The Complexity of Abduction for Equality Constraint Languages

Johannes Schmidt* **and Michał Wrona†**

Linköping University
IDA
Linköping
`{johannes.schmidt, michal.wrona}@liu.se`

Abstract

Abduction is a form of nonmonotonic reasoning that looks for an explanation for an observed manifestation according to some knowledge base. One form of the abduction problem studied in the literature is the propositional abduction problem parameterized by a structure Γ over the two-element domain. In that case, the knowledge base is a set of constraints over Γ, the manifestation and explanation are propositional formulas.

In this paper, we follow a similar route. Yet, we consider abduction over infinite domain. We study the equality abduction problem parameterized by a relational first-order structure Γ over the natural numbers such that every relation in Γ is definable by a Boolean combination of equalities, a manifestation is a literal of the form $(x = y)$ or $(x \neq y)$, and an explanation is a set of such literals. Our main contribution is a complete complexity characterization of the equality abduction problem. We prove that depending on Γ, it is Σ_2^P-complete, or NP-complete, or in P.

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Abduction, infinite structures, equality constraint languages, computational complexity, algebraic approach

Digital Object Identifier 10.4230/LIPIcs.CSL.2013.615

1 Introduction

Abduction is a form of logical inference that aims at finding explanations for observed manifestations, starting from some knowledge base. It found many different applications in artificial intelligence [21], in particular to explanation-based diagnosis (e.g. medical diagnosis [10]), text interpretation [18], and planning [17].

In this paper we are interested in the complexity of abduction in a well-defined framework explained below. In a certain sense we follow a series of papers concerning the complexity of propositional abduction [16, 15, 20]. Roughly speaking, an instance of a propositional abduction problem for a relational structure Γ over the two element domain consists of a *knowledge base* KB — a conjunction of constraints over Γ, a set of *hypotheses* H — propositional literals formed upon variables in KB, and a *manifestation* M — a propositional formula. The question is whether there exists an *explanation*, i.e., a set $E \subseteq H$ such that $(\mathrm{KB} \wedge \bigwedge E)$ is satisfiable and $(\mathrm{KB} \wedge \bigwedge E)$ entails M. For every Γ, this propositional abduction problem is in Σ_2^P [16]. Depending on the restrictions, e.g., on Γ, one can obtain variants which are polynomial, NP-complete, coNP-complete or Σ_2^P-complete [20].

* Supported by the National Graduate School in Computer Science (CUGS), Sweden.
† Supported by the *Swedish Research Council* (VR) under grant 621-2012-3239.

Computer Science Logic 2013 (CSL'13).
Editor: Simona Ronchi Della Rocca ; pp. 615–633
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Here, we follow a similar scheme. The difference is that Γ is not a structure over the two-element domain but a structure that has a first-order definition in $(\mathbb{N};=)$, that is, the set of natural numbers with equality only. In what follows we call such structures *equality (constraint) languages* and the corresponding abduction problem *the equality abduction problem.* Since all structures first-order definable in $(\mathbb{N};=)$ have also first-order definitions in all other infinite structures, it is natural to start classifying the complexity of abduction for infinite structures considering equality languages first. The motivation for studying abduction for infinite constraint languages is strong and presented below. Equality languages are also of independent interest. They were studied in the context of CSPs [5] and QCSPs [3], see also Section 4.1. In [4], these languages were classified with respect to primitive positive definability.

An instance of the equality abduction problem for Γ consists of a knowledge base ϕ that is a conjunction of constraints over Γ, a subset V of the set of variables occurring in ϕ, and a manifestation which is a literal $L(x, y)$ of the form $(x = y)$, or $(x \neq y)$. The question is whether there exists an explanation, i.e., a conjunction ψ of such literals formed upon variables in V such that $(\phi \wedge \psi)$ is satisfiable and $(\phi \wedge \psi)$ entails $L(x, y)$. For instance, a possible explanation for the knowledge base $((x_1 = y_1 \wedge x_2 = y_2) \rightarrow z = v)$ and the manifestation $(z = v)$ is $(x_1 = y_1 \wedge x_2 = y_2)$. We give a precise definition in Section 4. The notions used in the introduction are pretty standard. Most of them are, nevertheless, defined in Section 2.

The main contribution of this paper is a trichotomous complexity classification of the equality abduction problem. As we show, this problem is always in Σ_2^P. Moreover, depending on Γ, it may be Σ_2^P-hard, or NP-complete, or solvable in polynomial time.

This way of parameterizing computational problems by relational structures is also referred to as *Schaefer's framework* or *Schaefer's approach* and dates back to Schaefer's paper on the complexity of *constraint satisfaction problems (CSPs)* over the two-element domain [22]. Modern proofs of Schaefer's theorem (see, e.g., [12]), as well as many other classifications of the complexity of related problems (for a survey, see [14]) including also propositional abduction take advantage of the so-called *algebraic approach.* In this approach the complexity of the problem, e.g., a constraint satisfaction problem or an abduction problem for a fixed Γ, is related directly to the set of operations preserving Γ. To enjoy the benefits of the algebraic tools, it is not necessary to restrict to the two-element domain, neither to any finite domain. Indeed, algebraic tools are equivalently powerful in classifying the complexity of CSPs for certain infinite structures [2], called ω-categorical structures [19].

Very natural examples of such structures are $(\mathbb{N};=)$ but also $(\mathbb{Q};<)$, that is, the order of rational numbers. Several classifications of constraint satisfaction problems for ω-categorical structures were obtained in the literature [2]. All of them follow the following scheme. One starts from some ω-categorical structure Δ such as $(\mathbb{N};=)$ or $(\mathbb{Q};<)$, then considers the class of all structures Γ with a first-order definition in Δ, which are also ω-categorical. The cases where Δ is $(\mathbb{N};=)$ and $(\mathbb{Q};<)$ were treated in [5] and [6], respectively. In both situations it appeared that the problem CSP(Γ) is either in P, or it is NP-complete. Here, we adopt this framework to study the complexity of the equality abduction problem.

To motivate the study of CSPs for ω-categorical structures, it is worth to mention that many problems studied independently in temporal reasoning, e.g., network satisfaction problems for qualitative calculi such as the Point Algebra [23] or Allen's Interval Algebra [1] can be directly formulated in this framework. In fact the complexity classification of Γ with a first-order definition in $(\mathbb{Q};<)$ substantially generalizes the result on tractability for the network satisfaction problem for the Point Algebra. Furthermore, the network satisfaction problem for Allen's Interval Algebra may be also modelled as a CSP for an ω-categorical structure, see [2].

Back to the issue of abduction, a kind of this problem handling time dependencies between events is called temporal abduction. Among others it is studied in [8, 13] in the framework [9] based on already mentioned, formalisms: Point Algebra, and Allen's Interval Algebra. It motivates the study of the complexity of abduction for ω-categorical structures, which we initiate in this paper.

1.1 Outline of the paper

We start in Section 2 by providing some preliminaries. In Section 3 we give a general definition of an abduction problem for a relational structure Γ. This definition captures many variants of propositional abduction as well as the equality abduction problem we study in this paper. We believe that the general definition will be employed in our future research on abduction. In that section we also show that two primitive positive interdefinable structures Γ_1 and Γ_2 give rise to abduction problems that are polynomial-time equivalent. Using the Galois connection in [7], it follows that if Γ_1 and Γ_2 are ω-categorical and preserved by the same operations, then their abduction problems are polynomial-time equivalent. We conclude that part of the paper by proving a useful result which links the complexity of the abduction and the constraint satisfaction problem.

A set of operations preserving a given structure Γ forms an algebraic structure called a *clone*. Clones corresponding to equality languages were classified in [4]. To provide our classification, which is presented in detail in Section 4, we express it in terms of clones, see Section 5. Then it remains to prove the complexity results, which are provided in Sections 6, 7, and 8. The paper is concluded in Section 9, where also the issue of future work is adressed.

The appendix contains proofs of: Theorem 5, Proposition 26, Lemma 28, and Proposition 29.

2 Preliminaries

Always, when it is possible, the notation is consistent with [19], [11], and [2], which we recommend as further reading on model theory, ω-categoricity and CSPs over ω-categorical structures, respectively. We write $[n]$ to denote $\{1, \ldots, n\}$.

2.1 Structures and Formulas

In this paper, we consider relational structures, which are typically denoted here by capital Greek letters such as Γ, or Δ. A signature is usually denoted by τ. If it is not stated otherwise, then we assume that the signature is finite. For the sake of simplicity, we use the same symbols to denote relations and their corresponding relation symbols. We mainly focus on countably infinite and ω-categorical structures. We say that a countably infinite structure is *ω-categorical* if all countable models of its first-order theory are isomorphic.

Let σ and τ be signatures with $\sigma \subseteq \tau$. When Δ is a σ-structure and Γ is a τ-structure with the same domain such that $R^\Delta = R^\Gamma$ for all $R \in \sigma$, then Γ is called an *expansion* of Δ.

For a τ-structure Γ over the domain D we define $\Delta := \Gamma^k$, where $k > 0$ is a natural number, to be a k-fold direct product of Γ, that is, the τ-structure on the domain D^k such that for every n-ary relation symbol R in τ we have $((d_1^1, \ldots, d_k^1), \ldots, (d_1^n, \ldots, d_k^n)) \in R^\Delta$ iff $(d_i^1, \ldots, d_i^n) \in R^\Gamma$ for all $i \in [k]$.

In this paper, we say that a relational structure Γ is first-order definable in Δ if Γ has the same domain as Δ, and for every relation R of Γ there is a first-order formula ϕ in the signature of Δ such that ϕ holds exactly on those tuples that are contained in R. If Γ is first-order definable in Δ, then we say that Γ is a *first-order reduct* of Δ. We say that two formulas are equivalent if they are over the same variables and define the same relation.

We are in particular interested in *equality (constraint) languages*, that is, first-order reducts of $(\mathbb{N}; =)$. All equality languages are ω-categorical structures. Furthermore, since $(\mathbb{N}; =)$ has quantifier elimination, every equality language has a quantifier-free first-order definition in $(\mathbb{N}; =)$ in conjunctive normal form. Such formulas over the signature $\{=, \neq\}$ will be called *equality formulas*, and equality formulas of the form $(x = y)$ and $(x \neq y)$ will be called *(equality) literals*. The set of all literals that can be formed upon a set of variables V will be denoted by $\mathcal{L}(V)$.

