In: Proc. 12th Annual Conference of the EACSL (CSL 2003), J. Makowsky and M. Baaz, editors, LNCS, © 2003 Springer.

Generating all Abductive Explanations for Queries on Propositional Horn Theories*

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 Systems Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan, makino@sys.es.osaka-u.ac.jp

Abstract. Abduction is a fundamental mode of reasoning, which has taken on increasing importance in Artificial Intelligence (AI) and related disciplines. Computing abductive explanations is an important problem, and there is a growing literature on this subject. We contribute to this endeavor by presenting new results on computing multiple resp. all of the possibly exponentially many explanations of an abductive query from a propositional Horn theory represented by a Horn CNF. Here the issues are whether a few explanations can be generated efficiently and, in case of all explanations, 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. We explore these issues for queries in CNF and important restrictions thereof. Among the results, we show that computing all explanations for a negative query literal from a Horn CNF is not feasible in polynomial total time unless $\mathrm{P} = \mathrm{NP}$, which settles an open issue. However, we show how to compute under restriction to acyclic Horn theories polynomially many explanations in input polynomial time and all explanations in polynomial total time, respectively. Complementing and extending previous results, this draws a detailed picture of the computational complexity of computing multiple explanations for queries on Horn theories.

Keywords: Computational logic, abduction, propositional logic, Horn theories, polynomial total time computation, NP-hardness.

1 Introduction

Abduction is a fundamental mode of reasoning, which was extensively studied by C.S. Peirce [19]. It has taken on increasing importance in Artificial Intelligence (AI) and related disciplines, where it has been recognized as an important principle of commonsense reasoning (see e.g. [3]). Abduction has applications in many areas of AI and Computer Science including diagnosis, database updates, planning, natural language understanding, learning etc. (see e.g. references in [10]), where it is primarily used for generating explanations.

In a logic-based setting, abduction can be viewed as the task to find, given a set of formulas Σ (the *background theory*) and a formula χ (the *query*), a minimal set of formulas E (an *explanation*) from a set of hypotheses H such that Σ plus E is satisfiable

* 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.

and logically entails χ. Often considered is a scenario where Σ is a propositional Horn theory, χ is a single literal or a conjunction of literals, and H contains literals (see [24, 10] and references therein). 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 [20] or ATMS solvers [6, 22] have been devised. Since then, there has been a growing literature on this subject, indicating the need for efficient abductive procedures. We refer to [18], which gives an excellent survey on intimately closely related problems in computational logic. Note that much effort has been spent on studying various input restrictions, cf. [14, 4, 13, 25, 8, 7, 10, 23, 24].

While computing *some* explanation of a query χ has been studied extensively in the literature, the issue of computing multiple or even *all* explanations for χ has received less attention. 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 [24].

Computing some explanation for a query χ which is a literal from a Horn theory is a well-known polynomial problem. Selman and Levesque conjectured [24] that generating $O(n)$ many explanations for a positive literal is NP-hard, where n is the number of propositional atoms in the language, even if it is guaranteed that there are only few explanations overall. As shown in [11], this conjecture is not true unless $\mathrm{P}{=}\mathrm{NP}$. This follows from the result of [11] that all explanations for an atom can be generated in polynomial total time.

The status of generating all explanations for a negative literal $\chi = \overline{q}$ from a Horn CNF φ, however, remained open in [11]. Moreover, it was unclear whether a resolution-style procedure similar to the one for query atoms in [11] could solve the problem in polynomial total time. In this paper, we provide a negative answer to this question, by showing that given a collection of explanations for a query $\chi = \overline{q}$ from a Horn CNF φ, deciding whether there is an additional explanation is NP-complete. Consequently, the existence of a polynomial total time algorithm for computing all explanations implies $\mathrm{P}{=}\mathrm{NP}$. However, for the well-known class of acyclic Horn theories (see e.g. [5, 24, 21, 1]) we present an algorithm which enumerates all explanations for $\overline{q}$ with incremental polynomial delay (i.e., in time polynomial in the size of the input and output so far), and thus solves the problem in polynomial total time. Compared to explanations for an atomic query q, intuitively cyclic dependencies between atoms make the problem difficult. For completeness, a resolution-style procedure as in [11] needs to consider besides the input and output clauses also auxiliary clauses (see Example 7), whose derivation may cause a lot of overhead, since it is not a priori clear which such clauses are needed.

We furthermore address computing all explanations for queries χ beyond literals, where we consider 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 [22]. In the light of the negative results from above, we aim at elucidating the tractability frontier and present positive as well as negative results for such queries.

Our results shed new light on the computational nature of abduction and Horn theories in particular. They imply that, e.g., generating all minimal support clauses for a given clause (cf. [22]) from an acyclic Horn CNF is feasible in polynomial total time. The intractability result for negative literal queries $\overline{q}$ is somewhat unexpected, and the tractability result for acyclic Horn theories is more difficult to obtain than in case of atomic queries. As a byproduct, we also obtain results for computing all prime implicates of Horn theories containing a certain literal, which complement and refine previous results for computing all prime implicates of a Horn theory [2].

For space reasons, some proofs are omitted; we refer to the extended version [12].

2 Preliminaries and Notation

We assume a standard propositional language 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 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 the sets of atoms occurring positively and negated in c and $P(c) \cap N(c) = \emptyset$. Dually, a term is 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. We also view clauses and terms as sets of literals $P(c) \cup N(c)$ and $P(t) \cup N(t)$, respectively. A clause c is *Horn*, if $|P(c)| \leq 1$; *definite*, if $|P(c)| = 1$; and *negative* (resp., *positive*), if $|P(c)| = 0$ (resp., $|N(c)| = 0$). A conjunctive normal form (CNF) is a conjunction of clauses. It is *Horn* (resp., *definite, negative, positive*), if it contains only Horn clauses (resp., definite, negative, positive clauses). A *theory* Σ is any finite set of formulas; it is *Horn*, if it is a set of Horn clauses. As usual, we identify Σ with $\varphi = \bigwedge_{c \in \Sigma} c$, and write $c \in \varphi$ etc.

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 $i \in \{1, \ldots, n\}$. Satisfaction $v \models \varphi$ and logical consequence $\varphi \models c$, $\varphi \models \psi$ etc. are defined as usual (i.e., $\varphi(v) = 1$ etc.).

