given. We will show that $R^{(3)}(g(u), g(v), g(w))$ if and only if $R^{(3)}(f(u), f(v), f(w))$. Recall that $R^{(3)}(f(u), f(v), f(w))$ if and only if $R^{(3)}S^{(3)}S^{(3)}(u,v,w)$, $S^{(3)}R^{(3)}S^{(3)}(u,v,w)$, $S^{(3)}S^{(3)}R^{(3)}(u,v,w)$, or $R^{(3)}R^{(3)}R^{(3)}(u,v,w)$. This is in turn the case if and only if the cardinality of the set

$$E \cap \bigcup_{i \in \{1,2,3\}} \{(u_i, v_i), (u_i, w_i), (v_i, w_i)\}$$

is odd. This in turn is the case if and only if $E \cap \{(g(u), g(v)), (g(u), g(w)), (g(v), g(w))\}$ is odd, which is the case if and only if $R^{(3)}(g(u), g(v), g(w))$ holds.

By Corollary 54, f generates a binary canonical injection $s(x,y)$ of type p_1. Set $t(x,y,z) := s(x, s(y,z))$. As in the proof of Proposition 29 the function $p(x,y,z) := f(t(x,y,z), t(y,z,x), t(z,x,y))$ is still of type $R^{(3)}$-minority, and the function $q(x,y,z) := g(t(x,y,z), t(y,z,x), t(z,x,y))$ is still of type minority. Moreover, by the above we have $R^{(3)}(p(u), p(v), p(w))$ if and only if $R^{(3)}(q(u), q(v), q(w))$ for all $u, v, w \in V^3$, since t is injective. Therefore, the homogeneity of $(V; R^{(3)})$ implies that for all finite $S \subseteq V^3$ there exists a unary operation a generated by $\{\mathrm{sw}\}$ such that the ternary function $a(p(x,y,z))$ agrees with $q(x,y,z)$ on S. By local closure, q is thus generated by $\{f, \mathrm{sw}\}$. □

Lemma 62. *Let $\Gamma = (V; R^{(3)}, S^{(3)}, \ldots)$ be a reduct of G such that H_2 is not primitive positive definable. Then Γ has a ternary injective polymorphism which violates H_2.*

Proof. Since the relation H_2 consists of three orbits of 9-tuples in $\mathrm{Aut}(V; R^{(3)})$, Lemma 7 implies that f generates an at most ternary function that violates H_2, and hence we can assume that f itself is at most ternary; by adding a dummy variable if necessary, we may assume that f is actually ternary. Moreover, f must certainly be essential, since essentially unary operations that preserve $R^{(3)}$ and $S^{(3)}$ are generated by $\{\mathrm{sw}\}$ and hence also preserve H_2. Corollary 54 implies that Γ is preserved by a binary canonical injection g of type p_1. Consider

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

Then h is clearly injective, and still violates H_2 – the latter can easily be verified combining the facts that f violates H_2, g is of type p_1, and all tuples in H_2 have pairwise distinct entries. ☐

Proposition 63. *Let f be an operation on G that preserves $R^{(3)}$ and $S^{(3)}$ and violates H_2. Then $\{f, \mathrm{sw}\}$ generates a ternary canonical injection of type minority.*

Proof. The proof is similar to the proof of Proposition 29. By Lemma 62, we can assume that f is a ternary injection. Because f violates H_2, there are $x^1, x^2, x^3 \in H_2$ such that $f(x^1, x^2, x^3) \notin H_2$. In the following, we will write $x_i := (x_i^1, x_i^2, x_i^3)$ for $1 \leq i \leq 9$. So $(f(x_1), \ldots, f(x_9)) \notin H_2$. If there were a map a generated by sw such that $a(x^i) = x^j$ for $1 \leq i \neq j \leq 3$, then $\{f, \mathrm{sw}\}$ would generate a binary injection that still violates H_2. Proposition 58 asserts that all binary injections generated by $\{f, \mathrm{sw}\}$ are of type $R^{(3)}$-projection, so we have reached a contradiction since operations of type $R^{(3)}$-projection preserve H_2. By permuting arguments of f if necessary, we can therefore assume without loss of generality that

$$R^{(3)}S^{(3)}S^{(3)}(x_1, x_2, x_3),\ S^{(3)}R^{(3)}S^{(3)}(x_4, x_5, x_6), \text{ and } S^{(3)}S^{(3)}R^{(3)}(x_7, x_8, x_9).$$

We set

$$S := \{y \in V^3 \mid \mathrm{NNN}(x_i, y) \text{ for all } 1 \leq i \leq 9\} .$$

Consider the ternary relations $Q_1Q_2Q_3$ on V^3, where $Q_i \in \{R^{(3)}, S^{(3)}\}$ for $1 \leq i \leq 3$; each of these relations defines a 3-type in $(V; R^{(3)})$. We claim that for fixed $Q_1Q_2Q_3$, whether or not $R^{(3)}(f(u), f(v), f(w))$ holds for $u, v, w \in S$ with $Q_1Q_2Q_3(u, v, w)$ does not depend on u, v, w. We go through all possibilities of $Q_1Q_2Q_3$.

(1) $Q_1Q_2Q_3 = R^{(3)}S^{(3)}S^{(3)}$. Let $\alpha \in \mathrm{Aut}(V; R^{(3)})$ be such that $(x_1^2, x_2^2, x_3^2, u_2, v_2, w_2)$ is mapped to $(x_1^3, x_2^3, x_3^3, u_3, v_3, w_3)$; such an automorphism exists since $\mathrm{NNN}(x_1, u), \mathrm{NNN}(x_1, v), \mathrm{NNN}(x_1$ and since (x_1^2, x_2^2, x_3^2) has the same type as (x_1^3, x_2^3, x_3^3), and (u_2, v_2, w_2) has the same type as (u_3, v_3, w_3) in $(V; R^{(3)})$. By Proposition 58, the operation g defined by $g(x, y) := f(x, y, \alpha(y))$ must be of type $R^{(3)}$-projection. Hence, $R^{(3)}(g(u_1, u_2), g(v_1, v_2), g(w_1, w_2))$ iff $R^{(3)}(g(x_1^1, x_1^2), g(x_2^1, x_2^2), g(x_3^1, x_3^2))$. Combining this with the equations $(f(u), f(v), f(w)) = (g(u_1, u_2), g(v_1, v_2), g(w_1, w_2))$ and $(g(x_1^1, x_1^2), g(x_2^1, x_2^2), g(x_3^1, x_3^2)) = (f(x_1), f(x_2), f(x_3))$, we get that $R^{(3)}(f(u), f(v), f(w))$ iff $R^{(3)}(f(x_1), f(x_2), f(x_3))$, and so we are done.