A Γ-constraint is an atomic formula over the signature of Γ of the form $R(x_1, \dots, x_n)$. Of special interest for abduction are Γ-formulas which are conjunctions of Γ-constraints. Furthermore, *primitive positive formulas (pp-formulas)* over the signature of Γ are first-order formulas built exclusively from conjunction, existential quantifiers, Γ-constraints and atomic formulas of the form $(x = y)$. The set of relations with a pp-definition in Γ is denoted by $[\Gamma]$.

For a quantifier-free first-order formula ϕ, we write $\text{Var}(\phi)$ to denote the set of variables occurring in ϕ. Let ϕ_1 and ϕ_2 be two equality formulas over the same set of variables $\{v_1, \dots, v_n\}$. We say that ϕ_1 entails ϕ_2 if $(\mathbb{N}; =) \models (\forall v_1 \cdots \forall v_n . \phi_1 \to \phi_2)$.

2.2 Polymorphisms and Clones

Let Γ be a structure. Homomorphisms from Γ^k to Γ are called *polymorphisms* of Γ. When R is a relation over domain D, we say that $f \colon D^k \to D$ *preserves* R if f is a polymorphism of $(D; R)$, and that f *violates* R otherwise. The set of all polymorphisms of a relational structure Γ, denoted by $\text{Pol}(\Gamma)$, forms an algebraic object called a *clone*. A clone on some fixed domain D is a set of operations on D containing all projections and closed under composition. A clone $\mathcal{C}$ is *locally closed* iff for all natural numbers n, for all n-ary operations g on D, if for all finite $B \subseteq D^n$ there exists an n-ary $f \in \mathcal{C}$ which agrees with g on B, then $g \in \mathcal{C}$. We say that a set of operations F *(locally) generates* an operation f (or that an operation f is (locally) generated by F) if f is in the smallest locally closed clone containing F, denoted by $\langle F \rangle$. If $F = \{g\}$ then we also say that g generates f or that f is generated by g.

▶ Proposition 1 (see e.g. [2]). Let F be a set of operations on some domain D. Then the following are equivalent: (i) F is the polymorphism clone of a relational structure; and (ii) F is a locally closed clone.

For ω-categorical structures we have the following Galois connection.

▶ **Theorem 2** ([7]). *Let Γ_1, Γ_2 be ω-categorical structures. We have that $Pol(\Gamma_1) \subseteq Pol(\Gamma_2)$ if and only if $[\Gamma_2] \subseteq [\Gamma_1]$.*

A special kind of a polymorphism is an automorphism. Observe that the set of automorphisms of $(\mathbb{N}; =)$ is exactly $S_{\mathbb{N}}$, that is, the set of all permutations on $\mathbb{N}$. By the theorem of Engeler, Ryll-Nardzewski and Svenonius (see, e.g., [19]), it follows that a structure is an equality language if and only if it is preserved by $S_{\mathbb{N}}$. Thus, by Theorem 2 and results obtained in Section 3, studying the complexity of the equality abduction problem amounts to studying locally closed clones on $\mathbb{N}$ containing $S_{\mathbb{N}}$. In what follows, such clones will be called *equality clones*. These clones form a complete lattice, where the least element is the clone generated by $S_{\mathbb{N}}$ and the greatest element is the set of all operations on $\mathbb{N}$, denoted here by $\mathcal{O}$. By $\mathcal{O}^{(k)}$, we denote a subset of $\mathcal{O}$ containing the operations of arity k. For a given family of clones $(C_i)_{i\in I}$, the meet is just an intersection $\bigcap_{i\in I} C_i$, and the join is $\langle\bigcup_{i\in I} C_i\rangle$. The lattice of equality clones was described in [4]. In this paper, we take advantage of this classification. The clones which are important for us are recalled in Section 5.

2.3 Complexity Classes

In this paper we study decision problems. The complexity classes we deal with are P, NP and Σ_2^P. Recall that $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$ is the class of decision problems solvable in nondeterministic polynomial time with access to an NP-oracle. In general, we write $\mathcal{C}_1^{\mathcal{C}_2}$ for the class of languages solvable in $\mathcal{C}_1$ with access to a $\mathcal{C}_2$-oracle.

2.4 Propositional Abduction

Abduction has been intensively studied in the propositional case, see e.g., [15] and [20] for complexity classifications. Similarly as in this paper, these classifications are based on the closure properties of constraint languages. To prove hardness results on the equality abduction problem, we will use the complexity classification for a special kind of propositional abduction problem called PQ-ABDUCTION(Γ), where Γ is a structure over the two-element domain. By $\mathrm{Lit}(V)$ we denote the set of propositional literals that can be formed upon variables in V. An instance of PQ-ABDUCTION(Γ) is a triple (ϕ, V, q), where ϕ is a Γ-formula, $V \subseteq \mathrm{Var}(\phi)$, and $q \in \mathrm{Var}(\phi) \setminus V$. We ask whether there is a set of literals $\mathrm{Lit} \subseteq \mathrm{Lit}(V)$ such that $(\phi \wedge \bigwedge \mathrm{Lit})$ is satisfiable but $(\phi \wedge \bigwedge \mathrm{Lit} \wedge \neg q)$ is not. We will need the following version of Theorem 7.6 in [15]. Beforehand, however, consider the following operations over the two-element domain: i) $\mathrm{majority}(b_1, b_2, b_3) = (b_1 \wedge b_2) \vee (b_1 \wedge b_3) \vee (b_2 \wedge b_3)$, ii) $\mathrm{minority}(b_1, b_2, b_3) = (b_1 \wedge \neg b_2 \wedge \neg b_3) \vee (\neg b_1 \wedge b_2 \wedge \neg b_3) \vee (\neg b_1 \wedge \neg b_2 \wedge b_3) \vee (b_1 \wedge b_2 \wedge b_3)$, iii) $\min(b_1, b_2) = b_1 \wedge b_2$, iv) $\max(b_1, b_2) = b_1 \vee b_2$, v) $c_0(b) = 0$, vi) $c_1(b) = 1$, vii) $\mathrm{opzero}(b_1, b_2, b_3) = b_1 \wedge (b_2 \vee b_3)$, and viii) $\mathrm{opone}(b_1, b_2, b_3) = b_1 \vee (b_2 \wedge b_3)$.

▶ **Theorem 3** ([15]). *Let Γ be a structure over the two-element domain.*

- *If Γ is preserved by i) majority, ii) minority, iii) min and c_1, iv) opzero, or v) opone, then PQ-ABDUCTION(Γ) is in P.*
- *Otherwise, if Γ is preserved by min or max, the problem PQ-ABDUCTION(Γ) is NP-complete.*
- *In all other cases, we have that PQ-ABDUCTION(Γ) is Σ_2^P-hard.*

3 The Abduction Problem and Algebra

Let Δ be a relational structure over some domain D and let $\Gamma, \mathcal{HYP}, \mathcal{M}$ be three first-order reducts of Δ. Let $\Gamma_{\mathcal{HYP}}$ be an expansion of Γ by the relations in $\mathcal{HYP}$, that is, the structure whose relations are either from Γ or $\mathcal{HYP}$; and $\Gamma_{\mathcal{HYP},\mathcal{M}}$ be an expansion of $\Gamma_{\mathcal{HYP}}$ by the relations in $\mathcal{M}$.

▶ **Definition 4.** An instance of the abduction problem $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$ is a triple $T = (\phi, V, M)$, where:

- ϕ is a Γ-formula (the knowledge base),
- V is a subset of $\mathrm{Var}(\phi)$,
- M is an $\mathcal{M}$-constraint (the manifestation).

The triple $T = (\phi, V, M)$ is a positive instance of $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$ if there exists an *explanation* for T, that is, a $\mathcal{HYP}$-formula ψ built upon variables from V such that both of the following hold:

- $(\phi \wedge \psi)$ is satisfiable in $\Gamma_{\mathcal{HYP}}$,
- $(\phi \wedge \psi)$ entails M (or equivalently, $(\phi \wedge \psi \wedge \neg M)$ is not satisfiable in $\Gamma_{\mathcal{HYP},\mathcal{M}}$).

In this case ψ is called an *explanation* for T.

This definition allows to model many variants of the propositional abduction problem as defined in [20]. For instance, the basic problem PQ-ABDUCTION(Γ) discussed in Section 2 (called V-ABD(Γ, PosLits) in [20]) can be modelled in the following way. We start from $\Delta = (\{0,1\}; T, F)$, where $T = \{(1)\}$ and $F = \{(0)\}$, and consider Γ with a first-order definition in Δ, that is, Γ may be an arbitrary structure over the two-element domain. Then, we set $\mathcal{HYP}$ to $(\{0,1\}; T, F)$, and $\mathcal{M}$ to $(\{0,1\}; T)$.

We observe in the following that the algebraic approach is applicable to the abduction problem under consideration. We first show that when $\mathcal{HYP}$ and $\mathcal{M}$ are fixed, then the complexity of $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$ is fully determined by the set $[\Gamma]$, the closure of Γ under primitive positive definitions. For $\Gamma_1 \subseteq [\Gamma_2]$ and an instance $T_1 = (\phi_1, V, M)$ of $\mathrm{ABD}(\Gamma_1, \mathcal{HYP}, \mathcal{M})$, we create an instance $T_2 = (\phi_2, V, M)$ of $\mathrm{ABD}(\Gamma_2, \mathcal{HYP}, \mathcal{M})$ by transforming a Γ_1-formula ϕ_1 into a Γ_2-formula ϕ_2 in the following standard way: (1) replace in ϕ_1 every Γ_1-constraint by its pp-definition in Γ_2, (2) delete all existential quantifiers, (3) delete all equality constraints and identify variables that are linked by a sequence of =.

It is easily observed that this transformation preserves satisfiability. Similarly as in [20], one can show that an explanation for T_1 can be easily rewritten into an explanation for T_2, and vice versa.