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 vector $u = (0,1,0,1,0)$ is a model of φ. □

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.* [9]*).*

Recall that two clauses c and c' resolve on a pair of literals $x, \overline{x}$ if $x, \overline{x} \in c \cup c'$ and $c \cup c' \setminus \{x, \overline{x}\}$ is a legal clause (thus, x must occur in exactly one of c and c', and same for $\overline{x}$); c and c' resolve if their is a pair of literals $x, \overline{x}$ on which they resolve. Note that this pair, if it exists, is unique. In that case, we denote by $c \oplus c'$ the clause $c \cup c' \setminus \{x, \overline{x}\}$, which is their *resolvent* (otherwise, $c \oplus c'$ is undefined). A *resolution proof* of a clause

c from a CNF φ is a sequence $c_1, c_2, \dots, c_l$ of clauses such that $c_l = c$ and, for all $i = 1, \dots, l$, either $c_i \in \varphi$ or $c_i = c_j \oplus c_k$ for clauses c_j and c_k such that $j, k < i$. It is well-known that resolution proofs are sound and complete with respect to clause inference in the following sense (cf. [18]): For any CNF φ and clause c, $\varphi \models c$ holds iff there is a clause $c' \subseteq c$ which has a resolution proof from φ. For further background on resolution, we refer to [17, 16].

2.1 Abductive explanations

The notion of an abductive explanation can be formalized as follows (cf. [24, 10]).

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

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

Example 2. 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. $\Box$

More restricted forms of explanations require that E must be formed over a given set of abducible letters (cf. [24]); however, in such a setting, generating all explanations is easily seen to be coNP-hard for the cases that we consider from results in the literature.

The following characterization of explanations is immediate by the monotonicity of classical logic.

Proposition 2. *For any theory Σ, any query χ, and any $E \subseteq Lit$, E is an explanation for χ 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 $\ell \in E$.*

From Proposition 2, we thus obtain the following easy lemma.

Lemma 1. *Given a Horn CNF φ, a set $E \subseteq Lit$, and a CNF query χ, deciding whether E is an explanation for χ w.r.t. A is possible in polynomial time.*

3 Intractability of Negative Literal Queries

In this section, we show that computing all explanations of a negative query $\chi = \overline{q}$ is not possible in polynomial total time unless $\mathrm{P} = \mathrm{NP}$. This result follows by standard arguments from the following theorem.

Theorem 1. *Given a Horn CNF φ, a query $\chi = \overline{q}$, and explanations $E_1, E_2, \dots, E_k$ for χ, deciding whether χ has some additional explanation E_{k+1} different from each E_i, $1 \leq i \leq k$, is* NP*-complete. Hardness holds even if φ is definite Horn.*

In the proof of Theorem 1, we use the following well-known lemma, which links prime implicates of a theory to explanations. Recall that a *prime implicate* of a theory Σ is a minimal (w.r.t. inclusion) clause c such that $\Sigma \models c$. Let us call an explanation E for a literal query $\chi = \ell$ *trivial*, if $E = \{\ell\}$.

Lemma 2 (cf. [22, 15]). *Given a theory Σ, a set $E \subseteq Lit$ is a nontrivial explanation of a query literal χ iff the clause $c = \bigvee_{\ell \in E} \overline{\ell} \vee \chi$ is a prime implicate of Σ.*

Note $\chi = \ell$ has a trivial explanation iff $\Sigma \not\models \ell$ and $\Sigma \not\models \overline{\ell}$, which can be checked in polynomial time. Hence, as for NP-hardness we can without loss of generality focus on generating the nontrivial explanations of $\overline{q}$, i.e., all prime implicates containing $\overline{q}$.

Proof of Theorem 1. As for membership in NP, an additional explanation E_{k+1} can be guessed and, by Lemma 1, be verified in polynomial time.

We show the NP-hardness by a reduction from 3SAT. Let $\gamma = c_1 \wedge \cdots \wedge c_m$, $m \geq 2$, be a 3CNF over atoms $x_1, \ldots, x_n$, where $c_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$. We introduce for each clause c_i a new atom y_i, for each x_j a new atom x'_j (which intuitively corresponds to $\overline{x}_i$), and special atoms q and z. The Horn CNF φ contains the following clauses:

1. $c_{i,j} = \overline{q} \vee \overline{\ell^\star_{i,j}} \vee y_i$, for all $i = 1, \ldots, m$ and $j = 1, 2, 3$;
2. $d_{i,j} = \overline{\ell^\star_{i,j}} \vee y_i \vee \overline{y}_{i+1}$, for all $i = 1, \ldots, m$ and $j = 1, 2, 3$;
3. $\overline{x}_i \vee \overline{x'_i} \vee z$, for all $i = 1, \ldots, n$;
4. $e = \overline{y}_1 \vee \overline{y}_2 \vee \cdots \vee \overline{y}_m \vee z$,

where $\ell^\star_{i,j} = x_k$ if $\ell_{i,j} = x_k$ and $\ell^\star_{i,j} = x'_k$ if $\ell_{i,j} = \overline{x}_i$, and $y_{m+1} = y_1$.

Note that φ is definite Horn, and thus all prime implicates of φ are definite Horn. Informally, the clauses $c_{i,j}$ and $d_{i,j}$ stand for selection of literal $\ell_{i,j}$ in clause c_i. The clause in 4., which is needed to produce any negative prime implicate c containing $\overline{q}$, and the minimality of a prime implicate will effect that a literal is chosen from each clause c_i, and the clauses in 3. will ensure that the choice is consistent, such that γ is satisfied. Since the positive prime implicates containing $\overline{q}$ are just the clauses $c_{i,j}$, the a further prime implicate of φ containing $\overline{q}$ exists iff γ is satisfiable.

We establish the following properties of φ.

