New Polynomial Classes for Logic-Based Abduction (Technical Report)

Bruno Zanuttini zanutti@info.unicaen.fr
GREYC, Université de Caen, Boulevard du Maréchal Juin
14032 Caen Cédex, France

Abstract

We address the problem of propositional logic-based abduction, i.e., the problem of searching for a best explanation for a given propositional observation according to a given propositional knowledge base. We give a general algorithm, based on the notion of projection; then we study restrictions over the representations of the knowledge base and of the query, and find new polynomial classes of abduction problems. We also show that our algorithm unifies several previous results.[1]

1. Introduction

Abduction consists in searching for a plausible explanation for a given observation. For instance, if $p \models q$ then p is a plausible explanation for the observation q. More practically, abduction arises for instance when one wants to explain a system's failure, knowing its usual behaviour. It can also formalize the search for a set of actions to realize for achieving a given goal, or the search for a minimal number of points to check in order to make sure of another point, etc. To summarize, abduction consists in searching for a set of facts (the *explanation*) that, conjointly with a given *knowledge base*, imply a given *query* (the observation to be explained, or the goal to be achieved etc.). This process is also constrained by a set of *hypotheses* among which the explanations have to be chosen, and by a preference criterion among them, with respect to which we search for a *best* explanation.

The problem of abduction proved its practical interest in many domains. For instance, it has been used to formalize text interpretation (Hobbs et al., 1993), system (Coste-Marquis & Marquis, 1998; Stumptner & Wotawa, 2001) or medical diagnosis (Bylander et al., 1989, Section 6). It is also closely related to configuration problems (Amilhastre et al., 2002), to the ATMS/CMS (Reiter & de Kleer, 1987), to default reasoning (Selman & Levesque, 1990) and even to induction (Goebel, 1997).

In this paper, we are interested in the complexity of *propositional logic-based* abduction, which means we assume both the knowledge base and the query are represented by propositional formulas. Even in this framework, many different formalizations have been proposed in the literature (Eiter & Gottlob, 1995), mainly differing about the definition of an hypothesis and that of a best explanation. We assume here that the hypotheses are the conjunctions of literals formed upon a distinguished subset of the variables involved, and that a best explanation is one no proper subconjunction of which is an explanation (*subset-minimality* criterion).

1. This paper is an extended version of a research note published in *Journal of Artificial Intelligence Research* (vol. 19, pages 1–10, 2003), and is meant to provide detailed proofs and examples.

Our purpose is to exhibit new polynomial classes of abduction problems. We give a general algorithm for finding a best explanation in the framework defined above, independently from the syntactic form of the formulas representing the knowledge base and the query. Then we explore the syntactic forms that allow a polynomial running time for this algorithm. We find new polynomial classes of abduction problems, among which the one restricting the knowledge base to be represented by a Horn DNF and the query by a positive CNF, and the one restricting the knowledge base to be represented by an affine formula and the query by a disjunction of linear equations; finally, we show that our algorithm unifies several previous such results.

The paper is organized as follows. We first recall the useful notions of propositional logic, formalize the problems (Section 2) and briefly survey previous work about the complexity of abduction (Section 3). Then we give our algorithm (Section 4) and explore polynomial classes for it (Section 5). Finally, we discuss our results and perspectives (Section 6).

2. Preliminaries

We present in this section the notions that are useful for understanding the rest of the paper. We end with a motivating example that will be reused later on.

2.1 Notions of Propositional Logic

We assume a countable number of propositional variables $x_1, x_2 \ldots$ and the standard connectives $\neg, \wedge, \vee, \oplus, \rightarrow, \leftrightarrow$. A *literal* is either a variable x_i (*positive* literal) or its negation $\neg x_i$ (*negative* literal). A propositional formula is a well-formed formula built on a finite number of variables and on the connectives; $Var(\phi)$ denotes the set of variables that occur in the propositional formula ϕ. A *clause* is a finite disjunction of literals, and a propositional formula is in *Conjunctive Normal Form (CNF)* if it is written as a finite conjunction of clauses. For instance, $\phi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$ is in CNF. The dual notions of clause and CNF are the notions of *term* (finite conjunction of literals) and *Disjunctive Normal Form (DNF)* (finite disjunction of terms).

An *assignment* to a set of variables V is a set of literals m that contains exactly one literal per variable in V, and a *model* of a propositional formula ϕ is an assignment m to $Var(\phi)$ that satisfies ϕ in the usual way, where m assigns 1 to x_i iff $x_i \in m$; we also write m as a tuple, e.g., 0010 for $\{\neg x_1, \neg x_2, x_3, \neg x_4\}$. We denote by $m[i]$ the value assigned to x_i by m, and by $\mathcal{M}(\phi)$ the set of all the models of a propositional formula ϕ; ϕ is said to be *satisfiable* if $\mathcal{M}(\phi) \neq \emptyset$. A propositional formula ϕ is said to *imply* a propositional formula ϕ' (written $\phi \models \phi'$) if $\mathcal{M}(\phi) \subseteq \mathcal{M}(\phi')$. More generally, we identify sets of models with Boolean functions, and thus use the notations $\overline{\mathcal{M}}$ (negation), $\mathcal{M} \vee \mathcal{M}'$ (disjunction) and so on.

2.2 Projection

The notion of *projection* is very important for the rest of the paper. For m an assignment to a set of variables V and $A \subseteq V$, write $Select_A(m)$ for the set of literals in m that are formed upon A, e.g., $Select_{\{x_1,x_2\}}(0110) = 01$. Projecting a set of assignments onto a subset A of the variables intuitively consists in replacing each assignment m with $Select_A(m)$; for sake of simplicity however, we define the projection of a set of models $\mathcal{M}$ to be built upon the same set of variables as $\mathcal{M}$. This yields the following definition.