(2) $Q_1Q_2Q_3 = S^{(3)}R^{(3)}S^{(3)}$ or $Q_1Q_2Q_3 = S^{(3)}S^{(3)}R^{(3)}$. These cases are analogous to the previous case.

(3) $Q_1Q_2Q_3 = S^{(3)}R^{(3)}R^{(3)}$. Let α be defined as in the first case. By Proposition 58, the operation defined by $f(x, y, \alpha(y))$ must be of type projection. Reasoning as above, one gets that $R^{(3)}(f(u), f(v), f(w))$ iff $S^{(3)}(f(x_1), f(x_2), f(x_3))$.

(4) $Q_1Q_2Q_3 = R^{(3)}S^{(3)}R^{(3)}$ or $Q_1Q_2Q_3 = R^{(3)}R^{(3)}S^{(3)}$. These cases are analogous to the previous case.
(5) $Q_1Q_2Q_3 = R^{(3)}R^{(3)}R^{(3)}$ or $Q_1Q_2Q_3 = S^{(3)}S^{(3)}S^{(3)}$. These cases are trivial since f preserves $R^{(3)}$ and $S^{(3)}$.

To show that f generates an operation of type minority, by Proposition 28 it suffices to prove that f generates a function of type minority on S, since S is the product of isomorphic copies of G. We show this by another case distinction, based on the fact that $(f(x_1), \ldots, f(x_9)) \notin H_2$.

(1) Suppose that $R^{(3)}(f(x_1), f(x_2), f(x_3))$, $R^{(3)}(f(x_4), f(x_5), f(x_6))$, and $R^{(3)}(f(x_7), f(x_8), f(x_9))$. By the above, note that $R^{(3)}(f(u), f(v), f(w))$ for $u, v, w \in S$ if and only if $R^{(3)}S^{(3)}S^{(3)}(u, v, w)$, $S^{(3)}R^{(3)}S^{(3)}(u, v, w)$, $S^{(3)}S^{(3)}R^{(3)}(u, v, w)$, or $R^{(3)}R^{(3)}R^{(3)}(u, v, w)$. Hence, f behaves like an $R^{(3)}$-minority on S, and we are done by Lemma 61.
(2) Suppose that $S^{(3)}(f(x_1), f(x_2), f(x_3))$, $S^{(3)}(f(x_4), f(x_5), f(x_6))$, and $S^{(3)}(f(x_7), f(x_8), f(x_9))$. Then f behaves like an $R^{(3)}$-majority on S, which is impossible by Lemma 60.
(3) Suppose that $R^{(3)}(f(x_1), f(x_2), f(x_3))$, $R^{(3)}(f(x_4), f(x_5), f(x_6))$, and $S^{(3)}(f(x_7), f(x_8), f(x_9))$. Let e be a self-embedding of G such that for all $w \in V$, all $1 \leq j \leq 3$, and all $1 \leq i \leq 9$ we have that $N(x_i^j, e(w))$. Then $(u_1, u_2, e(f(u_1, u_2, u_3))) \in S$ for all $(u_1, u_2, u_3) \in S$. Hence, by the above, the ternary operation defined by $f(x, y, e(f(x, y, z)))$ is of type $R^{(3)}$-majority on S; but this is impossible by Lemma 60.
(4) Suppose that $R^{(3)}(f(x_1), f(x_2), f(x_3))$, $S^{(3)}(f(x_4), f(x_5), f(x_6))$, and $R^{(3)}(f(x_7), f(x_8), f(x_9))$ or $S^{(3)}(f(x_1), f(x_2), f(x_3))$, $R^{(3)}(f(x_4), f(x_5), f(x_6))$, and $R^{(3)}(f(x_7), f(x_8), f(x_9))$. These cases are analogous to the previous case.

Let $h(x, y, z)$ be a ternary injection of type minority generated by f; it remains to make h canonical. By Corollary 54, f generates a binary canonical injection $g(x, y)$ of type p_1. Set $t(x, y, z) := g(x, g(y, z))$. As in the proof of Proposition 29 the function $h(t(x, y, z), t(y, z, x), t(z, x, y))$ is still of type minority and canonical. □

of Theorem 48. Assume that H_2 is not primitive positive definable; by Theorem 4 there exists a polymorphism f of Γ that violates H_2. Since $\mathrm{Aut}(\Gamma)$ contains sw, the relations $R^{(3)}$ and $S^{(3)}$ consist of only one orbit of triples in Γ. Therefore, since they are preserved by all endomorphisms of Γ, it follows by Theorem 4 and Lemma 7 that these relations are primitive positive definable in Γ.

We can now apply Proposition 63 and obtain that $\{f, \mathrm{sw}\}$ generates a ternary injection of type minority which is canonical as a function from $(V; E)$ to $(V; E)$. Corollary 54 implies that Γ is preserved by a binary injection of type p_1 which is canonical as a function from $(V; E)$ to $(V; E)$, and the statement follows from Theorem 24. □

7.5. **When the endomorphisms of a reduct are generated by $\{-, \mathrm{sw}\}$.** We next consider Case (e) of Proposition 8. That is, we will assume that the endomorphisms of Γ are precisely the unary functions generated by $\{-, \mathrm{sw}\}$. In particular, $\mathrm{Aut}(\Gamma)$ contains $-, \mathrm{sw}$, and the automorphisms of Γ generate its endomorphisms. The proof for this case is similar to that for Case (c) of Proposition 8, presented in Section 7.3.

Definition 64. Let H_2' be the smallest 9-ary relation that is preserved by $-$ and contains H_2.

Proposition 65. $\mathrm{CSP}(V; H_2')$ *is NP-hard.*

Proof. If H_2' is primitive positive definable in Γ, then one can show similarly as in the proof of Proposition 21 that CSP(Γ) is NP-hard, by reduction from positive Not-all-three-equal-3SAT instead of positive 1-in-3-3SAT, and by simulating 1 with $R^{(3)}$ instead of E, and 0 with $S^{(3)}$ instead of N. $\square$

The following is an analog of Theorem 20 for the situation of this section.

