In search of the right extension

Jérôme Lang
IRIT-UPS
31062 Toulouse Cedex
France
lang@irit.fr

Pierre Marquis
CRIL/Université d'Artois
62307 Lens Cedex
France
marquis@cril.univ-artois.fr

Abstract

In this paper, the problem of purifying an assumption-based theory KB, i.e., identifying the right extension of KB using knowledge-gathering actions (tests), is addressed. Assumptions are just normal defaults without prerequisite. Each assumption represents all the information conveyed by an agent, and every agent is associated with a (possibly empty) set of tests. Through the execution of tests, the epistemic status of assumptions can change from "plausible" to "certainly true", "certainly false" or "irrelevant", and the KB must be revised so as to incorporate such a change. Because removing all the extensions of an assumption-based theory except one enables both identifying a larger set of plausible pieces of information and renders inference computationally easier, we are specifically interested in finding out sets of tests allowing to purify a KB (whatever their outcomes). We address this problem especially from the point of view of computational complexity.

1 Introduction

In the following, knowledge bases given by propositional assumption-based theories $KB = \langle W, \Delta \rangle$ (or equivalently, prerequisite-free normal default theories), are considered. W is a finite set of propositional formulas representing certain information about *the real world*, the latter being characterized by a specific (a priori unknown) model I. Δ is a finite set of defaults (or assumptions); defaults are just propositional formulas that are plausible, i.e., considered as true in absence of any contradictory information. Since defaults may be jointly contradictory, several belief states (called extensions) may result. Equivalently, we can also consider KB as a (typically inconsistent) stratified belief base, with two strata (the first one W being consistent).

The purpose of reasoning from KB is to identify formulas that are true in I, ideally all of them. Of course, in the general case, only some of the formulas true in I are accessible. Anyway, an important question is: what are the logical links between what is derivable from KB and what is true in I? When dealing with assumption-based theories, the following representation (meta)assumption is typically made:

(D) There exists a extension of KB (noted E_I) that is true in I.

Note that E_I is necessarily unique (because any two extensions of KB are jointly contradictory). It is important to note that (D) is a representation assumption (just like the one consisting in assuming that W is true in I). Eventually, it can be defeated at the light of certain information. Thus, there is no guarantee that every formula derived from KB through skeptical inference (i.e., every formula belonging to all the extensions of KB) is true in I. Nevertheless, such a (meta)assumption is mandatory to take defaults into account in reasoning. Without it, only the logical consequences of W would be considered as true in I and pure deduction from W is definitely an over-cautious form of inference in presence of defaults (since it does not exploit them at all).

Now, what can be done if we want to identify some other formulas as true in I? A classical way ([2]) consists in taking into consideration representation (meta)assumptions that are stronger than (D). Especially, if some priorities on defaults are available (e.g., a pre-order over Δ reflecting the fact that some defaults are more plausible than others is given), it can be valuable to take it into account and to consider that at least one *preferred* extension of KB is true in I. Indeed, restricting the set of extensions of KB by focus-

ing on the preferred ones may only lead to enlarge their intersection (hence the set of formulas that are considered as true in I). An extreme, most informative, case is achieved when all the extensions of KB are filtered out, except one. In this situation, every formula from the remaining extension is considered true in I. Additionally, provided that KB has only one extension, assumption-based reasoning becomes computationally easier (from the second level of the polynomial hierarchy to the first level)[1]. This is another reason that makes it highly desirable to have KB with one extension.

In this paper, a radically different but complementary approach is developed. We assume that every default from Δ represents all the pieces of information conveyed by one agent (a source of information); W represents the set of all pieces of information on which all agents agree (e.g., what must be true by nature). We also assume that each agent can perform some knowledge-gathering actions (informative tests). At the light of tests outcomes, the group of agents can change its mind about the epistemic status of defaults. As a consequence, revising KB with the outcome of a test can easily modify the set of its extensions. Specifically, contrariwise to what happens in the preferred extension approach, revising KB can lead to remove defaults in some extensions or even to generate new extensions, thus questioning the (meta)assumption (D).

In the following, each agent is associated with a single test whose outcome directly validates the corresponding default (changing its status from "plausible" to "certainly true"), invalidates the default (changing its status from "plausible" to "certainly false") or cancels it (changing its status to "irrelevant"). We show how such default verification tests can be generated from more general tests through a pre-processing phase, and we analyze the complexity of the pre-processing. *Purifying* a KB consists in determining a set of tests whose outcomes (whatever they are) generate an assumption-based theory with a unique extension after revision. We analyze the purification problem from the computational complexity point of view.