▶ **Theorem 5.** *Let Δ be a relational structure and $\Gamma_1, \Gamma_2, \mathcal{HYP}, \mathcal{M}$ be first-order reducts of Δ, where Γ_1 and Γ_2 are over finite signatures. If Γ_1 has a pp-definition in Γ_2, then ABD$(\Gamma_1, \mathcal{HYP}, \mathcal{M})$ reduces to ABD$(\Gamma_2, \mathcal{HYP}, \mathcal{M})$ in polynomial time.*

By Theorems 2 and 5, we have that the complexity of $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$ for ω-categorical Γ is fully captured by the set of polymorphisms preserving Γ.

▶ **Corollary 6.** *Let Δ be an ω-categorical structure and Γ_1, Γ_2, $\mathcal{HYP}$, and $\mathcal{M}$ first-order reducts of Δ, where Γ_1 and Γ_2 are over finite signatures. If Pol$(\Gamma_2) \subseteq$ Pol(Γ_1), then ABD$(\Gamma_1, \mathcal{HYP}, \mathcal{M})$ reduces to ABD$(\Gamma_2, \mathcal{HYP}, \mathcal{M})$ in polynomial time.*

We will conclude this section by providing a simple but useful link between the complexity of $\mathrm{CSP}(\Gamma_{\mathcal{HYP},\mathcal{M}})$ and $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$.

▶ Proposition 7. Let Δ be a relational structure and $\Gamma, \mathcal{HYP}, \mathcal{M}$ be first-order reducts of Δ, where $\mathcal{HYP}$ is over a finite signature. If $\mathrm{CSP}(\Gamma_{\mathcal{HYP},\mathcal{M}})$ is in the complexity class $\mathcal{C}$, then $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$ is in $\mathrm{NP}^{\mathcal{C}}$.

Proof. Let $T = (\phi, V, M)$ be an instance of $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$. By assumption, the signature of $\mathcal{HYP}$ is finite and therefore we can assume that if an explanation for T exists, then it is of polynomial length with respect to the number of variables in V, and thereby with respect to the length of T. Thus, we can guess a ψ and verify in polynomial time with two calls to the $\mathcal{C}$-oracle whether the instances $(\phi \wedge \psi)$ and $(\phi \wedge \psi \wedge \neg M)$ of $\mathrm{CSP}(\Gamma_{\mathcal{HYP},\mathcal{M}})$ are, respectively, satisfiable and not satisfiable in $\Gamma_{\mathcal{HYP},\mathcal{M}}$. ◀

4 Equality Abduction

In this paper, we treat a special case of the abduction problem $\mathrm{ABD}(\Gamma, \mathcal{HYP}, \mathcal{M})$. In the rest of the paper, Γ is always an equality language over a finite signature, and $\mathcal{HYP}$ and $\mathcal{M}$ are always $(\mathbb{N}; =, \neq)$.

We now formally define the equality abduction problem. Recall from Section 2 that equality literals are equality formulas of the form $(x = y)$ or $(x \neq y)$; and that the set of all equality literals that can be formed upon variables in V is denoted by $\mathcal{L}(V)$.

▶ **Definition 8** (Equality Abduction Problem)**.** The equality abduction problem $\mathrm{ABD}(\Gamma)$ for an equality language Γ (over a finite signature) is the computational problem, whose instance is a triple $T = (\phi, V, L(x, y))$, where:

- ϕ is a Γ-formula,
- V is a subset of $\mathrm{Var}(\phi)$,
- $L(x, y)$ is a literal and $x, y \in \mathrm{Var}(\phi)$.

The triple T is a positive instance of $\mathrm{ABD}(\Gamma)$ if there exists an explanation for T, that is, a set of literals $\mathcal{L} \subseteq \mathcal{L}(V)$ such that both of the following hold:

- $(\phi \wedge \bigwedge \mathcal{L})$ is satisfiable in $(\mathbb{N}; =, \neq)$,
- $(\phi \wedge \bigwedge \mathcal{L})$ entails $L(x, y)$ (or equivalently, $(\phi \wedge \bigwedge \mathcal{L} \wedge \neg L(x, y))$ is not satisfiable in $(\mathbb{N}; =, \neq)$).

We would like to remark that since Γ is always over a finite signature, the complexity of $\mathrm{ABD}(\Gamma)$ does not depend on the representation of relations in Γ.

Consider the following example. Let $\Gamma = (\mathbb{N}; \mathbb{I})$, where $\mathbb{I} = \{(x, y, z) \mid (x = y \rightarrow y = z)\}$, be an equality language. Consider the instance $T = (\phi, \{x, y, v\}, (z = w))$ of $\mathrm{ABD}(\Gamma)$ where ϕ is $((x = y \rightarrow y = z) \wedge (v = z \rightarrow v = w))$. Consider the set of literals $\mathcal{L} = \{x = y, y = v\}$. Observe that $(\phi \wedge \bigwedge \mathcal{L})$ is equivalent to $((x = y \rightarrow y = z) \wedge (v = z \rightarrow v = w)) \wedge (x = y) \wedge (y = v)$. It is straightforward to verify that this formula is satisfiable and that it entails $(z = w)$. Therefore, $\mathcal{L}$ is an explanation. As we will see, the problem $\mathrm{ABD}(\mathbb{N}; \mathbb{I})$ is NP-complete.

The following classes of equality languages are crucial to understand the complexity of the equality abduction problem.

▶ **Definition 9.** We say that a first-order formula is a *negative (equality) formula* if it is a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k) \text{ or } (x = y).$$

A relation R is called *negative* if it can be defined by a negative formula. An equality language Γ is *negative* if every its relation is negative.

▶ **Definition 10.** We say that a first-order formula is a *Horn (equality) formula* if it is a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x = y),$$

where it is permitted that $k = 0$ and the clause is simply an equality, i.e., of the form $x = y$. It is also permitted that we skip the equality and the clause is simply a disjunction of disequalities. A relation R is called *Horn* if it can be defined by a Horn formula. An equality language Γ is *Horn* if every its relation is Horn.

We will now present the main contribution of this paper. The following theorem completely classifies the complexity of the equality abduction problem.

▶ **Theorem 11** (Complexity Classification of the Equality Abduction Problem). *Let* Γ *be an equality language (over a finite signature). Then exactly one of the following holds.*
1. Γ *is negative and* $ABD(\Gamma)$ *is in P;*
2. Γ *is not negative but Horn and* $ABD(\Gamma)$ *is NP-complete;*
3. Γ *is not Horn and* $ABD(\Gamma)$ *is* Σ_2^P*-complete.*

We will now break the proof of Theorem 11 into smaller steps. Keeping in mind that NP^P is equal to NP, by Proposition 7 and the complexity results in [5], which are also discussed in Section 4.1, we have the following upper bounds.

▶ Proposition 12. Let Γ be an equality language. Then we have both of the following.
1. The problem ABD(Γ) is in Σ_2^P.
2. If Γ is Horn, then ABD(Γ) is in NP.

In the remainder of the paper, we focus on hardness and tractability results. We first characterize those equality languages for which the abduction problem is of the highest complexity.

▶ Proposition 13. Let Γ be an equality language. If Γ is not Horn, then ABD(Γ) is Σ_2^P-hard.

Then, we take care of those that are NP-hard.

▶ Proposition 14. Let Γ be an equality language. If Γ is not negative, then ABD(Γ) is NP-hard.

As we show, there is also a nontrivial class of equality abduction problems that are in P. For those, we will provide an appropriate algorithm.

▶ Proposition 15. Let Γ be an equality language. If Γ is negative, then ABD(Γ) is in P.

Propositions 13, 14, and 15 are proved in Sections 6, 7, and 8, respectively. Now assuming these propositions we will prove Theorem 11.

Proof of Theorem 11. It obviously holds exactly one of the following cases.
1. Γ is negative, or
2. Γ is not negative but Horn, or
3. Γ is not Horn.

We obtain the corresponding memberships by Propositions 15 and 12. The required hardness results follow by Propositions 13 and 14. ◀

4.1 Related Classifications on Equality Languages

As we already mentioned, equality languages were studied in the context of CSPs [5] and QCSPs [3]. Just to recall, an instance of CSP(Γ) may be seen as a Γ-formula where every variable is existentially quantified, and an instance of QCSP(Γ) as Γ-formula where every variable is either existentially or universally quantified. In both cases, the question is whether a given sentence is true in Γ.

The problem CSP(Γ) for an equality language Γ is always in NP, it is in P if Γ is Horn or it is preserved by a constant operation. Preservation under a constant function makes neither QCSP(Γ) nor ABD(Γ) tractable.

The problem QCSP(Γ) is always in PSPACE. Moreover, it is known to be in P if Γ is negative, and to be NP-hard if Γ is *positive*, that is, it may be defined as a conjunction of clauses of the form $(x_1 = y_1 \vee \cdots \vee x_k = y_k)$, but not negative. Otherwise QCSP(Γ) is coNP-hard. In particular QCSP($\mathbb{N}; \mathbb{I}$) is coNP-hard. We remark that the equality abduction problem for positive equality languages is Σ_2^P-hard unless it may be defined as a conjunction of equalities (i.e., unless it is negative).

5 Equality Clones

To prove Theorem 11 we take advantage of the classification of equality clones classified in [4]. Here, we recall only the definitions of clones that are relevant to our classification.

We say that an operation $f : \mathbb{N}^n \to \mathbb{N}$ is *essentially unary* if there exists $i \in [n]$ and $g : \mathbb{N} \to \mathbb{N}$ such that $f(x_1, \ldots, x_i, \ldots, x_k) = g(x_i)$.

▶ **Definition 16.** For every $i \in \mathbb{N}$, we define $\mathcal{K}_i$ to be the set of operations containing all essentially unary operations as well as all operations whose range has at most i elements.

We say that an operation $f : \mathbb{N}^n \to \mathbb{N}$ is *up to fictitious coordinates, injective* if there exists $\{i_1, \ldots, i_k\} \subseteq [n]$ where $i_1 < \ldots < i_k$ and an injective function $g : \mathbb{N}^k \to \mathbb{N}$ such that for all $(x_1, \ldots, x_n) \in \mathbb{N}^n$ we have that $f(x_1, \ldots, x_n) = g(x_{i_1}, \ldots, x_{i_k})$.

