Complexity of Propositional Abduction for Restricted Sets of Boolean Functions*

Nadia Creignou and **Johannes Schmidt**
Laboratoire d'Informatique Fondamentale, CNRS
Université d'Aix-Marseille II
163, avenue de Luminy, 13288 Marseille Cedex 9, France

Michael Thomas
Institut für Theoretische Informatik
Gottfried Wilhelm Leibniz Universität
Appelstr. 4, 30167 Hannover, Germany

Abstract

Abduction is a fundamental and important form of non-monotonic reasoning. Given a knowledge base explaining how the world behaves it aims at finding an explanation for some observed manifestation. In this paper we focus on propositional abduction, where the knowledge base and the manifestation are represented by propositional formulae. The problem of deciding whether there exists an explanation has been shown to be Σ_2^p-complete in general. We consider variants obtained by restricting the allowed connectives in the formulae to certain sets of Boolean functions. We give a complete classification of the complexity for all considerable sets of Boolean functions. In this way, we identify easier cases, namely NP-complete and polynomial cases; and we highlight sources of intractability. Further, we address the problem of counting the explanations and draw a complete picture for the counting complexity.

Introduction

Abduction is a fundamental and important form of non-monotonic reasoning. Assume that given a certain consistent knowledge about the world, we want to explain some observation. This task of finding an explanation or only telling if there is one, is called abduction. Today it has many application areas spanning medical diagnosis (Bylander et al. 1989), text analysis (Hobbs et al. 1993), system diagnosis (Stumptner and Wotawa 2001), configuration problems (Amilhastre, Fargier, and Marquis 2002), temporal knowledge bases (Bouzid and Ligeza 2000) and has connections to default reasoning (Selman and Levesque 1990).

There are several approaches to formalize the problem of abduction. In this paper, we focus on *logic based abduction* in which the knowledge base is given as a set Γ of propositional formulae. We are interested in deciding whether there exists an *explanation* E, *i.e.*, a set of literals consistent with Γ such that Γ and E together entail the observation.

From a complexity theoretic viewpoint, the abduction problem is very hard in the sense that it is Σ_2^p-complete and thus situated at the second level of the polynomial hierarchy (Eiter and Gottlob 1995). This intractability result raises the question for restrictions leading to fragments of lower complexity. Several such restrictions have been considered in previous works. One of the most famous amongst those is Schaefer's framework, where formulae are restricted to generalized conjunctive normal form with clauses from a fixed set of relations (Creignou and Zanuttini 2006; Nordh and Zanuttini 2005; 2008).

A similar yet different procedure is to rather require formulae to be constructed from a restricted set of Boolean functions B. Such formulae are called B-*formulae*. This approach has first been taken by Lewis, who showed that the satisfiability problem is NP-complete if and only if this set of Boolean functions has the ability to express the negation of implication connective $\not\rightarrow$ (Lewis 1979). Since then, this approach has been applied to a wide range of problems including equivalence and implication problems (Reith 2003; Beyersdorff et al. 2009a), satisfiability and model checking in modal and temporal logics (Bauland et al. 2006; 2008), default logic (Beyersdorff et al. 2009b), and circumscription (Thomas 2009), among others.

We follow this approach and show that Post's lattice allows to completely classify the complexity of propositional abduction for several variants and all possible sets of allowed Boolean functions. We first examine the case where the representation of the manifestation is a literal. We show that depending on the set B of allowed connectives the abduction problem is either Σ_2^p-complete, or NP-complete, or in P and $\oplus\mathrm{LOGSPACE}$-hard, or in $\mathrm{LOGSPACE}$. More precisely, we prove that the complexity of this abduction problem is Σ_2^p-complete as soon as B can express one of the functions $x \vee (y \wedge \neg z)$, $x \wedge (y \vee \neg z)$ or $(x \wedge y) \vee (x \wedge \neg z) \vee (y \wedge \neg z)$. It drops to NP-complete when all functions in B are monotonic and have the ability to express one of the functions $x \vee (y \wedge z)$, $x \wedge (y \vee z)$ or $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$. The problem becomes solvable in polynomial time and is $\oplus\mathrm{LOGSPACE}$-hard if B-formulae may depend on more than one variable while being representable as linear equations. Finally the complexity drops down to $\mathrm{LOGSPACE}$ in all remaining cases.

We then examine several variants of the propositional abduction problem. The variants considered are obtained by restricting representation of the manifestation to be respectively a clause, a term or a B-formula. We present a complete classification in all cases. An overview of the results is given

---

* Supported by ANR *Algorithms and complexity* 07-BLAN-0327-04 and DFG grant VO 630/6-1.

in Figure 1. Our results highlight the sources of intractability and exhibit properties of Boolean functions that lead to an increase of the complexity of abduction.

In (Creignou and Zanuttini 2006) the authors obtained a complexity classification of the abduction problem for formulae which are in generalized conjunctive normal form, with clauses from a fixed set of relations. The two classifications are in the same vein since they classify the complexity of abduction for local restrictions on the knowledge base. However the two results are incomparable, in the sense that no classification can be deduced from the other. They only overlap on the particular case of the linear connective $\oplus$, for which both types of sets of formulae can be seen as systems of linear equations. This special abduction case has been shown to be decidable in polynomial time in (Zanuttini 2003).

Besides the decision problem, another natural question is concerned with the number of explanations. This problem refers to the counting problem for abduction. The study of the counting complexity of abduction has been started by Hermann and Pichler (2007). We prove here a trichotomy theorem showing that counting the full explanations of propositional abduction problems is either $\#\cdot\mathrm{coNP}$-complete or $\#\mathrm{P}$-complete or in FP, depending on the set B of allowed connectives.

The rest of the paper is structured as follows. We first give the necessary preliminaries. Afterwards, we define the abduction problem considered herein. We then classify the complexity of the abduction of a single literal. These results are complemented with the complexity of the abduction problem for clauses, terms and restricted formulae. Next, we consider the counting problem and finally conclude with a discussion of the results.

Preliminaries

Complexity Theory We require standard notions of complexity theory. For the decision problems the arising complexity degrees encompass the classes $\mathrm{LOGSPACE}$, P, NP, and Σ_2^{p}. For more background information, the reader is referred to (Papadimitriou 1994). We furthermore require the class $\oplus\mathrm{LOGSPACE}$ defined as the class of languages L such that there exists a nondeterministic logspace Turing machine that exhibits an odd number of accepting paths if and only if $x \in L$, for all x (Buntrock et al. 1992). It holds that $\mathrm{LOGSPACE} \subseteq \oplus\mathrm{LOGSPACE} \subseteq \mathrm{P}$. For our hardness results we consider *logspace many-one reductions*, defined as follows: a language A is logspace many-one reducible to some language B (written $A \leq_{\mathrm{m}}^{\log} B$) if there exists a logspace-computable function f such that $x \in A$ if and only if $f(x) \in B$.

A *counting problem* is represented using a *witness function* w, which for every input x returns a finite set of witnesses. This witness function gives rise to the following counting problem: given an instance x, find the cardinality $|w(x)|$ of the witness set $w(x)$. The class $\#\mathrm{P}$ is the class of counting problems naturally associated with decision problems in NP. According to (Hemaspaandra and Vollmer 1995) if $\mathcal{C}$ is a complexity class of decision problems, we define $\#\cdot\mathcal{C}$ to be the class of all counting problems whose witness function is such that the size of every witness y of x is polynomially bounded in the size of x, and checking whether $y \in w(x)$ is in $\mathcal{C}$. Thus, we have $\#\mathrm{P} = \#\cdot\mathrm{P}$ and $\#\mathrm{P} \subseteq \#\cdot\mathrm{coNP}$. Completeness of counting problems is usually proved by means of Turing reductions. A stronger notion is the parsimonious reduction where the exact number of solutions is conserved by the reduction function.

