In: Proc. 6th International Conference on Discovery Science, LNAI 2843, © 2003 Springer.

Abduction and the Dualization Problem*

Thomas Eiter[1] and Kazuhisa Makino[2]

[1] Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Wien, Austria, eiter@kr.tuwien.ac.at
[2] Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan, makino@sys.es.osaka-u.ac.jp

Abstract. Computing abductive explanations is an important problem, which has been studied extensively in Artificial Intelligence (AI) and related disciplines. While computing some abductive explanation for a literal χ with respect to a set of abducibles A from a Horn propositional theory Σ is intractable under the traditional representation of Σ by a set of Horn clauses, the problem is polynomial under model-based theory representation, where Σ is represented by its characteristic models. Furthermore, computing all the (possibly exponentially) many explanations is polynomial-time equivalent to the problem of dualizing a positive CNF, which is a well-known problem whose precise complexity in terms of the theory of NP-completeness is not known yet. In this paper, we first review the monotone dualization problem and its connection to computing all abductive explanations for a query literal and some related problems in knowledge discovery. We then investigate possible generalizations of this connection to abductive queries beyond literals. Among other results, we find that the equivalence for generating all explanations for a clause query (resp., term query) χ to the monotone dualization problem holds if χ contains at most k positive (resp., negative) literals for constant k, while the problem is not solvable in polynomial total-time, i.e., in time polynomial in the combined size of the input and the output, unless P=NP for general clause resp. term queries. Our results shed new light on the computational nature of abduction and Horn theories in particular, and might be interesting also for related problems, which remains to be explored.

Keywords: abduction, monotone dualization, hypergraph transversals, Horn functions, model-based reasoning, polynomial total-time computation, NP-hardness.

1 Introduction

Abduction is a fundamental mode of reasoning, which was extensively studied by C.S. Peirce [54]. It has taken on increasing importance in Artificial Intelligence (AI) and related disciplines, where it has been recognized as an important principle of common-sense reasoning (see e.g. [9]). Abduction has applications in many areas of AI and

* This work was supported in part by the Austrian Science Fund (FWF) Project Z29-N04, by a TU Wien collaboration grant, and by the Scientific Grant in Aid of the Ministry of Education, Science, Sports, Culture and Technology of Japan.

Computer Science including diagnosis, database updates, planning, natural language understanding, learning etc. (see e.g. references in [21]), where it is primarily used for generating explanations. Specialized workshops have been held un the recent years, in which the nature and interrelation with other modes of reasoning, in particular induction and deduction, have been investigated.

In a logic-based setting, abduction can be seen as the task to find, given a set of formulas Σ (the *background theory*), a formula χ (the *query*), and a set of formulas A (the *abducibles* or *hypotheses*), a minimal subset E of A such that Σ plus E is satisfiable and logically entails χ (i.e., an *explanation*). A frequent scenario is where Σ is a propositional Horn theory, χ is a single literal or a conjunction of literals, and A contains literals. For use in practice, the computation of abductive explanations in this setting is an important problem, for which well-known early systems such as Theorist [55] or ATMS solvers [13, 56] have been devised. Since then, there has been a growing literature on this subject.

Computing some explanation for a query literal χ from a Horn theory Σ w.r.t. assumptions A is a well-known NP-hard problem [57], even if χ and A are positive. Much effort has been spent on studying various input restrictions, cf. [29, 11, 27, 16, 15, 21, 57–59], in order to single out tractable cases of abduction. For example, the case where A comprises all literals is tractable; such explanations are *assumption-free explanations*.

It turned out that abduction is tractable in model-based reasoning, which has been proposed as an alternative form of representing and accessing a logical knowledge base, cf. [14, 34–36, 42, 43]. Model-based reasoning can be seen as an approach towards Levesque's notion of "vivid" reasoning [44], which asks for a more straight representation of the background theory Σ from which common-sense reasoning is easier and more suitable than from the traditional formula-based representation. In model-based reasoning, Σ is represented by a subset S of its models, which are commonly called *characteristic models*, rather than by a set of formulas. Given a suitable query χ, the test for $\Sigma \models \chi$ becomes then as easy as to check whether χ is true in all models of S, which can be decided efficiently. We here mention that formula-based and the model-based approach are orthogonal, in the sense that while a theory may have small representation in one formalism, it has an exponentially larger representation in the other. The intertranslatability of the two approaches, in particular for Horn theories, has been addressed in [34–36, 40, 42]. Several techniques for efficient model-based representation of various fragments of propositional logic have been devised, cf. [35, 42, 43].

As shown by Kautz *et al.*, an explanation for a positive literal $\chi = q$ w.r.t. assumptions A from a Horn theory Σ, represented by its set of characteristic models, $char(\Sigma)$, can be computed in polynomial time [34, 35, 42]; this results extends to negative literal queries $\chi = \overline{q}$ as well, and has been generalized by Khardon and Roth [42] to other fragments of propositional logic. Hence, model-based representation is attractive from this view point of finding efficiently some explanation.

While computing *some* explanation of a query χ has been studied extensively in the literature, computing multiple or even *all* explanations for χ has received less attention. However, this problem is important, since often one would like to select one out of a set of alternative explanations according to a preference or plausibility relation; this relation may be based on subjective intuition and thus difficult to formalize. As easily seen, exponentially many explanations may exist for a query, and thus computing all explanations inevitably requires exponential time in general, even in propositional logic. However, it is of interest whether the computation is possible in *polynomial total-time* (or *output-polynomial time*), i.e., in time polynomial in the combined size of the input and the output. Furthermore, if exponential space is prohibitive, it is of interest to know whether a few explanations (e.g., polynomially many) can be generated in polynomial time, as studied by Selman and Levesque [58].

In general, computing all explanations for a literal χ (positive as well as negative) w.r.t. assumptions A from a Horn theory Σ is under formula-based representation not possible in polynomial total-time unless $\mathrm{P}{=}\mathrm{NP}$; this can be shown by standard arguments appealing to the NP-hardness of deciding the existence of some explanation. For generating all assumption-free explanations for a positive literal, a resolution-style procedure has been presented in [24] which works in polynomial total-time, while for a negative literal no polynomial total-time algorithm exists unless $\mathrm{P}{=}\mathrm{NP}$ [25].

However, under model-based representation, such results are not known. It turned out that generating all explanation for a literal is polynomial-time equivalent to the problem of dualizing a monotone CNF expression (cf. [2, 20, 28]), as shown in [24]. Here, polynomial-time equivalence means mutual polynomial-time transformability between deterministic functions, i.e., A reduces to B, if there are polynomial-time functions f, g such that for any input I of A, $f(I)$ is an input of B, and if O is the output for $f(I)$, then $g(O)$ is the output of I, cf. [52]; moreover, O is requested to have size polynomial in the size of the output for I (otherwise, trivial reductions may exist).

This result, for definite Horn theories and positive literals, is implicit also in earlier work on dependency inference [49, 50], and is closely related to results in [40].

The monotone dualization problem is an interesting open problem in the theory of NP-completeness (cf. [45, 53]), which has a number of applications in different areas of Computer Science, [2, 19], including logic and AI [22]; the problem is reviewed in Section 2.2 where also briefly related problems in knowledge discovery are mentioned.

In the rest of this paper, we first review the result on equivalence between monotone dualization and generating all explanations for a literal under model-based theory representation. We then consider possible generalizations of this result for queries χ beyond literals, where we consider DNF, CNF and important special cases such as a clause and a term (i.e., a conjunction of literals). Note that the explanations for single clause queries correspond to the minimal support clauses for a clause in Clause Management Systems [56, 38, 39]. Furthermore, we shall consider on the fly also some of these cases under formula-based theory representation. Our aim will be to elucidate the frontier of monotone dualization equivalent versus intractable instances, i.e., not solvable in polynomial total-time unless $\mathrm{P}{=}\mathrm{NP}$, of the problem. It turns out that indeed the results in [24] generalize to clause and term queries under certain restrictions. In particular, the equivalence for generating all explanations for a clause query (resp., term query) χ to the monotone dualization holds if χ contains at most k positive (resp., negative) literals for constant k, while the problem is not solvable in polynomial total-time unless $\mathrm{P}{=}\mathrm{NP}$ for general clause (resp., term) queries.

Our results shed new light on the computational nature of abduction and Horn theories in particular, and might be interesting also for related problems, which remains to be explored.