Definition 1 (projection) *Let $V = \{x_1, \ldots, x_n\}$ be a set of variables, $\mathcal{M}$ a set of assignments to V and $A \subseteq V$. The* projection *of $\mathcal{M}$ onto A is the set of assignments to V $\mathcal{M}_{|A} = \{m \mid \exists m' \in \mathcal{M}, Select_A(m') = Select_A(m)\}$.*

For instance, let $\mathcal{M} = \{0001, 0010, 0111, 1100, 1101\}$ be a set of assignments to $V = \{x_1, x_2, x_3, x_4\}$, and let $A = \{x_1, x_2\}$. Then it is easily seen that

$$\mathcal{M}_{|A} = \{0000, 0001, 0010, 0011\} \cup \{0100, 0101, 0110, 0111\} \cup \{1100, 1101, 1110, 1111\}$$

since $\{Select_A(m) \mid m \in \mathcal{M}\} = \{00, 01, 11\}$.

Remark that the projection of the set of models of a formula ϕ onto a set of variables A is the set of models of the most general consequence of ϕ that is independent of all the variables not in A. Note also that the projection of $\mathcal{M}(\phi)$ onto A is the set of models of the formula obtained from ϕ by forgetting its variables not occurring in A. For more details about variable forgetting and independence we refer the reader to the work by Lang et al. (Lang et al., 2002).

It is useful to note some straightforward properties of projection. Let $\mathcal{M}, \mathcal{M}'$ denote two sets of assignments to the set of variables V, and let $A \subseteq V$. First, projection is distributive over disjunction, i.e., $(\mathcal{M} \vee \mathcal{M}')_{|A} = \mathcal{M}_{|A} \vee \mathcal{M}'_{|A}$. Now it is distributive over conjunction when $\mathcal{M}$ does not depend on the variables $\mathcal{M}'$ depends on, i.e., when there exist $A, A' \subseteq V$, $A \cap A' = \emptyset$ with $\mathcal{M}_{|A} = \mathcal{M}$ ($\mathcal{M}$ does not depend on $V \backslash A$) and $\mathcal{M}'_{|A'} = \mathcal{M}'$, $(\mathcal{M} \wedge \mathcal{M}')_{|A} = \mathcal{M}_{|A} \wedge \mathcal{M}'_{|A}$ holds; note that this is not true in the general case. Note finally that in general $(\overline{\mathcal{M}})_{|A}$ is not the same as $\overline{\mathcal{M}_{|A}}$.

2.3 Our Model of Abduction

As said in the introduction, we assume in abduction problems that the knowledge base Σ and the query α are propositional formulas, that the hypotheses are the conjunctions of literals formed upon a distinguished subset A of $Var(\Sigma)$ (the set of *abducibles*), and that the best explanations are those no proper subconjunction of which is still an explanation. We formalize these notions below.

Definition 2 (abduction problem) *A triple $\Pi = (\Sigma, \alpha, A)$ is called an* abduction problem *if Σ and α are satisfiable propositional formulas and A is a set of variables with $Var(\alpha), A \subseteq Var(\Sigma)$; Σ is called the* knowledge base *of Π, α its* query *and A its* set of abducibles.

Remark that our definition allows an abducible (or its negation) to occur in the query; the running example below will justify this possibility, and for a longer discussion we refer the reader to Eiter and Gottlob's paper (Eiter & Gottlob, 1995).

Definition 3 (hypothesis,explanation) *Let $\Pi = (\Sigma, \alpha, A)$ be an abduction problem. An* hypothesis *for Π is a set of literals formed upon A (seen as their conjunction), and an hypothesis E for Π is called an* explanation *for Π if (i) $\Sigma \wedge E$ is satisfiable and (ii) $\Sigma \wedge E \models \alpha$. If no proper subconjunction of E satifies both points (i) and (ii), E is called a* best explanation *for Π.*

Note that this framework does not allow one to specify that a variable must occur unnegated (resp. negated) in an explanation. We do not think this is a prohibiting restriction, since abducibles are intuitively meant to represent the variables whose values can be modified, imposed, observed or repaired, and then no matter their sign in an explanation. We however note that this is a restriction, and that a more general framework could be defined where the abducibles are literals and the hypotheses, conjunctions of abducibles, as is done by Marquis (2000).

We are interested in the computational complexity of computing a best explanation of a given abduction problem, or asserting there is none at all. Following the usual model, we establish complexities with respect to the size of the representations of Σ and α and to the number of abducibles; for hardness results, the following associated decision problem is usually considered: is there at least one explanation for Π? Obviously, if this latter problem is hard, then the function problem also is.

2.4 A Motivating Example

We now give a simple example of an abduction problem that we will reuse later on. Let us consider a burglar who wishes to visit a house, but wants to make sure its inhabitants are in holidays, since on one hand he wants them to be far for a long time, and on the other hand he does not want them to have moved house, for otherwise he would not find anything to steal.

Thus our burglar has a knowledge base Σ_e encoding the habits of people, and he wants to know what points to check for being sure the query "inhabitants are far from here but did not moved house" holds. Of course, he cannot check directly whether they are far nor whether they have moved; but he can check, for instance, whether their shutters are closed or whether they answer the phone.

More formally, assume his knowledge base is built over the following propositional variables: sh, meaning the shutters are closed; d, meaning it is day (and not night); p, meaning the inhabitants of the house answer the phone; sl, meaning they are sleeping (in the house); e, meaning the house is empty; f, meaning the inhabitants are far from home. Now assume this knowledge base, Σ_e, is the conjunction of the following five formulas (the set of all the models of Σ_e is given in Figure 1 with examples of their intuitive meanings as situations):

- $sl \rightarrow (sh \wedge \neg d)$, i.e., if the inhabitants are sleeping then the shutters are closed and it is night
- $sl \rightarrow (\neg e \wedge \neg f)$, i.e., if the inhabitants are sleeping then their house is not empty and they are not far from home
- $(e \vee f) \rightarrow \neg p$, i.e., if the house is empty or if the inhabitants are far from home, then they will not answer the phone
- $(\neg d \wedge \neg e \wedge \neg f) \rightarrow sl$, i.e., if it is night, if the house is not empty and if the inhabitants are not far from home, then we can deduce they are sleeping
- $sh \rightarrow (sl \vee e \vee f)$, i.e., if the shutters are closed, then we can deduce that either the inhabitants are sleeping or their house is empty or they are far from it.