Lemma 3. *Any prime implicate c of φ such that $q \in N(c)$ and $P(c) \neq \{z\}$ is of the form $c_{i,j}$, where $i \in \{1, \ldots, m\}$ and $j \in \{1, 2, 3\}$.*

Lemma 4. *Any prime implicate c of φ such that $P(c) = \{z\}$ and $q \in N(c)$ satisfies (i) $\{x_i, x'_i\} \not\subseteq N(c)$, for all $i = 1, \ldots, n$, and (ii) $y_i \notin N(c)$, for all $i = 1, \ldots, m$.*

From Lemma 3, it is now easy to see that all prime implicates of φ given by the clauses in 1. correspond to nontrivial explanations of $\overline{q}$, and from Lemma 4 that an additional nontrivial explanation for $\overline{q}$ exists if and only if some prime implicate of φ of form $c = \overline{q} \vee \bigvee_{x_i \in X} \overline{x}_i \vee \bigvee_{x'_i \in X'} \overline{x'}_i \vee z$ exists iff the CNF γ is satisfiable. As for the last equivalence, note that for each smallest (w.r.t. $\subseteq$) choice $\ell_i \in c_i$ of a consistent collection of literals $\ell_1, \ldots, \ell_m$, we have an additional prime implicate of φ of form $\overline{q} \vee \bigvee_{\ell_i} \overline{\ell^\star_i} \vee z$. Conversely, each additional prime implicate c containing $\overline{q}$ gives rise to a consistent set of literals $\{x_j \mid x_{i,j} \in N(c)\} \cup \{\overline{x}_j \mid x'_{i,j} \in N(c)\}$ which satisfies γ. Clearly, φ is constructible in polynomial time from γ. Since φ is definite, this proves the NP-hardness under the asserted restriction. □

We note that φ in the hardness proof of Theorem 1 remains Horn upon switching the polarity of z. From this easily NP-completeness of deciding the existence of an explanation for $\chi = \overline{q}$ formed of only positive literals follows, even if all other explanations are given. This contrasts with respective tractability results for acyclic theories (implied by the next section) and for atomic queries $\chi = q$ on arbitrary Horn CNFs [11].

4 Negative Literal Queries on Acyclic Horn Theories

Since as shown in the previous section, a polynomial total time procedure for generating all explanations of a negative literal query is infeasible in general unless P=NP, it becomes an issue to find restricted input classes for which this is feasible. In this section, we show a positive result for the important class of acyclic Horn theories, which has been studied extensively in the literature (see, e.g., [5, 24, 21, 1]).

We first recall the concept of acyclic Horn theories (see e.g. [5, 24]).

Definition 2. *For any Horn CNF φ over atom set At, its dependency graph is the directed graph $G(\varphi) = (V, E)$, where $V = At$ and $E = \{x_i \to x_j \mid c \in \varphi, x_i \in N(c), x_j \in P(c)\}$, i.e., E contains an arc from each atom in a negative literal to the positive literal in a clause (if such a literal exists). A Horn CNF φ is* acyclic *if $G(\varphi)$ has no directed cycle.*

Example 3. As easily seen, the edges of $G(\varphi)$ for the CNF φ in Examples 1 and 2 are $x_1 \to x_2$, $x_4 \to x_1$, $x_5 \to x_1$, $x_2 \to x_3$, and $x_5 \to x_3$. Hence, φ is acyclic. □

Since the trivial explanation $E = \{\overline{q}\}$ can be easily generated (if it applies), we focus on generating all nontrivial explanations. For a negative query on an acyclic Horn theory, this is accomplished by Algorithm N-EXPLANATIONS in Figure 1. It first converts the input into an equivalent prime Horn CNF $\varphi^\star$, and then applies a restricted resolution procedure, in which pairs (c, c') of clauses are considered of which at least one is a prime implicate containing $\overline{q}$ and the other is either a clause of this form or a clause from the converted input $\varphi^\star$. In case their resolvent $d := c \oplus c'$ exists and, as implied by condition (ii) in Definition 1, includes only prime implicates containing $\overline{q}$, any such prime implicate d' is computed. If d' is recognized as a new prime implicate which has not been generated so far, a corresponding explanation is output and the set of candidate pairs is enlarged.

Example 4. Reconsider $\varphi = (\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_4 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_4 \vee \overline{x}_5 \vee x_1) \wedge (\overline{x}_2 \vee \overline{x}_5 \vee x_3)$, and apply N-EXPLANATIONS for $\chi = \overline{x}_1$. All clauses of φ are prime except $\overline{x}_4 \vee \overline{x}_5 \vee x_1$, which contains the prime implicate $\overline{x}_4 \vee \overline{x}_5$. Thus, $\varphi^\star = (\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_4 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_2 \vee \overline{x}_5 \vee x_3)$, and S contains the clauses $\overline{x}_1 \vee \overline{x}_4$ and $\overline{x}_1 \vee x_2$; the corresponding explanations $E_1 = \{x_4\}$ and $E_2 = \{\overline{x}_2\}$ are output. In Step 2, the pair $(\overline{x}_2 \vee \overline{x}_5 \vee x_3, \overline{x}_1 \vee x_2)$ is found in O which satisfies condition (i) in Def. 1. Moreover, (ii) is satisfied, since $\varphi^\star \not\models \overline{x}_5 \vee x_3$. Thus, a prime implicate within $d = \overline{x}_1 \vee \overline{x}_5 \vee x_3$ is computed; in fact, d is prime. Therefore, $E_3 = \{x_5, \overline{x}_3\}$ is output and d is added to S, and then O is enlarged. Eventually, the pair $(\overline{x}_3 \vee \overline{x}_4, \overline{x}_1 \vee \overline{x}_5 \vee x_3)$ from O, which satisfies condition (i), will be considered. However, $\varphi^\star \models \overline{x}_4 \vee \overline{x}_5$, and thus S remains unchanged. Hence, the output of N-EXPLANATIONS is E_1, E_2, and E_3. As can be seen, these are all nontrivial explanations for $\overline{x}_1$ from φ. □

We remark that our algorithm is similar in spirit to an algorithm for computing all prime implicates of a Horn CNF in polynomial total time [2]. Our algorithm solves a more constrained problem, though.

In the rest of this section, we show that Algorithm N-EXPLANATIONS generates all explanations in polynomial total time. For that, we first show its correctness, which splits into a soundness and completeness part, and then analyze its time complexity.

Algorithm N-EXPLANATIONS
Input: An acyclic Horn CNF φ and an atom q.
Output: All nontrivial explanations of the query $\chi = \overline{q}$ from φ.