2 Notation and Concepts

We assume a propositional (Boolean) setting with atoms $x_1, x_2, \ldots, x_n$ from a set At, where each x_i takes either value 1 (true) or 0 (false). Negated atoms are denoted by $\overline{x}_i$, and the opposite of a literal ℓ by $\overline{\ell}$. Furthermore, we use $\overline{A} = \{\overline{\ell} \mid \ell \in A\}$ for any set of literals A and set $Lit = At \cup \overline{At}$. A *theory* Σ is any finite set of formulas.

A clause is a disjunction $c = \bigvee_{p \in P(c)} p \vee \bigvee_{p \in N(c)} \overline{p}$ of literals, where $P(c)$ and $N(c)$ are respectively the sets of atoms occurring positively and negatively in c and $P(c) \cap N(c) = \emptyset$. Dually, a term is a conjunction $t = \bigwedge_{p \in P(t)} p \wedge \bigwedge_{p \in N(t)} \overline{p}$ of literals, where $P(t)$ and $N(t)$ are similarly defined. A conjunctive normal form (CNF) is a conjunction of clauses, and a disjunctive normal form (DNF) is a disjunction of terms. As common, we also view clauses c and terms t as the sets of literals ℓ they contain, and similarly CNFs φ and DNFs ψ as sets of clauses and terms, respectively, and write $\ell \in c$, $c \in \varphi$ etc.

A clause c is *prime* w.r.t. theory Σ, if $\Sigma \models c$ but $\Sigma \not\models c'$ for every $c' \subset c$. A CNF φ is *prime*, if each $c \in \varphi$ is prime, and irredundant, if $\varphi \setminus \{c\} \not\equiv \varphi$ for every $c \in \varphi$. Prime terms and irredundant prime DNFs are defined analogously.

A clause c is *Horn*, if $|P(c)| \leq 1$ and *negative* (resp., *positive*), if $|P(c)| = 0$ (resp., $|N(c)| = 0$). A CNF is *Horn* (resp., *negative*, *positive*), if it contains only Horn clauses (resp., negative, positive clauses). A theory Σ is *Horn*, if it is a set of Horn clauses. As usual, we identify Σ with $\varphi = \bigwedge_{c \in \Sigma} c$.

Example 1. The CNF $\varphi = (\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_4 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_4 \vee \overline{x}_5 \vee x_1) \wedge (\overline{x}_2 \vee \overline{x}_5 \vee x_3)$ over $At = \{x_1, x_2, \ldots, x_5\}$ is Horn. □

The following proposition is well-known.

Proposition 1. *Given a Horn CNF φ and a clause c, deciding whether $\varphi \models c$ is possible in polynomial time (in fact, in linear time, cf.* [18]*).*

Horn theories have a well-known semantic characterization. A *model* is a vector $v \in \{0,1\}^n$, whose i-th component is denoted by v_i. For $B \subseteq \{1, \ldots, n\}$, we let x^B be the model v such that $v_i = 1$, if $i \in B$ and $v_i = 0$, if $i \notin B$, for $i \in \{1, \ldots, n\}$. The notions of satisfaction $v \models \varphi$ of a formula φ and consequence $\Sigma \models \varphi$, $\psi \models \varphi$ etc. are as usual; the set of models of φ (resp., theory Σ), is denoted by $mod(\varphi)$ (resp., $mod(\Sigma)$). In the example above, the vector $u = (0,1,0,1,0)$ is a model of φ, i.e., $u \models \varphi$.

For models v, w, we denote by $v \leq w$ the usual componentwise ordering, i.e., $v_i \leq w_i$ for all $i = 1, 2, \ldots, n$, where $0 \leq 1$; $v < w$ means $v \neq w$ and $v \leq w$. For any set of models M, we denote by $\max(M)$, (resp., $\min(M)$) the set of all maximal (resp., minimal) models in M. We denote by $v \bigwedge w$ componentwise AND of vectors $v, w \in \{0,1\}^n$ (i.e., their intersection), and by $Cl_\wedge(S)$ the closure of $S \subseteq \{0,1\}^n$ under $\bigwedge$. Then, a theory Σ is Horn representable, iff $mod(\Sigma) = Cl_\wedge(mod(\Sigma))$.

Example 2. Consider $M_1 = \{(0101), (1001), (1000)\}$ and $M_2 = \{(0101),\ (1001),\ (1000),\ (0001),\ (0000)\}$. Then, for $v = (0101)$, $w = (1000)$, we have $w, v \in M_1$, while $v \bigwedge w = (0000) \notin M_1$; hence M_1 is not the set of models of a Horn theory. On the other hand, $Cl_\wedge(M_2) = M_2$, thus $M_2 = mod(\Sigma_2)$ for some Horn theory Σ_2.

As discussed by Kautz *et al.* [34], a Horn theory Σ is semantically represented by its characteristic models, where $v \in mod(\Sigma)$ is called *characteristic* (or *extreme* [14]), if $v \notin Cl_\wedge(mod(\Sigma) \setminus \{v\})$. The set of all such models, the *characteristic set of* Σ, is denoted by $char(\Sigma)$. Note that $char(\Sigma)$ is unique. E.g., $(0101) \in char(\Sigma_2)$, while $(0000) \notin char(\Sigma_2)$; we have $char(\Sigma_2) = M_1$. The following proposition is compatible with Proposition 1

Proposition 2. *Given* $char(\Sigma)$*, and a clause* c*, deciding whether* $\Sigma \models c$ *is possible in polynomial time* (*in fact, in linear time, cf.* [34, 26]).

The model-based reasoning paradigm has been further investigated e.g. in [40, 42], where also theories beyond Horn have been considered [42].

2.1 Abductive explanations

The notion of an abductive explanation can be formalized as follows.

Definition 1. Given a (Horn) theory Σ, called the background theory, a CNF χ (called *query*), and a set of literals $A \subseteq Lit$ (called *abducibles*), an *explanation of* χ *w.r.t.* A is a minimal set of literals E over A such that

(i) $\Sigma \cup E \models \chi$, and
(ii) $\Sigma \cup E$ is satisfiable.

If $A = Lit$, then we call E simply an *explanation of* χ.

The above definition generalizes the *assumption-based explanations* of [58], which emerge as $A{=}P' \cup \overline{P'}$ where $P' \subseteq P$ (i.e., A contains all literals over a subset P' of the letters) and $\chi = q$ for some atom q. Furthermore, in some texts (e.g., [21]) explanations must be sets of positive literals, and χ is restricted to a special form; in [21], χ is requested to be a conjunction of atoms.

The following characterization of explanations is immediate from the definition.

Proposition 3. *For any theory* Σ*, any query* χ*, and any* $E \subseteq A (\subseteq Lit)$*,* E *is an explanation for* χ *w.r.t.* A *from* Σ *iff the following conditions hold: (i)* $\Sigma \cup E$ *is satisfiable, (ii)* $\Sigma \cup E \models \chi$*, and (iii)* $\Sigma \cup (E \setminus \{\ell\}) \not\models \chi$*, for every* $\ell \in E$.

Example 3. Reconsider the Horn CNF $\varphi = (\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_4 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_4 \vee \overline{x}_5 \vee x_1) \wedge (\overline{x}_2 \vee \overline{x}_5 \vee x_3)$ from above. Suppose we want to explain $\chi = x_2$ from $A = \{x_1, x_4\}$. Then, we find that $E = \{x_1\}$ is an explanation. Indeed, $\Sigma \cup \{x_1\} \models x_2$, and $\Sigma \cup \{x_1\}$ is satisfiable; moreover, E is minimal. On the other hand, $E' = \{x_1, \overline{x}_4\}$ satisfies (i) and (ii) for $\chi = x_2$, but is not minimal. $\square$

We note that there is a close connection between the explanations of a literal and the prime clauses of a theory.

Proposition 4 (cf. [56, 32]). *For any theory Σ and literal χ, a set $E \subseteq A(\subseteq Lit)$ with $E \neq \{\chi\}$ is an explanation of χ w.r.t. A, iff the clause $c = \bigvee_{\ell \in E} \overline{\ell} \vee \chi$ is a prime clause of Σ.*

Thus, computing explanations for a literal is easily seen to be polynomial-time equivalent to computing prime clauses of a certain form. We refer here to [51] for an excellent survey of techniques for computing explanations via computing prime implicates and related problems.