sh	d	p	sl	e	f	intuitive meaning
0	0	0	0	0	1	in holidays
0	0	0	0	1	0	moved house
0	0	0	0	1	1	moved house
0	1	0	0	0	0	at work
0	1	0	0	0	1	in holidays
0	1	0	0	1	0	moved house
0	1	0	0	1	1	moved house
0	1	1	0	0	0	at home

sh	d	p	sl	e	f	intuitive meaning
1	0	0	0	0	1	in holidays
1	0	0	0	1	0	moved house
1	0	0	0	1	1	moved house
1	0	0	1	0	0	sleeping at home
1	0	1	1	0	0	sleeping at home
1	1	0	0	0	1	in holidays
1	1	0	0	1	0	moved house
1	1	0	0	1	1	moved house

Figure 1: The set of all the models of Σ_e

Our burglar would like to explain his query $\alpha_e = (\neg e \wedge f)$ over the set of abducibles $A_0 = \{sh, d, p\}$ (i.e., those points he can check directly). But the abduction problem $(\Sigma_e, \alpha_e, A_0)$ has no explanation, i.e., with the set of abducibles A_0, whathever the results of his observations are, the burglar cannot make sure the inhabitants are far from home and the house is not empty.

Now he must relax the set of abducibles, and he decides to take the risk to assume the house is not empty. Consequently, the set of abducibles A_0 becomes $A_e = A_0 \cup \{e\} = \{sh, d, p, e\}$, and we finally define our example abduction problem:

$$\Pi_e = (\Sigma_e, \alpha_e, A_e)$$

Now a best explanation for this problem is $E_e = (sh \wedge d \wedge \neg e)$, i.e., if the shutters are closed during the day and the house is not empty, then the burglar can be sure (as far as his knowledge base Σ_e is correct) that the house is not empty and that its inhabitants are far from it; indeed, it holds that $\Sigma_e \wedge sh \wedge d \wedge \neg e$ is satisfiable and implies $(\neg e \wedge f)$ (thus E_e is an *explanation* for Π_e), and no proper subconjunction of E_e has both these properties (thus E_e is a *best* explanation for Π_e). Remark that for this problem a variable occurs usefully in both the query and the set of abducibles; indeed, not only $\neg e$ explains $\neg e$ itself, but it also participates in explaining f.

3. Previous Work

The main general complexity results about propositional logic-based abduction with subset-minimality preference were stated by Eiter and Gottlob (1995). The authors show that deciding whether a given abduction problem has a solution at all is a Σ_2^P-complete problem, even if $A \cup Var(\alpha) = Var(\Sigma)$ and Σ is in CNF.

As stated as well by Selman and Levesque (1990), they also establish that this problem becomes "only" NP-complete when Σ is Horn, and even acyclic Horn; we refer the reader to their paper for definitions, but simply recall that the class of Horn CNFs is one of the most important ones for representing propositional knowledge, and that for most other reasoning tasks it admits polynomial algorithms: SAT, deduction (Dowling & Gallier, 1984), identification (Dechter & Pearl, 1992; Zanuttini & Hébrard, 2002) etc.

More generally, note that when SAT and deduction are polynomial with Σ the problem is obviously in NP. But in fact, very few classes of abduction problems are known to be polynomial for the search for explanations. As far as we know, the only such classes are those defined by the following restrictions (once again we refer the reader to the references for definitions):

- Σ is in 2CNF and α is in 2DNF (Marquis, 2000, Section 4.2)
- Σ is given as a monotone CNF and α as a clause (Marquis, 2000, Section 4.2)
- Σ is given as a definite Horn CNF and α as a conjunction of positive literals (Selman & Levesque, 1990; Eiter & Gottlob, 1995)
- Σ is given as an acyclic Horn CNF with pseudo-completion unit-refutable and α is a variable (Eshghi, 1993)
- Σ has bounded induced kernel width and α is given as a literal (del Val, 2000)
- Σ is represented by its set of characteristics models (with respect to a particular basis) and α is a variable (Khardon & Roth, 1996); note that a set of characteristic models is *not* a propositional formula, but that the result is however in the same vein as the other ones
- Σ is represented by the set of its models, or, equivalently, by a DNF with every variable occurring in each term, and α is any propositional formula.

The first two classes are proved polynomial with a general method for solving abduction problems with the notion of prime implicants, the last one is obvious since all the information is explicitely given in the input, and the four others are exhibited with *ad hoc* algorithms.

Let us also mention that Amilhastre et al. (2002) study most of the related problems in the more general framework of multivalued theories instead of propositional formulas, i.e., when the domain of the variables is not restricted to be $\{0, 1\}$. The authors mainly show, as far as this paper is concerned, that deciding whether there exists an explanation is still a Σ_2^P-complete problem (Amilhastre et al., 2002, Table 1).

Note that not all these results are stated in our exact framework in the papers cited above, but that they all still hold in it. Let us also mention that the problem of *enumerating* all the best explanations for a given abduction problem is of great interest; Eiter and Makino (2002) provide a discussion and some first results about it, mainly in the case when the knowledge base is Horn.

4. A General Algorithm

We now give the principle of our algorithm. Let us stress first that, as well as Marquis' construction (Marquis, 2000, Section 4.2) for instance, its outline matches point by point the definition of a best explanation for an abduction problem; our ideas and Marquis' are anyway rather close, as will be discussed in Section 5.3.