Propositional formulae We assume familiarity with propositional logic. The set of all propositional formulae is denoted by $\mathcal{L}$. A *model* for a formula φ is a truth assignment to the set of its variables that satisfies φ. Further we denote by $\varphi[\alpha/\beta]$ the formula obtaine from φ by replacing all occurrences of α with β. For a given set Γ of formulae, we write $\mathrm{Vars}(\Gamma)$ to denote the set of variables occurring in Γ. We identify finite Γ with the conjunction of all the formulae in Γ, $\bigwedge_{\varphi\in\Gamma}\varphi$. For any formula $\varphi \in \mathcal{L}$, we write $\Gamma \models \varphi$ if Γ entails φ, *i.e.*, if every model of Γ also satisfies φ.

A *literal* l is a variable x or its negation $\neg x$; x is called the *atom* of l and is denoted by $|l|$. Given a set of variables V, $\mathrm{Lits}(V)$ denotes the set of all literals formed upon the variables in V, *i.e.*, $\mathrm{Lits}(V) = V \cup \{\neg x \mid x \in V\}$. A *clause* is a disjunction of literals and a *term* is a conjunction of literals.

Clones of Boolean Functions A *clone* is a set of Boolean functions that is closed under superposition, *i.e.*, it contains all projections (that is, the functions $f(a_1,\dots,a_n) = a_k$ for $1 \leq k \leq n$ and $n \in \mathbb{N}$) and is closed under arbitrary composition. Let B be a finite set of Boolean functions. We denote by $[B]$ the smallest clone containing B and call B a *base* for $[B]$. All closed classes of Boolean functions were identified by Post (1941). Post also found a finite base for each of them and detected their inclusion structure, hence the name of *Post's lattice* (see Figure 1).

In order to define the clones, we require the following notions, where f is an n-ary Boolean function:

- f is *c-reproducing* if $f(c,\dots,c) = c$, $c \in \{0,1\}$.
- f is *monotonic* if $a_1 \leq b_1, a_2 \leq b_2, \dots, a_n \leq b_n$ implies $f(a_1,\dots,a_n) \leq f(b_1,\dots,b_n)$.
- f is *c-separating of degree* k if for all $A \subseteq f^{-1}(c)$ of size $|A| = k$ there exists an $i \in \{1,\dots,n\}$ such that $(a_1,\dots,a_n) \in A$ implies $a_i = c$, $c \in \{0,1\}$.
- f is *c-separating* if f is c-separating of degree $|f^{-1}(c)|$.
- f is *self-dual* if $f \equiv \neg f(\neg x_1,\dots,\neg x_n)$.
- f is *affine* if $f \equiv x_1 \oplus \cdots \oplus x_n \oplus c$ with $c \in \{0,1\}$.

A list of all clones with definitions and finite bases is given in Table 1, see also *e.g.*, (Böhler et al. 2003). A propositional formula using only functions from B as connectives is called a *B-formula*. The set of all B-formulae is denoted by $\mathcal{L}(B)$. Let f be an n-ary Boolean function. A B-formula φ such that $\mathrm{Vars}(\varphi) = \{x_1,\dots,x_n,y_1,\dots,y_m\}$ is a *B-representation* of f if for all $a_1,\dots,a_n,b_1,\dots,b_m \in \{0,1\}$ it holds that $f(a_1,\dots,a_n) = 1$ if and only if every $\sigma\colon \mathrm{Vars}(\varphi) \longrightarrow \{0,1\}$ with $\sigma(x_i) = a_i$ and $\sigma(y_i) = b_i$ for all relevant i, satisfies φ. Such a B-representation exists for every $f \in [B]$. Yet, it may happen that the B-representation of some function uses some input variable more than once.

Example 1. Let $h(x, y) = x \wedge \neg y$. An $\{h\}$-representation of the function $x \wedge y$ is $h(x, h(x, y))$.

Observe that if B_1 and B_2 are two sets of Boolean functions such that $B_1 \subseteq [B_2]$, then every function of B_1 can be expressed by a B_2-formula, its so-called *B_2-representation*.

The Abduction Problem

Let B be a finite set of Boolean functions. We are interested in a propositional abduction problem parameterized by the set B of allowed connectives. We define the *abduction problem for B-formulae* as

Problem: $\textsc{Abd}(B)$
Instance: $\mathcal{P} = (\Gamma, A, \varphi)$, where

- Γ is a set of B-formulae, $\Gamma \subseteq \mathcal{L}(B)$,
- A is a set of variables, $A \subseteq \mathrm{Vars}(\Gamma)$,
- φ is a formula, $\varphi \in \mathcal{L}$ with $\mathrm{Vars}(\varphi) \subseteq \mathrm{Vars}(\Gamma) \setminus A$.

Question: Is there a set $E \subseteq \mathrm{Lits}(A)$ such that $\Gamma \wedge E$ is satisfiable and $\Gamma \wedge E \models \varphi$ (or equivalently $\Gamma \wedge E \wedge \neg\varphi$ is unsatisfiable)?

The set Γ represents the *knowledge base*. The set A is called the set of *hypotheses* and φ is called *manifestation* or *query*. Furthermore, if such a set E exists, it is called an *explanation* or a *solution* of the abduction problem. It is called a *full explanation* if $\mathrm{Vars}(E) = A$. Observe that every explanation can be extended to a full one.

We will consider several restrictions on the manifestations of this problem. To indicate these restrictions, we introduce a second argument $\mathcal{M}$: in the abduction problem $\textsc{Abd}(B, \mathcal{M})$, φ is required to be a single literal if $\mathcal{M} = \mathrm{Q}$, a clause if $\mathcal{M} = \mathrm{C}$, a term if $\mathcal{M} = \mathrm{T}$, and a B-formula if $\mathcal{M} = \mathcal{L}(B)$.

Let us start with a lemma that makes clear the role of the two constants 0 and 1 in our problem.

Lemma 2. *Let B be a finite set of Boolean functions*

1. *If $\mathcal{M} \in \{\mathrm{Q}, \mathrm{C}, \mathrm{T}, \mathcal{L}(B)\}$, then*
$$\textsc{Abd}(B, \mathcal{M}) \equiv_{\mathrm{m}}^{\log} \textsc{Abd}(B \cup \{1\}, \mathcal{M})$$
2. *If $\mathcal{M} \in \{\mathrm{Q}, \mathrm{C}, \mathrm{T}\}$ and $\vee \in [B]$, then*
$$\textsc{Abd}(B, \mathcal{M}) \equiv_{\mathrm{m}}^{\log} \textsc{Abd}(B \cup \{0\}, \mathcal{M})$$

Proof. To reduce $\textsc{Abd}(B \cup \{1\}, \mathcal{M})$ to $\textsc{Abd}(B, \mathcal{M})$ we transform any instance of the first problem in replacing every occurrence of 1 by a fresh variable t and adding the unit clause (t) to the knowledge base. To prove $\textsc{Abd}(B \cup \{0\}, \mathcal{M}) \equiv_{\mathrm{m}}^{\log} \textsc{Abd}(B, \mathcal{M})$, let $\mathcal{P} = (\Gamma, A, \psi)$ be an instance of the first problem and f be a fresh variable. If $\mathcal{M} \in \{\mathrm{Q}, \mathrm{C}, \mathrm{T}\}$, then we can suppose w.l.o.g. that ψ does not contain 0. We map $\mathcal{P}$ to $\mathcal{P}' = (\Gamma', A \cup \{f\}, \psi)$, where Γ' is the B-representation of $\{\varphi[0/f] \vee f \mid \varphi \in \Gamma\}$.