▶ **Definition 17.** (The Horn clone $\mathcal{H}$). We define $\mathcal{H}$ to be the set of operations which are, up to fictitious coordinates, injective.

Let $i \in [n]$. We call an operation $f \in \mathcal{O}^{(n)}$ *injective in the i-th direction* if $f(a) \neq f(b)$ whenever $a, b \in \mathbb{N}^n$ and $a_i \neq b_i$. We say that $f \in \mathcal{O}^{(n)}$ is *injective in one direction* if there is an $i \in [n]$ such that f is injective in the i-th direction.

▶ **Definition 18.** (Richard $\mathcal{R}$). We define $\mathcal{R}$ to be the set of operations injective in one direction.

Let $f_3 \in \mathcal{O}^{(3)}$ be any operation satisfying the following.

- For all $a \in \mathbb{N}$ we have $f_3(a, 1, 1) = 1$, $f_3(2, a, 2) = 2$, and $f_3(3, 3, a) = 3$.
- For all other arguments, the function arbitrarily takes a value that is distinct from all other function values.

▶ **Definition 19.** (The odd clone $\mathcal{S}$). We define $\mathcal{S}$ to be the set of operations generated by f_3, and $\mathcal{S}^+$ to be the superset of $\mathcal{S}$ containing additionally all constant operations.

In [4], one can find the proof that the sets of operations $\mathcal{S}, \mathcal{S}^+, \mathcal{R}, \mathcal{H}$ and $\mathcal{K}_i$ for every $i \in \mathbb{N}$ are locally closed clones. The following theorem is a direct consequence of Theorems 8, 13, and 15 in that paper.

▶ **Theorem 20.** *Let Γ be an equality language. Then either*
1. *Pol(Γ) is contained in $\mathcal{K}_i$ for some $i \in \mathbb{N}$, or*
2. *Pol(Γ) contains $\mathcal{H}$. In this case:*
   1. *either Pol(Γ) is contained in $\mathcal{S}^+$, or*
   2. *Pol(Γ) contains $\mathcal{R}$.*

Further, we get from [4] (Propositions 43 and 68) the algebraic characterizations for negative constraint languages and Horn constraint languages.

▶ Proposition 21. Let Γ be an equality constraint language. Then:
1. Γ is negative if and only if $\mathcal{R} \subseteq \mathrm{Pol}(\Gamma)$;
2. Γ is Horn if and only if $\mathcal{H} \subseteq \mathrm{Pol}(\Gamma)$.

We will use the following version of Corollary 6.

▶ Proposition 22. Let Γ_1 and Γ_2 be equality languages such that $\mathrm{Pol}(\Gamma_2) \subseteq \mathrm{Pol}(\Gamma_1)$, then $\mathrm{ABD}(\Gamma_1)$ has a polynomial time reduction to $\mathrm{ABD}(\Gamma_2)$.

6 Σ_2^P-hard Equality Abduction Problems

We start by presenting an infinite family of relations $\mathbb{H}_2, \mathbb{H}_3, \ldots$ that give rise to the abduction problems whose complexity meets the upper bound from Proposition 12.

Let $i \in \mathbb{N} \setminus \{0,1\}$. We define $\mathbb{H}_i$ to be an equality relation of arity $(i+4)$ which is the union of: (1) $\{(b_0,\ldots,b_i,x,y,z) \in \mathbb{N}^{i+4} \mid (|\{b_0,\ldots,b_i,x,y,z\}| < i+1)\}$ and (2) $\{(b_0,\ldots,b_i,x,y,z) \in \mathbb{N}^{i+4} \mid \bigwedge_{k\neq l;k,l\in\{0,\ldots,i\}}(b_k \neq b_l) \wedge (x = b_0 \vee x = b_1) \wedge (y = b_0 \vee y = b_1) \wedge (z = b_0 \vee z = b_1) \wedge (b_0 = x \vee b_0 = y \vee b_0 = z) \wedge (b_1 = x \vee b_1 = y \vee b_1 = z)\}$.

Observe that item (1) has a first-order definition over $(\mathbb{N};=)$. Indeed, one can define it by a conjunction of the formulas of the form $\neg(\bigwedge_{v,w\in S} v \neq w)$ where $S \subseteq \{b_0,\ldots,b_i,x,y,z\}$ is of size greater or equal than $(i+1)$.

The real purpose of this chapter is, however, to prove that ABD(Γ) is Σ_2^P-complete whenever Pol(Γ) $\subseteq \mathcal{K}_i$ for some i. The next lemma reduces that problem to showing that every ABD($\mathbb{N};\mathbb{H}_i$) for every i is Σ_2^P-complete. The lemma also explains why item (1) is included in the definition of $\mathbb{H}_i$: assure that $\mathbb{H}_i$ is preserved by all operations in $\mathcal{K}_i$.

▶ **Lemma 23.** *Let $i \in \mathbb{N} \setminus \{0,1\}$. Then $\mathbb{H}_i$ is preserved by all operations in $\mathcal{K}_i$.*

Proof. Directly from the definition of $\mathbb{H}_i$, it follows that this relation contains all tuples with at most i pairwise different entries. Thus it is preserved by all operations with range of at most i elements. It remains to show that $\mathbb{H}_i$ is preserved also by all essentially unary operations. We therefore consider some $f : \mathbb{N} \to \mathbb{N}$ and $t \in \mathbb{H}_i$. Observe that either $f(t) = \alpha(t)$ for some automorphism α of $(\mathbb{N};=)$, or the number of pairwise different entries in $f(t)$ is strictly smaller than in t. In the first case $f(t)$ is certainly in $\mathbb{H}_i$. Further, we observe that t has at most $(i+1)$ pairwise different entries. Hence in the second case, the tuple $f(t)$ has at most i pairwise different entries. Thus in this case, we are done by the observation from the first sentence of the proof. ◀

Let NAE $= \{(1,0,0),(0,1,0),(0,0,1),(0,1,1),(1,0,1),(1,1,0)\}$. To prove that ABD($\Gamma$), where $\Gamma = (\mathbb{N};\mathbb{H}_i)$, is Σ_2^P-hard we reduce from the problem PQ-ABDUCTION(Δ) such that $\Delta = (\{0,1\};\text{NAE})$. By Theorem 3, this problem is Σ_2^P-hard. Indeed, it is straightforward to verify that Δ is preserved by none of the following operations: majority, minority, min, max, opzero, opone. Let i be a natural number greater than or equal to 2. Observe that a tuple $(b_0,\ldots,b_i,x_p,x_r,x_s)$ of $\mathbb{H}_i$ with $b_0,\ldots,b_i$ pairwise different may be easily translated into a tuple (p,r,s) of NAE such that the value of $t \in \{p,r,s\}$ is $k \in \{0,1\}$ if and only if x_t is assigned to the same value as b_k. Analogously, one can find a tuple in $\mathbb{H}_i$ with $b_0,\ldots,b_i$ pairwise different corresponding to a tuple (p,r,s) of NAE. We produce an instance T_Γ of ABD(Γ) from an instance T_Δ of PQ-ABDUCTION(Δ) by replacing in the knowledge base every constraint NAE(p,r,s) with $\mathbb{H}_i(b_0,\ldots,b_i,x_p,x_r,x_s)$ and setting the manifestation to $(x_q \neq b_0)$, where q is the manifestation in T_Δ. Translating an explanation for T_Δ into an explanation for T_Γ, we ensure that all $b_0,\ldots,b_i$ are forced to be pairwise different. Converting the other way, it turns out that in general every explanation for T_Γ enforces $b_0,\ldots,b_i$ to take pairwise distinct values. We now give more details on that.

▶ Proposition 24. Let $i \in \mathbb{N} \setminus \{0,1\}$. Then the problem ABD($\mathbb{H}_i$) is Σ_2^P-hard.

Proof. Let $T_\Delta = (\phi_\Delta, V_\Delta, q)$ be an instance of PQ-ABDUCTION(Δ). We will now construct an instance $T_\Gamma = (\phi_\Gamma, V_\Gamma, L(x,y))$ of ABD(Γ) from it. First, for every propositional variable p occurring in ϕ_Δ, we introduce a variable x_p ranging over $\mathbb{N}$. Besides, we have $(i+1)$ extra variables $b_0,\ldots,b_i$ in Var(ϕ_Γ). The formula ϕ_Γ is a conjunction of constraints of the form $\mathbb{H}_i(b_0,\ldots,b_i,x_p,x_r,x_s)$ such that NAE(p,r,s) occurs in ϕ_Δ. The set V_Γ we define to be equal to $\{x_p \mid p \in V_\Delta\} \cup \{b_0, \ldots, b_i\}$, and $L(x, y)$ equal to $(x_q \neq b_0)$. This construction may be certainly performed in polynomial time. We will now prove that $T_\Delta \in$ PQ-ABDUCTION(Δ) if and only if $T_\Gamma \in$ ABD(Γ). We start from the following facts.