Theorem 66. *Let Γ be a reduct of G whose endomorphisms are precisely the unary functions generated by $\{-, \mathrm{sw}\}$. Then H_2' is primitive positive definable in Γ, or (b) or (d) from Theorem 20 applies.*

Proof. Note that H_2' consists of three orbits of 9-tuples in $\mathrm{Aut}(\Gamma)$, and hence, if H_2' is not primitive positive definable in Γ, then there exists by Theorem 4 and Lemma 7 a ternary polymorphism f of Γ that violates H_2'. That is, there are $t^1, t^2, t^3 \in H_2'$ such that $f(t^1, t^2, t^3) \notin H_2'$. Note that for each t^j, either t^j or $-t^j \in H_2$. In the first case we set g_j to be the identity function on V, in the second case we let g_j be the operation $-$. Now consider the function f' defined by $f'(x_1, x_2, x_3) := f(g_1(x_1), g_2(x_2), g_3(x_3))$. We have that $s^j := g_j^{-1}(t^j) \in H_2$, but $f'(s^1, s^2, s^3) = f(t^1, t^2, t^3)$ is not in H_2', and therefore not in H_2 either. Hence, f' violates H_2. The function $h(x) := f'(x, x, x)$ is generated by $\{-, \mathrm{sw}\}$, and hence h either preserves $R^{(3)}$ and $S^{(3)}$, or it flips them. Since $f'(s^1, s^2, s^3)$ is not in H_2', neither is $-f'(s^1, s^2, s^3)$, and in particular not in H_2, so also $-f'$ violates H_2. Hence, by replacing f' with $-f'$ if necessary, we may assume that h preserves $R^{(3)}$ and $S^{(3)}$.

We claim that f' preserves $R^{(3)}$ and $S^{(3)}$. Suppose for contradiction that there are $u, v, w \in V^3$ with $R^{(3)}(u_i, v_i, w_i)$ for all $i \in \{1, 2, 3\}$ such that $R^{(3)}(f'(u), f'(v), f'(w))$ does not hold; the case where f' violates $S^{(3)}$ can be treated similarly. If (u_1, v_1, w_1), (u_2, v_2, w_2), and (u_3, v_3, w_3) all lie in the same orbit of triples in G, then we choose $a, b, c \in V$ with $R^{(3)}(a, b, c)$ such that $N(x, y)$ for $x \in \{a, b, c\}$ and $y \in \{u_1, v_1, w_1, u_2, v_2, w_2, u_3, v_3, w_3\}$. Then by the homogeneity of G there is for each $i \in \{2, 3\}$ a unary operation $\alpha_i \in \mathrm{Aut}(G)$ such that $\alpha_i(u_1, v_1, w_1, a, b, c) = (u_i, v_i, w_i, a, b, c)$. We then have that the unary function $g(x) := f'(x, \alpha_2(x), \alpha_3(x))$ maps $(u_1, v_1, w_1) \in R^{(3)}$ to $(f'(u), f'(v), f'(w)) \notin R^{(3)}$. But g and the function h above agree on $\{a, b, c\}$, and hence g preserves $R^{(3)}$ on $\{a, b, c\}$, but violates it on $\{u_1, v_1, w_1\}$. This contradicts the assumption that g is generated by $\{-, \mathrm{sw}\}$.

So suppose in the following that $R^{(3)}(f'(u), f'(v), f'(w))$ for all $u, v, w \in V^3$ with $R^{(3)}(u_i, v_i, w_i)$ for all $i \in \{1, 2, 3\}$ such that u, v, w belong to the same orbit of triples in G. We now show that $R^{(3)}(f'(u), f'(v), f'(w))$ for all $u, v, w \in V^3$ with $R^{(3)}(u_i, v_i, w_i)$ for all $i \in \{1, 2, 3\}$. To this end, note that for each $i \in \{2, 3\}$ there is a subset S_i of $\{u_i, v_i, w_i\}$ such that $(\mathrm{sw}_{S_i}(u_i), \mathrm{sw}_{S_i}(v_i), \mathrm{sw}_{S_i}(w_i))$ and (u_1, v_1, w_1) belong to the same orbit in G. Hence, there is $\beta_i \in \mathrm{Aut}(G)$ such that $\beta_i(\mathrm{sw}_{S_i}(u_1)) = u_i$, $\beta_i(\mathrm{sw}_{S_i}(v_1)) = v_i$, and $\beta_i(\mathrm{sw}_{S_i}(w_1)) = w_i$. Pick $a, b, c \in V \setminus \bigcup_{i \in \{1,2,3\}} \{u_i, v_i, w_i\}$. Note that for both $i \in \{2, 3\}$ we have that the triples (a, b, c) and $(\mathrm{sw}_{S_i}(a), \mathrm{sw}_{S_i}(b), \mathrm{sw}_{S_i}(c))$ lie in the same orbit. We then have that the function $x \mapsto f'(x, \beta_2(\mathrm{sw}_{S_2}(x)), \beta_3(\mathrm{sw}_{S_3}(x)))$ maps $(u_1, v_1, w_1) \in R^{(3)}$ to $(f'(u), f'(v), f'(w)) \notin R^{(3)}$. But the same unary function also maps $(a, b, c) \in R^{(3)}$ to a tuple in $R^{(3)}$ since f' by assumption preserves $R^{(3)}$ on tuples $R^{(3)}$ that lie in the same orbit, and indeed we have that for $i \in \{2, 3\}$ the triples (a, b, c) and $(\beta_i(\mathrm{sw}_{S_i}(a)), \beta_i(\mathrm{sw}_{S_i}(b)), \beta_i(\mathrm{sw}_{S_i}(c)))$ lie in the same orbit. This again contradicts the assumption that the unary function is generated by $\{-, \mathrm{sw}\}$.

We therefore have that f' preserves $R^{(3)}$ and $S^{(3)}$. Since it violates H_2, Proposition 49 implies that $\{f', \mathrm{sw}\}$ generates a ternary canonical injection of type minority, and we are done. □

8. Algorithms

We now prove that if one of the Cases (b) to (f) of Proposition **??** holds for a reduct $\Gamma = (V; E, N, \neq, \dots)$ of G with a finite language, then CSP(Γ) is in P. Tractability of Cases (b) and (c) is shown in Subsection 8.1, tractability of Case (d) in Subsection 8.2, of Case (e) in Subsection 8.3, and finally tractability of Case (f) in Subsection 8.4.