[1] When KB has only one extension, skeptical inference from KB (as well as argumentative and brave inference since, in this case, they all coincide) becomes computationally easier; indeed, while skeptical inference is Π_2^p-complete in the general case [7] [21], the complexity falls down to coNP-complete, when KB has a unique extension E_S (once its generating scenario S has been found out – and this can be achieved using only $card(\Delta)$ calls to an NP oracle).

2 Basics of assumption-based reasoning

We assume the reader acquainted with basic notions of propositional logic. Let $PROP_{PS}$ be a propositional language built up in the usual way from a set of variables PS. Throughout this paper, knowledge bases are represented by propositional assumption-based theories [16].

Definition 1 (assumption-based reasoning)

- *An* assumption-based theory KB *is an ordered pair* $KB = \langle W, \Delta \rangle$, *where* W *is a (conjunctively-interpreted) finite subset of* $PROP_{PS}$, *and* $\Delta = \{\phi_1, \ldots, \phi_n\}$ *is a finite subset of* $PROP_{PS}$ [2].
- *A* nogood *of* KB *is a minimal subset* N *of* Δ *(w.r.t.* $\subseteq$*) such that* $N \cup W$ *is not satisfiable.* $Nogood(KB)$ *is the set of all nogoods of* KB.
- *A* maximal scenario *of* KB *is a maximal subset* S *of* Δ *(w.r.t.* $\subseteq$*) such that* $S \cup W$ *is satisfiable.* $MaxScen(KB)$ *is the set of all maximal scenarios of* KB.
- *An* extension E_S *of* KB *is the deductive closure* $Cn(S \cup W)$, *where* S *is a maximal scenario of* KB. S *is called the* generating scenario *of* E_S. $Ext(KB)$ *is the set of all extensions of* KB.

Example 1 *Let* $KB = \langle W, \Delta \rangle$ *with* $W = \emptyset$ *and* $\Delta = \{a \wedge b, \neg a, \neg b \wedge c, d\}$.
We have $Nogood(KB) = \{\{a \wedge b, \neg a\}, \{a \wedge b, \neg b \wedge c\}\}$, $MaxScen(KB) = \{\{a \wedge b, d\}, \{\neg a, \neg b \wedge c, d\}\}$, *and* $Ext(KB) = \{Cn(\{a \wedge b, d\}), Cn(\{\neg a, \neg b \wedge c, d\})\}$.

Every assumption-based theory KB has at least a maximal scenario whenever W is satisfiable, and more than one whenever it has a nogood containing more than one assumption. It is easy to see that $MaxScen(KB)$ and $Ext(KB)$ are two similar sets since every extension of KB is generated by a unique maximal scenario of KB (the maximality requirement ensures that the union of two distinct maximal scenarios of KB is unsatisfiable with W). Accordingly, determining whether KB has only one extension can be decided in linear time when either $Nogood(KB)$ or $MaxScen(KB)$ are given (but both sets may contain exponentially many elements w.r.t. the size of KB in

[2] At a first glance, considering Δ as a set can appear too restrictive since it can be the case that two different agents convey the same piece of information: hence, a bag (or multiset) would be more convenient. Nevertheless, for simplicity and without loss of generality, Δ will be considered as a set: if two identical defaults are to be considered, it is sufficient to make them syntactically different, for instance by indexing them with integers, while preserving equivalence.

the worst case). It is well-known that $Nogood(KB)$ can be generated from $MaxScen(KB)$ and *vice-versa*, through the computation of minimal hitting sets [18] and set-theoretic complements in Δ.

3 Revising defaults using tests

In the following, we assume that each default ϕ_i of Δ represents all the information conveyed by the agent (or source) i. Some of the defaults ϕ_i are associated with a test t_i (at most one per default). Executing t_i can change the epistemic status of ϕ_i (considered plausible before execution) in one of the following ways:

- ϕ_i becomes certainly true, noted $ok(i)$.
- ϕ_i becomes certainly false, noted $bad(i)$.
- ϕ_i becomes irrelevant, noted $forget(i)$.