2.2 Dualization problem

Dualization of Boolean functions (i.e., given a formula φ defining a function f, compute a formula ψ for the dual function f^d) is a well-known computational problem. The problem is trivial if ψ may be any formula, since we only need to interchange $\vee$ and $\wedge$ in φ (and constants 0 and 1, if occurring). The problem is more difficult if ψ should be of special form. In particular, if φ is a CNF and ψ should be a irredundant prime CNF (to avoid redundancy); this problem is known as DUALIZATION [22]. For example, if $\varphi = (x_1 \vee x_3)(x_2 \vee \overline{x}_3)$, then a suitable ψ would be $(x_2 \vee x_3)(x_1 \vee \overline{x}_3)$, since $(x_1 \wedge x_3) \vee (x_2 \wedge \overline{x}_3) \equiv (x_1 \vee x_2)(x_2 \vee x_3)(x_1 \vee \overline{x}_3)$ simplifies to it.

Clearly, ψ may have size exponential in the size of φ, and thus the issue is here whether a polynomial total-time algorithms exists (rather than one polynomial in the input size). While it is easy to see that the problem is not solvable in polynomial total unless P=NP, this result could neither be established for the important class of positive (monotone) Boolean functions so far, nor is a polynomial total-time algorithm known to date, cf. [23, 28, 37]. Note that for this subproblem, denoted MONOTONE DUALIZATION, the problem looks simpler: all prime clauses of a monotone Boolean function f are positive and f has a unique prime CNF, which can be easily computed from any given CNF (just remove all negative literals and non-minimal clauses). Thus, in this case the problem boils down to convert a prime positive DNF φ' constructed from φ into the equivalent prime (monotone) CNF.

An important note is that MONOTONE DUALIZATION is intimately related to its decisional variant, MONOTONE DUAL, since MONOTONE DUALIZATION is solvable in polynomial total-time iff MONOTONE DUAL is solvable in polynomial time cf. [2]. MONOTONE DUAL consists of deciding whether a pair of CNFs φ, ψ whether ψ is the prime CNF for the dual of the monotone function represented by φ (strictly speaking, this is a promise problem [33], since valid input instances are not recognized in polynomial time. For certain instances such as positive φ, this is ensured).

A landmark result on MONOTONE DUAL was [28], which presents an algorithm solving the problem in time $n^{o(\log n)}$. More recently, algorithms have been exhibited [23, 37] which show that the complementary problem can be solved with limited nondeterminism in polynomial time, i.e, by a nondeterministic polynomial-time algorithm that makes only a poly-logarithmic number of guesses in the size of the input. Although it is still open whether MONOTONE DUAL is polynomially solvable, several relevant tractable classes were found by various authors (see e.g. [8, 12, 17, 20, 30, 47, 47] and references therein).

A lot of research efforts have been spent on MONOTONE DUALIZATION and MONOTONE DUAL (see survey papers, e.g. [45, 53, 22]), since a number of problems turned out to be polynomial-time equivalent to this problem; see e.g. [2, 19, 20] and the more paper [22]. *Polynomial-time equivalence* of computation problems Π and Π' is here understood in the sense that problem Π reduces to Π' and vices versa, where Π reduces to Π', if there a polynomial functions f, g s.t. for any input I of Π, $f(I)$ is an input of Π', and if O is the output for $f(I)$, then $g(O)$ is the output of I, cf. [52]; moreover, O is requested to have size polynomial in the size of the output for I (if not, trivial reductions may exist).

Of the many problems to which MONOTONE DUALIZATION is polynomially equivalent, we mention here computing the transversal hypergraph of a hypergraph (known as TRANSVERSAL ENUMERATION (TRANS-ENUM)) [22]. A *hypergraph* $\mathcal{H} = (V, E)$ is a collection E of subsets $e \subseteq V$ of a finite set V, where the elements of E are called *hyperedges* (or simply *edges*). A *transversal* of $\mathcal{H}$ is a set $t \subseteq V$ that meets every $e \in E$, and is *minimal*, if it contains no other transversal properly. The *transversal hypergraph* of $\mathcal{H}$ is then the unique hypergraph $Tr(\mathcal{H}) = (V, T)$ where T are all minimal transversals of $\mathcal{H}$. Problem TRANS-ENUM is then, given a hypergraph $\mathcal{H} = (V, E)$, to generate all the edges of $Tr(\mathcal{H})$; TRANS-HYP is deciding, given $\mathcal{H} = (V, E)$ and a set of minimal transversals T, whether $Tr(\mathcal{H}) = (V, T)$.

There is a simple correspondence between MONOTONE DUALIZATION and TRANS-ENUM: For any positive CNF φ on At representing a Boolean function f, the prime CNF ψ for the dual of f consists of all clauses c such that $c \in Tr(At, \varphi)$ (viewing φ as set of clauses). E.g., if $\varphi = (x_1 \vee x_2)x_3$, then $\psi = (x_1 \vee x_3)(x_2 \vee x_3)$.

As for computational learning theory, MONOTONE DUAL resp. MONOTONE DUALIZATION are of relevance in the context of exact learning, cf. [2, 31, 47, 48, 17], which we briefly review here.

Let us consider the exact learning of DNF (or CNF) formulas of monotone Boolean functions f by membership oracles only, i.e., the problem of identifying a prime DNF (or prime CNF) of an unknown monotone Boolean function f by asking queries to an oracle whether $f(v)$=1 holds for some selected models v. It is known [1] that monotone DNFs (or CNFs) are not exact learnable with membership oracles alone in time polynomial in the size of the target DNF (or CNF) formula, since information theoretic barriers impose a $|\mathrm{CNF}(f)| + |\mathrm{DNF}(f)|$ lower bound on the number of queries needed, where $|\mathrm{CNF}(f)|$ and $|\mathrm{DNF}(f)|$ denote the numbers of prime implicates and prime implicants of f, respectively. This fact raises the following question:

- Can we identity both the prime DNF and CNF of an unknown monotone function f by membership oracles alone in time polynomial in $|\mathrm{CNF}(f)| + |\mathrm{DNF}(f)|$?

Since the prime DNF (resp., prime CNF) corresponds one-to-one to the set of all minimal true models (resp., all maximal false models) of f, the above question can be restated in the following natural way [2, 47]: Can we compute the *boundary* between true and false areas of an unknown monotone function in polynomial total-time ? There should be a simple algorithm for the problem as follows, which uses a DNF h and a CNF h' consisting of some prime implicants and prime implicates of f, respectively, such that $h \models \varphi$ and $\varphi \models h'$, for any formula φ representing f:

Step 1. Set h and h' to be empty (i.e., falsity and truth).
Step 2. **while** $h \not\equiv h'$ **do**

Take a counterexample x of $h \equiv h'$;
if $f(x) = 1$ **then**
begin
Minimize $t = \bigwedge_{i:x_i=1} x_i$ to a prime implicant t^* of f;
$h := h \vee t^*$ (i.e., add t^* to h);
end
else /* $f(x) = 0$ */
begin
Minimize $c = \bigvee_{i:x_i=0} x_i$ to a prime implicate c^* of f;
$h' := h' \wedge c^*$ (i.e., add c^* to h');
end

Step 3. Output h and h'.

This algorithm needs $O(n(|\mathrm{CNF}(f)| + |\mathrm{DNF}(f)|))$ many membership queries. If $h \equiv h'$ (i.e., the pair (h^d, h) is a Yes instance of MONOTONE DUAL) can always be decided in polynomial time, then the algorithm is polynomial in n, $\mathrm{CNF}(f)$, and $\mathrm{DNF}(f)$. (The converse is also known [2], i.e., if the above exact learning problem is solvable in polynomial total time, then MONOTONE DUAL is polynomially solvable.) Of course, other kinds of algorithms exist; for example, [31] derived an algorithm with different behavior and query bounds.

Thus, for the classes $\mathcal{C}$ of monotone Boolean functions which enjoying that

(i) MONOTONE DUAL is polynomially solvable and a counterexample is found in polynomial time in case (which is possible under mild conditions, cf. [2]), and
(ii) the family of prime DNFs (or CNFs) is *hereditary*, i.e., if a function with the prime DNF $\phi = \bigvee_{i \in I} t_i$ is in $\mathcal{C}$, then any function with the prime DNF $\phi_S = \bigvee_{i \in S} t_i$, where $S \subseteq I$, is in $\mathcal{C}$,

the above is a simple polynomial time algorithm which uses polynomially many queries within the optimal bound (assuming that $|\mathrm{CNF}(f)| + |\mathrm{DNF}(f)|$ is at least of order n). For many classes of monotone Boolean functions, we thus can get the learnability results from the results of MONOTONE DUAL, e.g., k-CNFs, k-clause CNFs, read-k CNFs, and k-degenerate CNFs [20, 23, 17].