▶ Observation 25.
- Let $a_\Delta : \mathrm{Var}(\phi_\Delta) \to \{0,1\}$, and let $F_{\Delta,\Gamma}(a_\Delta) : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ be such that $F_{\Delta,\Gamma}(a_\Delta)(b_k) = k$ for all $k \in \{0, \ldots, i\}$ and $F_{\Delta,\Gamma}(a_\Delta)(x_p) = k$ if and only if $a_\Delta(p) = k$ for all $p \in \mathrm{Var}(\phi_\Delta)$ and $k \in \{0,1\}$. Then, if a_Δ satisfies ϕ_Δ, then $F_{\Delta,\Gamma}(a_\Delta)$ satisfies ϕ_Γ.
- Let $a_\Gamma : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ be such that $a_\Gamma(b_k) = k$ for $k \in \{0, \ldots, i\}$ and $a_\Gamma(x_p) \in \{0,1\}$ for all $x_p \notin \{b_0, \ldots, b_k\}$. Let $F_{\Gamma,\Delta}(a_\Gamma) : \mathrm{Var}(\phi_\Delta) \to \{0,1\}$ such that $F_{\Gamma,\Delta}(a_\Gamma)(p) = k$ if and only if $a_\Gamma(x_p) = k$ for all $p \in \mathrm{Var}(\phi_\Delta)$ and $k \in \{0,1\}$. Then, if a_Γ satisfies ϕ_Γ, then $F_{\Gamma,\Delta}(a_\Gamma)$ satisfies ϕ_Δ. ◀

Suppose that T_Δ has an explanation $\mathrm{Lit}_\Delta \subseteq \mathrm{Lit}(V_\Delta)$ so that $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta)$ is satisfiable by some assignment $a_\Delta : \mathrm{Var}(\phi_\Delta) \to \{0,1\}$ and $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$ is not satisfiable. We set $\mathcal{L}_\Gamma \subseteq \mathcal{L}(V_\Gamma)$ to be the union of $\{(x_p = b_1) \mid p \in \mathrm{Lit}_\Delta\}$, and $\{(x_p = b_0) \mid (\neg p) \in \mathrm{Lit}_\Delta\}$, and $\bigcup_{k \neq l; k,l \in \{0,\ldots,i\}} \{(b_k \neq b_l)\}$. We will now prove that $\mathcal{L}_\Gamma$ is an explanation for T_Γ. By Observation 25, the assignment $F_{\Delta,\Gamma}(a_\Delta)$ satisfies $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$. Assume towards contradiction that $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma \wedge (x_q = b_0))$ is also satisfiable by some a_Γ. By the construction of $\mathcal{L}_\Gamma$, the image of a_Γ has at least $(i+1)$ elements. Indeed, every b_k for $k \in \{0, \ldots, i\}$ has to be set to a different element. We assume without loss of generality that $a_\Gamma(b_k) = k$ for all $k \in \{0, \ldots, i\}$. By the construction of ϕ_Γ, for every $p \in \mathrm{Var}(\phi_\Delta)$ we have that $a_\Gamma(x_p) \in \{0,1\}$. Hence, by Observation 25, the assignment $F_{\Gamma,\Delta}(a_\Gamma)$ satisfies $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$. It contradicts the assumption and thus we are done with the left-to-right implication.

Suppose now that there is some explanation $\mathcal{L}_\Gamma \subseteq \mathcal{L}(V_\Gamma)$ for T_Γ, that is, (i) the formula $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$ is satisfiable, and (ii) $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma \wedge (x_q = b_0))$ is not satisfiable. Observe first that every assignment a satisfying $\mathcal{L}_\Gamma$ has at least $(i+1)$ elements in the image. Indeed, suppose that there is some $a : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ with less than $(i+1)$ elements in the image. Then a can be extended to a' without increasing the size of the range of the assignment so that every variable not occurring in $\mathcal{L}_\Gamma$ is set to the same value as b_0. By the definition of the PQ-ABDUCTION problem and the construction of ϕ_Γ, the variable x_q is not in V_Γ, and hence a' satisfies $(x_q = b_0)$. By the definition of $\mathbb{H}_i$, the assignment a' also satisfies all constraints in ϕ_Γ. But this contradicts (ii). Now, by item (2) of the definition of $\mathbb{H}_i$, every assignment satisfying $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$ has exactly $(i+1)$ elements in the image. Indeed, for every $p \in \mathrm{Var}(\phi_\Delta)$, such an assignment assigns to x_p the same value as to b_0 or b_1. We can therefore assume that there is an assignment a_Γ satisfying $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$ such that $a_\Gamma(b_k) = k$ for all $k \in \{0, \ldots, i\}$. We now set the explanation Lit_Δ for T_Δ to be the union of $\{(p) \mid p \in V_\Delta \wedge a_\Gamma(x_p) = 1\}$ and $\{(\neg p) \mid p \in V_\Delta \wedge a_\Gamma(x_p) = 0\}$. To complete the proof we have to show that (a) $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta)$ is satisfiable, and (b) $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$ is not satisfiable. Point (a) follows by (i) and Observation 25. To prove that (b) holds, we suppose that $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$ is satisfiable by some a_Δ. From Observation 25, it follows that $F_{\Delta,\Gamma}(a_\Delta)$ satisfies $(\phi_\Gamma \wedge (x_q = b_0))$. It is also easy to see that $F_{\Delta,\Gamma}(a_\Delta)(x) = a_\Gamma(x)$ for every x occurring in $\mathcal{L}_\Gamma$, hence $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma \wedge (x_q = b_0))$ is satisfiable by $F_{\Delta,\Gamma}(a_\Delta)$. But it contradicts (ii) and thus completes the proof. ◀

This section will be concluded by proving Proposition 13.

Proof of Proposition 13. Let Γ be not Horn. By Theorem 20 and Proposition 21 we know that there is an $i \in \mathbb{N}$ such that $\mathrm{Pol}(\Gamma) \subseteq \mathcal{K}_i$. From Lemma 23 it follows that $\mathrm{Pol}(\Gamma) \subseteq \mathcal{K}_i \subseteq \mathrm{Pol}(\mathbb{N}; \mathbb{H}_i)$ for some $i \in \mathbb{N}$. By Proposition 22 and Proposition 24 we conclude that ABD(Γ) is Σ_2^P-hard. ◀

7 Equality Horn Languages that are NP-hard

In this section we prove Proposition 14. It turns out that already a very simple Horn relation

$$\mathbb{I} = \{(x, y, z) \in \mathbb{N}^3 \mid (x = y \rightarrow y = z)\}$$

gives rise to an abduction problem which is NP-hard. In fact we provide a hardness proof for a structure $\Gamma = (\mathbb{N}; \mathbb{I}_4)$, where $\mathbb{I}_4 = \{(a, b, c, d) \in \mathbb{N}^4 \mid ((a = b \wedge b = c) \rightarrow (a = d))\}$. Observe that $\exists z\ (\mathbb{I}(x, y, z) \wedge \mathbb{I}(v, z, w))$ pp-defines $\mathbb{I}_4(x, y, v, w)$.

We reduce from the propositional abduction problem PQ-ABDUCTION(Δ), where $\Delta = (\{0, 1\}; R_{\mathrm{A3}})$ and $R_{\mathrm{A3}} = \{(x, y, z) \mid \neg x \wedge \neg y \rightarrow \neg z\}$. By Theorem 3, this problem is NP-hard. Indeed, it is straightforward to verify that Δ is preserved by none of the following operations: majority, minority, min, opzero, opone.

The idea of the proof is similar to what we had in the preceding section. Observe that every tuple (b_0, x_p, x_r, x_s) of $\mathbb{I}_4$ may be translated into a tuple (p, r, s) of R_{A3} such that $t \in \{p, r, s\}$ is 0 if x_t and b_0 are assigned to the same value and t is 1 otherwise. In the analogical way, one can find a tuple (b_0, x_p, x_r, x_s) of $\mathbb{I}_4$ for every (p, r, s) in R_{A3} by setting b_0 to 0, and x_t to the same value which is assigned to $t \in \{p, r, s\}$. We construct an instance of T_Γ from T_Δ by replacing in the knowledge base every constraint of the form $R_{\mathrm{A3}}(p, r, s)$ with $\mathbb{I}_4(b_0, x_p, x_r, x_s)$, and setting the manifestation to $(x_q \neq b_0)$ where q is the manifestation in T_Δ. Now an explanation for T_Γ may be obtained from an explanation for T_Δ when t and $(\neg t)$ are replaced with $(x_t \neq b_0)$ and $(x_t = b_0)$, respectively. Converting the explanation back is analogous.

▶ Proposition 26. The problem ABD($\mathbb{N}; \mathbb{I}$) is NP-hard.

We will conclude this section by proving Proposition 14.

Proof of Proposition 14. Let Γ be not negative. It suffices to concentrate on the case where Γ is Horn (if Γ is not Horn, we conclude with Proposition 13). We obtain then by Theorem 20 and Proposition 21 that $\mathrm{Pol}(\Gamma) \subseteq \mathcal{S}^+$.

By Proposition 62 in [4], we have that if R has a pp-definition by $\mathrm{ODD}_3 = \{(a, b, c) \in \mathbb{N}^3 \mid a = b = c \vee |\{a, b, c\}| = 3\}$, then $\mathcal{S} \subseteq \mathrm{Pol}(\mathbb{N}; R)$. The relation $\mathbb{I}$ has a pp-definition by ODD_3, this follows by Lemma 8.6 in [3]. Further, since $\mathbb{I}$ is preserved by all constant operations, we have that $\mathrm{Pol}(\Gamma) \subseteq \mathcal{S}^+ \subseteq \mathrm{Pol}(\mathbb{N}; \mathbb{I})$. Hence, by Proposition 22, there is a polynomial-time reduction from ABD($\mathbb{N}; \mathbb{I}$) to ABD(Γ). Thus, by Proposition 26, the problem ABD(Γ) is NP-hard. ◀

8 Abduction for Negative Languages is in P

Recall negative equality languages provided in Definition 9. In this section we prove Proposition 15, that is, we show that if Γ is a negative equality language, then ABD(Γ) is in P. The algorithm is presented in Fig. 1. We will first discuss the first line of the procedure. There an instance $T = (\phi, V, L(x, y))$ of ABD(Γ) is transformed into an instance $T_A = (\phi_A, V_A, L_A)$ of the problem $\mathrm{ABD}_{\mathrm{no\_eq}}$ defined below. The instance T_A is equivalent to T w.r.t. existence of explanations but is such that ϕ_A contains no equalities.

▶ **Definition 27.** An instance of the computational problem $\mathrm{ABD}_{\mathrm{no\_eq}}$ consists of:

- a conjunction of disjunctions of disequalities ϕ,
- a subset V of $\mathrm{Var}(\phi)$, and
- an equality literal $L(x, y)$, with $\{x, y\} \subseteq \mathrm{Var}(\phi)$, of the form $(x = y)$, or $(x \neq y)$.

The question is whether there is an explanation $\mathcal{L} \subseteq \mathcal{L}(V)$ such that:

1. $(\phi \wedge \bigwedge \mathcal{L})$ is satisfiable, and
2. $(\phi \wedge \bigwedge \mathcal{L} \wedge \neg L(x, y))$ is not satisfiable.