Definition 2 (default verification test)
A default verification test t_i *consists of a subset of* $\{ok(i), bad(i), forget(i)\}$ *containing* $ok(i)$ *and at least one additional element. The elements* r_i *of* t_i *are its* possible outcomes. *To each test* t_i *is attached a positive integer* k_i *representing the* cost *of performing* t_i.

Since the very purpose of testing a default ϕ_i is to validate it, $ok(i)$ is required to be among the possible outcomes of it.

While such tests can be considered quite restricted at a first glance, they nevertheless capture several important notions of verification:

- *default verification*: the piece of information itself (ϕ_i) given by the source i is verified. Assume for example that undergrad student Tweety programmed an algorithm for testing primality, which resulted after some laborious hours in software S. Running S on number 348754697 returns "yes" (ϕ_i is thus "348754697 is prime"). Tweety's ability in programming is known to be average, therefore, ϕ_i should be considered a default and not a fact. Now, Lea is an expert in prime numbers. She are fully reliable, thus, her answer to the question whether a given number is prime is always correct. Furthermore, Lea always gives an answer whenever the input number does not exceed a given maximum number N_{max} (much larger than 348754697), and sometimes gives an answer when the input number exceeds N_{max}. Asking Lea about the primality of 348754697 is a *default verification test* of the form $\{ok(i), bad(i)\}$: its outcome is $ok(i)$ (resp. $bad(i)$) if Lea answers "yes" (resp. "no"). Some years later, Tweety improves his software so as to deal with very large integers. Running the improved software on $N = 495729^{50876} + 34532^{3486}$ (which exceeds N_{max}) returns "yes", which leads to considering the default ϕ_j "N is prime". Asking Lea about the primality of N is a default verification test of the form $\{ok(j), bad(j), forget(j)\}$, because $N > N_{max}$. If Lea is not able to answer, i.e., if the outcome of the test is $forget(j)$, then we will not be ready to consider N as prime any longer, because Lea is an expert and the fact that she was not able to answer is a reason strong enough for changing the status of ϕ_j from plausible to irrelevant.

- *source verification*: the reliability of the source, thus the *relevance* of the piece of information, is checked. Not the piece of information itself. Bob is Tweety's teacher. Asking Bob to look carefully on Tweety's first program S returns yes or no whether S is valid or not. If it is, then the information S gave was correct and 348754697 is prime. If it is not, then anything could happen. The program might have been something like generating "yes" or "no" randomly. This cancels the default information that ϕ_i is true but certainly does not tell us that ϕ_i is false. It could well be true in I, *by chance* (here, 348754697 may well be prime even if S is not valid). Asking Bob is a test of the form $\{ok(i), forget(i)\}$.

Now, once t_i has been executed, KB must be revised so as to incorporate the information given by its resulting outcome r_i into KB. The revision process aims at characterizing the new assumption-based theory $KB * r_i$ obtained by incorporating the information given by r_i within KB:

Definition 3 (revising KB with a test outcome)
Let $KB = \langle W, \Delta \rangle$ *an assumption-based theory, and* t_i *be a test concerning default* ϕ_i *of* Δ. *We have:*

- **default validation**

$$KB * ok(i) = \langle W \cup \{\phi_i\}, \Delta \setminus \{\phi_i\} \rangle$$

The execution of t_i *has validated* ϕ_i. *The default is definitely true in* I. *It can be soundly added to* W *and removed from* Δ[3].

- **default invalidation**

$$KB * bad(i) = \langle W \cup \{\neg\phi_i\}, \Delta \setminus \{\phi_i\} \rangle$$

[3]Note that letting ϕ_i in Δ or not once ϕ_i has been added to W has no influence on $Ext(KB * ok(i))$.

The execution of t_i has invalidated ϕ_i. The default is definitely false in I. Its negation can be soundly added to W and ϕ_i can be removed from Δ [4].

- **default cancellation**

$$KB * forget(i) = \langle W, \Delta \setminus \{\phi_i\} \rangle$$

The default is considered irrelevant, it is removed from Δ. A frequent reason for cancelling a default is a source verification that fails (as explained above); the failure can be viewed as an argument against the default, but of course, this argument is not sufficiently strong for invalidating the default (we just consider it as irrelevant).

Tests are assumed reliable (their outcome is true in I) and non-intrusive (they have no effect on the world). Accordingly, revising KB by a sequence of test outcomes depends only on the outcome of each individual test, and is independent from the ordering in which tests are performed.