8.1. **Tractability of types minority / majority with unbalanced projections.** We show tractability of the CSP for reducts Γ as in Cases (b) and (c) of Proposition **??**.

Proposition 67. *Let $\Gamma = (V; E, N, \neq, \dots)$ be a finite language reduct of G, and assume that* Pol(Γ) *contains a ternary injection of type minority or majority, as well as a binary injection which is of type p_1 and either E-dominated or N-dominated in the second argument. Then* CSP(Γ) *is tractable.*

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

With the goal of reducing the CSP to injective structures, we define *injectivizations* for relations, atomic formulas, and structures.

Definition 69.
- Let R be any relation. Then the *injectivization of* R, denoted by inj(R), is the largest injective relation contained in R.
- 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 $R_\phi^{inj}(x_1, \dots, x_n)$, where R_ϕ^{inj} is a relation symbol which stands for the injectivization of the relation defined by ϕ.
- The *injectivization* of a relational structure Γ, denoted by inj(Γ), 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_ϕ^{inj}.

Note that inj(Γ) 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 70. Let Γ be a structure in a finite language, Δ be the injectivization of Γ, and Φ be an instance of CSP(Γ). Then the *injectivization of* Φ, denoted by inj(Φ), is the instance Ψ of CSP(Δ) obtained from Φ by replacing each conjunct $\phi(x_1, \dots, x_n)$ of Φ by $R_\phi^{inj}(x_1, \dots, x_n)$.

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 $R(x_1, \dots, x_n)$ 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).

```
// 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 1. Polynomial-time Turing reduction from the CSP(Γ) for Γ closed under an unbalanced binary injection, to the CSP of its injectivization Δ.

Lemma 71. *Let* Γ *be a finite language reduct of* G *which is preserved by a binary injection* f *of type* p_1 *that is* E*-dominated or* N*-dominated in the second argument. Then the algorithm shown in Figure 1 is a polynomial-time reduction of* CSP(Γ) *to* CSP(Δ)*, where* Δ *is the injectivization of* Γ*.*

Proof. By duality, we may in the following assume that f is E-dominated in the second argument.

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 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 of Ψ 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 $\phi = R(x_1, \ldots, x_n)$ be a constraint in Ψ. Since we are at the final stage of the algorithm, we can conclude that ϕ does not imply equality of any of the variables $x_1, \ldots, x_n$, and so there is for all $1 \leq i < j \leq n$ an n-tuple $t^{(i,j)}$ such that $R(t^{(i,j)})$ and $t_i \neq t_j$ hold. Since $R(x_1, \ldots, x_n)$ 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 $R(x_1, \ldots, x_n)$ contains an injective tuple $t = (t_1, \ldots, t_n)$.

Consider the mapping $r : \{x_1, \ldots, x_n\} \to G$ 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 $E(r(x_i), r(x_j))$, and if $N(s(x_i), s(x_j))$ then $N(r(x_i), r(x_j))$, because f is of type p_1 and because the entries of t are distinct. 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, $(s'(x_1), \ldots, s'(x_n))$ and $(r(x_1), \ldots, r(x_n))$ 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

$R(x_1,\dots,x_n)$. Hence, s' satisfies $R(x_1,\dots,x_n)$ as well. In this fashion we see that s' satisfies all the constraints of Ψ, proving our claim. □

To reduce the CSP for injective structures to Boolean CSPs, we make the following definition.

Definition 72. Let t be a k-tuple of distinct vertices of G, and let q be $\binom{k}{2}$. Then Boole(t) is the q-tuple $(a_{1,2}, a_{1,3}, \dots, a_{1,k},\ a_{2,3}, \dots, a_{k-1,k}) \in \{0,1\}^q$ 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 Boole(R) is the q-ary Boolean relation $\{\text{Boole}(t) \mid t \in R\}$. If ϕ is a formula that defines a relation R over G, then we also write Boole(ϕ) instead of Boole(inj(R)). Finally, for an injective reduct Γ, we write Boole(Γ) for the structure over a Boolean domain which has the relations of the form Boole(R), where R is a relation of Γ.

Lemma 73. *Let Γ be a finite language reduct of G which is injective. Then* CSP(Γ) *can be reduced to* CSP(Boole(Γ)) *in polynomial time.*

Proof. Let Φ be an instance of CSP(Γ), with variable set W. We create an instance Ψ of CSP(Boole(Γ)) as follows. The variable set of Ψ is the set of unordered pairs of variables from Φ. When $\phi = R(x_1,\dots,x_k)$ is a constraint in Φ, then Ψ contains the constraint

$$\text{Boole}(R)(x_{1,2}, x_{1,3}, \dots, x_{1,k}, x_{2,3}, \dots, x_{k-1,k}).$$

It is straightforward to verify that Ψ can be computed from Φ in polynomial time, and that Φ is a satisfiable instance of CSP(Γ) if and only if Ψ is a satisfiable instance of CSP(Boole(Γ)). □

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 $f(x,x,y) = f(x,y,x) = f(y,x,x) = y$.

Lemma 74. *Let Γ be a finite language reduct of G which is injective, and suppose it has an polymorphism of type minority (majority). Then* Boole(Γ) *has a minority (majority) polymorphism, and hence* CSP(Boole(Γ)) *can be solved in polynomial time.*

Proof. It is straightforward to show that Boole(Γ) has a minority (majority) polymorphism, and well-known (see [26]) that CSP(Boole(Γ)) can then be solved in polynomial time. □

Lemmas 71, 73, and 74 together provide a proof of Proposition 67.

8.2. **Tractability of type minority with balanced projections.** We move on to reducts as in Case (d) of Proposition **??**.

Proposition 75. *Let $\Gamma = (V; E, N, \neq, \dots)$ be a finite language reduct of G, and assume that* Pol(Γ) *contains a ternary injection of type minority, as well as a binary injection which is of type p_1 and balanced. Then* CSP(Γ) *is tractable.*

We start by proving that the relations of the reducts under consideration can be defined in G by first-order formulas of a certain restricted syntactic form; this normal form will later be 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. [16]).

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

Definition 76. 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 77. A ternary operation $f : V^3 \to V$ is called *straight* 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.

We remark that the existence of straight operations and even straight minority injections follows from the fact that G contains all countable graphs as induced subgraphs.

Proposition 78. *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 ternary injection which is of type minority and straight;*
(3) *R is preserved by some ternary injection injection of type minority, and some 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 straight injections 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 straight, 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 $t_2(u_i) \neq t_2(v_i)$. If also $t_3(u_i) \neq t_3(v_i)$, 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. If on the other hand $t_3(u_i) = t_3(v_i)$, then again $E(t_0(u_i), t_0(v_i))$ if and only if $E(t_2(u_i), t_2(v_i))$ since g is balanced. In either case, t_0 satisfies ϕ. This shows that f preserves ϕ, and hence also Ψ.

The implication from 2 to 3 is trivial, since every straight injection 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 straight 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 76:

- 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 $R(x_1, \dots, x_n)$ 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 an operation of type minority, the relation $\mathrm{Boole}(\mathrm{inj}(R))$ 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 $\mathrm{inj}(R)$ 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 $t^i := (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) \notin R_{i,j}$. 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 ϕ 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. 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. □

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

Definition 79. Let Γ be a finite language reduct of G which has only 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 76 has a