We are first interested in the hypotheses in which every abducible $x \in A$ occurs (either negated or unnegated); let us call them *full hypotheses*. Note indeed that every explanation E for an abduction problem is a subconjunction of a full explanation F; indeed, since E is by definition such that $\Sigma \wedge E$ is satisfiable and implies α, it suffices to let F be $Select_A(m)$ for a model m of $\Sigma \wedge E \wedge \alpha$. Minimization of F will be discussed later on.

Our algorithm's principle is then based on the following proposition. The idea is that a full explanation F must be $Select_A(m)$ for an assignment m to $Var(\Sigma)$ such that (i) $m \notin (\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}$, otherwise m would be a model of $\Sigma \wedge F \wedge \overline{\alpha}$, thus $\Sigma \wedge F \models \alpha$ would not hold, and (ii) $m \models \Sigma$, to ensure that $\Sigma \wedge F$ is satisfiable.

Proposition 1 *Let* $\Pi = (\Sigma, \alpha, A)$ *be an abduction problem, and* F *a full hypothesis of* Π. *Then* F *is an explanation for* Π *if and only if there exists an assignment* m *to* $Var(\Sigma)$ *with* $F = Select_A(m)$ *and* $m \in \mathcal{M}(\Sigma) \wedge \overline{(\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}}$.

Proof Assume first F is an explanation for Π. Then (i) there exists an assignment m to $Var(\Sigma)$ with $m \models \Sigma \wedge F$, thus $F = Select_A(m)$ and $m \in \mathcal{M}(\Sigma)$, and (ii) $\Sigma \wedge F \models \alpha$, i.e., $\Sigma \wedge F \wedge \overline{\alpha}$ is unsatisfiable, thus $F \notin \{Select_A(m) \mid m \in \mathcal{M}(\Sigma \wedge \overline{\alpha})\}$, thus $m \notin (\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}$, thus $m \in \overline{(\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}}$.

Conversely, if $m \in \mathcal{M}(\Sigma) \wedge \overline{(\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}}$ let $F = Select_A(m)$. Then we have (i) since $m \in \mathcal{M}(\Sigma)$, $\Sigma \wedge F$ is satisfiable, and (ii) since $m \notin (\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}$, there is no $m' \in \mathcal{M}(\Sigma \wedge \overline{\alpha})$ with $Select_A(m') = F$, thus $\Sigma \wedge F \wedge \overline{\alpha}$ is unsatisfiable, thus $\Sigma \wedge F \models \alpha$. □

Thus we have characterized the full explanations for an abduction problem. Now since, as remarked above, every explanation is a subset of a full one, once we have found a convenient F there is only left to minimize it. It is easily seen that the following greedy procedure (Selman & Levesque, 1990) reduces F into a best explanation for Π:

For every literal $\ell \in F$ **do**
  If $\Sigma \wedge F \backslash \{\ell\} \models \alpha$ **then** $F \leftarrow F \backslash \{\ell\}$ **endif**;
Endfor;

Note that depending on the order in which the literals $\ell \in F$ are considered the result may be different, but that in all cases it will be a best explanation for Π.

Finally, we can give our general algorithm for computing a best explanation for a given abduction problem $\Pi = (\Sigma, \alpha, A)$; its correctness follows directly from Proposition 1:

$\Sigma' \leftarrow$ a propositional formula with $\mathcal{M}(\Sigma') = \mathcal{M}(\Sigma) \wedge \overline{(\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}}$;
If Σ' is unsatisfiable **then**
  return "No explanation";
Else
  $m \leftarrow$ a model of Σ';
  $F \leftarrow Select_A(m)$;
  minimize F;
  return F;
Endif;

Let us demonstrate this algorithm on our running example. We first compute a formula Σ_e^1 with set of models $(\mathcal{M}(\Sigma_e \wedge \overline{\alpha_e}))_{|A_e}$, which yields for instance $\Sigma_e^1 = ((\neg sh \wedge \neg d) \rightarrow e) \wedge (p \rightarrow (sh \vee d)) \wedge (p \rightarrow \neg e) \wedge ((sh \wedge d) \rightarrow e)$; one of the models of Σ_e^1 is 101000, which means that the full hypothesis $Select_{A_e}(101000) = \{sh, \neg d, p, \neg e)$ cannot explain α_e. Now we compute Σ'_e with set of models $\mathcal{M}(\Sigma_e) \wedge \overline{(\mathcal{M}(\Sigma_e \wedge \overline{\alpha_e}))_{|A_e}} = \mathcal{M}(\Sigma_e) \wedge \overline{\mathcal{M}(\Sigma_e^1)}$, which yields for instance $\Sigma'_e = (sh \leftrightarrow d) \wedge \neg p \wedge \neg sl \wedge \neg e \wedge f$; its models are 000001 and 110001, thus the full hypotheses $(\neg sh \wedge \neg d \wedge \neg p \wedge \neg e)$ and $(sh \wedge d \wedge \neg p \wedge \neg e)$ are exactly the full explanations for Π_e. Finally, if we choose the full explanation $(sh \wedge d \wedge \neg p \wedge \neg e)$ for Π_e, we can get the best explanation $E_e = (sh \wedge d \wedge \neg e)$.

We are interested in the next Section in classes of abduction problems, defined by the syntactic forms of the representations of Σ and α, for which the running time of this algorithm is polynomial. Let us simply remark for the moment that it suffices that these restrictions allow to process efficiently the following operations: projecting $\Sigma \wedge \overline{\alpha}$ onto A, deciding the satifiability of Σ' and computing one of its models, and finally deciding $\Sigma \wedge \bigwedge F \backslash \{\ell\} \models \alpha$. The other operations are purely syntactic computations with linear complexity.

5. Polynomial Classes

We first explore the new polynomial classes of abduction problems that our algorithm allows to exhibit (Sections 5.1 and 5.2) and then briefly explore what previously known classes are encompassed by our framework (Section 5.3). Throughout the section, n denotes the number of variables in $Var(\Sigma)$.