Definition 4 (verification problems)

- *A* verification problem *is an ordered pair $P = \langle KB, T \rangle$, where $KB = \langle W, \Delta \rangle$ is an assumption-based theory and $T = \{t_1, \ldots, t_k\}$ is the available set of tests. For every $\phi_i \in \Delta$, ϕ_i is* testable *iff $t_i \in T$; without loss of generality, we assume that only the first k defaults $\phi_1 \ldots \phi_k$ of Δ are testable ($0 \leq k \leq card(\Delta)$).*
- *A* set of tests $X = \{t_1, \ldots t_l\}$ *(given P) is any subset of T.*
- *Executing a set of tests $X = \{t_1, \ldots t_l\}$ results in a* state *(for P), i.e., a set $R = \{r_1, \ldots, r_l\}$ where each r_i ($i \in 1 \,..\, l$) is an element of t_i (one of its possible outcomes). It is convenient to partition every state R in $\{Ok(R), Bad(R), Forget(R)\}$ with $Ok(R) = \{i \mid ok(i) \in R\}$, $Bad(R) = \{i \mid bad(i) \in R\}$, $Forget(R) = \{i \mid forget(i) \in R\}$. $State(X)$ is the set of states that can possibly result from the execution of X.*
- *A state R (for P) is* consistent *iff $W \cup \{\phi_i \mid i \in Ok(R)\} \cup \{\neg\phi_i \mid i \in Bad(R)\}$ is satisfiable. Clearly, since tests are assumed reliable, all states in $State(X)$ are consistent.*
- *The* revision *of KB by state $R = \{r_1, \ldots, r_l\}$, noted $KB * R$, is the assumption-based theory defined by $KB * r_1 * \ldots * r_l$.*

[4] Note that letting ϕ_i in Δ or not once $\neg\phi_i$ has been added to W has no influence on $Ext(KB * bad(i))$.

For our purpose, it is important to identify the effect of each possible default revision (validation, invalidation, cancellation) on $Ext(KB)$. Since it is possible to determine efficiently whether KB has a unique extension given $Nogood(KB)$ or $MaxScen(KB)$, it is also important to analyze how both sets evolve after a revision.

The case of default validation corresponds to a "soft" knowledge expansion. By "soft" we mean that there is no *surprise*: what was considered as a plausible piece of information becomes now a certain piece of information. The effects are a (possible) removing of some of the current extensions, but never the generation of a new extension (point 3 below).

We make use of the following notations: if A is a set of subsets of a given set, then
$A_{min\subset} = \{B \in A \mid \text{there is no } B' \in A \text{ s.t. } B' \subset B\}$ and
$A_{min\supset} = \{B \in A \mid \text{there is no } B' \in A \text{ s.t. } B' \supset B\}$.

Proposition 1 (effects of a default validation)

1. $Nogood(KB * ok(i)) = \{N \setminus \{\phi_i\} \mid N \in Nogood(KB)\}_{min\subset}$.
2. $MaxScen(KB * ok(i)) = \{S \setminus \{\phi_i\} \mid S \in MaxScen(KB) \text{ and } \phi_i \in S\}$.
3. $Ext(KB * ok(i)) \subseteq Ext(KB)$.

Thus, the extensions of $KB * ok(i)$ are exactly those extensions of $Ext(KB)$ which contain ϕ_i. The other ones are removed. Consider Example 1 again with $\phi_1 = a \wedge b$. We have $Nogood(KB * ok(1)) = \{\{\neg a\}, \{\neg b \wedge c\}\}$ and $MaxScen(KB * ok(1)) = \{\{d\}\}$. Accordingly, $KB * ok(1)$ has a unique extension $Cn(\{a \wedge b, d\})$. Clearly enough, point 3. can be easily generalized (by induction) to *sequences of validation*: namely, if a state R contains only $ok(i)$ variables, then $Ext(KB * R) \subseteq Ext(KB)$.

Proposition 2 (effects of a default cancellation)

1. $Nogood(KB * forget(i)) = \{N \in Nogood(KB) \mid \phi_i \notin N\}$.
2. $MaxScen(KB * forget(i)) = \{S \setminus \{\phi_i\} \mid S \in MaxScen(KB)\}_{min\supset}$.
3. $\forall E \in Ext(KB * forget(i))$, *there is a* $E' \in Ext(KB)$ *such that* $E \subseteq E'$.