In knowledge discovery, MONOTONE DUALIZATION and MONOTONE DUAL are relevant in the context of several problems. For example, it is encountered in computing maximal frequent and minimal infrequent sets [6], in dependency inference and key computation from databases [49, 50, 19], which we will address in Sections 3 and 4.2 below, as well as in translating between models of a theory and formula representations [36, 40]. Moreover, their natural generalizations have been studied to model various interesting applications [4, 5, 7].

3 Explanations and Dualization

Deciding the existence of some explanation for a literal $\chi = \ell$ w.r.t. an assumption set A from a Horn Σ is NP-complete under formula representation (i.e., Σ is given by a

Horn CNF), for both positive and negative ℓ, cf. [57, 24]; hence, generating some resp. all explanations is intractable in very elementary cases (however, under restrictions such as $A = Lit$ for positive ℓ, the problem is tractable [24]).

In the model-based setting, matters are different, and there is a close relationship between abduction and monotone dualization. If we are given $char(\Sigma)$ of a Horn theory Σ and an atom q, computing an explanation E for q from Σ amounts to computing a minimal set E of letters such that (i) at least one model of Σ (and hence, a model in $char(\Sigma)$) satisfies E and q, and that (ii) each model of Σ falsifying q also falsifies E; this is because an atom q has only *positive* explanations E, i.e., it contains only positive literals (see e.g. [42] for a proof). Viewing models as $v = x^B$, then (ii) means that E is a minimal transversal of the hypergraph (V, M) where V corresponds to the set of the variables and M consists of all $V - B$ such that $x^B \in char(\Sigma)$ and $x^B \not\models q$.

This led Kautz *et al.* [34] to an algorithm for computing an explanation E for $\chi = q$ w.r.t. a set of atoms $A \subseteq At$ which essentially works as follows:

1. Take a model $v \in char(\Sigma)$ such that $v \models q$.
2. Let $V := A \cap B$ and $M = \{V \setminus B' \mid x^{B'} \in char(\Sigma),\ q \notin B'\}$, where $v = x^B$.
3. if $\emptyset \notin M$, compute a minimal transversal E of $\mathcal{H} = (V, M)$, and output E; otherwise, select another v in Step 1 (if no other is left, terminate with no output).

In this way, some explanation E for q w.r.t. A can be computed in polynomial time, since computing some minimal transversal of a hypergraph is clearly polynomial. Recall that under formula-based representation, this problem is NP-hard [57, 58]. The method above has been generalized to arbitrary theories represented by models using Bshouty's Monotone Theory [10] and positive abducibles A, as well as for other settings, by Khardon and Roth [42] (cf. also Section 4.2).

Also all explanations of q can be generated in the way above, by taking all models $v \in char(\Sigma)$ and all minimal transversals of (V, M). In fact, in Step 1 v can be restricted to the maximal vectors in $char(\Sigma)$. Therefore, computing all explanations reduces to solving a number of transversal computation problems (which trivially amount to monotone dualization problems) in polynomial time. As shown in [24], the latter can be polynomially reduced to a single instance.

Conversely, monotone dualization can be easily reduced to explanation generation, cf. [24]. This established the following result.

Theorem 1. *Given $char(\Sigma)$ of a Horn theory Σ, a query q, and $A \subseteq Lit$, computing the set of all explanations for q from Σ w.r.t. A is polynomial-time equivalent to* MONOTONE DUALIZATION.

A similar result holds for negative literal queries $\chi = \overline{q}$ as well. Also in this case, a polynomial number of transversal computation problems can be created such that each minimal transversal corresponds to some explanation. However, matters are more complicated here since a query $\overline{q}$ might also have non-positive explanations. This leads to a case analysis and more involved construction of hypergraphs.

We remark that a connection between dualization and abduction from a Horn theory represented by $char(\Sigma)$ is implicit in dependency inference from relational databases. An important problem there is to infer, in database terminology, a prime cover of the set F_r^+ of all functional dependencies (FDs) $X \rightarrow A$ which hold on an instance r of a relational schema $U = \{A_1, \ldots, A_n\}$ where the A_i are the attributes. A functional dependency $X \rightarrow A$, $X \subseteq U$, $A \in U$, is a constraint which states that for every tuples t_1 and t_2 occurring in the same relation instance r, it holds that $t_1[A] = t_2[A]$ whenever $t_1[X] = t_2[X]$, i.e., coincidence of t_1 and t_2 on all attributes in X implies that t_1 and t_2 also coincide on A. A *prime cover* is a minimal (under $\subseteq$) set of non-redundant FDs $X \rightarrow A$ (i.e., $X' \rightarrow A$ is violated for each $X' \subset X$) which is logically equivalent to F_r^+.

In our terms, a non-redundant FD $X \rightarrow A$ corresponds to a prime clause $\bigvee_{B \in X} \overline{B} \vee A$ of the CNF $\varphi_{F_r^+}$, where for any set of functional dependencies F, φ_F is the CNF $\varphi_F = \bigwedge_{X \rightarrow A \in F} \Big(\bigvee_{B \in X} \overline{B} \vee A \Big)$, where the attributes in U are viewed as atoms and F_r^+ is the set of all FDs which hold on r. Thus, by Proposition 4, the set X is an explanation of A from φ_r. As shown in [49], so called *max sets* for all attributes A are polynomial-time computable from r, which in totality correspond to the characteristic models of the (definite) Horn theory Σ_r defined by $\varphi_{F_r^+}$ [41]. Computing the explanations for A is then reducible to an instance of TRANS-ENUM [50], which establishes the result for generating all assumption-free explanations from definite Horn theories. We refer to [41] for an elucidating discussion which reveals the close correspondence between concepts and results in database theory on Armstrong relations and in model-based reasoning, which can be exploited to derive results about space bounds for representation and about particular abduction problems. The latter will be addressed in Section 4.2.

Further investigations on computing prime implicates from model-based theory representation appeared in [36] and in [40], which exhibited further problems equivalent to MONOTONE DUALIZATION. In particular, Khardon has shown, inspired by results in [49, 50, 19], that computing all prime implicates of a Horn theory Σ represented by its characteristic models is, under Turing reducibility (which is more liberal than the notion of reducibility we employ here in general), polynomial-time equivalent to TRANS-ENUM. Note, however, that by Proposition 4, we are here concerned with computing particular prime implicates rather than all.

4 Possible Generalizations

The results reported above deal with queries χ which are a single literal. As already stated in the introduction, often queries will be more complex, however, and consist of a conjunction of literals, of clauses [56], etc.

This raises the question about possible extensions of the above results for queries of a more general form, and in particular whether we encounter other kinds of problem instances which are equivalent to MONOTONE DUALIZATION.

4.1 General formulas and CNFs

Let us first consider how complex finding abductive explanations can grow. It is known [21] that deciding the existence of an explanation for literal query χ w.r.t. a set A is Σ_2^P-complete (i.e., complete for $\mathrm{NP}^{\mathrm{NP}}$), if the background theory Σ is a set of arbitrary clauses (not necessarily Horn). For Horn Σ, we get a similar result if the query χ is an arbitrary formula.

Proposition 5. *Given a Horn CNF φ, a set $A \subseteq Lit$, and a query χ, deciding whether χ has some explanation w.r.t A from φ is (i) Σ_2^P-complete, if χ is arbitrary (even if it is a DNF), (ii) NP-complete, if χ is a CNF, and (iii) NP-complete, if $A = Lit$.*

Intuitively, an explanation E for χ can be guessed and then, by exploiting Proposition 3, be checked in polynomial time with an oracle for propositional consequence. The Σ_2^P-hardness in case (i) can be shown by an easy reduction from deciding whether a quantified Boolean formula (QBF) of form $F = \exists X \forall Y \alpha$, where X and Y are disjoint sets of Boolean variables and α is a DNF over $X \cup Y$. Indeed, just let $\Sigma = \emptyset$, and $A = \{x, \overline{x} \mid x \in X\}$. Then, $\chi = \alpha$ has an explanation w.r.t. A iff formula F is valid. On the other hand, if χ is a CNF, then deciding consequence $\Sigma \cup S \models \chi$ is polynomial for every set of literals S; hence, in case (ii) the problem has lower complexity and is in fact in NP. As for case (iii), if $A = Lit$, then an explanation exists iff $\Sigma \cup \{\chi\}$ has a model, which can be decided in NP. The hardness parts for (ii) and (iii) are immediate by a simple reduction from SAT (given a CNF β, let $\Sigma{=}\emptyset$, $\chi{=}\beta$, and $A{=}Lit$).