Step 1. $\varphi^\star := \emptyset$, $S := \emptyset$, and $O := \emptyset$;
Step 2. **for each** $c \in \varphi$ **do**
  add any prime implicate $c' \subseteq c$ of φ to $\varphi^\star$;
  for each $c' \in \varphi^\star$ with $q \in N(c')$ and $c' \notin S$ **do**
  begin output $(\{\overline{\ell} \mid \ell \in c' \setminus \{\overline{q}\}\}$;
    $S := S \cup \{c'\}$; $O := O \cup \{(c, c') \mid c \in \varphi^\star, q \notin P(c)\}$
  end;
Step 3. **while** some $(c, c') \in O$ exists **do**
  begin $O := O \setminus \{(c, c')\}$;
    if (1) c and c' resolve and (2) $\varphi^\star \not\models (c \oplus c' \setminus \{\overline{q}\})$
    then begin $d := c \oplus c'$;
      compute any prime implicate $d' \subseteq d$ of φ;
      if $d' \notin S$ **then**
      begin output $(\{\overline{\ell} \mid \ell \in d' \setminus \{\overline{q}\}\}$; $S := S \cup \{d'\}$;
        $O := O \cup \{(d'', d') \mid d'' \in \varphi^\star, q \notin P(d'')\} \cup \{(d'', d') \mid d'' \in S\}$
      end
    end
  end. □

Fig. 1. Algorithm computing all nontrivial explanations of a query χ=$\overline{q}$ on an acyclic Horn theory

As for soundness, it is easily seen that Algorithm N-EXPLANATIONS produces output only if d' is some prime implicate of $\varphi^\star$ (and thus of φ) such that $q \in N(d')$. Thus, from Lemma 2, we immediately obtain

Lemma 5 (Soundness of N-EXPLANATIONS). *Algorithm* N-EXPLANATIONS *outputs only nontrivial explanations for $\overline{q}$ from φ.*

It is much more difficult to show the completeness, i.e., that Algorithm N-EXPLANATIONS actually generates all nontrivial explanations. Intuitively, the difficulty stems from the fact that the restricted resolution procedure retains only prime clauses containing $\overline{q}$, and, moreover, may skip relevant prime implicates $d' \subseteq c \oplus c'$ in Step 3 if condition (ii) fails, i.e., $c \oplus c'$ is an implicate of φ (which is tantamount to the condition that $c \oplus c'$ contains some prime implicate of φ that does not contain $\overline{q}$). To see that no explanation is missed requires a careful analysis of how the desired explanations are generated, and leads to a nontrivial argument which takes the complex interaction between clauses into account.

We need a number of preliminary technical lemmas on which our proof builds, which are interesting in their own right. In what follows, we call a Horn clause c *definite*, if $P(c) \neq \emptyset$. Furthermore, for any literal ℓ, a clause c is a *ℓ-clause* if c contains ℓ.

The following propositions are well-known.

Proposition 3. *Let c_1, c_2 be Horn implicates of a Horn CNF φ that resolve. Then, $c = c_1 \oplus c_2$ is Horn, and if c_1 contains a negative implicate of φ, then also $c_1 \oplus c_2$ contains a negative implicate of φ.*

Proposition 4 (cf. [2]). *Every prime implicate c of a Horn CNF φ has an input resolution proof from it, i.e., a resolution proof $c_1, c_2, \ldots, c_l\ (= c)$ such that either $c_i \in \varphi$ or $c_i = c_j \oplus c_k$ where $j, k < l$ and either $c_j \in \varphi$ or $c_k \in \varphi$, for all $i \in \{1, \ldots, l\}$.*

We start with the following lemma.

Lemma 6. *Let φ be a prime Horn CNF, and let c be any prime implicate of φ such that $c \notin \varphi$. Then, $c = c_1 \oplus c_2$, where c_1 is a prime implicate contained in φ, and either (i) c_2 is a prime implicate of φ, or (ii) $c_2 = c \cup \{\ell\}$ where $c_1 \setminus \{\bar{\ell}\} \subset c$ and c is the unique prime implicate of φ contained in c_2.*

Note that item (ii) is needed in this lemma, as shown by the following example.

Example 5. Consider the Horn CNF $\varphi = (\overline{x}_0 \vee \overline{x}_1 \vee x_2)(\overline{x}_2 \vee \overline{x}_3)(\overline{x}_3 \vee x_0)$. As easily checked, φ is prime and has a further prime implicate $\overline{x}_1 \vee \overline{x}_3$, which can not be derived as the resolvent of any two prime implicates of φ. Note that φ is acyclic. □

Next we state some important properties of acyclic Horn CNFs under resolution.

Proposition 5. *Let φ be an acyclic Horn CNF, and let $c = c_1 \oplus c_2$ where $c_1, c_2 \in \varphi$. Then, $\varphi' = \varphi \wedge c$ is acyclic Horn, and the dependency graphs $G(\varphi)$ and $G(\varphi')$ have the same transitive closure. Furthermore, any subformula $\varphi'' \subseteq \varphi$ is acyclic Horn.*

Thus, adding repeatedly clauses derived by resolution preserves the acyclicity of a CNF, and, moreover, the possible topological sortings of the dependency graph.

The following proposition captures that for an acyclic Horn CNF, resolution cannot be blocked because of multiple resolving pairs of x_i and $\overline{x}_i$ of literals.

Proposition 6. *Let φ be an acyclic Horn CNF, and let c_1 and c_2 be any implicates of φ derived from φ by resolution. Then, c_1 and c_2 do not resolve iff $P(c_1) \cap N(c_2) = \emptyset$ and $P(c_2) \cap N(c_1) = \emptyset$.*

We define an ordering on Horn clauses as follows. Suppose that $\leq$ imposes a total ordering on the atoms ($x_{i_1} \leq x_{i_2} \leq \cdots \leq x_{i_n}$). Then, for any Horn clauses c_1 and c_2, define $c_1 \leq c_2$ iff $c_1 = c_2$ or one of the following conditions holds:

(i) $P(c_1) \neq \emptyset$ and $P(c_2) = \emptyset$;
(ii) $P(c_1) = \{x_i\}$ and $P(c_2) = \{x_j\}$ and $x_i < x_j$;
(iii) $P(c_1) = P(c_2)$ and $\max N(c_1) \triangle N(c_2) \in c_1$, where "$\triangle$" denotes standard symmetric difference (i.e., $S_1 \triangle S_2 = (S_1 \cup S_2) \setminus (S_1 \cap S_2)$).

As usual, we write $c_1 < c_2$ if $c_1 \leq c_2$ and $c_1 \neq c_2$, $c_1 > c_2$ for $c_2 < c_1$ etc. Note that $\leq$ orders first all definite Horn clauses along their positive literals, followed by the negative clauses. Notice that $c_1 \subset c_2$ implies $c_2 < c_1$, for any Horn clauses c_1 and c_2.