The Complexity of $\textsc{Abd}(B, \mathrm{Q})$

Theorem 3. *Let B be a finite set of Boolean functions. Then, the abduction problem for propositional B-formulae, $\textsc{Abd}(B, \mathrm{Q})$, is*

1. *Σ_2^{p}-complete if $\mathsf{S}_{02} \subseteq [B]$ or $\mathsf{S}_{12} \subseteq [B]$ or $\mathsf{D}_1 \subseteq [B]$,*
2. *NP-complete if $\mathsf{S}_{00} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$,*
3. *in P and $\oplus$LOGSPACE-hard if $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$, and*
4. *in LOGSPACE in all other cases.*

Remark 4. For such a classification a natural question is: given B, how hard is it to determine the complexity of $\textsc{Abd}(B, \mathrm{Q})$? Solving this task requires checking whether certain clones are included in $[B]$ (for lower bounds) and whether B itself is included in certain clones (for upper bounds). As shown in (Vollmer 2009), the complexity of checking whether certain Boolean functions are included in a clone depends on the representation of the Boolean functions. If all functions are given by their truth table then the problem is in quasi-polynomial-size AC^0, while if the input functions are given in a compact way, *i.e.*, by circuits, then the above problem becomes coNP-complete.

We split the proof of Theorem 3 into several propositions.

Proposition 5. *Let B be a finite set of Boolean functions such that $[B] \subseteq \mathsf{E}$ or $[B] \subseteq \mathsf{N}$ or $[B] \subseteq \mathsf{V}$. Then $\textsc{Abd}(B, \mathrm{Q}) \in$ LOGSPACE.*

Proof. Let $\mathcal{P} = (\Gamma, A, q)$ be an instance of $\textsc{Abd}(B, \mathrm{Q})$.

For $[B] = \mathsf{N}$ or E, Γ is equivalent to a set of literals, hence $\mathcal{P}$ has the empty set as a solution if $\mathcal{P}$ possesses a solution at all. Finally notice that satisfiability of a set of N-formulae can be tested in logarithmic space (Schnoor 2005).

For $[B] = \mathsf{V}$ each formula $\varphi \in \Gamma$ is equivalent to either a constant or disjunction. It holds that (Γ, A, q) has a solution if and only if Γ contains a formula $\varphi \equiv q \vee x_1 \vee \cdots \vee x_k$ such that $X := \{x_1, \ldots x_k\} \subseteq A$, and $\Gamma[X/0]$ is satisfiable. This can be tested in logarithmic space, as substitution of symbols and evaluation of V-formulae can all be performed in logarithmic space.

Proposition 6. *Let B be a finite set of Boolean functions such that $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$. Then $\textsc{Abd}(B, \mathrm{Q})$ is $\oplus$LOGSPACE-hard and contained in P.*

Proof. In this case, deciding whether an instance of $\textsc{Abd}(B, \mathrm{Q})$ has a solution logspace reduces to the problem of deciding whether a propositional abduction problem in which the knowledge base is a set of linear equations has a solution. This has been shown to be decidable in polynomial time in (Zanuttini 2003).

As for the $\oplus$LOGSPACE-hardness, let B be such that $[B] = \mathsf{L}_2$. Consider the $\oplus$LOGSPACE-complete problem to determine whether a system of linear equations S over $GF(2)$ has a solution (Buntrock et al. 1992). Note that $\oplus$LOGSPACE is closed under complement, so deciding whether such a system has no solution is also $\oplus$LOGSPACE-complete. Let $S = \{s_1, \ldots, s_m\}$ be such a system of linear equations over variables $\{x_1, \ldots, x_n\}$. Then, for all $1 \leq i \leq m$, the equation s_i is of the form $x_{i_1} + \cdots + x_{i_{n_i}} = c_i \pmod 2$ with $c_i \in \{0, 1\}$ and

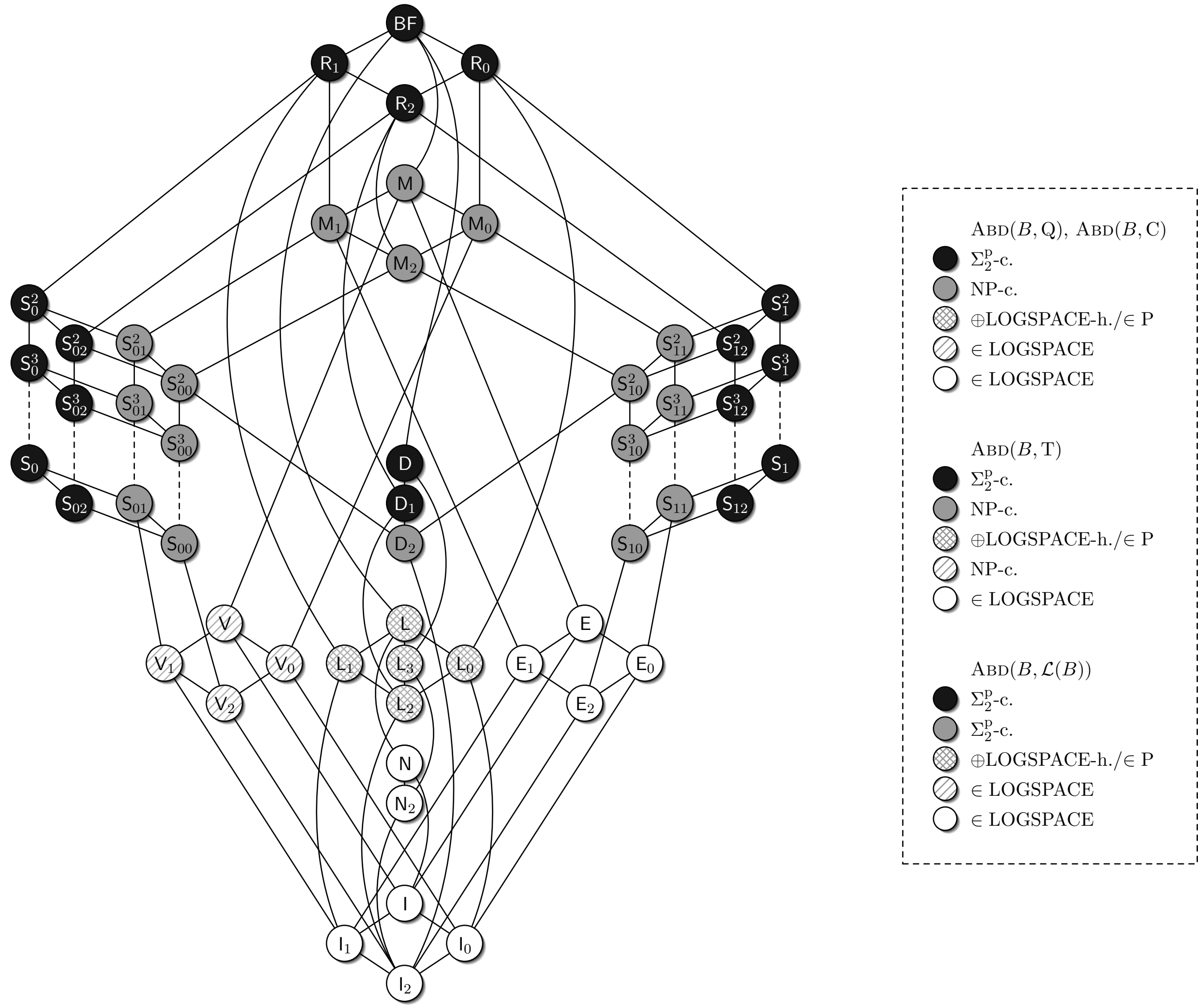

Fig. 1. Post's lattice showing the complexity of the abduction problem $\mathrm{ABD}(B, \mathcal{M})$ for all sets B of Boolean functions and considered restrictions $\mathcal{M}$ of the manifestations.