```
// 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 2. A polynomial-time algorithm for CSP(Γ) when Γ is preserved by a straight operation of type minority.

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 CSP(Γ) can be computed in linear time from Φ.

Definition 80. Let Γ be a finite language reduct of G which has only edge affine relations, and let Φ be an instance of CSP(Γ). 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 76 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 $\{u_1, v_1\} \oplus \cdots \oplus \{u_l, v_l\} = p$.

Proposition 75 now follows from the following lemma and Proposition 78.

Lemma 81. *Let Γ be a finite language reduct of G which has only edge affine relations. Then the algorithm shown in Figure 2 solves* CSP(Γ) *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. 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. □

8.3. **Tractability of type majority with balanced projections.** We turn to reducts as in Case (e) of Proposition **??**.

Proposition 82. *Let* $\Gamma = (V; E, N, \neq, \dots)$ *be a finite language reduct of* G*, and assume that* $\mathrm{Pol}(\Gamma)$ *contains a ternary injection of type majority, as well as a binary injection which is of type* p_1 *and balanced. Then* $\mathrm{CSP}(\Gamma)$ *is tractable.*

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. [16]).

Definition 83. 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

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

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

Proposition 84. *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 ternary injection which is of type majority and straight;*
(3) *R is preserved by some ternary injection of type majority and some binary balanced injection of type p_1.*

Proof. The proof is very similar to the proof of Proposition 78. We first show the implication from 1 to 2, that relations that are graph bijunctive are preserved by straight injections 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 $\mathrm{Boole}(\phi)$ 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 78.

For the implication from 2 to 3, observe that straight injections of type majority exist since G contains all countable graphs, 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 injection 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 83.

- 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 $R(x_1, \ldots, x_n)$ 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 $\exists x_i.R(x_1, \ldots, x_n)$. 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 Boole(inj(R)) is preserved by the Boolean majority operation, and hence is bijunctive. From this definition it is straightforward to obtain a definition $\Phi(x_1, \ldots, x_n)$ of 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 $u_1, u_2, v_1, v_2 \in \{x_1, \ldots, x_n\}$, 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 $i, j \in \{1, \ldots, n\}$ with $i < j$, let $R_{i,j}$ be the relation that holds for the tuple $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ iff $R(x_1, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_n)$ 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 $t \notin \text{inj}(R)$, 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 78.
(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 lemma with Proposition 84 gives us a proof of Proposition 82.

Lemma 85. *Let* Γ *be a reduct of* G *with a finite signature all of whose relations are graph bijunctive. Then* $\mathrm{CSP}(\Gamma)$ *can be solved in polynomial time.*

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

8.4. **Tractability of types max and min.** We are left with proving tractability of the CSP for reducts Γ as in case Case (f) of Proposition **??**, i.e., for reducts which have a canonical binary injective polymorphism of type max or min. We first observe 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 86. *Let* Γ *be a reduct of* G. *If* Γ *has a canonical binary injective polymorphism of type* max*, then it also has a canonical binary injective polymorphism of type* max *which is balanced or* E*-dominated. If it has a canonical binary injective polymorphism of type* min*, then it also has a canonical binary injective polymorphism of type* min *which is balanced or* 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 $h(x, y) := p(x, p(x, y))$ 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 87. *Let* Γ *be an* ω*-categorical homogeneous structure such that* $\mathrm{CSP}(\hat{\Gamma})$ *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* $\mathrm{CSP}(\Delta)$ *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 87 to our setting as follows.

Proposition 88. *Let Γ 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* $\mathrm{CSP}(\Gamma)$ *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 87, because

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

can be solved in polynomial time. One way to see this is to verify that all relations are preserved by a straight polymorphism of type majority, and to use the algorithm presented in Section 8.3. ☐

This completes the proof of the dichotomy statement of Theorem 1!

9. Classification

We have proven so far that all reducts of the random graph with finitely many relations define a CSP which is either tractable or NP-complete. This section is devoted to a more explicit description of the border between tractable and hard reducts.

Definition 89. Let B be a behavior for functions from G^2 to G. A ternary injection $f\colon V^3 \to V$ 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 **??** of Section 8.2: A ternary function is balanced if and only if it is hyperplanely balanced and of type p_1. Let us now define some more behaviors of binary functions which will appear hyperplanely in ternary functions in our classification.

Definition 90. A binary injection $f\colon V^2 \to V$ is

- *E-constant* if the image of f is a clique;
- *N-constant* if the image of f is an independent set;
- of type 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;
- of type 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.

Observe that if two canonical functions $f, g\colon V^n \to V$ satisfy the same type conditions, then they generate the same clone. This follows easily from the homogeneity of G and by local closure.

Let E_6 be the 6-ary relation defined by

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

It is easy to see that $\mathrm{Pol}(E_6)$ contains precisely the essentially unary operations which after deletion of all dummy variables are injective.