The following proposition is not difficult to establish:

Proposition 7. *Every total ordering $\leq$ of the atoms At induces a total ordering $\leq$ of all Horn clauses over At as described.*

With respect to acyclic Horn CNFs φ, in the rest of this paper we assume an arbitrary but fixed total ordering $\leq$ of the atoms which is compatible with some topological sorting of the dependency graph $G(\varphi)$.

Proposition 8. *Let c_1 and c_2 be Horn clauses such that $c = c_1 \oplus c_2$ exists. Then, $c_1 < c$ and $c_2 < c$ hold.*

Corollary 1. *Let φ be an acyclic Horn CNF, and let c, c_1 and c_2 be any implicates of φ derived from φ such that $c \subseteq c_1 \oplus c_2$. Then, $c > c_1$ and $c > c_2$ holds.*

Consequently, in any input resolution proof of a clause from an acyclic Horn CNF the derived clauses increase monotonically. As for the derivation of prime implicates, we find for such CNFs a more general form than in Lemma 6:

Lemma 7. *Let φ be an acyclic prime Horn CNF, and let c be any prime implicate of φ such that $c \notin \varphi$. Then, there are prime implicates c_1 and c_2 of φ and, for some $k \geq 0$, prime implicates $d_1, d_2, \ldots, d_k$ and literals $\ell_1, \ell_2, \ldots, \ell_k$, respectively, such that: (i) $c_1, d_1, \ldots, d_k \in \varphi$, and (ii) $c = c_1 \oplus e_1$, where $e_i = c \cup \{\ell_i\} = d_i \oplus e_{i+1}$, for $i \in \{1, \ldots, k\}$, and $e_{k+1} = c_2$, such that e_i contains the single prime implicate c.*

An immediate consequence of this result is that prime implicates of an acyclic Horn CNF can be generated from two prime implicates as follows.

Corollary 2. *Let φ be an acyclic prime Horn CNF, and let c be any prime implicate of φ such that $c \notin \varphi$. Then, there exist prime implicates c_1 and c_2 of φ which resolve such that either (i) $c = c_1 \oplus c_2$ or (ii) $c_1 \oplus c_2 = c \cup \{\ell\}$, where $\ell \notin c$ and c is the unique prime implicate of φ contained in $c_1 \oplus c_2$.*

In Example 5, the further prime implicate $\overline{x}_1 \vee \overline{x}_3$ can be derived as in case (ii) of Corollary 2: For $c_1 = \overline{x}_0 \vee \overline{x}_1 \vee x_2$ and $c_2 = \overline{x}_2 \vee \overline{x}_3$, we have $c_1 \oplus c_2 = \overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_0$, and $c = \overline{x}_1 \vee \overline{x}_3$ is the unique prime implicate of φ contained in $c_1 \oplus c_2$.

After the preparatory results, we now show that Algorithm N-EXPLANATIONS is complete. Using an inductive argument on clause orderings, we show that all explanations are generated by taking into account possible derivations of prime implicates as established in Lemma 7 and Corollary 2. However, an inductive proof along $\leq$ encounters two major difficulties: First, the resolvent $c = c_1 \oplus c_2$ of two clauses is *larger* than c_1 and c_2, thus we cannot simply rearrange resolution steps and appeal to smaller clauses. Second, Algorithm N-EXPLANATIONS does not generate prime implicates d' by a resolution step alone, but using *minimization* in Step 3; that is, a prime implicate *included* in the resolvent $d = c \oplus c'$. A respective statement is much more difficult to prove than the one if d were prime.

In order to overcome these difficulties, we use a more sophisticated ordering of clause pairs (c, c') and establish as a stepping stone the following key lemma. For ease of reference, let us say that resolvable implicates c_1 and c_2 of a Horn CNF φ satisfy the (technical) property $(*)$, if the following conditions hold:

1. At least one of c_1 and c_2 is prime.
2. If c_i is not prime, then it is of form $c_i = c' \cup \{\ell\}$, where c' is the unique prime implicate of φ contained in c_i ($i \in \{1, 2\}$), and c_i occurs in some derivation of c' as in Lemma 7.
3. There is no implicate $c'_1 \subset c_1$ (resp., $c'_2 \subset c_2$) of φ such that $c = c'_1 \oplus c_2$ (resp., $c = c_1 \oplus c'_2$).

Lemma 8 (Key Lemma). *Let φ be a prime acyclic Horn CNF, and let c_1 and c_2 be resolvable clauses satisfying $(*)$ such that $\overline{q} \in c := c_1 \oplus c_2$. Suppose that $c_i \in S$ if c_i is prime and $q \in N(c_i)$ (resp., $c_i' \in S$ if $c_i = c_i' \cup \{\ell_i\}$ where c_i' is prime and $q \in N(c_i')$) for $i \in \{1, 2\}$. Then at least one of the following conditions hold:* (i) *$c \setminus \{\overline{q}\}$ is an implicate of φ, or* (ii) *c contains a $\overline{q}$-clause from S.*

Proof. (Outline) We prove the statement using an inductive argument which involves clause orderings and takes into account how the clauses c_1 and c_2 are recursively generated. Depending on the shape of c_1 and c_2, we consider different cases.

Consider first the case in which both c_1 and c_2 contain $\overline{q}$. Then, w.l.o.g. $c = \overline{q} \vee \overline{a} \vee \overline{b}\,(\vee x_i)$, $c_1 = \overline{q} \vee \overline{a} \vee \overline{x}\,(\vee x_i)$, and $c_2 = \overline{b} \vee x \vee \overline{q}$. Here, $\overline{a}$ and $\overline{b}$ are disjunctions of negative literals, while x is a single atom; "$(\vee x_i)$" means the optional presence of x_i.

Both c_1 and c_2 contain a unique prime implicate c_1' resp. c_2' of φ (where possibly $c_i' = c_i$). If $\overline{q} \in c_i'$, then by assertion we have $c_i' \in S$. Thus, if both c_1' and c_2' contain $\overline{q}$, Algorithm N-EXPLANATIONS considers $c_1' \oplus c_2'$, which implies the statement. No other cases are possible, since either c_1' or c_2' must contain $\overline{q}$ (since c_1 or c_2 is prime) and condition 3 of $(*)$ excludes that exactly one of c_1' and c_2' contains $\overline{q}$. This proves the statement if both c_1 and c_2 contain $\overline{q}$.