5.1 Affine Formulas

A propositional formula is said to be *affine* (or in *XOR-CNF*) (Schaefer, 1978; Kavvadias & Sideri, 1998; Zanuttini, 2002) if it is written as a finite conjunction of linear equations over the two-element field, e.g., $\phi = (x_1 \oplus x_3 = 1) \wedge (x_1 \oplus x_2 \oplus x_4 = 0)$. As can be seen, equations play the same role in affine formulas as clauses do in CNFs; roughly, affine formulas represent conjunctions of parity or equivalence constraints. This class proves very interesting for knowledge representation, since on one hand it is tractable for most of the common reasoning tasks, and on the other hand the affine approximations of a knowledge base can be made very small and are efficiently learnable (Zanuttini, 2002). The task of projecting an affine formula onto a subset of its variables appears to be quite easy too, as stated in the next lemma and demonstrated in the following example ($|S|$ denotes the number of elements in a set S).

Lemma 1 *Let ϕ be an affine formula containing k equations, and $A \subseteq Var(\phi)$. Then an affine formula ψ with $\mathcal{M}(\psi) = (\mathcal{M}(\phi))_{|A}$ and containing at most k equations can be computed in time $O(k^2|Var(\phi)|)$.*

Proof Let $<$ be a total order on $Var(\phi)$ such that $\forall x \in A, y \notin A, x > y$. First sort the variables inside each equation according to $<$, and triangulate ϕ into an equivalent affine formula ϕ' with the elimination method of Gauss in time $O(k^2|Var(\phi)|)$ (Curtis, 1984). Let ψ be the conjunction of all the equations of ϕ' that contain only variables of A; we prove that $\mathcal{M}(\psi) = (\mathcal{M}(\phi))_{|A}$. Indeed, if $m \models \psi$ then since ϕ' is triangular and the variables of ψ (i.e., in A) are the greatest with respect to $<$, $Select_A(m)$ can be extended into a model of ϕ' (i.e., of ϕ), thus $m \in (\mathcal{M}(\phi))_{|A}$. Conversely, if $m \in (\mathcal{M}(\phi))_{|A}$ then there is a M with $M \models \phi$, thus $M \models \phi'$, and $Select_A(M) = Select_A(m)$, and it follows that m satisfies every equation of ψ. $\square$

Example 1 *Let* $\phi = (x_1 \oplus x_2 \oplus x_3 \oplus x_4 = 0) \wedge (x_2 \oplus x_3 = 1) \wedge (x_1 \oplus x_3 = 1) \wedge (x_1 \oplus x_4 = 1)$ *and* $A = \{x_1, x_2\}$. *A convenient order on the variables is* $x_3 < x_4 < x_1 < x_2$, *and we rewrite* ϕ *in the following manner:*

$$\left(\begin{array}{cccc|c} x_3 & x_4 & x_1 & x_2 & 0 \\ x_3 & & & x_2 & 1 \\ x_3 & & x_1 & & 1 \\ & x_4 & x_1 & & 1 \end{array}\right)$$

Applying the elimination method of Gauss we get the equivalent triangular formula ϕ':

$$\left(\begin{array}{cccc|c} x_3 & x_4 & x_1 & x_2 & 0 \\ & x_4 & x_1 & & 1 \\ & & x_1 & x_2 & 0 \end{array}\right)$$

and we get $\psi = (x_1 \oplus x_2 = 0)$.

Now we can state the result for abduction. Even if we give it with a quite general form for α, for sake of completeness, we wish to remark that variables, literals and clauses are all special cases of disjunctions of linear equations.

Proposition 2 *If* Σ *is represented by an affine formula containing* k *equations and* α *by a disjunction of* k' *linear equations, and* A *is a subset of* $Var(\Sigma)$, *then searching for a best explanation for* $\Pi = (\Sigma, \alpha, A)$ *can be done in time* $O((k+k')((k+1)^2 + |A|(k+k'))n)$.

Proof We detail the computations of the algorithm of Section 4. Since α is a disjunction of linear equations, an affine formula for $\overline{\alpha}$ can be computed in time $O(k'|Var(\alpha)|)$ by replacing $\vee$ with $\wedge$ between the equations and inverting the right member in each one of them. This yields an affine formula (containing $k'+k$ equations and n variables) for $\Sigma \wedge \overline{\alpha}$; this formula can be projected onto A in time $O((k+k')^2 n)$ (Lemma 1), and this yields an affine formula Σ^1 of at most $k+k'$ equations for $(\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A}$ and, straightforwardly, a disjunction of at most $k+k'$ linear equations for $\overline{\Sigma^1}$.

Now we show that an affine formula for Σ' can be obtained in polynomial time. Write $\overline{\Sigma^1} = E_1 \vee E_2 \vee \ldots \vee E_{k+k'}$, where for all i, $E_i = (x_{i1} \oplus \ldots \oplus x_{ik_i} = b_i)$ is a linear equation; by distributivity of $\wedge$ over $\vee$, $\Sigma \wedge \overline{\Sigma^1}$ can be represented by $(\Sigma \wedge E_1) \vee (\Sigma \wedge E_2) \vee \ldots \vee (\Sigma \wedge E_{k+k'})$; thus the satisfiability of Σ' can be decided by solving at most $k+k'$ satisfiability problems for affine formulas; recall that SAT can be solved in time $O(k^2 n)$ for an affine formula of k equations over n variables by the elimination method of Gauss (Curtis, 1984). The model m can be computed at the same time, in overall time $O((k+k')(k+1)^2 n)$. There is only left to minimize F, but since deciding $\Sigma \wedge \bigwedge F \backslash \{\ell\} \models \alpha$ is the same as deciding whether $\Sigma \wedge \bigwedge F \backslash \{\ell\} \wedge \overline{\alpha}$ is unsatisfiable, this can be done in time $|A| \times O((k+k')^2 n)$. □

5.2 DNFs

Though the class of DNF formulas has very good computational properties, abduction remains a hard problem for it as a whole, even with additional restrictions. Recall that the TAUTOLOGY problem is the one of deciding whether a given DNF formula represents the identically true function, and that this problem is coNP-complete.