Theorem 91. *Let Γ be a reduct of G. Then either one of the relations E_6, H_1, H_1', H_2, or H_2' has a primitive positive definition in Γ, or Γ has a canonical polymorphism of one of the following 17 types.*

(1) *A constant operation.*
(2) *A balanced binary injection of type* max*.*
(3) *A balanced binary injection of type* min*.*
(4) *An E-dominated binary injection of type* max*.*
(5) *An N-dominated binary injection of type* min*.*
(6) *A ternary injection of type majority which is hyperplanely balanced and of type projection.*
(7) *A ternary injection of type majority which is hyperplanely E-constant.*
(8) *A ternary injection of type majority which is hyperplanely N-constant.*
(9) *A ternary injection of type majority which is hyperplanely of type* max *and E-dominated.*
(10) *A ternary injection of type majority which is hyperplanely of type* min *and N-dominated.*
(11) *A ternary injection of type minority which is hyperplanely balanced and of type projection.*
(12) *A ternary injection of type minority which is hyperplanely of type projection and E-dominated.*
(13) *A ternary injection of type minority which is hyperplanely of type projection and N-dominated.*
(14) *A ternary injection of type minority which is hyperplanely balanced of type* xnor*.*
(15) *A ternary injection of type minority which is hyperplanely balanced of type* xor*.*
(16) *A binary injection which is E-constant.*
(17) *A binary injection which is N-constant.*

Proof. Assume that none of the relations E_6, H_1, H_1', H_2, or H_2' has a primitive positive definition in Γ. Then Γ has a polymorphism which violates E_6; this polymorphism must be essential. By Lemma 5.3.10 in [2], Γ also has a binary essential polymorphism f.

We apply Proposition 9. There is nothing to show when the first case of that proposition holds, i.e., when Γ has a constant endomorphism.

Assume the second case holds, i.e., Γ has the endomorphism e_E or e_N; without loss of generality, we consider the case where e_E preserves Γ. Then consider the structure Δ induced in Γ on the image $e_E[V]$. This structure Δ is invariant under all permutations of its domain, and hence is first-order definable in $(e_E[V];=)$. It follows from the results in [6] that it either has a constant polymorphism, or a binary injection, or all polymorphisms of Δ are essentially unary. The structure Δ cannot have a constant endomorphism as otherwise also Γ has a constant polymorphism by composing the constant of Δ with e_E. We now show that Δ has an essential operation. Suppose that $f(a,a) = f(a,b)$ for all $a,b \in V$ with $E(a,b)$. We claim that $f(u,u) = f(u,v)$ for *every* $u,v \in V$. To see this, let $w \in V$ be such that

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, balanced.
Balanced in 1st, N-dom. in 2nd arg.	Hp. type min, N-dom.	Hp. type xor, balanced.

FIGURE 3. Minimal tractable canonical functions of type majority / minority and their corresponding canonical binary injections of type projection.

$E(u,w)$ and $E(v,w)$. Then $f(u,u) = f(u,w) = f(u,v)$, as required. It follows that f does not depend on its first variable, a contradiction. Hence, there exist $a, b \in V$ such that $E(a,b)$ and $f(a,a) \neq f(a,b)$. Similarly, there exist $c, d \in V$ such that $E(c,d)$ and $f(c,c) \neq f(d,c)$. Let T be an infinite clique adjacent to a, b, c, d. Then f is either essential on $T \cup \{a,b\}$ or on $T \cup \{c,d\}$, both cliques. Suppose without loss of generality that f is essential on $C = T \cup \{a,b\}$. Since all operations with the same behavior as e_E generate each other, we can also assume that the image of e_E is C. Then the restriction f' of $(x_1,x_2) \mapsto e_E(f(x_1,x_2))$ to $e_E[V]$ is an essential polymorphism of Δ. Hence, the above-mentioned result from [6] implies that Δ has a binary injective polymorphism h'. Then $h(x,y) := h'(e_E(x), e_E(y))$ is a polymorphism of Γ. But h is a binary canonical injection which is E-constant, and so Γ has a polymorphism from Item 16 of our list. The argument when Γ is preserved by e_N is similar, with Item 17 instead of Item 16.

It remains to discuss the last four cases of Proposition 8. Consider the very last case, i.e., where the endomorphisms of Γ are generated by $\mathrm{Aut}(G)$. Then Theorem 20 applies, and recall that we assume that H_1 has no primitive positive definition in Γ, excluding the first case of that theorem. If Γ has a binary canonical injective polymorphism of type max or min, then by Proposition 86 one of the operations from Item 2 to 5 applies. Otherwise, Γ has a ternary injective polymorphism t of type minority or majority, and one of the binary canonical injective polymorphisms of type projection listed in Theorem 20 – denote it by p. Set $s(x,y,z) := t(p(x,y), p(y,z), p(z,x))$ and $w(x,y,z) := s(p(x,y), p(y,z), p(z,x))$. Then the function w has one of the behaviors that describe functions from Items 6 to 15 – which of the behaviors depends on the precise behavior of p, and is shown in Figure 3. We leave the verification to the reader.

When the endomorphisms of Γ are generated by the function $-\colon V \to V$, then we may refer to Theorem 44, which brings us back to the preceding case. Similarly, when the endomorphisms of Γ are generated by sw or by $\{-, \mathrm{sw}\}$, then we may refer to Theorems 48 and 66, respectively, concluding the proof. □

The following is an operational tractability criterion for reducts of G.

Corollary 92. *Let Γ be a reduct of G with finite relational signature. Then:*

- *either Γ has a canonical polymorphism of one of the 17 types listed in Theorem 91, and* $\mathrm{CSP}(\Gamma)$ *is tractable, or*
- *one of the relations E_6, H_1, H_1', H_2, H_2' has a primitive positive definition in Γ, and* $\mathrm{CSP}(\Gamma)$ *is NP-complete.*

Proof. First suppose that one of the relationsE_6, H_1, H_1', H_2, H_2' has a primitive positive definition in Γ. In the case of H_1, NP-hardness of $\mathrm{CSP}(\Gamma)$ follows from Proposition 21, in the case of H_1' from Proposition 45, in the case of H_2 from Proposition 49, in the case of H_2' from Proposition 65, and in the case of E_6, NP-hardness of CSP(Γ) follows from [6].