For the other cases, assume that $\overline{q} \in c_1$ and $\overline{q} \notin c_2$ and prove the statement by induction along the lexicographic ordering of the pairs (c_1, c_2), where the clauses c_1 are in *reverse* ordering $\geq$ and the clauses c_2 in regular ordering $\leq$. We distinguish the following cases:

Definite/Negative Case 1 (DN1): $c = \overline{q} \vee \overline{a} \vee \overline{b}\,(\vee x_i)$, $c_1 = \overline{q} \vee \overline{a} \vee \overline{x}\,(\vee x_i)$, and $c_2 = \overline{b} \vee x$. That is, the $\overline{q}$-clause c is generated by resolving a $\overline{q}$-clause c_1 with a non-$\overline{q}$-clause c_2, where the positive resolution literal x is in c_2.

Definite/Negative Case 2 (DN2): $c = \overline{q} \vee \overline{a} \vee \overline{b}\,(\vee x_i)$, $c_1 = \overline{q} \vee \overline{a} \vee x$, and $c_2 = \overline{b} \vee \overline{x}\,(\vee x_i)$. That is, the $\overline{q}$-clause c is generated by resolving a $\overline{q}$-clause c_1 with a non-$\overline{q}$-clause c_2, where the positive resolution literal x is in c_1.

The statement is shown by a careful analysis of parent clauses of c_1 and c_2, and by reordering and adapting resolution steps. DN1 recursively only involves cases of the same kind (in fact, for negative c_1 we need to appeal only to smaller instances (c_1', c_2') where c_1' is negative), while DN2 recursively involves itself as well as DN1. □

By combining Lemma 8 with Proposition 8 and Corollary 2, we obtain by an inductive argument on the clause ordering $\leq$ the desired completeness result.

Lemma 9 (Completeness of N-EXPLANATIONS). *Algorithm* N-EXPLANATIONS *outputs all nontrivial explanations for a query $\chi = \overline{q}$ from an acyclic Horn CNF φ.*

Proof. We prove by induction on $\leq$ that S contains each $\overline{q}$-prime implicate c of φ.
(Basis) Let c be the least prime implicate of φ which contains $\overline{q}$. From Proposition 8 and Corollary 2, we conclude that $c \in \varphi$ must hold. Hence, $c \in S$.
(Induction) Suppose the claim holds for all $\overline{q}$-prime implicates c' of φ such that $c' < c$, and consider c. By Corollary 2, there exist prime implicates c_1 and c_2 such that either (i) $c = c_1 \oplus c_2$ or (ii) c is the unique prime implicate contained in $c_1 \oplus c_2 = c \cup \{\ell\}$ where $\ell \notin c$. By Proposition 8 and the induction hypothesis, we have $c_i \in S$ if $q \in N(c_i)$

holds for $i \in \{1, 2\}$. Consequently, c_1 and c_2 satisfy the conditions of Lemma 8. Hence, either (a) $c_1 \oplus c_2 \setminus \{\overline{q}\}$ is an implicate of φ, or (b) $c_1 \oplus c_2$ contains a $\overline{q}$-clause c' from S. Since $q \in N(c)$ and c is the unique prime implicate contained in $c_1 \oplus c_2$, we have (b). It follows from the uniqueness of c that $c' = c$, which proves the statement. □

We are now in a position to establish the main result of this section. Let $\|\varphi\|$ denote the size (number of symbols) of any CNF φ.

Theorem 2. *Algorithm* N-EXPLANATIONS *incrementally outputs, without duplicates, all nontrivial explanations of $\chi = \overline{q}$ from φ. Moreover, the next output (respectively termination) occurs within $O(s \cdot (s + m) \cdot n \cdot \|\varphi\|)$ time, where m is the number of clauses in φ, n the number of atoms, and s the number of explanations output so far.*

Proof. By Lemmas 5 and 9, it remains to verify the time bound. Computing a prime implicate $c' \subseteq c$ and $d' \subseteq d$ of φ in Steps 2 and 3, respectively, is feasible in time $O(n \cdot \|\varphi\|)$ (cf. Proposition 1), and thus the outputs in Step 2 occur with $O(m \cdot n \cdot \|\varphi\|)$ delay. As for Step 3, note that O contains only pairs (c, c') where $c \in \varphi^\star \cup S$ and $c' \in S$ such that the explanation corresponding to c' was generated, and each such pair is added to O only once. Thus, the next output or termination follows within $s \cdot (s + m)$ runs of the while-loop, where s is the number of solutions output so far. The body of the loop can be done, using proper data structures, in $O(n \cdot \|\varphi\|)$ time (for checking $d' \notin S$ efficiently, we may store S in a prefix tree). Thus, the time until the next output resp. termination is bounded by $O(s \cdot (s + m) \cdot n \cdot \|\varphi\|)$. □

Corollary 3. *Computing polynomially many explanations for a negative query $\chi = \overline{q}$ from an acyclic Horn CNF φ is feasible in polynomial time (in the size of the input).*

We conclude this section with some remarks on Algorithm N-EXPLANATIONS.

(1) As for implementation, standard data structures and marking methods can be used to realize efficient update of the sets O and S, to determine resolvable clauses, and to eliminate symmetric pairs (c, c') an (c', c) in O.

(2) Algorithm N-EXPLANATIONS is incomplete for cyclic Horn theories, as shown by the following example.

Example 6. Consider the Horn CNF $\varphi = (\overline{x}_0 \vee \overline{x}_1 \vee x_2)(\overline{x}_0 \vee \overline{x}_1 \vee x_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee \overline{x}_3)(\overline{x}_2 \vee \overline{x}_3 \vee x_4)$ over $x_0, \ldots, x_4$. Note that all clauses in φ are prime, and that x_2 and x_3 are symmetric. There are three further prime implicates, viz. $c_1 = \overline{x}_1 \vee \overline{x}_2 \vee x_4$, $c_2 = \overline{x}_1 \vee \overline{x}_3 \vee x_4$, and $c_3 = \overline{x}_0 \vee \overline{x}_1 \vee x_4$. Thus, $\overline{q} = \overline{x}_0$ has the nontrivial explanations $E_1 = \{x_1, \overline{x}_2\}$, $E_2 = \{x_1, \overline{x}_3\}$, and $E_3 = \{x_1, \overline{x}_4\}$. Apply then algorithm N-EXPLANATIONS on input φ and $q = x_0$. While it outputs E_1 and E_2, it misses explanation E_3. □

Algorithm N-EXPLANATIONS may be extended to handle this example and others correctly by adding in Step 2 prime implicates to $\varphi^\star$ which are generated in polynomial time (e.g., by minimizing clauses derived by resolution proofs from $\varphi^\star$ whose number of steps is bounded by a constant).