We get a similar picture under model-based representation. Here, inferring a clause c from $char(\Sigma)$ is feasible in polynomial time, and hence also inferring a CNF. On the other hand, inferring an arbitrary formula (in particular, a DNF) α, is intractable, since to witness $\Sigma \not\models \alpha$ we intuitively need to find proper models $v_1, \ldots, v_l \in char(\Sigma)$ such that $\bigwedge_i v_i \not\models \alpha$.

Proposition 6. *Given $char(\Sigma)$, a set $A \subseteq Lit$, and a query χ, deciding whether χ has some explanation w.r.t A from Σ is (i) Σ_2^P-complete, if χ is arbitrary (even if it is a DNF), (ii) NP-complete, if χ is a CNF, and (iii) NP-complete, if $A = Lit$.*

As for (iii), we can guess a model v of χ and check whether v is also a model of Σ from $char(\Sigma)$ in polynomial time (indeed, check whether $v = \bigwedge\{w \in char(\Sigma) \mid v \leq w\}$ holds). The hardness parts can be shown by slight adaptations of the constructions for the formula based case, since $char(\Sigma)$ for the empty theory is easily constructed (it consists of x^{At} and all $x^{At \setminus \{i\}}$, $i \in \{1, \ldots, n\}$).

So, like in the formula-based case, also in the model-based case we loose the immediate computational link of computing all explanations to MONOTONE DUALIZATION if we generalize queries to CNFs and beyond. However, it appears that there are interesting cases between a single literal and CNFs which are equivalent to MONOTONE DUALIZATION.

As for the formula-based representation, recall that generating all explanations is polynomial total-time for χ being a positive literal (thus, tractable and "easier" than MONOTONE DUALIZATION), while it is coNP-hard for χ being a CNF (and in fact, a negative literal); so, somewhere between the transition from tractable to intractable might pass instances which are equivalent to MONOTONE DUALIZATION.

Unfortunately, restricting to positive CNFs (this is the first idea) does not help, since the problem remains coNP-hard, even if all clauses have small size; this can be shown by a straightforward reduction from the well-known EXACT HITTING SET problem [25]. However, we encounter monotone dualization if Σ is empty.

Proposition 7. *Given a set $A \subseteq Lit$, and a positive CNF χ, generating all explanations of χ w.r.t A from $\Sigma = \emptyset$ is polynomial-time equivalent to dualizing a positive CNF, under both model-based and formula based representation.*

This holds since, as easily seen, every explanation E must be positive, and moreover, must be a minimal transversal of the clauses in χ. Conversely, every minimal transversal T of χ after removal of all positive literals that do not belong to A (viewed as hypergraphs), is an explanation. Note that this result extends to those CNFs which are *unate*, i.e., convertible to a positive CNF by flipping the polarity of some variables.

4.2 Clauses and Terms

Let us see what happens if we move from general CNFs to the important subcases of a single clause and a single term, respectively. As shown in [25], generating all explanations remains under formula-based polynomial total-time for χ being a positive clause or a positive term, but is intractable as soon as we allow a negative literal. Hence, we do not see an immediate connection to monotone dualization.

More promising is model-based representation, since for a single literal query, equivalence to monotone dualization is known. It appears that we can extend this result to clauses and terms of certain forms.

Clauses Let us first consider the clause case, which is important for Clause Management Systems [56, 38, 39]. Here, the problem can be reduced to the special case of a single literal query as follows. Given a clause c, introduce a fresh letter q. If we would add the formula $c \Rightarrow \overline{q}$ to Σ, then the explanations of $\overline{q}$ would be, apart from the trivial explanation $\overline{q}$ in case, just the explanations of c. We can rewrite $c \Rightarrow \overline{q}$ to a Horn CNF $\alpha = \bigwedge_{x \in P(c)} (\overline{x} \vee \overline{q}) \wedge \bigwedge_{x \in N(c)} (x \vee \overline{q})$, so adding α maintains Σ Horn and thus the reduction works in polynomial time under formula-based representation. (Note, however, that we reduce this to a case which is intractable in general.)

Under model-based representation, we need to construct $char(\Sigma \cup \{\alpha\})$, however, and it is not evident that this is always feasible in polynomial time. We can establish the following relationship, though.

Let $P(c) = \{q_1, \ldots, q_k\}$ and $N(c) = \{q_{k+1}, \ldots, q_m\}$; thus, α is logically equivalent to $q \Rightarrow \overline{q}_1 \wedge \cdots \overline{q}_k \wedge q_{k+1} \wedge \cdots \wedge q_m$.

Claim. For $\Sigma' = \Sigma \cup \{\alpha\}$, we have

$$\begin{aligned} char(\Sigma') \subseteq \{v@(0) \mid v \in char(\Sigma)\} \cup \\ \left\{ (v_1 \wedge \cdots \wedge v_k)@(1) \mid v_i \in char(\Sigma), v_i \models \overline{q}_i \wedge \textstyle\bigwedge_{j=k+1}^{m} q_j, \text{ for } 1 \leq i \leq k \right\} \cup \\ \left\{ \begin{array}{c} (v_0 \wedge v_1 \wedge \cdots \wedge v_k)@(1) \mid v_i \in char(\Sigma), \text{ for } 0 \leq i \leq k, \\ v_0 \models \bigwedge_{i=1}^{m} q_i, \ v_i \models \overline{q}_i \wedge \bigwedge_{j=k+1}^{m} q_j, \text{ for } 1 \leq i \leq k \end{array} \right\}. \end{aligned}$$

where "@" denotes concatenation and q is assumed to be the last in the variable ordering.

Indeed, each model of Σ, extended to q, is a model of Σ' if we set q to 0; all models of Σ' of this form can be generated by intersecting characteristic models of Σ extended in this way. On the other hand, any model v of Σ' in which q is 1 must have also $q_{k+1}, \ldots, q_m$ set to 1 and $q_1, \ldots, q_k$ set to 0. We can generate such v by expanding the intersection of some characteristic models $v_1, \ldots, v_l$ of Σ (where $l \leq |char(\Sigma)|$) in which $q_{k+1}, \ldots, q_m$ are set to 1, and where each of q_1, q_2,..., q_k is made 0 by intersection with at least one of these vectors. By associativity and commutativity of intersection, we can generate v then by expanding the intersection of vectors of the form given in the above equation.

From the set RHS on the right hand side, $char(\Sigma')$ can be easily computed by eliminating those vectors v which are generated by the intersection of other vectors (i.e., such that $v = \bigwedge\{w \in RHS \mid v < w\}$).

In general, RHS will have exponential size; however, computing RHS is polynomial if k is bounded by a constant; clearly, computing $char(\Sigma')$ from RHS is feasible in time polynomial in the size of RHS, and hence in polynomial time in the size of $char(\Sigma)$. Thus, the reduction from clause explanation to literal explanation is computable in polynomial time. We thus obtain the following result.

Theorem 2. *Given $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ, a clause query c, and $A \subseteq Lit$, computing all explanations for c from Σ w.r.t. A is polynomial-time equivalent to* MONOTONE DUALIZATION*, if $|P(c)| \leq k$ for some constant k.*

Note that the constraint on $P(c)$ is necessary, in the light of the following result.

Theorem 3. *Given $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ, a positive clause query c, and some explanations $E_1, \ldots, E_l$ for c from Σ, deciding whether there exists some further explanation is* NP*-complete.*

The NP-hardness can be shown by a reduction from the well-known 3SAT problem.

By standard arguments, we obtain from this the following result for computing multiple explanations.