Otherwise, by Theorem 91 the reduct Γ has a polymorphism of one of 17 described types, and we have to prove that CSP(Γ) is in P. In Item 1, that is if Γ is preserved by a constant polymorphism, then CSP(Γ) is trivially tractable as already stated in Proposition 8. In Item 2 to 5, CSP(Γ) is tractable by Proposition 88. If Γ is preserved by a function of type majority or minority (Item 6 to 15) then CSP(Γ) is tractable by Propositions 67, 75 and 82. In those cases, certain binary canonical injections of type projection are required – these are obtained by identifying the first two variables of the function of type majority / minority, and possibly exchanging the two arguments – Figure 3 shows which function of type majority / minority yields which type of binary injection. We leave the verification to the reader.

Finally, suppose that Γ is preserved by an operation f which is an E-constant binary injection from Item 16. 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 first-order definable in $(g[V]; =)$; such structures definable by equality only have been called *equality constraint languages* in [6], and their computational complexity has been classified. The structure Δ has a binary injection among its polymorphisms, namely, the restriction of f to Δ. It then follows from the results in [6] that CSP(Δ) is tractable. Hence, by Proposition 6 CSP(Γ) tractable as well, since Γ and Δ are homomorphically equivalent. □

Clearly, if we add relations to a reduct Γ, then the CSP of the structure thus obtained is computationally at least as complex as the CSP of Γ. On the other hand, by Lemma 3, adding relations with a primitive positive definition to a reduct does not increase the computational complexity of the corresponding CSP more than polynomially. Therefore, it makes sense to call a reduct *primitive positive closed* if it contains all relations that are primitive positive definable from it, and work with such reducts. Observe that primitive positive closed reducts will have infinitely many relations, and hence do not define a CSP; however, as we have already discussed in Section 3, it is convenient to consider a primitive positive closed reduct Γ tractable if and only if every reduct which has finitely many relations, all taken from Γ, has a tractable CSP.

The primitive positive closed reducts of G form a complete lattice, in which the meet of an arbitrary set S of reducts is their *intersection*, i.e., the reduct which has precisely those relations that are relations of all reducts in S. Call a primitive positive closed reduct *maximal tractable* if it is tractable and any extension of it by relations that are first-order definable in G is not tractable anymore. Under the assumption that P does not equal NP, we will now list the maximal tractable reducts of G; there are 17 of them. Since any chain C of tractable elements of the lattice of primitive positive closed reducts is bounded from above by a tractable element (namely, by the reduct which has all relations of all members of C), it then follows from Zorn's lemma that a reduct of G is tractable if and only if its relations are contained in the relations of one of the reducts of our list.

Recall the notion of a *clone* from Section 3. It follows from Theorem 4 and Proposition 5 that 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. We can use Corollary 92 to determine the minimal tractable clones; the maximal tractable reducts then are those with relations $\mathrm{Inv}(\mathcal{C})$ for a minimal tractable clone $\mathcal{C}$.

Corollary 93. *Assume $P \neq NP$. There are 17 minimal tractable clones that contain* $\mathrm{Aut}(G)$*; equivalently, there are 17 maximal tractable reducts of G.*

Proof. By Corollary 92 and the previous discussion, every minimal tractable clone that contains $\mathrm{Aut}(G)$ must contain an operation from one of the 17 types of operations listed in Theorem 91. Also recall that each operation of one of those 17 types generates a clone that contains every other operation with the same type. It therefore suffices to verify that all of these 17 clones are incomparable (i.e., no clone of the list contains another clone of the list), and hence that the clones in our list are indeed minimal.

This task is automatically verifiable: all functions in a clone generated by a set of canonical functions from a finite power of G to G are canonical – this can be shown by a straightforward induction over terms, since type conditions propagate through composition. Given a finite set $\mathcal{F}$ of canonical functions in form of their behaviors, for fixed $n \geq 1$ we can calculate all behaviors of the n-ary functions generated by $\mathcal{F}$ by composing the behaviors in all possible ways until we do not obtain any new behaviors. By this method, an algorithm can check that indeed, the behaviors of the ternary functions of each of the clones in our list are distinct. $\square$

Figure 4 shows the border between the clones of reducts with hard, and those with tractable CSP. 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. Lines between the circles that symbolize clones indicate containment (however, 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 23, 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 91. "E-semidominated" refers to "balanced in the first and E-dominated in the second argument".

We conclude by giving the argument for the decidability claim of Theorem 1.

Proposition 94. *There is an algorithm which given a finite set Ψ of graph formulas decides whether or not the problem Graph-SAT(Ψ) is tractable.*

Proof. By Corollary 92, the algorithm only has to check whether one of the canonical functions in Theorem 91 preserves all formulas ψ in Ψ. To do so it applies the canonical operation to orbit representatives from tuples satisfying ψ in all possible ways, and checks whether the result satisfies ψ, too. $\square$

We remark that it also follows from the more recent and more general result in [12] that it is decidable whether or not one of the relations in Corollary 92 has a primitive positive definition from a given finite language reduct Γ of G (of which the relations are given as graph formulas). This again yields Proposition 94.

Observe that the algorithm in the proof of Proposition 94 even decides tractability of Graph-SAT(Ψ) in polynomial time if the formulas ψ in Ψ are given as follows: if R is the, say, k-ary relation defined by ψ in G, then for every orbit of k-tuples in G that is contained in R the representation of ψ has a k-tuple representing this orbit (with the information which relations E, N, and $=$ hold on the tuple). Now since the operations the algorithm has to