Let $\sim$ be an equivalence relation on $\mathrm{Var}(\phi)$ such that for all $x_1, x_2 \in \mathrm{Var}(\phi)$ we have $x_1 \sim x_2$ if and only if ϕ entails $(x_1 = x_2)$. We construct $\phi_A, V_A, L_A(x_A, y_A)$ by first replacing in $\phi, V, L(x, y)$, respectively, every variable from $\mathrm{Var}(\phi)$ by its equivalence class in $\mathrm{Var}(\phi)/\sim$. Then, we remove all equalities and disequalities of the form $(v \neq v)$ in ϕ_A.

▶ **Lemma 28.** *Let Γ be a negative language and $T = (\phi, V, L(x, y))$ be an instance of $ABD(\Gamma)$. Then there exists an instance T_A of ABD_{no_eq} such that $T \in ABD(\Gamma)$ if and only if $T_A \in ABD_{no\_eq}$. Moreover, T_A can be obtained from T in polynomial time.*

Algorithm for ABD(Γ), where Γ is a negative structure.

INPUT: An instance $T = (\phi, V, L(x, y))$ of $\mathrm{ABD}(\Gamma)$, where
- ϕ is a Γ-formula
- $V, \{x, y\} \subseteq \mathrm{Var}(\phi)$, and
- $L(x, y)$ is $(x = y)$, or $(x \neq y)$.

Let $T_A = (\phi_A, V_A, L_A(x_A, y_A))$ be an instance of $\mathrm{ABD}_{\mathrm{no\_eq}}$ from Lemma 28.
if ϕ_A is unsatisfiable **then return** FALSE
if $L_A(x_A, y_A)$ is $(x_A = y_A)$ **then**
  if x_A and y_A is the same variable **then return** TRUE
  else if $x_A, y_A \in V_A$ and $(\phi_A \wedge x_A = y_A)$ is satisfiable **then return** TRUE
  else return FALSE
end if
// from now on we can assume that $L_A(x_A, y_A)$ is $(x_A \neq y_A)$.
if x_A and y_A is the same variable **then return** FALSE
if $x_A, y_A \in V_A$ **then return** TRUE
if $z \in \{x_A, y_A\}$ is in V_A, and
  $v \in \{x_A, y_A\} \setminus \{z\}$ is not in V_A, and
  ϕ_A contains a clause equivalent to $(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee w \neq v)$ such that
  - $\{x_1, y_1, \ldots, x_k, y_k, w\} \subseteq V_A$, and
  - $(\phi \wedge \bigwedge_{i \in [k]} x_i = y_i \wedge z = w)$ is satisfiable

  then return TRUE
if $x_A, y_A \notin V_A$ and
  ϕ_A contains a clause equivalent to $(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_A \neq y_A)$ such that
  - $\{x_1, y_1, \ldots, x_k, y_k\} \subseteq V_A$, and
  - $(\phi \wedge \bigwedge_{i \in [k]} x_i = y_i)$ is satisfiable

  then return TRUE
return FALSE

Figure 1 Algorithm for Abduction for Negative Languages.

The following proposition states that the algorithm presented in Fig. 1 is correct and complete.

▶ Proposition 29. Let Γ be a negative equality language, and $(\phi, V, L(x, y))$ be an instance of $\mathrm{ABD}(\Gamma)$. Then $(\phi, V, L(x, y)) \in \mathrm{ABD}(\Gamma)$ if and only if the algorithm in Fig. 1 returns TRUE.

By Lemma 28, the first line of the algorithm in Fig. 1 can be performed in polynomial time. Apart from that, the procedure amounts to checking the satisfiability of a number of formulas which are obtained from negative formulas by adding conjuncts, which are equalities. Formulas of this form are always Horn formulas, and hence by the result in [5], each such check may be performed in polynomial time. Since the number of the satisfiability checks is readily polynomial with respect to the size of the input, we have the following.

▶ Proposition 30. Let Γ be a negative equality language. Then, for a given instance $(\phi, V, L(x, y))$ of $\mathrm{ABD}(\Gamma)$, the algorithm in Fig. 1 works in polynomial time in the size of $(\phi, V, L(x, y))$.

We are now ready to conclude this section.

Proof of Proposition 15. The statement follows by Propositions 29 and 30. ◀

9 Conclusion and Future Work

In this paper, we have initiated the study of the abduction problem parameterized by an ω-categorical relational structure Γ. We proved that as in the case of CSPs for these structures, the complexity of the abduction problem is fully captured by the set of operations preserving Γ. We have classified the complexity of the abduction problem parameterized by Γ with a first-order definition in $(\mathbb{N}; =)$ under the assumption that a manifestation is a literal of the form $(x = y)$ or $(x \neq y)$ and an explanation is a set of such literals over a given set of variables.

Our future work will concern similar classifications for first-order reducts of other ω-categorical structures. A natural choice for the next structure to study is $(\mathbb{Q}; <)$. Let Γ be a first-order reduct of $(\mathbb{Q}; <)$. In this case an instance of an abduction problem consists of a *temporal knowledge base* ϕ — a set of Γ-constraints — that describes point-based temporal dependencies between a finite number of events. A manifestation might be, for instance, of the form $(x < y)$, where x, y are events from ϕ. We can ask for an explanation that is a partial order on events in ϕ described by a conjunction of literals ψ of the form $(x \leq y)$ and $(x \neq y)$ such that ψ is consistent with ϕ ($(\phi \wedge \psi)$ is satisfiable) and ordering events from ϕ as described in ψ entails that x has to take place before y ($(\phi \wedge \psi)$ entails $(x < y)$).

Abduction problems for first-order reducts of $(\mathbb{Q}; <)$ do not only form a class of natural computational problems but also are plausible to be classified. The complexity of CSPs for these structures was classified in [6].

References

1 James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

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

3 Manuel Bodirsky and Hubie Chen. Quantified equality constraints. *SIAM Journal on Computing*, 39(8):3682–3699, 2010. A preliminary version of the paper appeared in the proceedings of LICS'07.

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

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

6 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).
7 Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.
8 Vittorio Brusoni, Luca Console, Paolo Terenziani, and Daniele Theseider Dupré. A spectrum of definitions for temporal model-based diagnosis. *Artif. Intell.*, 102(1):39–79, 1998.
9 Vittorio Brusoni, Luca Console, Paolo Terenziani, and Barbara Pernici. Later: Managing temporal information efficiently. *IEEE Expert*, 12(4):56–64, 1997.
10 Tom Bylander, Dean Allemang, Michael C. Tanner, and John R. Josephson. The computational complexity of abduction. *Artif. Intell.*, 49(1-3):25–60, 1991.
11 Peter J. Cameron. *Oligomorphic Permutation Groups*. Cambridge University Press, Cambridge, 1990.
12 Hubie Chen. A rendezvous of logic, complexity, and algebra. *ACM Comput. Surv.*, 42(1), 2009.
13 Luca Console, Paolo Terenziani, and Daniele Theseider Dupré. Local reasoning and knowledge compilation for efficient temporal abduction. *IEEE Trans. Knowl. Data Eng.*, 14(6):1230–1248, 2002.
14 Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer, editors. *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, volume 5250 of *Lecture Notes in Computer Science*. Springer, 2008.
15 Nadia Creignou and Bruno Zanuttini. A complete classification of the complexity of propositional abduction. *SIAM J. Comput.*, 36(1):207–229, 2006.
16 Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. *J. ACM*, 42(1):3–42, 1995.
17 A. Herzig, J. Lang, and Pierre Marquis. Planning as abduction. In *IJCAI-01 Workshop on Planning under Uncertainty and Incomplete Information*, Seattle, Washington, USA, aug 2001.
18 Jerry R. Hobbs, Mark E. Stickel, Douglas E. Appelt, and Paul A. Martin. Interpretation as abduction. *Artif. Intell.*, 63(1-2):69–142, 1993.
19 Wilfrid Hodges. *Model theory*. Cambridge University Press, 1993.
20 Gustav Nordh and Bruno Zanuttini. What makes propositional abduction tractable. *Artif. Intell.*, 172(10):1245–1284, 2008.
21 Harry E. Pople. On the mechanization of abductive logic. In *IJCAI*, pages 147–152, 1973.
22 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.
23 Marc Vilain, Henry Kautz, and Peter van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. *Reading in Qualitative Reasoning about Physical Systems*, pages 373–381, 1989.

A Proof of Theorem 5

Proof of Theorem 5. By assumption, it follows that every n-ary relation R in the structure Γ_1 has a pp-definition $(\exists y_1 \cdots \exists y_m . \phi_R(y_1, \ldots, y_m, x_1, \ldots, x_n))$ in Γ_2.

Let $T_1 = (\phi_1, V_1, M_1)$ be an instance of $\mathrm{ABD}(\Gamma_1, \mathcal{HYP}, \mathcal{M})$. We will now construct an instance $T_2 = (\phi_2, V_2, M_2)$ of $\mathrm{ABD}(\Gamma_2, \mathcal{HYP}, \mathcal{M})$. To obtain ϕ_2 from ϕ_1, we first transform ϕ_1 to a $(\Gamma_2 \cup \{=\})$-formula $\phi_=$. To that end, we replace every Γ_1-constraint in ϕ_1 of the form $R(x_1, \ldots, x_n)$ with $\phi_R(x_1, \ldots, x_n, y_1, \ldots, y_m)$ so that every time $y_1, \ldots, y_m$ are fresh variables. Observe that by the construction of $\phi_=$, we have $\mathrm{Var}(\phi_1) \subseteq \mathrm{Var}(\phi_=)$.

Consider now a partition $\Xi = \{X_1, \ldots, X_n\}$ of $\mathrm{Var}(\phi_=)$ such that $x, y \in \mathrm{Var}(\phi_=)$ are in the same block X_i if and only if $(x = y)$ is entailed by the conjunction of equalities occurring in $\phi_=$. Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be fresh variables. If $x \in \mathrm{Var}(\phi_=)$ is in X_i, then we say that v_i is the representative of x. We obtain a Γ_2-formula ϕ_2 from $\phi_=$ by first removing all $\{=\}$-constraints and then replacing every occurrence of every variable by its representative in $\mathcal{V}$. Similarly, to obtain V_2 and M_2, we replace all variables in V_1 and M_1, respectively, with their representatives in $\mathcal{V}$. It is easy to see that this procedure can be performed in polynomial time. We will now prove that T_1 is a positive instance of $\mathrm{ABD}(\Gamma_1, \mathcal{HYP}, \mathcal{M})$ if and only if T_2 is a positive instance of $\mathrm{ABD}(\Gamma_2, \mathcal{HYP}, \mathcal{M})$.