Proposition 3 *Deciding whether there is at least one explanation for a given abduction problem* (Σ, α, A) *is NP-complete when* Σ *is given in DNF, even if* α *is a variable and* $A \cup \{\alpha\} = Var(\Sigma)$*.*

Proof Membership in NP is obvious, since deduction with DNFs is polynomial. To establish completeness, we show that a DNF Σ is tautological if and only if the abduction problem $\Pi = (\Sigma \vee (x), x, Var(\Sigma))$ has no explanation, where x is a variable not occuring in Σ. First note that $\Sigma \vee (x)$ is in DNF by construction. Now if Σ is tautological, then every hypothesis H over $Var(\Sigma)$ satisfies Σ, and thus $H \wedge \{\neg x\}$ satisfies $\Sigma \vee (x)$; finally, H is not an explanation for Π. Conversely, if Σ is not tautological then we can find an assignment E to $Var(\Sigma)$ with $E \not\models \Sigma$; consequently, we have $E \wedge \{\neg x\} \not\models \Sigma \vee (x)$ and $E \wedge \{x\} \models \Sigma \vee (x)$, thus E is an explanation for Π. □

However, when Σ is represented by a DNF the task is easier than in the general case; indeed, for projecting Σ onto A it suffices to cancel its literals that are not formed upon A.

Lemma 2 *Let* ϕ *be a DNF containing* k *terms, and* $A \subseteq Var(\phi)$*. Then a DNF* ψ *with* $\mathcal{M}(\psi) = (\mathcal{M}(\phi))_{|A}$ *and containing at most* k *terms can be computed in time* $O(k|Var(\phi)|)$*.*

Proof Obvious since projection is distributive over $\vee$, and over $\wedge$ inside each term since the literals in a term are built upon different variables. □

Now we study classes of abduction problems that are polynomial for our algorithm and where Σ is restricted to be in a subclass of the class of DNFs. The first result is quite general, but we wish to note that its assumptions are satisfied by natural classes of DNFs: e.g., that of *Horn* (resp. *reverse Horn*) DNFs, i.e., those DNFs with at most one positive (resp. negative) literal per term; similarly, that of Horn-renamable DNFs, i.e., those that can be turned into a Horn DNF by replacing some variables with their negation, and simplifying double negations, everywhere in the formula; 2DNFs, those DNFs with at most two literals per term; and finally, *positive* (resp. *negative*) DNFs, those DNFs containing only positive (resp. negative) literals. For ϕ a DNF (resp. CNF), write $N(\phi)$ for the CNF (resp. DNF) obtained from ϕ by replacing $\vee$ with $\wedge$, $\wedge$ with $\vee$ and every literal with its negation.

Proposition 4 *Let* $\mathcal{D}$ *be a class of DNFs that is stable under removal of occurrences of literals and for which the TAUTOLOGY problem is polynomial. If* Σ *is restricted to belong to* $\mathcal{D}$*,* α *is a clause and* A *is a subset of* $Var(\Sigma)$*, then searching for a best explanation for* $\Pi = (\Sigma, \alpha, A)$ *can be done in polynomial time.*

Proof Write $\Sigma = T_1 \vee \ldots \vee T_k$, where each T_i is a term, and $\alpha = (\ell_1 \vee \ldots \vee \ell_{k'})$, where each ℓ_i is a literal. Then $\overline{\alpha}$ can be represented by $N(\alpha) = \overline{\ell_1} \wedge \ldots \wedge \overline{\ell_{k'}}$. Now write $\Sigma[\overline{\alpha}]$ the DNF obtained from Σ by propagating the $\overline{\ell_i}$'s, i.e., replacing every occurrence of a literal ℓ in Σ with 1 if ℓ is a literal of $N(\alpha)$ and with 0 if $\overline{\ell}$ is a literal of $N(\alpha)$, and simplifying the constants. It is easily seen that $\Sigma \wedge \overline{\alpha}$ is represented by $\Sigma[\overline{\alpha}] \wedge N(\alpha)$, and since $Var(\Sigma[\overline{\alpha}]) \cap Var(N(\alpha)) = \emptyset$ (by construction), that $(\mathcal{M}(\Sigma \wedge \overline{\alpha}))_{|A} = (\mathcal{M}(\Sigma[\overline{\alpha}]))_{|A} \wedge (\mathcal{M}(N(\alpha)))_{|A}$. Now since $\Sigma[\overline{\alpha}]$ is in DNF, it can be projected onto A in polynomial time (Lemma 2), yielding a DNF $\phi \in \mathcal{D}$ since $\mathcal{D}$ is closed under removal of occurrences of literals. Since $N(\alpha)$ is a term, its projection onto A is logically equivalent to the conjunction ψ of its literals formed on A. Thus we can compute $\Sigma' = \Sigma \wedge (N(\phi) \vee N(\psi))$, which is logically equivalent to $(\Sigma \wedge N(\phi)) \vee (\Sigma \wedge N(\psi))$. As for the SATISFIABILITY problem for $(\Sigma \wedge N(\phi))$, we can distribute $\wedge$ over $\vee$ for obtaining k satisfiability problems for formulas of the form $T_i \wedge N(\phi)$; since $\phi \in \mathcal{D}$ and thus the SATISFIABILITY problem for $N(\phi)$ is polynomial (it is equivalent to the TAUTOLOGY problem for ϕ), the satisfiability of $\Sigma \wedge N(\phi)$ can be decided, and one of its models computed, in polynomial time; now since $N(\psi)$ is a clause, the SATISFIABILITY problem for $\Sigma \wedge N(\psi)$ is polynomial as well. Finally, there is only left to minimize F, which can be done in polynomial time: indeed, deciding $\Sigma \wedge \bigwedge F \backslash \{\ell\} \models \alpha$ is equivalent to deciding whether $\Sigma \wedge \bigwedge F \backslash \{\ell\} \wedge \overline{\alpha}$ is unsatisfiable, i.e., to deciding whether for all terms T of Σ it holds that $T \wedge \bigwedge F \backslash \{\ell\} \wedge \overline{\alpha}$ is unsatisfiable. □

Thus we can establish that abduction is tractable if (among others) Σ is in Horn-renamable DNF (including the Horn and reverse Horn cases) or in 2DNF, and α is a clause.