$i_1, \ldots, i_{n_i} \in \{1, \ldots, n\}$. We map S to a set of affine formulae $\Gamma = \{\varphi_1, \ldots, \varphi_m\}$ over variables $\{x_1, \ldots, x_n, q\}$ via

$$\begin{aligned} \varphi_i &:= x_{i_1} \oplus \cdots \oplus x_{i_{n_i}} \oplus 1 && \text{if } c_i = 0, \text{ and} \\ \varphi_i &:= x_{i_1} \oplus \cdots \oplus x_{i_{n_i}} && \text{if } c_i = 1. \end{aligned}$$

Now define

$$\begin{aligned} \Gamma' := &\{\varphi_i \oplus q \mid \varphi_i \in \Gamma \text{ such that } \varphi_i(1, \ldots, 1) = 0\} \\ &\cup \{\varphi_i \mid \varphi_i \in \Gamma \text{ such that } \varphi_i(1, \ldots, 1) = 1\}. \end{aligned}$$

Γ' is obviously satisfied by the assignment mapping all propositions to 1. It furthermore holds that S has no solution if and only if $\Gamma' \wedge \neg q$ is unsatisfiable. Hence, we obtain that S has no solution if and only if the propositional abduction problem $(\Gamma', \emptyset, q)$ has an explanation.

It remains to transform Γ' into a set of B-formulae in logarithmic space. Since $[B] = \mathsf{L}_2$, we have $x \oplus y \oplus z \in [B]$. We insert parentheses in every formula φ of Γ' in such a way that we get a ternary $\oplus$-tree of logarithmic depth whose leaves are either a proposition or the constant 1. Then we replace every node $\oplus$ by its equivalent B-formula. Thus we get a $(B \cup \{1\})$-formula of size polynomial in the size of the original one. Lemma 2 allows to conclude.

Note that the B-formulae replacing the connectives might use some input variable more than once. Therefore, the logarithmic depth tree is built in order to avoid an exponential explosion of the formula size during the replacement.

Observe that the abduction problem for B-formulae is self-reducible for the above cases, *i.e.*, for $[B] \subseteq \mathsf{L}$, $[B] \subseteq \mathsf{E}$ and $[B] \subseteq \mathsf{V}$. Roughly speaking this means, given an instance $\mathcal{P}$ and a literal l, we can compute efficiently an instance $\mathcal{P}'$ such that the question whether there exists an explanation E with $l \in E$ reduces to the question whether $\mathcal{P}'$ admits solutions. It is well-known that for self-reducible problems whose decision problem is in P, the lexicographically first solution can be computed in FP. It is an easy exercise to extend this algorithm to enumerate all solutions in lexicographical order with polynomial delay and polynomial space. Thus, the explanations of $\textsc{Abd}(B, \mathrm{Q})$ can be enumerated with polynomial delay and polynomial space if $[B] \subseteq \mathsf{L}$ or $[B] \subseteq \mathsf{E}$ or $[B] \subseteq \mathsf{V}$, according to Proposition 5 and 6.

Proposition 7. *Let B be a finite set of Boolean functions such that $\mathsf{S}_{00} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$. Then $\textsc{Abd}(B, \mathrm{Q})$ is* NP*-complete.*

Proof. We first show that $\textsc{Abd}(B, \mathrm{Q})$ is efficiently verifiable. Let $\mathcal{P} = (\Gamma, A, q)$ be an $\textsc{Abd}(B, \mathrm{Q})$-instance and $E \subseteq \mathrm{Lits}(A)$ be a candidate for an explanation. Define Γ' as the set of formulae obtained from Γ by replacing each occurrence of the proposition x with 0 if $\neg x \in E$, and each occurrence of the proposition x with 1 if $x \in E$. It holds that E is a solution for $\mathcal{P}$ if Γ' is satisfiable and $\Gamma'[q/0]$ is not. These tests can be performed in polynomial time, because Γ' is a set of monotonic formulae (Lewis 1979). Hence, $\textsc{Abd}(B, \mathrm{Q}) \in \mathrm{NP}$.

Next we give a reduction from the NP-complete problem 2-In-3-Sat, *i.e.*, the problem to decide whether there exists an assignment that satisfies exactly two propositions in each clause of a given formula in conjunctive normal form with exactly three positive propositions per clause, see (Schaefer 1978). Let $\varphi := \bigwedge_{i \in I} c_i$ with $c_i = x_{i1} \vee x_{i2} \vee x_{i3}$, $i \in I$, be the given formula. We map φ to the following instance $\mathcal{P} = (\Gamma, A, q)$. Let q, q_i, $i \in I$, be fresh, pairwise distinct propositions and let $A := \mathrm{Vars}(\varphi) \cup \{q_i \mid i \in I\}$. We define Γ as

$$\begin{aligned}\Gamma := &\{c_i | i \in I\} && (1)\\ &\cup \{x_{i1} \vee x_{i2} \vee q_i, x_{i1} \vee x_{i3} \vee q_i, x_{i2} \vee x_{i3} \vee q_i | i \in I\} && (2)\\ &\cup \{\textstyle\bigvee_{i \in I} \bigwedge_{j=1}^{3} x_{ij} \vee \bigvee_{i \in I} q_i \vee q\} && (3)\end{aligned}$$

We show that there is an assignment that sets to true exactly two propositions in each clause of φ if and only if $\mathcal{P}$ has a solution. First, suppose that there exists an assignment σ such that for all $i \in I$, there is a permutation π_i of $\{1, 2, 3\}$ such that $\sigma(x_{i\pi_i(1)}) = 0$ and $\sigma(x_{i\pi_i(2)}) = \sigma(x_{i\pi_i(3)}) = 1$. Thus (1) and (2) are satisfied, and (3) is equivalent to $\bigvee_{i \in I} q_i \vee q$. From this, it is readily observed that $\{\neg x \mid \sigma(x) = 0\} \cup \{\neg q_i \mid i \in I\}$ is a solution to $\mathcal{P}$.

Conversely, suppose that $\mathcal{P}$ has an explanation E that is w.l.o.g. full. Then $\Gamma \wedge E$ is satisfiable and $\Gamma \wedge E \models q$. Let $\sigma\colon \mathrm{Vars}(\Gamma) \to \{0, 1\}$ be an assignment that satisfies $\Gamma \wedge E$. Then, for any $x \in A$, $\sigma(x) = 0$ if $\neg x \in E$, and $\sigma(x) = 1$ otherwise. Since $\Gamma \wedge E$ entails q and as the only occurrence of q is in (3), we obtain that σ sets to 0 each q_i and at least one proposition in each clause of φ. Consequently, from (2) follows that σ sets to 1 at least two propositions in each clause of φ. Therefore, σ sets to 1 exactly two propositions in each clause of φ.

It remains to show that $\mathcal{P}$ can be transformed into an $\textsc{Abd}(B, \mathrm{Q})$-instance for all considered B. Observe that $\vee \in [B \cup \{1\}]$ and $[\mathsf{S}_{00} \cup \{0, 1\}] = [\mathsf{D}_2 \cup \{0, 1\}] = [\mathsf{S}_{10} \cup \{0, 1\}] = \mathsf{M}$. Therefore due to Lemma 2 it suffices to consider the case $[B] = \mathsf{M}$. Using the associativity of $\vee$ rewrite (3) as an $\vee$-tree of logarithmic depth and replace all the connectives in Γ by their B-representation ($\vee, \wedge \in [B]$).

Proposition 8. *Let B be a finite set of Boolean functions such that $\mathsf{S}_{02} \subseteq [B]$ or $\mathsf{S}_{12} \subseteq [B]$ or $\mathsf{D}_1 \subseteq [B]$. Then $\textsc{Abd}(B, \mathrm{Q})$ is Σ_2^{p}-complete.*

Proof. Membership in Σ_2^{p} is easily seen to hold: given an instance (Γ, A, q), guess an explanation E and subsequently verify that $\Gamma \wedge E$ is satisfiable and $\Gamma \wedge E \wedge \neg q$ is not.