Corollary 1. *Given $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ, a set $A \subseteq Lit$ and a clause c, computing a given number resp. all explanations for c w.r.t. A from Σ is not possible in polynomial total-time unless* P=NP*. The hardness holds even for $A = Lit$, i.e., for assumption-free explanations.*

Terms Now let us finally turn to queries which are given by terms, i.e., conjunctions of literals. With a similar technique as for clause queries, explanations for a term t can be reduced to explanations for a positive literal query in some cases. Indeed, introduce a fresh atom q and consider $t \Rightarrow \overline{q}$; this formula is equivalent to a Horn clause α if $|N(t)| \leq 1$ (in particular, if t is positive). Suppose $t = \overline{q}_0 \wedge q_1 \wedge \cdots \wedge q_m$, and let $\Sigma' = \Sigma \cup \{\alpha\}$ (where $\alpha = \overline{q} \vee \overline{q}_1 \vee \cdots \vee \overline{q}_m \vee q_0$). Then we have

$$\begin{aligned} char(\Sigma') \subseteq \{v@(0) \mid v \in char(\Sigma)\} \cup \\ \{\, v@(1) \mid v \in char(\Sigma),\ v \models q_0 \vee \overline{q}_1 \vee \cdots \vee \overline{q}_m\} \cup \\ \{(v \wedge v')@(1) \mid v, v' \in char(\Sigma),\ v \models \overline{q}_1 \vee \cdots \vee \overline{q}_m, v' \models \overline{q}_0 \wedge q_1 \wedge \cdots \wedge q_m\}, \end{aligned}$$

from which $char(\Sigma')$ is easily computed. The explanations for t from Σ then correspond to the explanations for $\overline{q}$ from Σ' modulo the trivial explanation $\overline{q}$ in case.

This implies polynomial-time equivalence of generating all explanations for a term t to MONOTONE DUALIZATION if t contains at most one negative literal. In particular, this holds for the case of positive terms t.

Note that the latter case is intimately related to another important problem in knowledge discovery, namely inferring the keys of a relation r, i.e., the minimal (under $\subseteq$) sets of attributes $K \subseteq U = \{A_1, A_2, \ldots, A_n\}$ whose values uniquely identify the rest of any tuple in the relation. The keys for a relation instance r over attributes U amount to the assumption-free explanations of t from the Horn CNF $\varphi_{F_r^+}$ defined as above in Section 2.2. Thus, abduction procedures can be used for generating keys. Note that $char(F_r^+)$ is computable in polynomial time from r (cf. [41]), and hence generating all keys polynomially reduces to MONOTONE DUALIZATION; the converse has also been shown [19] [19]. Hence, generating all keys is polynomially equivalent to MONOTONE DUALIZATION and also to generating all explanations for a positive term from $char(\Sigma)$.

A similar reduction can be used to compute all keys of a generic relation scheme (U, F) of attributes U and functional dependencies F, which amount to the explanations for the term $t = A_1 A_2 \cdots A_n$ from the CNF φ_F. Note that Khardon *et al.* [41] investigate computing keys for given F and more general Boolean constraints ψ by a simple reduction to computing all nontrivial explanations E of a fresh letter q (i.e., $E \neq \{q\}$; use $\psi \wedge \left(\bigwedge_{A_i \in U} \overline{A_i} \vee q\right)$), which can thus be done in polynomial total-time as follows from results in [24]; for FDs (i.e., $\psi = \varphi_F$) this is a classic result in database theory [46]. Furthermore, [41] also shows how to compute an abductive explanation using an oracle for key computation.

We turn back to abductive explanations for a term t, and consider what happens if we have multiple negative literals in t. Without constraints on $N(t)$, we face the intractability, since deciding the existence of a nontrivial explanation for t is already difficult, where an explanation E for t is nontrivial if $E \neq P(t) \cup \{\overline{q} \mid q \in N(t)\}$.

Theorem 4. *Given $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ and a term t, deciding whether (i) there exists a nontrivial assumption-free explanation for t from Σ is* NP-*complete; (ii) there exists an explanation for t w.r.t. a given set $A \subseteq Lit$ from Σ is* NP-*complete. In both cases, the* NP-*hardness holds even for negative terms t.*

The hardness parts of this theorem can be shown by a reduction from 3SAT.

Corollary 2. *Given $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ, a set $A \subseteq Lit$ and a term t, computing a given number resp. all explanations for t w.r.t. A from Σ is not possible in polynomial total-time, unless* P=NP. *The hardness holds even for $A = Lit$, i.e., for assumption-free explanations.*

While the above reduction technique works for terms t with a single negative literal, it is not immediate how to extend it to multiple negative literals such that we can derive polynomial equivalence of generating all explanations to MONOTONE DUALIZATION if $|N(t)|$ is bounded by a constant k. However, this can be shown by a suitable extension of the method presented in [24], transforming the problem into a polynomial number of instances of MONOTONE DUALIZATION, which can be polynomially reduced to a single instance [24].

Proposition 8. *For any Horn theory Σ and $E = \{\overline{x}_1, \ldots, \overline{x}_k, x_{k+1}, \ldots, x_m\}$ ($\subseteq Lit$), $char(\Sigma \cup E)$ is computable from $char(\Sigma)$ by $char(\Sigma \cup E) = char(M_1 \cup M_2)$, where*

$$M_1 = \{v_1 \wedge \cdots \wedge v_k \mid v_i \in char(\Sigma), v_i \models \overline{x}_i \wedge \bigwedge_{j=k+1}^{m} x_j \text{ for } 1 \leq i \leq k\},$$
$$M_2 = \{v \wedge v_0 \mid v \in M_1, v_0 \in char(\Sigma), v_0 \models \bigwedge_{j=1}^{m} x_j\},$$

and $char(S) = \{v \in S \mid v \notin Cl_\wedge(S \setminus \{v\})\}$ for every $S \subseteq \{0,1\}^n$. This can be done in polynomial time if $k = |E \cap \overline{At}|$ is bounded by some constant.

For any model v and any $V \subseteq At$, let $v[V]$ denote the projection of v on V, and for any theory Σ and any $V \subseteq At$, $\Sigma[V]$ denotes the projection of Σ on V, i.e., $mod(\Sigma[V]) = \{v[V] \mid V \in mod(\Sigma)\}$.

Proposition 9. *For any Horn theory Σ and any $V \subseteq At$, $char(\Sigma[V])$ can be computed from $char(\Sigma)$ in polynomial time by $char(\Sigma[V]) = char(char(\Sigma)[V])$.*

For any model v and any set of models M, let $\max_v(M)$ denote the set of all the models in M that is maximal with respect to $\leq_v$. Here, for models w and u, we write $w \leq_v u$ if $w_i \leq u_i$ if $v_i = 0$, and $w_i \geq u_i$ if $v_i = 1$.

Proposition 10. *For any Horn theory Σ and any model v, $\max_v(\Sigma)$ can be computed from $char(\Sigma)$ by $\max_v(\{w \in M_S \mid S \subseteq \{x_i \mid v_i = 1\}\})$, where $M_\emptyset = char(\Sigma)$ and for $S = \{x_1, \ldots, x_k\}$,*

$$M_S = \{v_1 \wedge \cdots \wedge v_k \mid v_i \in char(\Sigma), v_i \models \overline{x}_i \text{ for } 1 \leq i \leq k\}.$$

This can be done in polynomial time if $|\{x_i \mid v_i = 1\}|$ is bounded by some constant.

Theorem 5. *Given $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ, a term query t, and $A \subseteq Lit$, computing all explanations for t from Σ w.r.t. A is polynomial-time equivalent to* MONOTONE DUALIZATION, *if $|N(t)| \leq k$ for some constant k.*

Proof. (Sketch) We consider the following algorithm.

Algorithm TERM-EXPLANATIONS
Input: $char(\Sigma) \subseteq \{0,1\}^n$ of a Horn theory Σ, a term t, and $A \subseteq Lit$.
Output: All explanations for t from Σ w.r.t. A.

Step 1. Let $\Sigma' = \Sigma \cup P(t) \cup \{\overline{q} \mid q \in N(t)\}$. Compute $char(\Sigma')$ from $char(\Sigma)$.
Step 2. For each $x_i \in N(t)$, let $\Sigma_{x_i} = \Sigma \cup \{x_i\}$, and for each $x_i \in P(t)$, let $\Sigma_{\overline{x}_i} = \Sigma \cup \{\overline{x}_i\}$. Compute $char(\Sigma_{x_i})$ from $char(\Sigma)$ for $x_i \in N(t)$, and compute $char(\Sigma_{\overline{x}_i})$ from $char(\Sigma)$ for $x_i \in P(t)$.
Step 3. For each $B = B_- \cup B_+$, where $B_- \subseteq A \cap \overline{At}$ with $|B_-| \leq |N(t)|$ and $B_+ = (A \cap At) \setminus \{q \mid \overline{q} \in B_-\}$, let $C = B_+ \cup \{q \mid \overline{q} \in B_-\}$.

(3-1) Compute $char(\Sigma'[C])$ from $char(\Sigma')$.