(3) Algorithm N-EXPLANATIONS is no longer complete if we constrain the resolution process to input resolution, i.e., consider only pairs (c, c') in Step 3 where at least one of c and c' is from φ (which means that in the update of O in Step 3, the part "$\{(d'', d') \mid d'' \in S\}$" is omitted). This is shown by the following example.

Example 7. Consider the Horn CNF $\varphi = (\overline{x}_0 \lor x_1)(\overline{x}_1 \lor \overline{x}_2 \lor x_3)(\overline{x}_1 \lor \overline{x}_3 \lor x_4)$ over $x_0, \dots, x_4$. As easily seen, φ is acyclic. Moreover, φ is prime. There are three further prime implicates containing $\overline{x}_0$, viz. $c_1 = \overline{x}_0 \lor \overline{x}_2 \lor x_3$, $c_2 = \overline{x}_0 \lor \overline{x}_3 \lor x_4$, and $c_3 = \overline{x}_0 \lor \overline{x}_2 \lor x_4$. Hence, $\overline{q} = \overline{x}_0$ has the nontrivial explanations $E_1 = \{\overline{x}_1\}$, $E_2 = \{x_2, \overline{x}_3\}$, $E_3 = \{x_3, \overline{x}_4\}$, and $E_4 = \{x_2, \overline{x}_4\}$. If at least one of the clauses (c, c') in Step 3 must be from φ, then E_2 and E_3 are generated from $(\overline{x}_1 \lor \overline{x}_2 \lor x_3, \overline{x}_0 \lor x_1)$ and $(\overline{x}_1 \lor \overline{x}_3 \lor x_4, \overline{x}_0 \lor x_1)$, respectively, while E_4 is missed: The pairs $(\overline{x}_1 \lor \overline{x}_3 \lor x_4, \overline{x}_0 \lor \overline{x}_2 \lor x_3)$ and $(\overline{x}_1 \lor \overline{x}_2 \lor x_3, \overline{x}_0 \lor \overline{x}_3 \lor x_4)$ yield the same resolvent $\overline{x}_0 \lor \overline{x}_1 \lor \overline{x}_2 \lor x_4$, for which $\varphi^\star \not\models (c \oplus c' \setminus \{\overline{q}\})$ fails since $\overline{x}_1 \lor \overline{x}_2 \lor x_4$, which is the resolvent of the last two clauses in φ, is an implicate. Note that E_4 is generated from each of the excluded symmetric pairs $(\overline{x}_0 \lor \overline{x}_2 \lor x_3, \overline{x}_0 \lor \overline{x}_3 \lor x_4)$ and $(\overline{x}_0 \lor \overline{x}_3 \lor x_4, \overline{x}_0 \lor \overline{x}_2 \lor x_3)$. $\square$

In terms of generating prime implicates, this contrasts with the cases of computing all prime implicates of a Horn CNF and all prime implicates that contain a positive literal q , for which input-resolution style procedures are complete, cf. [2, 11].

5 Compound Queries

In this section, we consider generating all explanations for queries beyond literals. Theorem 1 implies that this problem is intractable for any common class of CNF queries which admits a negative literal. However, also for positive CNFs, it is intractable.

Theorem 3. *Deciding whether a given CNF χ has an explanation from a Horn CNF φ is* NP*-complete. Hardness holds even if χ is positive and φ is negative (thus acyclic).*

Proof. Membership in NP easily follows from Lemma 1. Hardness is shown via a reduction from the classical EXACT HITTING SET problem. Let $\mathcal{S} = \{S_1, \dots, S_m\}$ be a collection of subsets $S_i \subseteq U$ of a finite set U. Construct $\chi = \bigwedge_i \big(\bigvee_{u \in S_i}\big)$ and $\varphi = \bigwedge_i \bigwedge_{x \neq y \in S_i} (\overline{x} \lor \overline{y})$. Then χ has an explanation from φ iff there exists an exact hitting set for $\mathcal{S}$, i.e., a set $H \subseteq U$ such that $|H \cap S_i| = 1$ for all $i \in \{1, \dots, m\}$. $\square$

For important special cases of positive CNFs, we obtain positive results. In particular, this holds if the query χ is restricted to be a clause or a term.

Theorem 4. *Computing polynomially many (resp., all) explanations for a query χ which is either a positive clause or a positive term from a Horn CNF φ is feasible in polynomial time (resp., polynomial total time).*

Proof. Let us first consider the case in which χ is a positive clause $c = \bigvee_{x \in P(c)} x$. Then let $\varphi^* = \varphi \land \bigwedge_{x \in P(c)} (\overline{x} \lor x^*)$, where x^* is a new letter. As easily seen, φ^* is a Horn CNF and there is a one-to-one correspondence between explanations for a query χ from φ and the ones for x^* form φ^* (except for a trivial explanation x^*). This, together with the result in [11] that all explanations for a query $\chi = q$ where q is an atom from a Horn CNF can be generated with incremental polynomial delay, proves the theorem.

Similarly, if χ is a positive term $t = \bigwedge_{x \in P(t)} x$, one can consider explanations for x^* from the Horn CNF $\varphi^* = \varphi \land (\bigvee_{x \in P(t)} \overline{x} \lor x^*)$, where x^* is a new letter. $\square$

In case of acyclic Horn theories, the positive result holds also in the case where negative literals are present in a clause query.

Query χ	CNF		single literal		single clause		single term	
Horn theory Σ, by	general	positive	atom q	$\overline{q}$	positive	general	positive	general
Horn CNF φ	NPTT	NPTT	PTT[a]	NPTT	PTT	NPTT	PTT	NPTT
Acyclic Horn CNF φ	NPTT	NPTT	PTT[a]	PTT	PTT	PTT	PTT	–

[a] By the results of [11].