Observe that $\vee \in [B \cup \{1\}]$. By virtue of Lemma 2 and the fact that $[\mathsf{S}_{02} \cup \{0, 1\}] = [\mathsf{S}_{12} \cup \{0, 1\}] = [\mathsf{D}_1 \cup \{0, 1\}] = \mathsf{BF}$, it suffices to consider the case $[B] = \mathsf{BF}$. In (Eiter and Gottlob 1995) it has been shown that the propositional abduction problem remains Σ_2^{p}-complete when the knowledge base Γ is a set of CNF-formulae. From such an instance (Γ, A, q) we build an instance of $\textsc{Abd}(B, \mathrm{Q})$ by rewriting first each formula as a tree of logarithmic depth and then replacing all the connectives $\wedge$,$\vee$ and $\neg$ by their B-representation, thus concluding the proof.

Complexity of the Variants

We now turn to the study of the complexity of some variants of the abduction problem. It is obvious that $\textsc{Abd}(B, \mathrm{Q}) \leq_{\mathrm{m}}^{\log} \textsc{Abd}(B, \mathrm{C})$ and that $\textsc{Abd}(B, \mathrm{Q}) \leq_{\mathrm{m}}^{\log} \textsc{Abd}(B, \mathrm{T})$. Therefore, all hardness results still hold for the variants $\textsc{Abd}(B, \mathrm{C})$ and $\textsc{Abd}(B, \mathrm{T})$. Also, it can be easily checked that the hardness results in the previous sections still hold when the query is required to be a positive literal. For this reason the hardness results also carry over to the variant $\textsc{Abd}(B, \mathcal{L}(B))$.

It is an easy exercise to prove that all algorithms that have been developed for a single query can be naturally extended to clauses. Therefore, the complexity classification for the problem $\textsc{Abd}(B, \mathrm{C})$ is exactly the same as for $\textsc{Abd}(B, \mathrm{Q})$.

Theorem 9. *Let B be a finite set of Boolean functions. Then, the abduction problem for propositional B-formulae, $\textsc{Abd}(B, \mathrm{C})$, is*

1. *Σ_2^{p}-complete if $\mathsf{S}_{02} \subseteq [B]$ or $\mathsf{S}_{12} \subseteq [B]$ or $\mathsf{D}_1 \subseteq [B]$,*
2. *NP-complete if $\mathsf{S}_{00} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$,*
3. *in P and $\oplus$LOGSPACE-hard if $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$, and*
4. *in LOGSPACE in all other cases.*

More interestingly, we will prove in the next section that allowing terms as manifestations increases the complexity for the clones V (from membership in LOGSPACE to NP-completeness), while allowing B-formulae as manifestations makes the classification dichotomous, P/Σ_2^{p}-complete, thus skipping the intermediate NP level.

The Complexity of $\textsc{Abd}(B, \mathrm{T})$

Proposition 10. *Let B be a finite set of Boolean functions such that $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$. Then $\textsc{Abd}(B, \mathrm{T})$ is* NP*-complete.*

Proof. Let B be a finite set of Boolean functions such that $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{V}$ and let $\mathcal{P} = (\Gamma, A, t)$ be an instance of $\textsc{Abd}(B, \mathrm{T})$. Hence, Γ is a set of B-formulae and t is a term, $t = \bigwedge_{i=1}^{n} l_i$. Observe that E is a solution for $\mathcal{P}$ if $\Gamma \wedge E$ is satisfiable and for every $i = 1, \ldots, n$, $\Gamma \wedge E \wedge \neg l_i$ is not. Given a set $E \subseteq \mathrm{Lits}(A)$, these verifications, which require substitution of symbols and evaluation of an $\vee$-formula, can be performed in polynomial time, thus proving membership in NP.

To prove NP-hardness, we give a reduction from 3SAT. Let φ be a 3-CNF-formula, $\varphi := \bigwedge_{i\in I} c_i$. Let $x_1,\dots,x_n$ enumerate the variables occurring in φ. Let $x'_1,\dots,x'_n$ and $q_1,\dots,q_n$ be fresh, pairwise distinct variables. We map φ to $\mathcal{P} = (\Gamma, A, t)$, where

$$\begin{aligned}\Gamma &:= \{c_i[\neg x_1/x'_1,\dots,\neg x_n/x'_n] \mid i \in I\}\\ &\quad\cup \{x_i \vee x'_i, x_i \vee q_i, x'_i \vee q_i \mid 1 \le i \le n\},\\ A &:= \{x_1,\dots,x_n,x'_1,\dots,x'_n\},\\ t &:= q_1 \wedge \dots \wedge q_n.\end{aligned}$$

We show that φ is satisfiable if and only if $\mathcal{P}$ has a solution. First assume that φ is satisfied by the assignment $\sigma\colon \{x_1,\dots,x_n\} \to \{0,1\}$. Define $E := \{\neg x_i \mid \sigma(x_i) = 0\} \cup \{\neg x'_i \mid \sigma(x_i) = 1\}$ and $\hat\sigma$ as the extension of σ mapping $\hat\sigma(x'_i) = \neg\sigma(x_i)$ and $\hat\sigma(q_i) = 1$ for all $1 \le i \le n$. Obviously, $\hat\sigma \models \Gamma \wedge E$. Furthermore, $\Gamma \wedge E \models q_i$ for all $1 \le i \le n$, because any satisfying assignment of $\Gamma \wedge E$ sets to 0 either x_i or x'_i and thus $\{x_i \vee q_i, x'_i \vee q_i\} \models q_i$. Hence E is an explanation for $\mathcal{P}$.

Conversely, suppose that $\mathcal{P}$ has a full explanation E. The facts that $\Gamma \wedge E \models q_1 \wedge \dots \wedge q_n$ and that each q_i occurs only in the clauses $x_i \vee q_i, x'_i \vee q_i$ enforce that, for every i, E contains $\neg x_i$ or $\neg x'_i$. Because of the clause $x_i \vee x'_i$, it cannot contain both. Therefore in E the value of x'_i is determined by the value of x_i and is its dual. From this it is easy to conclude that the assignment $\sigma\colon \{x_1,\dots,x_n\} \to \{0,1\}$ defined by $\sigma(x_i) = 0$ if $\neg x_i \in E$, and 1 otherwise, satisfies φ. Finally $\mathcal{P}$ can be transformed into an $\textsc{Abd}(B,\mathrm{T})$-instance, because every formula in Γ is the disjunction of at most three variables and $\vee \in [B]$.

Theorem 11. *Let B be a finite set of Boolean functions. Then, the abduction problem for propositional B-formulae, $\textsc{Abd}(B,\mathrm{T})$, is*

1. *Σ_2^{p}-complete if $\mathsf{S}_{02} \subseteq [B]$ or $\mathsf{S}_{12} \subseteq [B]$ or $\mathsf{D}_1 \subseteq [B]$,*
2. *NP-complete if $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ or $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$,*
3. *in P and $\oplus$LOGSPACE-hard if $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$, and*
4. *in LOGSPACE in all other cases.*

The Complexity of $\textsc{Abd}(B,\mathcal{L}(B))$

Proposition 12. *Let B be a finite set of Boolean functions such that $\mathsf{S}_{00} \subseteq [B]$ or $\mathsf{S}_{10} \subseteq [B]$ or $\mathsf{D}_2 \subseteq [B]$. Then $\textsc{Abd}(B,\mathcal{L}(B))$ is Σ_2^{p}-complete.*

Proof. We prove Σ_2^{p}-hardness by giving a reduction from the Σ_2^{p}-hard problem $\textsc{Qsat}_2$ (Wrathall 1977). Let an instance of $\textsc{Qsat}_2$ be given by a closed formula $\chi := \exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m \varphi$ with φ a 3-DNF-formula. First observe that $\exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m \varphi$ is true if and only if there exists a consistent set $X \subseteq \mathrm{Lits}(\{x_1,\dots,x_n\})$ such that $X \cap \{x_i, \neg x_i\} \neq \emptyset$, for all $1 \le i \le n$, and $\neg X \vee \varphi$ is (universally) valid (or equivalently $\neg\varphi \wedge X$ is unsatisfiable).