(3-2) Let $v \in \{0,1\}^C$ be the model with $v_i = 1$ if $x_i \in B_-$, and $v_i = 0$ if $x_i \in B_+$. Compute $\max_v(\Sigma'[C])$ from $char(\Sigma'[C])$.

(3-3) For each $w \in \max_v(\Sigma'[C])$, let $C_{v,w}=\{x_i \mid v_i \neq w_i\}$ and let $v^*=v[C_{v,w}]$.

(3-3-1) Compute $\max_{v^*}(\Sigma_{x_i}[C_{v,w}])$ and $\max_{v^*}(\Sigma_{\overline{x}_i}[C_{v,w}])$ from $char(\Sigma_{x_i})$ and $char(\Sigma_{\overline{x}_i})$, respectively. Let

$$M_{v,w} = \bigcup_{x_i \in N(t)} \max\nolimits_{v^*}(\Sigma_{x_i}[C_{v,w}]) \cup \bigcup_{x_i \in P(t)} \max\nolimits_{v^*}(\Sigma_{\overline{x}_i}[C_{v,w}]).$$

(3-3-2) Dualize the CNF $\varphi_{v,w} = \bigwedge_{u \in M_{v,w}} c_u$, where

$$c_u = \bigvee_{i:u_i=v_i=0} x_i \vee \bigvee_{i:u_i=v_i=1} \overline{x}_i.$$

Each prime clause c of $\varphi^d_{v,w}$ corresponds to an explanation $E = P(c) \cup \{\overline{x}_j \mid x_j \in N(c)\}$, which is output. Note that $\varphi_{v,w}$ is *unate*, i.e., convertible to a positive CNF by flipping the polarity of some variables.

Informally, the algorithm works as follows. The theory Σ' is used for generating candidate sets of variables C on which explanations can be formed; this corresponds to condition (i) of an explanation, which combined with condition (ii) amounts to consistency of $\Sigma \cup E \cup \{t\}$. These sets of variables C are made concrete in Step 3 via B, where the easy fact is taken into account that any explanation of a term t can contain at most $|N(t)|$ negative literals. The projection of $char(\Sigma')$ to the relevant variables C, computed in Step 3-1, serves then as the basis for a set of variables, $C_{v,w}$, which is a largest subset of C on which some vector in $\Sigma'[C]$ is compatible with the selected literals B; any explanation must be formed on variables included in some $C_{v,w}$. Here, the ordering of vectors under $\leq_v$ is relevant, which respects negative literals. The explanations over $C_{v,w}$ are then found by excluding every countermodel of t, i.e., all the models of Σ_{x_i}, $x_i \in N(t)$ resp. $\Sigma_{\overline{x}_i}$, $x_i \in P(t)$, with a smallest set of literals. This amounts to computing minimal transversals (where only maximal models under $\leq_{v^*}$ need to be considered), or equivalently, to dualization of the given CNF $\varphi_{v,w}$.

More formally, it can be shown that the algorithm above computes all explanations. Moreover, from Propositions 8, 9, and 10, we obtain that computing all explanations reduces in polynomial time to (parallel) dualization of positive CNFs if $|N(t)| \leq k$, which can be polynomially reduced to dualizing a single positive CNF [24]. Since as already known, MONOTONE DUALIZATION reduces in polynomial-time to computing all explanations of a positive literal, the result follows. □

We remark that algorithm TERM EXPLANATIONS is closely related to results by Khardon and Roth [42] about computing some abductive explanation for a query χ from a (not necessarily Horn) theory represented by its characteristic models, which are defined using Monotone Theory [10]. In fact, Khardon and Roth established that computing some abductive explanation for a Horn CNF query χ w.r.t. a set A containing at most k negative literals from a theory Σ is feasible in polynomial time, provided that Σ is represented by an appropriate type of characteristic models (for Horn Σ, the

Query χ / Horn theory Σ	general/ DNF	CNF	literal		clause			term		
			pos	neg	pos	Horn	general	pos	neg	general
Horn CNF φ	Π_2^P [d]	coNP	coNP[a]	coNP[c]	coNP[a]	coNP[c]	coNP[c]	coNP[a]	coNP[c]	coNP[c]
$char(\Sigma)$	Π_2^P [d]	coNP	DUAL		nPTT[b,c]	DUAL	nPTT[b,c]	DUAL	coNP[b,c]	coNP[b,c]

[a] polynomial total-time for assumption-free explanations ($A = Lit$).

[b] DUAL for k-positive clauses resp. k-negative terms, k bounded by a constant.

[c] nPTT for assumption-free explanations ($A = Lit$).

[d] coNP (resp. nPTT) for assumption-free explanations ($A = Lit$) and general χ (resp. DNF χ).

Table 1. Complexity of computing all abductive explanations for a query from a Horn theory

characteristic models $char_{k+1}(\Sigma)$ with respect to $(k + 1)$-quasi Horn functions will do, which are those functions with a CNF φ such that $|P(c)| \leq k + 1$ for every $c \in \varphi$). Proposition 10 implies that $char_{k+1}(\Sigma)$ can be computed in polynomial time from $char(\Sigma)$. Hence, by a detour through characteristic models conversion, some explanation for a Horn CNF w.r.t. A as above can be computed from a Horn Σ represented by $char(\Sigma)$ in polynomial time using the method of [42]. This can be extended to computing all explanations for χ, and exploiting the nature of explanations for terms to an algorithm similar to TERM EXPLANATIONS.

Furthermore, the results of [42] provides a basis for obtaining further classes of abduction instances Σ, A, χ polynomially equivalent to MONOTONE DUALIZATION where Σ is not necessarily Horn. However, this is not easy to accomplish, since roughly non-Horn theories lack in general the useful property that every prime implicate can be made monotone by flipping the polarity of some variables, where the admissible flipping sets induce a class of theories in Monotone Theory. Explanations corresponding to such prime implicates might not be covered by a simple generalization of the above methods.

5 Conclusion

In this paper, we have considered the connection between abduction and the well-known dualization problems, where we have reviewed some results from recent work and added some new; a summary picture is given in Table 1.

In this table, "nPTT" stands for "not polynomial total-time unless P=NP," and "coNP" resp. "Π_2^P" stands for for deciding whether the output is empty (i.e., no explanation exists) is coNP-complete resp. Π_2^P-complete (which trivially implies nPTT); DUAL denotes polynomial-time equivalence to MONOTONE DUALIZATION. In order to elucidate the role of abducibles, the table highlights also results for assumption-free explanations ($A = Lit$) when they deviate from an arbitrary set A of abducibles.

As can be seen from the table, there are several important classes of instances which are equivalent to MONOTONE DUALIZATION. In particular, this holds for generating all explanations for a clause query (resp., term query) χ if χ contains at most k positive (resp., negative) literals for constant k. It remains to be explored how these results, via the applications of abduction, lead to the improvement of problems that appear in applications. In particular, the connections to problems in knowledge discovery remain to be explored. Furthermore, an implementation of the algorithms and experiments are left for further work.

We close by pointing out that besides MONOTONE DUALIZATION, there are related problems whose precise complexity status in the theory of NP-completeness is not known to date. A particular interesting one is the dependency inference problem which we mentioned above, i.e., compute a prime cover of the set F_r^+ of all functional dependencies (FDs) $X{\rightarrow}A$ which hold on an instance r of a relational schema [49] (recall that a prime cover is a minimal (under $\subseteq$) set of non-redundant FDs which is logically equivalent to F_r^+). There are other problems which are polynomial-time equivalent to this problem [40] under the more liberal notion of Turing-reduction used there; for example, one of these problems is computing the set of all characteristic models of a Horn theory Σ from a given Horn CNF φ representing it.

Dependency inference contains MONOTONE DUALIZATION as a special case (cf. [20]), and is thus at least as hard, but to our knowledge there is no strong evidence that it is indeed harder, and in particular, it is yet unknown whether a polynomial total-time algorithm for this problem implies $\mathrm{P}{=}\mathrm{NP}$. It would be interesting to see progress on the status of this problem, as well as possible connections to abduction.