Table 1. Complexity of computing all abductive explanations for a query χ from a Horn theory (PTT = polynomial total time, NPTT = not polynomial total time unless P=NP)

Theorem 5. *Computing polynomially many (resp., all) explanations for a query $\chi = c$ where c is a clause from an acyclic Horn CNF φ is feasible in polynomial time (resp., polynomial total time).*

Proof. Let $\chi = \bigvee_{x \in P(c)} x \vee \bigvee_{x \in N(c)} \overline{x}$. Then let $\varphi^* = \varphi \wedge \bigwedge_{x \in P(c)} (\overline{x} \vee \overline{x^*}) \wedge \bigwedge_{x \in N(c)} (x \vee \overline{x^*})$, where x^* is a new letter. It is not difficult to see that φ^* is an acyclic Horn CNF, and there is a ono-to-one correspondence between explanations for a query χ from φ and the ones for $\overline{x^*}$ from φ^* (except for a trivial explanation $\overline{x^*}$). This together with Theorem 2 proves the theorem. □

Note that explanations for a single clause query $\chi = c$ correspond to the minimal support clauses for c as used in Clause Management Systems [22]. Thus, from Theorems 1 and 5 we obtain that while in general, generating all minimal support clauses for a given clause c is not possible in polynomial total time unless $\mathrm{P} = \mathrm{NP}$, it is feasible with incremental polynomial delay for acyclic Horn theories.

The presence of negative literals in a query $\chi = t$ for a term t from an acyclic Horn theory is more involved; a similar reduction technique as for a clause to a single literal seems not to work. We can show that generating all nontrivial explanations E (i.e., $E \cap \chi = \emptyset$) for a term is intractable; the case of all explanations is currently open.

6 Conclusion

We considered computing all abductive explanations for a query χ from a propositional Horn CNF φ, which is an important problem that has many applications in AI and Computer Science. We presented a number of new complexity results, which complement and extend previous results in the literature; they are compactly summarized in Table 1.

We showed the intractability of computing all abductive explanations for a negative literal query χ from a general Horn CNF φ (thus closing an open issue), while we presented a polynomial total time algorithm for acyclic Horn CNFs. Since this amounts to computing all prime implicates of φ which contain $\overline{q}$, we have obtained as a byproduct also new results on computing all such prime implicates from a Horn CNF. Note that our intractability result contrasts with the result in [2] that all prime implicates of a Horn CNF are computable in polynomial total time. Furthermore, our results on clause queries imply analogous results for generating all minimal support clauses for a clause in a Clause Management System [22].

It remains for further work to complete the picture and to find further meaningful input classes of cyclic Horn theories which permit generating a few resp. all explanations in polynomial total time. For example, this holds for clause queries from quadratic

Horn CNFs (i.e., each clause is Horn and has at most 2 literals) and for literal queries from Horn CNFs in which each clause contains the query literal. Another issue is a similar study for the case of predicate logic.

Acknowledgments. We thank the reviewers for helpful suggestions.

References

1. K. Apt and M. Bezem. Acyclic programs. *New Generation Comp.*, 9(3-4):335–364, 1991.
2. E. Boros, Y. Crama, and P. L. Hammer. Polynomial-time inference of all valid implications for Horn and related formulae. *Ann. Math. and Artif. Int.*, 1:21–32, 1990.
3. G. Brewka, J. Dix, and K. Konolige. *Nonmonotonic Reasoning – An Overview*. Number 73 in CSLI Lecture Notes. CSLI Publications, Stanford University, 1997.
4. T. Bylander. The monotonic abduction problem: A functional characterization on the edge of tractability. In *Proc. KR-91*, 70-77. Morgan Kaufmann, 1991.
5. L. Console, D. Theseider Dupré, and P. Torasso. On the relationship between abduction and deduction. *J. Logic and Computation*, 1(5):661–690, 1991.
6. J. de Kleer. An assumption-based truth maintenance system. *Artif. Int.*, 28:127–162, 1986.
7. A. del Val. On some tractable classes in deduction and abduction. *Artif. Int.*, 116(1-2):297–313, 2000.
8. A. del Val. The complexity of restricted consequence finding and abduction. In *Proc. National Conference on Artificial Intelligence (AAAI-2000)*,pp. 337–342, 2000.
9. W. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn theories. *J. Logic Programming*, 3:267–284, 1984.
10. T. Eiter and G. Gottlob. The complexity of logic-based abduction. *JACM*, 42(1):3–42, 1995.
11. T. Eiter and K. Makino. On computing all abductive explanations. In *Proc. AAAI-02*, pp. 62–67. AAAI Press, July 2002.
12. T. Eiter and K. Makino. Generating all abductive explanations for queries on propositional Horn theories. Tech. Rep. INFSYS RR-1843-03-09, Institute of Inf.Sys., TU Vienna, 2003.
13. K. Eshghi. A tractable class of abduction problems. In *Proc. IJCAI-93*, pp. 3–8, 1993.
14. G. Friedrich, G. Gottlob, and W. Nejdl. Hypothesis classification, abductive diagnosis, and therapy. In *Proc. Int'l Workshop on Expert Sys. in Engineering*, LNCS 462, pp. 69–78, 1990.
15. K. Inoue. Linear resolution for consequence finding. *Artif. Int.*, 56(2-3):301–354, 1992.
16. H. Kleine Büning and T. Lettmann. *Aussagenlogik - Deduktion und Algorithmen*. B.G. Teubner, Stuttgart, 1994.
17. A. Leitsch. *The Resolution Calculus*. Springer, 1997.
18. P. Marquis. Consequence finding algorithms. In D. Gabbay, Ph. Smets (eds), *Handbook of Defeasible Reasoning and Uncertainty Management Systems* (V), pp. 41-145. Kluwer, 2000.
19. C. S. Peirce. Abduction and induction. In J. Buchler, editor, *Philosophical Writings of Peirce*, chapter 11. Dover, New York, 1955.
20. D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(1):97–110, 1989.
21. D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artif. Int.*, 64:81-130, 1993.
22. R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: Preliminary report. In *Proc. AAAI-87*, pp. 183–188, 1982.
23. B. Selman and H. J. Levesque. Abductive and default reasoning: A computational core. In *Proc. AAAI-90*, pp. 343–348, 1990.
24. B. Selman and H. J. Levesque. Support Set Selection for Abductive and Default Reasoning. *Artif. Int.*, 82:259–272, 1996.
25. B. Zanuttini. New Polynomial Classes for Logic-Based Abduction. *J. Artificial Intelligence Research*, 2003. To appear.