Finally, let us point out that with a very similar proof we can obtain polynomiality for some problems obtained by strengthening the restriction of Proposition 4 over Σ, but weakening that over α.

Proposition 5 *If Σ is represented by a Horn (resp. reverse Horn) DNF of k terms and α by a positive (resp. negative) CNF of k' clauses, and A is a subset of $Var(\Sigma)$, then searching for a best explanation for $\Pi = (\Sigma, \alpha, A)$ can be done in time $O((k + |A|)kk'n)$. The same holds if Σ is represented by a positive (resp. negative) DNF of k terms and α by a Horn (resp. reverse Horn) CNF of k' clauses.*

Proof We prove the result for Σ in Horn DNF and α in positive CNF; the other case is dual. Since α is in positive CNF, one can compute in time $O(k'|Var(\alpha)|)$ a negative DNF for $\overline{\alpha}$ as above; then distributing $\wedge$ over $\vee$ (each term of the DNF for Σ being combined with a term of the DNF for $\overline{\alpha}$) yields a Horn DNF of at most kk' terms for $\Sigma \wedge \overline{\alpha}$ in time $O(kk'n)$; indeed, the combined terms are always one Horn term and one negative term, thus the resulting term contains at most one positive literal. It is easily seen that projecting this DNF onto A yields a Horn DNF Σ^1 of at most kk' terms in time $O(kk'n)$; thus a reverse Horn CNF for $\overline{\Sigma^1}$ can be computed in time $O(kk'n)$, and as for Proposition 4 we distribute $\wedge$ over $\vee$ for obtaining k satisfiability problems for formulas of the form $\overline{\Sigma^1} \wedge T_i$, where $\Sigma = T_1 \vee T_2 \vee \ldots \vee T_k$, each T_i being a term; since $\overline{\Sigma^1}$ is in reverse Horn CNF each of these satisfiability problems can be solved in linear time $O(kk'n)$ (Dowling & Gallier, 1984), and m can be computed at the same time. Finally, F can be minimized in overall time $O(|A|kk'n)$ as in the proof of Proposition 4. □

Once again note that variables, literals and terms are all special cases of (reverse) Horn CNFs, and that variables, positive (resp. negative) clauses and positive (resp. negative) terms are all special cases of positive (resp. negative) CNFs.

5.3 Previously Known Classes

Finally, we wish to emphasize that our algorithm runs in polynomial time as well for previously known polynomial classes of abduction problems. Indeed, this is true for Σ given as a 2CNF (i.e., a CNF with at most two literals per term) and α as a 2DNF, since in this case $\Sigma \wedge \overline{\alpha}$ is in 2CNF and can be projected onto A by testing all the candidates 2clauses formed upon A, which are only polynomially many (Marquis, 2000, Section 3.4.1); this projection is

representation of Σ	representation of α	References
affine formula	$\vee$ of linear equations	*
Horn-renamable DNF	clause	*
2DNF	clause	*
Horn DNF	positive CNF	*
reverse Horn DNF	negative CNF	*
negative DNF	reverse Horn CNF	*
positive DNF	Horn CNF	*
2CNF	2DNF	(Marquis, 2000), *
monotone CNF	clause	(Marquis, 2000), *
definite Horn CNF	positive term	(Eiter & Gottlob, 1995)
acyclic Horn CNF with...	variable	(Eshghi, 1993)
bounded induced kernel width	literal	(del Val, 2000)
characteristic models wrt $B \cup B_H$	positive formula	(Khardon & Roth, 1996)
set of its models	any formula	*

Table 1: Summary of the polynomial classes for abduction

in 2CNF, thus the satifiability problem of the algorithm is equivalent to a linear number of 2SATISFIABILITY problems by distributing $\wedge$ over $\vee$; finally, minimizing F is polynomial because deduction with 2CNFs is.

For similar reasons, it is easily seen that our algorithm is polynomial if Σ is given as a monotone CNF and α as a clause. More generally, Marquis (2000) shows that abduction is tractable for these two classes of problems because the set of all the prime implicates of Σ (resp. $\Sigma \wedge \overline{\alpha}$) can be computed efficiently. It is easily seen that when this is the case, our algorithm is polynomial as well and its behaviour is exactly the same.

Finally, it is also easily seen that our algorithm is polynomial when Σ is represented by the set of its models, whatever propositional formula α is.

6. Discussion and Perspectives

We have given an algorithm for propositional logic-based abduction and studied restrictions over the knowledge base and the query that allow it to run in polynomial time. Table 1 summarizes the main known polynomial restrictions of abduction problems (where each line corresponds to one restriction, and '*' in the last column means our algorithm is polynomial). The new polynomial classes our algorithm allows us to identify include the one restricting Σ to be given as an affine formula and α as a disjunction of linear equations, and the one restricting Σ to be given as a Horn DNF and α as a positive CNF; moreover, our algorithm is polynomial on other, previously known classes. Finally, even if there is no gaurantee for efficiency in the general case, its presentation does not depend on the syntactic form of Σ or α, and it uses only standard operations on Boolean functions (projection, conjunction, negation).

Another interesting feature of this algorithm is that before minimization it computes the explanations *intentionnally*. Thus all the full explanations can be enumerated with roughly the same delay that the models of the formula representing them (Σ'). However, there is no guarantee that two of them would not be minimized into the same *best* explanation, which prevents from concluding that our algorithm can enumerate all the *best* explanations; trying to extend it into this direction would be an interesting problem. For more details about enumeration we refer the reader to Eiter and Makino's work (Eiter & Makino, 2002).