Denote by $\overline{\varphi}$ the negation normal form of $\neg\varphi$ and let $\overline{\varphi}'$ be obtained from $\overline{\varphi}$ by replacing all occurrences of $\neg x_i$ with a fresh proposition x'_i, $1 \le i \le n$, and all occurrences of $\neg y_i$ with a fresh proposition y'_i, $1 \le i \le m$. That is, $\overline{\varphi}' \equiv \overline{\varphi}[\neg x_1/x'_1,\dots,\neg x_n/x'_n,\neg y_1/y'_1,\dots,\neg y_m/y'_m]$. Thus $\overline{\varphi}' = \bigwedge_{i\in I} c'_i$ where every c'_i is a disjunction of three propositions. To χ we associate the propositional abduction problem $\mathcal{P} = (\Gamma, A, \psi)$ defined as follows:

$$\begin{aligned}\Gamma &:= \{c'_i \vee q \mid i \in I\}\\ &\quad\cup \{x_i \vee x'_i \mid 1 \le i \le n\} \cup \{y_i \vee y'_i \mid 1 \le i \le m\}\\ &\quad\cup \{f_i \vee x_i, t_i \vee x'_i, f_i \vee t_i \mid 1 \le i \le n\},\\ A &:= \{t_i, f_i \mid 1 \le i \le n\},\\ \psi &:= q \vee \textstyle\bigvee_{1\le i\le n}(x_i \wedge x'_i) \vee \bigvee_{1\le i\le m}(y_i \wedge y'_i).\end{aligned}$$

Suppose that χ is true. Then there exists an assignment $\sigma\colon \{x_1,\dots,x_n\} \to \{0,1\}$ such that no extension $\sigma'\colon \{x_1,\dots,x_n\} \cup \{y_1,\dots,y_m\} \to \{0,1\}$ of σ satisfies $\neg\varphi$. Define X as the set of literals over $\{x_1,\dots,x_n\}$ set to 1 by σ. Defining $E := \{\neg f_i, t_i \mid x_i \in X\} \cup \{\neg t_i, f_i \mid \neg x_i \in X\}$, we obtain with abuse of notation

$$\begin{aligned}&\Gamma \wedge E \wedge \neg\psi\\ \equiv\ &\textstyle\bigwedge_{i\in I} c'_i \wedge \bigwedge_{1\le i\le n}(x_i \oplus x'_i) \wedge \bigwedge_{1\le i\le m}(y_i \oplus y'_i) \wedge\\ &\textstyle\bigwedge_{1\le i\le n,\sigma(x_i)=1} x_i \wedge \bigwedge_{1\le i\le n,\sigma(x_i)=0} x'_i\\ \equiv\ &\neg\varphi \wedge X,\end{aligned}$$

which is unsatisfiable by assumption. As $\Gamma \wedge E$ is satisfied by any assignment setting in addition all x_i, x'_i, $1 \le i \le n$, and all y_j, y'_j, $1 \le i \le m$, to 1, we have proved that E is an explanation for $\mathcal{P}$.

Conversely, suppose that $\mathcal{P}$ has an explanation E. Due to the clause $(f_i \vee t_i)$ in Γ, we also may assume that $|E \cap \{\neg t_i, \neg f_i\}| \le 1$ for all $1 \le i \le n$.

Setting $X := \{x_i \mid \neg f_i \in E\} \cup \{\neg x_i \mid \neg t_i \in E\}$ we now obtain $\bigwedge_{1\le i\le n}\big((f_i \vee x_i) \wedge (t_i \vee \neg x_i) \wedge (f_i \vee t_i)\big) \wedge E \equiv X$ and $\Gamma \wedge E \wedge \neg\psi \equiv \neg\varphi \wedge X$ as above. Hence, $\neg\varphi \wedge X$ is unsatisfiable, which implies the existence of an assignment $\sigma\colon \{x_1,\dots,x_n\} \to \{0,1\}$ such that no extension $\sigma'\colon \{x_1,\dots,x_n\} \cup \{y_1,\dots,y_m\} \to \{0,1\}$ of σ satisfies $\neg\varphi$. Therefore, we have proved that χ is true if and only if $\mathcal{P}$ has an explanation.

It remains to show that $\mathcal{P}$ can be transformed into an $\textsc{Abd}(B,\mathcal{L}(B))$-instance for any relevant B. Since $[\mathsf{S}_{00} \cup \{1\}] = \mathsf{S}_{01}$, $[\mathsf{S}_{10} \cup \{1\}] = \mathsf{M}_1$, $[\mathsf{D}_2 \cup \{1\}] = \mathsf{S}_{01}^2$ and $\mathsf{S}_{01} \subseteq \mathsf{S}_{01}^2 \subseteq \mathsf{M}_1$, by Lemma 2 it suffices to consider the case $[B] = \mathsf{S}_{01}$. Observe that $\vee,\ x \vee (y \wedge z) \in [B]$. The transformation can be done in polynomial time by replacement, rewriting ψ as $\bigvee_{1\le i\le n} q \vee (x_i \wedge x'_i) \vee \bigvee_{1\le i\le m} q \vee (y_i \wedge y'_i)$ and using the associativity of $\vee$.

Theorem 13. *Let B be a finite set of Boolean functions. Then, the abduction problem for propositional B-formulae, $\textsc{Abd}(B,\mathcal{L}(B))$, is*

1. *Σ_2^{p}-complete if $\mathsf{S}_{00} \subseteq [B]$ or $\mathsf{S}_{10} \subseteq [B]$ or $\mathsf{D}_2 \subseteq [B]$,*
2. *in P and $\oplus$LOGSPACE-hard if $\mathsf{L}_2 \subseteq [B] \subseteq \mathsf{L}$, and*
3. *in LOGSPACE in all other cases.*

Counting complexity

The counting problem $\#\text{ABD}(B, \text{Q})$ we are interested in is the following: given an instance $\mathcal{P} = (\Gamma, A, \varphi)$ of $\text{ABD}(B, \text{Q})$, compute $\#\text{Sol}(\mathcal{P})$, the number of full explanations of $\mathcal{P}$.

Theorem 14. *Let B be a finite set of Boolean functions. Then, the abduction problem for propositional B-formulae, $\#\text{ABD}(B, \text{Q})$, is*

1. $\#\cdot\text{coNP}$*-complete if* $\mathsf{S}_{02} \subseteq [B]$ *or* $\mathsf{S}_{12} \subseteq [B]$ *or* $\mathsf{D}_1 \subseteq [B]$,
2. $\#\text{P}$*-complete if* $\mathsf{V}_2 \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{S}_{10} \subseteq [B] \subseteq \mathsf{M}$ *or* $\mathsf{D}_2 \subseteq [B] \subseteq \mathsf{M}$,
3. *in* FP *in all other cases.*

Proof. The $\#\cdot\text{coNP}$-membership for $\#\text{ABD}(B, \text{Q})$ follows from the fact that checking whether a set of literals is indeed an explanation for an abduction problem is in $\text{P}^{\text{NP}} = \Delta_2^{\text{p}}$ and from the equality $\#\cdot\Delta_2^{\text{p}} = \#\cdot\text{coNP}$, see (Hemaspaandra and Vollmer 1995).