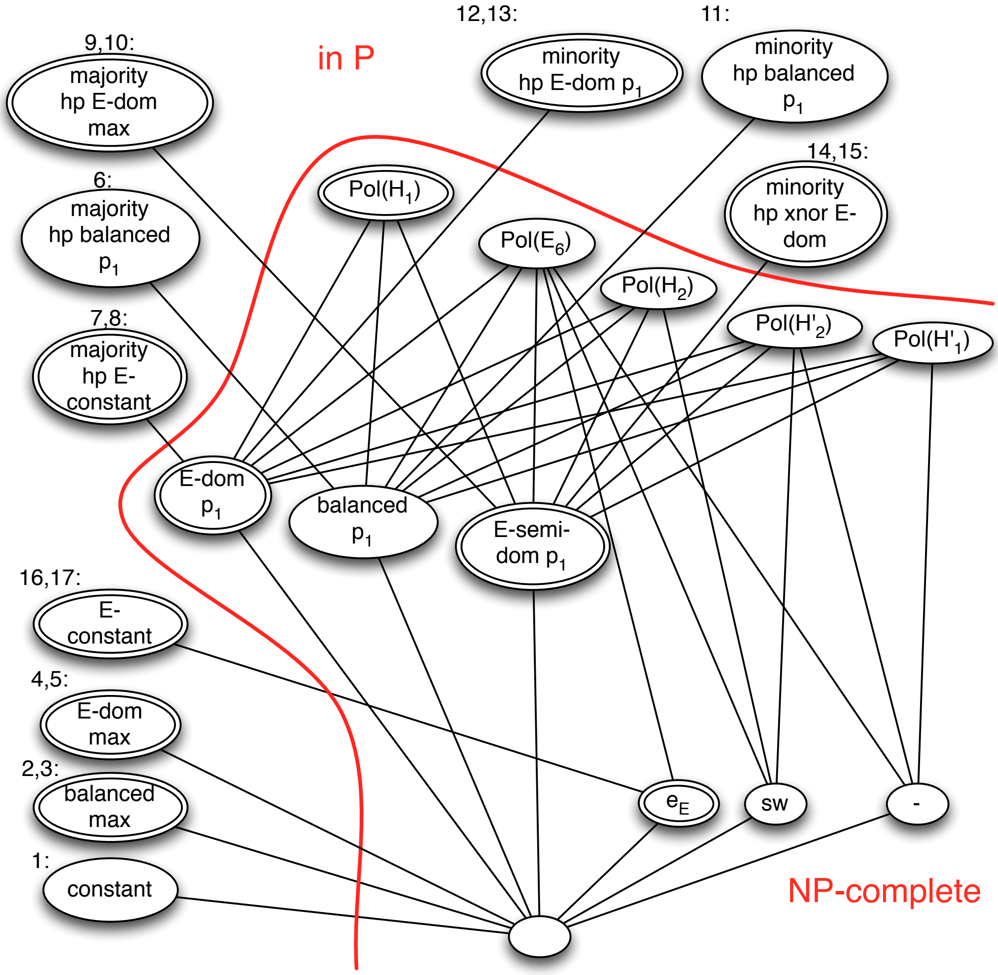

FIGURE 4. The border: Minimal tractable and maximal hard clones containing Aut(G).

consider are at most ternary, the number of possibilities for applying a canonical function to orbit representatives is at most cubic in the number of orbits satisfying ψ, which equals the representation size of ψ under this assumption.

REFERENCES

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

[2] Manuel Bodirsky. Complexity classification in infinite-domain constraint satisfaction. Mémoire d'habilitation à diriger des recherches, Université Diderot – Paris 7. Available at arXiv:1201.0856, 2012.

[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, Peter Jonsson, and Timo von Oertzen. Horn versus full first-order: a complexity dichotomy for algebraic constraint satisfaction problems. *Journal of Logic and Computation*, 22(3):643–660, 2011.
[6] 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 Computer Science Russia (CSR'06).
[7] Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):1–41, 2009. An extended abstract appeared in the Proceedings of the Symposium on Theory of Computing (STOC'08).
[8] Manuel Bodirsky and Jan Kára. A fast algorithm and Datalog inexpressibility for temporal reasoning. *ACM Transactions on Computational Logic*, 11(3), 2010.
[9] Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.
[10] Manuel Bodirsky and Michael Pinsker. Reducts of Ramsey structures. *AMS Contemporary Mathematics, vol. 558 (Model Theoretic Methods in Finite Combinatorics)*, pages 489–519, 2011.
[11] Manuel Bodirsky and Michael Pinsker. Minimal functions on the random graph. *Israel Journal of Mathematics*, 200(1):251–296, 2014.
[12] Manuel Bodirsky, Michael Pinsker, and Todor Tsankov. Decidability of definability. *Journal of Symbolic Logic*, 78(4):1036–1054, 2013. A conference version appeared in the Proceedings of LICS 2011, pages 321–328.
[13] Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
[14] Peter J. Cameron. The random graph. *Algorithms and Combinatorics*, 14:333–351, 1997.
[15] Peter J. Cameron. The random graph revisited. In *Proceedings of the European Congress of Mathematics*, volume 201, pages 267–274. Birkhäuser, 2001.
[16] Hubie Chen. A rendezvous of logic, complexity, and algebra. *SIGACT News*, 37(4):85–114, 2006.
[17] David A. Cohen, Peter Jeavons, Peter Jonsson, and Manolis Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.
[18] Ivo Duentsch. Relation algebras and their application in temporal and spatial reasoning. *Artificial Intelligence Review*, 23:315–357, 2005.
[19] Geoffrey Exoo. A lower bound for $R(5,5)$. *Journal of Graph Theory*, 13:97–98, 1989.
[20] Michael Garey and David Johnson. *A guide to NP-completeness*. CSLI Press, Stanford, 1978.
[21] Martin Goldstern and Michael Pinsker. A survey of clones on infinite sets. *Algebra Universalis*, 59:365–403, 2008.
[22] Wilfrid Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.
[23] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.
[24] Jaroslav Nešetřil. Ramsey theory. *Handbook of Combinatorics*, pages 1331–1403, 1995.
[25] Jaroslav Nešetřil and Vojtěch Rödl. Ramsey classes of set systems. *Journal of Combinatorial Theory, Series A*, 34(2):183–201, 1983.
[26] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.
[27] Ágnes Szendrei. *Clones in universal algebra*. Séminaire de Mathématiques Supérieures. Les Presses de l'Université de Montréal, 1986.
[28] Simon Thomas. Reducts of the random graph. *Journal of Symbolic Logic*, 56(1):176–181, 1991.
[29] Simon Thomas. Reducts of random hypergraphs. *Annals of Pure and Applied Logic*, 80(2):165–193, 1996.
[30] 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.

Institut für Algebra, TU Dresden, 01062 Dresden, Germany
E-mail address: Manuel.Bodirsky@tu-dresden.de
URL: http://www.math.tu-dresden.de/~bodirsky/

Équipe de Logique Mathématique, Université Diderot - Paris 7, UFR de Mathématiques, 75205 Paris Cedex 13, France
E-mail address: marula@gmx.at
URL: http://dmg.tuwien.ac.at/pinsker/