As identified by Selman and Levesque (1990), central to the task is the notion of projection onto a set of variables, and our algorithm isolates this subtask. However, as pointed out before our notion of projection only concerns variables, and not literals, which prevents from imposing a sign to the literals the hypotheses are formed upon, contrariwise to more general formalizations proposed for abduction, as Marquis' (Marquis, 2000). Even if we think this is not a prohibiting restriction, as argued before, it would be interesting to try to fix that weakness of our algorithm while preserving its polynomial classes.

Another problem of interest is the behaviour of our algorithm when Σ and α are not only propositional formulas, but more generally *multivalued theories*, in which the domain of variables is not restricted to be $\{0,1\}$: e.g., signed formulas (Beckert et al., 1999). This framework is used, for instance, for configuration problems by Amilhastre et al. (2002). It is easily seen that our algorithm is still correct in this framework; however, there is still left to study in which cases its running time is polynomial.

Finally, problems of great interest are those of deciding the *relevance* or the *necessity* of an abducible (Eiter & Gottlob, 1995). An abducible x is said to be *relevant* to an abduction problem Π if there is at least one best explanation for Π containing x or $\neg x$, and *necessary* to Π if all best explanations for Π contain x or $\neg x$. It is easily seen that x is necessary for $\Pi = (\Sigma, \alpha, A)$ if and only if $\Pi' = (\Sigma, \alpha, A \backslash \{x\})$ has no explanation, hence showing that polynomial restrictions for the search for explanations are polynomial as well for deciding the necessity of an hypothesis as soon as they are stable under the substitution of $A \backslash \{x\}$ for A, which is the case for all restrictions considered in this paper. Contrastingly, we do not know of any such relation for relevance, and the study of this problem would also be of great interest.

Acknowledgments

A much shorter version of this paper appeared in French in *Proc. Huitièmes Journées Nationales sur la Résolution Pratique de Problèmes NP-Complets (JNPC'02)* (2002) 255–268; the author wishes to thank an anonymous referee of this version for very valuable and constructive comments. He also wishes to thank Jean-Jacques Hébrard for very useful discussions about this work and its redaction. Finally, he wishes to thank the referees of the version published in JAIR for constructive comments.

References

Amilhastre, J., Fargier, H., & Marquis, P. (2002). Consistency restoration and explanations in dynamic CSPs — application to configuration. *Artificial Intelligence*, *135*(1–2), 199–234.

Beckert, B., Hähnle, R., & Manyà, F. (1999). Transformations between signed and classical clause logic. In *Proc. 29th International Symposium on Multiple-Valued Logics (ISMVL'99)*, pp. 248–255. IEEE Computer Society Press.

Bylander, T., Allemang, D., Tanner, M., & Josephson, J. (1989). Some results concerning the computational complexity of abduction. In *Proc. 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pp. 44–54. Morgan Kaufmann.

Coste-Marquis, S., & Marquis, P. (1998). Characterizing consistency-based diagnoses. In *Proc. 5th International Symposium on Artificial Intelligence and Mathematics (AIMATH'98)*.

Curtis, C. (1984). *Linear algebra. An introductory approach.* Springer Verlag.

Dechter, R., & Pearl, J. (1992). Structure identification in relational data. *Artificial Intelligence*, *58*, 237–270.

del Val, A. (2000). The complexity of restricted consequence finding and abduction. In *Proc. 17th National Conference on Artificial Intelligence (AAAI'00)*, pp. 337–342. AAAI Press/MIT Press.

Dowling, W., & Gallier, J. (1984). Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, *3*, 267–284.

Eiter, T., & Gottlob, G. (1995). The complexity of logic-based abduction. *Journal of the ACM*, *42*(1), 3–42.

Eiter, T., & Makino, K. (2002). On computing all abductive explanations. In *Proc. 18th National Conference on Artificial Intelligence (AAAI'02)*, pp. 62–67. AAAI Press.

Eshghi, K. (1993). A tractable class of abduction problems. In *Proc. 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, pp. 3–8. Morgan Kaufmann.

Goebel, R. (1997). Abduction and its relation to constrained induction. In *Proc. IJCAI'97 workshop on abduction and induction in AI.*

Hobbs, J., Stickel, M., Appelt, D., & Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence*, *63*, 69–142.

Kavvadias, D., & Sideri, M. (1998). The inverse satisfiability problem. *SIAM Journal on Computing*, *28*(1), 152–163.

Khardon, R., & Roth, D. (1996). Reasoning with models. *Artificial Intelligence*, *87*, 187–213.

Lang, J., Liberatore, P., & Marquis, P. (2002). Conditional independence in propositional logic. *Artificial Intelligence*, *141*, 79–121.

Marquis, P. (2000). Consequence finding algorithms. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems (DRUMS)*, Vol. 5, pp. 41–145. Kluwer Academic.

Reiter, R., & de Kleer, J. (1987). Foundations of assumption-based truth maintenance systems: preliminary report. In *Proc. 6th National Conference on Artificial Intelligence (AAAI'87)*, pp. 183–188. AAAI Press/MIT Press.

Schaefer, T. (1978). The complexity of satisfiability problems. In *Proc. 10th Annual ACM Symposium on Theory Of Computing (STOC'78)*, pp. 216–226. ACM Press.

Selman, B., & Levesque, H. (1990). Abductive and default reasoning: a computational core. In *Proc. 8th National Conference on Artificial Intelligence (AAAI'90)*, pp. 343–348. AAAI Press.

Stumptner, M., & Wotawa, F. (2001). Diagnosing tree-structured systems. *Artificial Intelligence*, *127*, 1–29.

Zanuttini, B. (2002). Approximating propositional knowledge with affine formulas. In *Proc. 15th European Conference on Artificial Intelligence (ECAI'02)*, pp. 287–291. IOS Press.

Zanuttini, B., & Hébrard, J.-J. (2002). A unified framework for structure identification. *Information Processing Letters*, *81*(6), 335–339.