We show the $\#\cdot\text{coNP}$-hardness by giving a parsimonious reduction from the following $\#\cdot\text{coNP}$-complete problem: Count the number of satisfying assignments of $\psi(x_1, \dots, x_n) = \forall y_1 \cdots \forall y_m \varphi(x_1, \dots, x_n, y_1, \dots, y_m)$, where φ is a DNF-formula (see, *e.g.*, (Durand, Hermann, and Kolaitis 2005)). Let $x'_1, \dots, x'_n$, $r_1, \dots, r_n, t$, and q be fresh, pairwise distinct propositions. We define the propositional abduction problem $\mathcal{P} = (\Gamma, A, q)$ as follows:

$$\begin{aligned} \Gamma &:= \{x_i \to r_i, x'_i \to r_i, \neg x_i \vee \neg x'_i \mid 1 \leq i \leq n\} \\ &\quad \cup \{\varphi \to t\} \cup \{\textstyle\bigwedge_{i=1}^n r_i \wedge t \to q\}, \\ A &:= \{x_1, \dots, x_n\} \cup \{x'_1, \dots, x'_n\}. \end{aligned}$$

Observe that the manifestation q occurs only in the formula $\bigwedge_{i=1}^n r_i \wedge t \to q$. This together with the formulae $x_i \to r_i, x'_i \to r_i, \neg x_i \vee \neg x'_i$, $1 \leq i \leq n$, enforces that every full explanation of $\mathcal{P}$ has to select for each i either x_i and $\neg x'_i$, or $\neg x_i$ and x'_i. By this the value of x'_i is fully determined by the value of x_i and is its dual. Moreover, it is easy to see that there is a one-to-one correspondence between the models of ψ and the full explanations of $\mathcal{P}$.

Observe that since the reductions in Lemma 2 are parsimonious, we can suppose w.l.o.g. that B contains the two constants 1 and 0. Therefore, analogously to Proposition 8 it suffices to consider the case $[B] = \mathsf{BF}$. Since Γ can be written in a normal form, it can be transformed in logarithmic space into an equivalent set of B-formulas. This provides a parsimonious reduction from the $\#\cdot\text{coNP}$-complete problem we considered to $\#\text{ABD}(B, \text{Q})$.

Note that the proposed reduction is very similar to the one proposed in (Hermann and Pichler 2007). It also proves hardness of counting the positive explanations (since there is an one-to-one correspondence between full explanations and purely positive explanations), as well as hardness of counting the subset-minimal explanations, (since all solutions are incomparable and hence subset-minimal).

Let us now consider the $\#\text{P}$-complete cases. When $[B] \subseteq \mathsf{M}$, checking whether a set of literals E is an explanation for an abduction problem with B-formulae is in P (see Proposition 7). This proves membership in $\#\text{P}$. For the hardness result, it suffices to consider the case $[B] = \mathsf{V}_2$, because the reduction provided in Lemma 2 is parsimonious and $\mathsf{V}_2 \subseteq [\mathsf{S}_{10} \cup \{1\}]$. We provide a Turing reduction from the problem $\#\text{POSITIVE-2-SAT}$, which is known to be $\#\text{P}$-complete (Valiant 1979). Let $\varphi = \bigwedge_{i=1}^k (p_i \vee q_i)$ be an instance of this problem, where p_i and q_i are propositional variables from the set $X = \{x_1, \dots, x_n\}$. Let q be a fresh proposition. Define the propositional abduction problem $\mathcal{P} = (\Gamma, A, q)$ as follows:

$$\Gamma := \{p_i \vee q_i \vee q \mid 1 \leq i \leq k\}, \qquad A := \{x_1, \dots, x_n\}.$$

It is easy to check that the number of satisfying assignments for φ is equal to $2^n - \#\text{Sol}(\mathcal{P})$. Finally, since $\vee \in [B] = \mathsf{V}_2$, $\mathcal{P}$ can easily be transformed in logarithmic space into an $\text{ABD}(B, \text{Q})$-instance.

As for the tractable cases, the clones E and N are easy; and finally, for $[B] \subseteq \mathsf{L}$, the number of full explanations is polynomial time computable according to (Hermann and Pichler 2007, Theorem 8).

Concluding Remarks

In this paper we provided a complete classification of the propositional abduction problem, $\text{ABD}(B, \mathcal{M})$ for every set B of allowed connectives. We gave results for several restrictions over the representation of manifestations. For instance our results show that when the knowledge base formulae are positive clauses (clone V) then the abduction problem is very easy (solvable in LOGSPACE), even when the manifestations are also represented by positive clauses. But its complexity jumps to NP-completeness when manifestations are represented by positive terms. When looking at the so-called monotonic fragment (clone M) the abduction problem is NP-complete when manifestations are represented by clauses or terms. Allowing also the manifestation to be a monotonic formulae as the knowledge base, it becomes Σ_2^{p}-complete. This can be intuitively explained as follows. The complexity of the abduction rests on two sources: finding a candidate explanation and checking that it is indeed a solution. The NP-complete cases that occur in our classification hold for problems for which verification can be performed in polynomial time. If both the knowledge base and the manifestation are represented by monotonic formulae, verifying a candidate explanation is coNP-complete. Further our results also show that, except for the clones L and L_c with $c \in \{0, 1, 2, 3\}$, all tractable cases are even trivial.

In order to get a more detailed picture of the complexity of abduction in this framework it would now be interesting to consider restrictions on hypotheses. A classification in which both types of restrictions are considered would help to understand how restrictions on the representation of hypotheses and manifestations influence the complexity of the propositional abduction problem. When restricting the hypotheses, as for instance when requiring the explanation to consist of positive literals only, it may happen that only one maximal candidate has to be considered, thus leading