As a first step towards this goal, we will show that $T_1 = (\phi_1, V_1, M_1)$ is a positive instance of $\mathrm{ABD}(\Gamma_1, \mathcal{HYP}, \mathcal{M})$ if and only if $T_= = (\phi_=, V_1, M_1)$ is a positive instance of $\mathrm{ABD}(\Gamma_2 \cup \{=\}, \mathcal{HYP}, \mathcal{M})$. Observe that the claim is a consequence of the following two facts. First, every assignment $a_1 : \mathrm{Var}(\phi_1) \to \mathbb{N}$ satisfying ϕ_1 may be extended to $a_= : \mathrm{Var}(\phi_=) \to \mathbb{N}$ satisfying $\phi_=$. Second, every assignment $a_= : \mathrm{Var}(\phi_=) \to \mathbb{N}$ satisfying $a_=$ restricted to variables in $\mathrm{Var}(\phi_1)$ satisfies ϕ_1.

Now, it remains to prove that $T_= = (\phi_=, V_1, M_1)$ is a positive instance of $\mathrm{ABD}(\Gamma_2 \cup \{=\}, \mathcal{HYP}, \mathcal{M})$, if and only if $T_2 = (\phi_2, V_2, M_2)$ is a positive instance of $\mathrm{ABD}(\Gamma_2, \mathcal{HYP}, \mathcal{M})$. We start from an easy observation.

▶ Observation 31. Let $a_= : \mathrm{Var}(\phi_=) \to \mathbb{N}$. If $a_=$ satisfies $\phi_=$, then for all $X_i \in \Xi$ and all $x, y \in X_i$ we have that $a_=(x) = a_=(y)$. ◀

Suppose first that $\psi_=$ is an explanation for $T_=$. Construct ψ_2 from $\psi_=$ by replacing every variable by its representative in $\mathcal{V}$. Since $(\phi_= \wedge \psi_=)$ is satisfiable, by Observation 31, it follows that $(\phi_2 \wedge \psi_2)$ is also satisfiable. Furthermore, if $(\phi_2 \wedge \psi_2 \wedge \neg M_2)$ was satisfiable, we would have that $(\phi_= \wedge \psi_= \wedge \neg M_1)$ is satisfiable. It contradicts the assumption and completes the proof of this implication.

Suppose now that ψ_2 is an explanation for T_2. Construct $\psi_=$ from ψ_2 by replacing each variable v_i by some, always the same, variable $x \in X_i \cap V_1$. Observe that by the construction of V_2, the set $X_i \cap V_1$ is not empty. Since $(\phi_2 \wedge \psi_2)$ is satisfiable, it easily follows that $(\phi_= \wedge \psi_=)$ is also satisfiable. To conclude the proof, observe that $(\phi_= \wedge \psi_= \wedge \neg M_1)$ is not satisfiable. Indeed, if it was satisfiable, then by Observation 31, we would have that $(\phi_2 \wedge \psi_2 \wedge \neg M_2)$ is satisfiable. It contradicts the assumption and completes the proof of the theorem. ◀

B Proof of Proposition 26

Proof of Proposition 26. Consider the relation $\mathbb{I}_4 = \{(a, b, c, d) \in \mathbb{N}^4 \mid ((a = b \wedge b = c) \to (a = d))\}$. Observe that $\exists z\, (\mathbb{I}(x, y, z) \wedge \mathbb{I}(v, z, w))$ pp-defines $\mathbb{I}_4(x, y, v, w)$. By Theorem 5, it is therefore enough to show that $\mathrm{ABD}(\Gamma)$, where $\Gamma = (\mathbb{N}; \mathbb{I}_4)$, is NP-hard.

We reduce from the propositional abduction problem PQ-ABDUCTION(Δ), where $\Delta = (\{0, 1\}; R_{\mathrm{A3}})$ and $R_{\mathrm{A3}} = \{(x, y, z) \mid \neg x \wedge \neg y \to \neg z\}$. By Theorem 3, this problem is NP-hard. Indeed, it is straightforward to verify that Δ is preserved by none of the following operations: majority, minority, min, opzero, opone.

Let $T_\Delta = (\phi_\Delta, V_\Delta, q)$ be an instance of PQ-ABDUCTION(Δ). We will now construct an instance $T_\Gamma = (\phi_\Gamma, V_\Gamma, L(x, y))$ of $\mathrm{ABD}(\Gamma)$. First, for every Boolean variable $p \in \mathrm{Var}(\phi_\Delta)$, we introduce a variable x_p ranging over $\mathbb{N}$. Besides, we have also one extra variable b_0 in $\mathrm{Var}(\phi_\Gamma)$. Then, we set ϕ_Γ to be a conjunction of atomic formulas of the form $\mathbb{I}_4(b_0, x_p, x_r, x_s)$ such that $R_{\mathrm{A3}}(p, r, s)$ occurs in ϕ_Δ; and V_Γ to $\{x_p \mid p \in V_\Delta\} \cup \{b_0\}$. To complete the reduction we set $L(x, y)$ to be equal to $(x_q \neq b_0)$.

The reduction may certainly be performed in polynomial time. To complete the proof, we will now show that $T_\Delta \in$ PQ-ABDUCTION(Δ) if and only if $T_\Gamma \in$ ABD(Γ).

We start from the following facts.

▶ Observation 32. Let $a_\Gamma : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ be any assignment satisfying ϕ_Γ. Then a'_Γ, obtained from a_Γ so that for every $x \in \mathrm{Var}(\phi_\Gamma)$ we have that $a'_\Gamma(x) = 0$ iff $a_\Gamma(x) = a_\Gamma(b_0)$ and $a'_\Gamma(x) = 1$ otherwise, also satisfies ϕ_Γ. ◀

▶ Observation 33.
- Let $a_\Delta : \mathrm{Var}(\phi_\Delta) \to \{0, 1\}$, and let $F_{\Delta,\Gamma}(a_\Delta) : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ be such that $F_{\Delta,\Gamma}(a_\Delta)(b_0) = 0$ and $F_{\Delta,\Gamma}(a_\Delta)(x_p) = k$ if and only if $a_\Delta(p) = k$ for $k \in \{0, 1\}$. Then, if a_Δ satisfies ϕ_Δ, then $F_{\Delta,\Gamma}(a_\Delta)$ satisfies ϕ_Γ.
- Let $a_\Gamma : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ such that $a_\Gamma(b_0) = 0$ and for every $p \in \mathrm{Var}(\phi_\Delta)$, we have $a_\Gamma(x_p) \in \{0, 1\}$. Define $F_{\Gamma,\Delta}(a_\Gamma) : \mathrm{Var}(\phi_\Delta) \to \{0, 1\}$ so that for every $p \in \mathrm{Var}(\phi_\Delta)$ and $k \in \{0, 1\}$, we have that $F_{\Gamma,\Delta}(a_\Gamma)(x_p) = k$ iff $a_\Gamma(p) = k$. Then, if a_Γ satisfies ϕ_Γ, then $F_{\Gamma,\Delta}(a_\Gamma)$ satisfies ϕ_Δ. ◀

Suppose first that there exists a set of propositional literals $\mathrm{Lit}_\Delta \subseteq \mathrm{Lit}(V_\Delta)$ such that $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta)$ is satisfiable by some assignment $a_\Delta : \mathrm{Var}(\phi_\Delta) \to \{0, 1\}$ and $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$ is not satisfiable. We set the explanation $\mathcal{L}_\Gamma \subseteq \mathcal{L}(V_\Gamma)$ for T_Γ to be the union of $\bigcup_{p \in \mathrm{Lit}_\Delta} \{x_p \neq b_0\}$ and $\bigcup_{(\neg p) \in \mathrm{Lit}_\Delta} \{x_p = b_0\}$. We will now show that $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$ is satisfiable and $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma \wedge (b_0 \neq x_q))$ is not satisfiable. The former follows from Observation 33. Indeed, the formula $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$ is satisfiable by $F_{\Delta,\Gamma}(a_\Delta)$. To prove the latter, assume on the contrary that $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma \wedge (x_q = b_0))$ is satisfied by some $a_\Gamma : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$. By Observation 32, there exists $a'_\Gamma : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ satisfying ϕ_Γ that assigns 0 to b_0 as well as sends every variable to 0, or 1. Thus, by Observation 33, we have that $F_{\Gamma,\Delta}(a'_\Gamma)$ satisfies $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$. This contradicts the assumption and completes the proof of the left-to-right implication.

Suppose now that there is an explanation $\mathcal{L}_\Gamma \subseteq \mathcal{L}(V_\Gamma)$ of T_Γ, that is, the formula $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma)$ is satisfiable by some $a_\Gamma : \mathrm{Var}(\phi_\Gamma) \to \mathbb{N}$ and $(\phi_\Gamma \wedge \bigwedge \mathcal{L}_\Gamma \wedge (x_q = b_0))$ is not satisfiable. By Observation 32, we can assume that a_Γ sends every variable to 0 or 1 and b_0 to 0. We set the explanation $\mathrm{Lit}_\Delta \subseteq \mathrm{Lit}(V_\Delta)$ for T_Δ to be the union of $\{(p) \mid p \in \mathrm{Lit}(V_\Delta) \wedge a_\Gamma(x_p) = 1\}$ and $\{(\neg p) \mid p \in \mathrm{Lit}(V_\Delta) \wedge a_\Gamma(x_p) = 0\}$. We will now show that $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta)$ is satisfiable and $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$ is not satisfiable. The former holds by Observation 33. Indeed, we have that $F_{\Gamma,\Delta}(a_\Gamma)$ satisfies $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta)$. Finally, assume on the contrary that $(\phi_\Delta \wedge \bigwedge \mathrm{Lit}_\Delta \wedge \neg q)$ is satisfied by $a_\Delta : \mathrm{Var}(\phi_\Delta) \to \{0, 1\}$. By Observation 33, the assignment $F_{\Delta,\Gamma}(a_\Delta)$ satisfies $(\phi_\Gamma \wedge (x_q = b_0))$. It is also easy to see that for every $p \in \mathrm{Var}(\phi_\Delta)$ it holds $F_{\Delta,\Gamma}(a_\Delta)(x_p) = a_\Gamma(x_p)$. Thus $F_{\Delta,\Gamma}(a_\Delta)$ satisfies $\bigwedge \mathcal{L}_\Gamma$ and in consequence, by Observation 33, $(\phi_\Gamma \wedge \mathcal{L}_\Gamma \wedge (x_q = b_0))$. But this contradicts the assumption and hence completes the proof. ◀

C Proof of Lemma 28

Proof of Lemma 28. Let $T_A = (\phi_A, V_A, L_A(x_A, y_A))$ be an instance of $\mathrm{ABD}_{\mathrm{no\_eq}}$ obtained from $T = (\phi, V, L(x, y))$ as it was described before the formulation of the lemma. It is easily observed that T_A can be obtained from T in polynomial time.