References

1. D. Angluin. Queries and Concept Learning. *Machine Learning*, 2:319–342, 1996.
2. C. Bioch and T. Ibaraki. Complexity of identification and dualization of positive Boolean functions. *Information and Computation*, 123:50–63, 1995.
3. E. Boros, Y. Crama, and P. L. Hammer. Polynomial-time inference of all valid implications for Horn and related formulae. *Ann. Mathematics and Artificial Intelligence*, 1:21–32, 1990.
4. E. Boros, V. Gurvich, L. Khachiyan and K. Makino. Dual-bounded generating problems: Partial and multiple transversals of a hypergraph. *SIAM J. Computing*, 30:2036–2050, 2001.
5. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan and K. Makino. Dual-bounded generating problems: All minimal integer solutions for a monotone system of linear inequalities. *SIAM Journal on Computing*, 31:1624-1643, 2002.
6. E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. On the complexity of generating maximal frequent and minimal infrequent sets. In *Proc. 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS-02)*, LNCS 2285, pp. 133–141, 2002.
7. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. An intersection inequality for discrete distributions and related generation problems. In *Proc. 30th Int'l Coll. on Automata, Languages and Programming (ICALP 2003)*, LNCS 2719, pp. 543-555, 2003.
8. E. Boros, V. Gurvich, and P. L. Hammer. Dual subimplicants of positive Boolean functions. *Optimization Methods and Software*, 10:147–156, 1998.
9. G. Brewka, J. Dix, and K. Konolige. *Nonmonotonic Reasoning – An Overview*. Number 73 in CSLI Lecture Notes. CSLI Publications, Stanford University, 1997.
10. N. H. Bshouty. Exact Learning Boolean Functions via the Monotone Theory. *Information and Computation*, 123:146–153, 1995.
11. T. Bylander. The monotonic abduction problem: A functional characterization on the edge of tractability. In *Proc. 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*, pp. 70–77, 1991.

12. Y. Crama. Dualization of regular boolean functions. *Discrete App. Math.*, 16:79–85, 1987.
13. J. de Kleer. An assumption-based truth maintenance system. *Artif. Int.*, 28:127–162, 1986.
14. R. Dechter and J. Pearl. Structure identification in relational data. *Artificial Intelligence*, 58:237–270, 1992.
15. A. del Val. On some tractable classes in deduction and abduction. *Artificial Intelligence*, 116(1-2):297–313, 2000.
16. A. del Val. The complexity of restricted consequence finding and abduction. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 337–342, 2000.
17. C. Domingo, N. Mishra, and L. Pitt. Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries. *Machine Learning*, 37:89–110, 1999.
18. W. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn theories. *Journal of Logic Programming*, 3:267–284, 1984.
19. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. Technical Report CD-TR 91/16, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, January 1991.
20. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, December 1995.
21. T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, January 1995.
22. T. Eiter and G. Gottlob. Hypergraph transversal computation and related problems in logic and AI. In *Proc. 8th European Conference on Logics in Artificial Intelligence (JELIA 2002)*, LNCS 2424, pp. 549–564. Springer, 2002.
23. T. Eiter, G. Gottlob, and K. Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM Journal on Computing*, 32(2):514–537, 2003. Preliminary paper in Proc. ACM STOC 2002.
24. T. Eiter and K. Makino. On computing all abductive explanations. In *Proc. 18th National Conference on Artificial Intelligence (AAAI '02)*, pp. 62–67, 2002. Preliminary Tech. Rep. INFSYS RR-1843-02-04, Institut für Informationssysteme, TU Wien, April 2002.
25. T. Eiter and K. Makino. Generating all abductive explanations for queries on propositional Horn theories. In *Proc. 12th Annual Conference of the EACSL (CSL 2003), August 25-30 2003, Vienna, Austria*. LNCS, Springer, 2003.
26. T. Eiter, T. Ibaraki, and K. Makino. Computing intersections of Horn theories for reasoning with models. *Artificial Intelligence*, 110(1-2):57–101, 1999.
27. K. Eshghi. A tractable class of abduction problems. In *Proc. 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pp. 3–8, 1993.
28. M. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
29. G. Friedrich, G. Gottlob, and W. Nejdl. Hypothesis classification, abductive diagnosis, and therapy. In *Proc. International Workshop on Expert Systems in Engineering*, LNCS/LNAI 462, pp. 69–78. Springer, 1990.
30. D.R. Gaur and R. Krishnamurti. Self-duality of bounded monotone boolean functions and related problems. In *Proc. 11th International Conference on Algorithmic Learning Theory (ALT 2000)*, LNCS 1968, pp. 209-223, 2000.
31. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data mining, hypergraph transversals, and machine learning. In *Proc. 16th ACM Symposium on Principles of Database Systems (PODS-96)*, pp. 209–216, 1993.
32. K. Inoue. Linear resolution for consequence finding. *Artif. Int.*, 56(2-3):301–354, 1992.
33. D. S. Johnson. A Catalog of Complexity Classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2. Elsevier, 1990.
34. H. Kautz, M. Kearns, and B. Selman. Reasoning With Characteristic Models. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-93)*, pp. 34–39, 1993.

35. H. Kautz, M. Kearns, and B. Selman. Horn approximations of empirical data. *Artificial Intelligence*, 74:129–245, 1995.
36. D. Kavvadias, C. Papadimitriou, and M. Sideri. On Horn envelopes and hypergraph transversals. In *Proc. 4th International Symposium on Algorithms and Computation (ISAAC-93)*, LNCS 762, pp. 399–405, Springer, 1993.
37. D. J. Kavvadias and E. C. Stavropoulos. Monotone Boolean dualization is in co-NP[$\log^2$n]. *Information Processing Letters*, 85:1–6, 2003.
38. A. Kean and G. Tsiknis. Assumption based reasoning and Clause Management Systems. *Computational Intelligence*, 8(1):1–24, 1992.
39. A. Kean and G. Tsiknis. Clause Management Systems (CMS). *Computational Intelligence*, 9(1):11–40, 1992.
40. R. Khardon. Translating between Horn representations and their characteristic models. *Journal of Artificial Intelligence Research*, 3:349–372, 1995.
41. R. Khardon, H. Mannila, and D. Roth. Reasoning with examples: Propositional formulae and database dependencies. *Acta Informatica*, 36(4):267–286, 1999.
42. R. Khardon and D. Roth. Reasoning with models. *Artif. Int.*, 87(1/2):187–213, 1996.
43. R. Khardon and D. Roth. Defaults and relevance in model-based reasoning. *Artificial Intelligence*, 97(1/2):169–193, 1997.
44. H. Levesque. Making believers out of computers. *Artificial Intelligence*, 30:81–108, 1986.
45. L. Lovász. Combinatorial optimization: Some problems and trends. DIMACS Technical Report 92-53, RUTCOR, Rutgers University, 1992.
46. C. L. Lucchesi and S. Osborn. Candidate Keys for Relations. *Journal of Computer and System Sciences*, 17:270–279, 1978.
47. K. Makino and T. Ibaraki. The maximum latency and identification of positive Boolean functions. *SIAM Journal on Computing*, 26:1363–1383, 1997.
48. K. Makino and T. Ibaraki, A fast and simple algorithm for identifying 2-monotonic positive Boolean functions. *Journal of Algorithms*, 26:291–305, 1998.
49. H. Mannila and K.-J. Räihä. Design by Example: An application of Armstrong relations. *Journal of Computer and System Sciences*, 22(2):126–141, 1986.
50. H. Mannila and K.-J. Räihä. Algorithms for inferring functional dependencies. Technical Report A-1988-3, University of Tampere, CS Dept., Series of Publ. A, April 1988.
51. P. Marquis. Consequence Finding Algorithms. In D. Gabbay and Ph.Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume V: Algorithms for Uncertainty and Defeasible Reasoning, pp. 41–145. Kluwer Academic, 2000.
52. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
53. C. H. Papadimitriou. NP-Completeness: A retrospective, In *Proc. 24th Int'l Coll. on Automata, Languages and Programming (ICALP 1997)*, pp. 2–6, LNCS 1256. Springer, 1997.
54. C. S. Peirce. Abduction and Induction. In J. Buchler, editor, *Philosophical Writings of Peirce*, chapter 11. Dover, New York, 1955.
55. D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(1):97–110, 1989.
56. R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: Preliminary report. In *Proc. 6th National Conference on Artificial Intelligence (AAAI-87)*, pp. 183–188, 1982.
57. B. Selman and H. J. Levesque. Abductive and default reasoning: A computational core. In *Proc. 8th National Conference on Artificial Intelligence (AAAI-90)*, pp. 343–348, July 1990.
58. B. Selman and H. J. Levesque. Support set selection for abductive and default reasoning. *Artificial Intelligence*, 82:259–272, 1996.
59. B. Zanuttini. New polynomial classes for logic-based abduction. *Journal of Artificial Intelligence Research*, 2003. To appear.