Name	Definition	Base
BF	All Boolean functions	$\{x \wedge y, \neg x\}$
R_0	$\{f \mid f$ is 0-reproducing$\}$	$\{x \wedge y, x \oplus y\}$
R_1	$\{f \mid f$ is 1-reproducing$\}$	$\{x \vee y, x \oplus y \oplus 1\}$
R_2	$\mathsf{R}_0 \cap \mathsf{R}_1$	$\{\vee, x \wedge (y \oplus z \oplus 1)\}$
M	$\{f \mid f$ is monotonic$\}$	$\{x \vee y, x \wedge y, 0, 1\}$
M_0	$\mathsf{M} \cap \mathsf{R}_0$	$\{x \vee y, x \wedge y, 0\}$
M_1	$\mathsf{M} \cap \mathsf{R}_1$	$\{x \vee y, x \wedge y, 1\}$
M_2	$\mathsf{M} \cap \mathsf{R}_2$	$\{x \vee y, x \wedge y\}$
S_0^n	$\{f \mid f$ is 0-separating of degree $n\}$	$\{x \rightarrow y, \mathrm{dual}(h_n)\}$
S_0	$\{f \mid f$ is 0-separating$\}$	$\{x \rightarrow y\}$
S_1^n	$\{f \mid f$ is 1-separating of degree $n\}$	$\{x \wedge \neg y, h_n\}$
S_1	$\{f \mid f$ is 1-separating$\}$	$\{x \wedge \neg y\}$
S_{02}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2$	$\{x \vee (y \wedge \neg z), \mathrm{dual}(h_n)\}$
S_{02}	$\mathsf{S}_0 \cap \mathsf{R}_2$	$\{x \vee (y \wedge \neg z)\}$
S_{01}^n	$\mathsf{S}_0^n \cap \mathsf{M}$	$\{\mathrm{dual}(h_n), 1\}$
S_{01}	$\mathsf{S}_0 \cap \mathsf{M}$	$\{x \vee (y \wedge z), 1\}$
S_{00}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \vee (y \wedge z), \mathrm{dual}(h_n)\}$
S_{00}	$\mathsf{S}_0 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \vee (y \wedge z)\}$
S_{12}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2$	$\{x \wedge (y \vee \neg z), h_n\}$
S_{12}	$\mathsf{S}_1 \cap \mathsf{R}_2$	$\{x \wedge (y \vee \neg z)\}$
S_{11}^n	$\mathsf{S}_1^n \cap \mathsf{M}$	$\{h_n, 0\}$
S_{11}	$\mathsf{S}_1 \cap \mathsf{M}$	$\{x \wedge (y \vee z), 0\}$
S_{10}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \wedge (y \vee z), h_n\}$
S_{10}	$\mathsf{S}_1 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \wedge (y \vee z)\}$
D	$\{f \mid f$ is self-dual$\}$	$\{(x \wedge \neg y) \vee (x \wedge \neg z) \vee (\neg y \wedge \neg z)\}$
D_1	$\mathsf{D} \cap \mathsf{R}_2$	$\{(x \wedge y) \vee (x \wedge \neg z) \vee (y \wedge \neg z)\}$
D_2	$\mathsf{D} \cap \mathsf{M}$	$\{(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)\}$
L	$\{f \mid f$ is affine$\}$	$\{x \oplus y, 1\}$
L_0	$\mathsf{L} \cap \mathsf{R}_0$	$\{x \oplus y\}$
L_1	$\mathsf{L} \cap \mathsf{R}_1$	$\{x \oplus y \oplus 1\}$
L_2	$\mathsf{L} \cap \mathsf{R}_2$	$\{x \oplus y \oplus z\}$
L_3	$\mathsf{L} \cap \mathsf{D}$	$\{x \oplus y \oplus z \oplus 1\}$
V	$\{f \mid f$ is a disjunction of variables or constants$\}$	$\{x \vee y, 0, 1\}$
V_0	$\mathsf{V} \cap \mathsf{R}_0$	$\{x \vee y, 0\}$
V_1	$\mathsf{V} \cap \mathsf{R}_1$	$\{x \vee y, 1\}$
V_2	$\mathsf{V} \cap \mathsf{R}_2$	$\{x \vee y\}$
E	$\{f \mid f$ is a conjunction of variables or constants$\}$	$\{x \wedge y, 0, 1\}$
E_0	$\mathsf{E} \cap \mathsf{R}_0$	$\{x \wedge y, 0\}$
E_1	$\mathsf{E} \cap \mathsf{R}_1$	$\{x \wedge y, 1\}$
E_2	$\mathsf{E} \cap \mathsf{R}_2$	$\{x \wedge y\}$
N	$\{f \mid f$ depends on at most one variable$\}$	$\{\neg x, 0, 1\}$
N_2	$\mathsf{N} \cap \mathsf{R}_2$	$\{\neg x\}$
I	$\{f \mid f$ is a projection or a constant$\}$	$\{\mathrm{id}, 0, 1\}$
I_0	$\mathsf{I} \cap \mathsf{R}_0$	$\{\mathrm{id}, 0\}$
I_1	$\mathsf{I} \cap \mathsf{R}_1$	$\{\mathrm{id}, 1\}$
I_2	$\mathsf{I} \cap \mathsf{R}_2$	$\{\mathrm{id}\}$

Table 1. The list of all Boolean clones with definitions and bases, where $h_n := \bigvee_{i=1}^{n+1} \bigwedge_{j=1, j \neq i}^{n+1} x_j$ and $\mathrm{dual}(f)(a_1, \dots, a_n) = \neg f(\neg a_1 \dots, \neg a_n)$.

to further trivial but also some coNP-complete cases. Also note that the upper bounds in the case of the clone L relies on Gaussian elimination (Zanuttini 2003). This method fails when restricting the hypothesis to be positive. Determining the complexity of abduction for the clone L when manifestations are represented by L-formulae and explanations have to be positive might hence prove to be a challenging task (notice that this case remained unclassified in circumscriptive inference (Thomas 2009)).

References

Amilhastre, J.; Fargier, H.; and Marquis, P. 2002. Consistency restoration and explanations in dynamic CSPs. *Artif. Intell.* 135(1-2):199–234.

Bauland, M.; Hemaspaandra, E.; Schnoor, H.; and Schnoor, I. 2006. Generalized modal satisfiability. In *Proc. 23rd STACS*, volume 3884 of *LNCS*, 500–511.

Bauland, M.; Schneider, T.; Schnoor, H.; Schnoor, I.; and Vollmer, H. 2008. The complexity of generalized satisfiability for linear temporal logic. In *Logical Methods in Computer Science*, volume 5.

Beyersdorff, O.; Meier, A.; Thomas, M.; and Vollmer, H. 2009a. The complexity of propositional implication. *Information Processing Letters* 109(18):1071–1077.

Beyersdorff, O.; Meier, A.; Thomas, M.; and Vollmer, H. 2009b. The complexity of reasoning for fragments of default logic. In *Proc. 12th SAT*, volume 5584 of *LNCS*, 51–64.

Böhler, E.; Creignou, N.; Reith, S.; and Vollmer, H. 2003. Playing with Boolean blocks I: Post's lattice with applications to complexity theory. *SIGACT News* 34(4):38–52.

Bouzid, M., and Ligeza, A. 2000. Temporal causal abduction. *Constraints* 5(3):303–319.

Buntrock, G.; Damm, C.; Hertrampf, U.; and Meinel, C. 1992. Structure and importance of logspace MOD-classes. *Mathematical Systems Theory* 25:223–237.

Bylander, T.; Allemang, D.; Tanner, M. C.; and Josephson, J. R. 1989. Some results concerning the computational complexity of abduction. In *Proc. 1st KR*, 44–54.

Creignou, N., and Zanuttini, B. 2006. A complete classification of the complexity of propositional abduction. *SIAM J. Comput.* 36(1):207–229.

Durand, A.; Hermann, M.; and Kolaitis, P. G. 2005. Subtractive reductions and complete problems for counting complexity classes. *Theoretical Computer Science* 340(3):496–513.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *J. ACM* 42(1):3–42.

Hemaspaandra, L., and Vollmer, H. 1995. The satanic notations: counting classes beyond #P and other definitional adventures. *Complexity Theory Column 8, ACM-SIGACT News* 26(1):2–13.

Hermann, M., and Pichler, R. 2007. Counting complexity of propositional abduction. In *Proc. 20th IJCAI*, 417–422.

Hobbs, J. R.; Stickel, M. E.; Appelt, D. E.; and Martin, P. A. 1993. Interpretation as abduction. *Artif. Intell.* 63(1-2):69–142.

Lewis, H. 1979. Satisfiability problems for propositional calculi. *Mathematical Systems Theory* 13:45–53.

Nordh, G., and Zanuttini, B. 2005. Propositional abduction is almost always hard. In *Proc. 19th IJCAI*, 534–539.

Nordh, G., and Zanuttini, B. 2008. What makes propositional abduction tractable. *Artif. Intell.* 172(10):1245–1284.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley.

Post, E. 1941. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies* 5:1–122.

Reith, S. 2003. On the complexity of some equivalence problems for propositional calculi. In *Proc. 28th MFCS*, volume 2747 of *LNCS*, 632–641.

Schaefer, T. J. 1978. The complexity of satisfiability problems. In *Proc. 10th STOC*, 216–226.

Schnoor, H. 2005. The complexity of the Boolean formula value problem. Technical report, Theoretical Computer Science, University of Hannover.

Selman, B., and Levesque, H. 1990. Abductive and default reasoning: A computational core. In *Proc. 8th AAAI*, 343–348.

Stumptner, M., and Wotawa, F. 2001. Diagnosing tree-structured systems. *Artif. Intell.* 127(1):1–29.

Thomas, M. 2009. The complexity of circumscriptive inference in Post's lattice. In *Proc. 10th LPNMR*, volume 5753 of *Lecture Notes in Computer Science*, 290–302.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM J. Comput.* 8(3):411–421.

Vollmer, H. 2009. The complexity of deciding if a boolean function can be computed by circuits over a restricted basis. *Theory of Computing Systems* 44(1):82–90.

Wrathall, C. 1977. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science* 3:23–33.

Zanuttini, B. 2003. New polynomial classes for logic-based abduction. *J. Artif. Intell. Res.* 19:1–10.