We will now show that $T \in$ ABD(Γ) if and only if $T_A \in \mathrm{ABD}_{\mathrm{no\_eq}}$. As we will argue, it is basically a consequence of the following two facts. Let $X_1, \ldots, X_k$ be equivalence classes of $\mathrm{Var}(\phi)/\sim$.

▶ Observation 34. Let $a_\phi : \mathrm{Var}(\phi) \to \mathbb{N}$ be an assignment satisfying ϕ. Then for every $i \in [k]$ and all $v, z \in X_i$ we have that $a_\phi(v) = a_\phi(z)$. ◀

▶ Observation 35.
- Let $a_\phi : \mathrm{Var}(\phi) \to \mathbb{N}$ be such that for every $i \in [k]$ and all $v, z \in X_i$ we have that $a_\phi(v) = a_\phi(z)$, and let $F_{\phi,\phi_A}(a_\phi) : \mathrm{Var}(\phi)/\sim \to \mathbb{N}$ be such that for all $i \in [k]$ and all $z \in X_i$ we have that $F_{\phi,\phi_A}(a_\phi)(X_i) = a_\phi(z)$. Then if a_ϕ satisfies ϕ, then $F_{\phi,\phi_A}(a_\phi)$ satisfies ϕ_A.
- Let $a_{\phi_A} : \mathrm{Var}(\phi)/\sim \to \mathbb{N}$, and $F_{\phi_A,\phi}(a_{\phi_A}) : \mathrm{Var}(\phi) \to \mathbb{N}$ be such that for all $i \in [k]$ and $z \in X_i$ we have $F_{\phi_A,\phi}(a_{\phi_A})(z) = a_{\phi_A}(X_i)$. Then, if a_{ϕ_A} satisfies ϕ_A, then $F_{\phi_A,\phi}$ satisfies ϕ. ◀

Suppose first that $T \in \mathrm{ABD}(\Gamma)$. Then there exists $\mathcal{L} \subseteq \mathcal{L}(V)$ such that $(\phi \wedge \bigwedge \mathcal{L})$ is satisfiable by some $a_\phi : \mathrm{Var}(\phi) \to \mathbb{N}$ and $(\phi \wedge \bigwedge \mathcal{L} \wedge \neg L(x, y))$ is not satisfiable. We define $\mathcal{L}_A$ to be the set that contains all literals of the form $(X_i \circ X_j)$, where $i, j \in [k]$ and $\circ \in \{=, \neq\}$, such that $\mathcal{L}$ contains $(x' \circ y')$ for some $x' \in X_i, y' \in X_j$. From Observations 34 and 35, we easily obtain that $(\phi_A \wedge \bigwedge \mathcal{L}_A)$ is satisfiable by $F_{\phi,\phi_A}(a_\phi)$. Also, if $(\phi_A \wedge \bigwedge \mathcal{L}_A \wedge \neg L_A(x_A, y_A))$ was satisfiable, then by Observation 35 we would have that $(\phi \wedge \bigwedge \mathcal{L} \wedge \neg L(x, y))$ is satisfiable. It contradicts the assumption and completes the proof of the left-to-right implication.

Suppose now that there is $\mathcal{L}_A$ such that both $(\phi_A \wedge \bigwedge \mathcal{L}_A)$ is satisfiable by some $a_{\phi_A} : \mathrm{Var}(\phi)/\sim \to \mathbb{N}$ and $(\phi_A \wedge \bigwedge \mathcal{L}_A \wedge \neg L(x_A, y_A))$ is not satisfiable. We set $\mathcal{L}$ to contain all literals of the form $(v \circ z)$, where $\circ \in \{=, \neq\}$, such that $v \in X_i \cap V$ and $z \in X_j \cap V$ and $X_i \circ X_j$ is in $\mathcal{L}_A$. By Observation 35, we have that $(\phi \wedge \mathcal{L})$ is satisfiable by $F_{\phi_A,\phi}(a_{\phi_A})$. On the other hand, if $(\phi \wedge \bigwedge \mathcal{L} \wedge \neg L(x, y))$ was satisfiable by some a_ϕ, then by Observations 34 and 35, we would have that $(\phi_A \wedge \bigwedge \mathcal{L}_A \wedge \neg L_A(x_A, y_A))$ is satisfiable, which contradicts the assumption and completes the proof of the lemma. ◀

D Proof of Proposition 29

Proof of Proposition 29. By Lemma 28, it is enough to show that the algorithm returns TRUE if and only if $(\phi_A, V_A, L_A(x_A, y_A)) \in \mathrm{ABD}_{\mathrm{no\_eq}}$.

We will first show the proof of the right-to-left implication. If $L_A(x_A, y_A)$ is $(x_A = y_A)$, then the algorithm returns TRUE if ϕ_A is satisfiable and either x_A and y_A are the same variable, or $\{x_A, y_A\} \subseteq V_A$. In the first case an empty set of literals works as an explanation, while in the other, we can take $\mathcal{L}$ equal to $(x_A = y_A)$. If $L_A(x_A, y_A)$ is $(x_A \neq y_A)$, then the algorithm may return TRUE in lines 10, 11 and 12. In line 10, we set $\mathcal{L}$ to $\{x_A \neq y_A\}$. In line 11 to $\bigcup_{i \in [k]} \{x_i = y_i\} \cup \{z = w\}$, while in line 12 to $\bigcup_{i \in [k]} \{x_i = y_i\}$.

We now turn to the left-to-right implication. Suppose that there is a set of literals $\mathcal{L}$ such that both Points 1 and 2 in Definition 27 hold. Consider a formula ψ equal to $(\phi_A \wedge \bigwedge \mathcal{L})$. Let $X = \{X_1, \ldots, X_k\}$ be a partition of $\mathrm{Var}(\psi)$ such that ψ entails $(s = t)$ if and only if there exists $i \in [k]$ such that both s and t are in X_i. Then, for $i \in [k]$, we choose one element s_i from every X_i to be a representative of all elements in X_i. Then, in all disjunctions of disequalities, we first replace all variables with their representatives, and then remove all disequalities of the form $(s_i \neq s_i)$. Since all these transformations preserve the satisfiability of the formula, in the end we get no empty clauses. Denote the formula obtained in this way by ψ'.

Consider first the case where $L_A(x_A, y_A)$ is $(x_A = y_A)$. The case where x_A and y_A are the same variable is handled by the procedure in line 4. Thus we can assume that they are different. Since $(\phi_A \wedge \bigwedge \mathcal{L})$ entails $(x_A = y_A)$, and ϕ_A does not contain equalities, the formula $(x_A = y_A)$ must be entailed by $\bigwedge \mathcal{L}$. Indeed, suppose this is not the case, then there is an assignment $a : Var(\phi_A) \to \mathbb{N}$ satisfying $(\bigwedge \mathcal{L} \wedge x_A \neq y_A)$. Let $a' : Var(\phi_A) \to \mathbb{N}$ be a satisfying assignment to $(\phi_A \wedge \bigwedge \mathcal{L})$. We claim that $b(a, a')$ where $b : \mathbb{N}^2 \to \mathbb{N}$ is a binary injective operation satisfies $(\phi_A \wedge \bigwedge \mathcal{L} \wedge x_A \neq y_A)$. It clearly satisfies $(x_A \neq y_A)$, it satisfies

$\bigwedge \mathcal{L}$ since it is Horn. To see that $b(a, a')$ satisfies ϕ_A we use the following property of a negative equality formula ϕ: let $a, a' : \mathrm{Var}(\phi) \to \mathbb{N}$ such that a' satisfies ϕ and b be a binary injection, then $b(a, a')$ satisfies ϕ. Thus, we proved that $(x_A = y_A)$ must be entailed by $\bigwedge \mathcal{L}$. Hence x_A and y_A are in V_A. Since also $\phi_A \wedge (x_A = y_A)$ is satisfiable, the procedure returns TRUE in line 5.

Assume now that $L_A(x_A, y_A)$ is $(x_A \neq y_A)$. Since $(\phi_A \wedge L_A(x_A, y_A))$ is satisfiable, we have that x_A and y_A are in different blocks X_a, and X_b, respectively, of X. Assume without loss of generality that they are the representatives of their own blocks. Observe that ψ' contains a clause of the form $(x_A \neq y_A)$: otherwise $(\psi \wedge x_A = y_A)$ would be satisfiable by the assignment $a_X : \mathrm{Var}(\psi) \to \mathbb{N}$ sending all variables from X_i where $i \neq b$ to i, and all variables from X_b to a, i.e., no explanation would exist. So, we can assume that ψ' contains $(x_A \neq y_A)$. In this case either (i) x_A and y_A are both in V, or (ii) there is $z \in \{x_A, y_A\}$ which is in V, and $v \in \{x_A, y_A\} \setminus \{z\}$ which is not, and ϕ contains a clause equivalent to $(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee w \neq v)$ such that for every i both x_i and y_i as well as z and w are in the same block of X, or (iii) both x_A, y_A are not in V and ϕ contains a clause equivalent to $(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_A \neq y_A)$ such that for every i both x_i and y_i are in the same block of X. Observe that cases (i), (ii), and (iii) are handled by the procedure in lines 10, 11, and 12, respectively. ◀