Default cancellation can be considered as a simple form of belief contraction. However, a default ϕ_i can survive its contraction in some sense; indeed, it can be the case that ϕ_i is a (skeptical) consequence of $KB * forget(i)$ (just consider the case where $\Delta = \{\phi_i, \phi_j\}$, $W \cup \{\phi_j\}$ is satisfiable and $\phi_j \models \phi_i$ holds). This is not an undesirable effect: if ϕ_i is a logical consequence of W

when some defaults of Δ are validated (or only considered as true), the fact that the corresponding verification test t_i has failed is not a sufficient argument for considering ϕ_i as irrelevant. The discrepancy with the belief contraction situation can be explained by the fact that, while both cancellation and contraction change the status of some information to "irrelevant", the status of a default before change is "plausible", not "certainly true".

With default cancellation, preservation of extensions is not guaranteed; indeed, it may be the case that one of the extensions of KB leads to a weaker one. Consider Example 1 again, where $\phi_2 = \neg a$. We have $Nogood(KB * forget(2)) = \{\{a \wedge b, \neg b \wedge c\}\}$ and $MaxScen(KB * forget(2)) = \{\{a \wedge b, d\}, \{\neg b \wedge c, d\}\}$. Thus $KB * forget(2)$ has two extensions, $E_1 = Cn(\{a \wedge b, d\})$ and $E_2 = Cn(\{\neg b \wedge c, d\})$. While E_1 is an extension of KB, E_2 is only a proper subset of the extension $Cn(\{\neg a, \neg b \wedge c, d\})$ of KB.

When it comes to sequential revision, generalizing points 3. of both Propositions 1 and 2 by induction on $card(R)$ easily gives the following result: if a state R is "bad-free", i.e., $Bad(R) = \emptyset$, then $\forall E \in Ext(KB * R)$, there is a $E' \in Ext(KB)$ s.t. $E \subseteq E'$. As a corollary, we have that $card(Ext(KB * R)) \leq card(Ext(KB))$: revising a KB with a "bad-free" state R does not increase the *number* of extensions of KB.

Default invalidation can be considered as a knowledge expansion with the negation of the default. Nevertheless, it is trickier than default validation: there may be new extensions appearing, and the number of extensions may increase. These more chaotic effects are in agreement with the intuition which tells that a default invalidation change the status of "plausible" to "certainly false", which leads to a *surprise* (contrariwise to default validation). Consider Example 1 again. We have $Nogood(KB * bad(2)) = \{\{a \wedge b, \neg b \wedge c\}\}$ and $MaxScen(KB * bad(2)) = \{\{a \wedge b, d\}, \{\neg b \wedge c, d\}\}$, thus $KB * bad(2)$ has two extensions: E_1 as above, and $E_3 = Cn(\{a, \neg b \wedge c, d\})$ *which is a new extension* – incompatible with any extension of KB.

Furthermore, while Propositions 1 and 2 show how both the nogoods and the maximal scenarios of a revised KB can be computed (in polynomial time) from, respectively, the nogoods and the maximal scenarios of KB whenever the revision operation is a default validation or a default cancellation, no similar results can be achieved when considering a default invalidation. Indeed, there is not enough information in $Nogood(KB)$ (or $MaxScen(KB)$) to compute $Nogood(KB * bad(i))$ (or $MaxScen(KB * bad(i))$) in the general case:

Example 2 *Let* $KB = \langle W, \Delta \rangle$, *with* $W = \{a \vee \neg b \vee \neg c\}$ *and* $\Delta = \{a, b, c\}$. *We have* $Ext(KB) = \{Cn(W \cup \{a, b, c\})\}$, $MaxScen(KB) = \{\{a, b, c\}\}$, *and* $Nogood(KB) = \emptyset$. *Now, given that* $\phi_1 = a$, $Ext(KB * bad(1)) = \{Cn(W \cup \{\neg a\} \cup \{b\}), Cn(W \cup \{\neg a\} \cup \{c\})\}$, $MaxScen(KB * bad(1)) = \{\{b\}, \{c\}\}$, *and* $Nogood(KB * bad(1)) = \{\{b, c\}\}$. *Let* $KB' = \langle \emptyset, \Delta \rangle$. *We have* $MaxScen(KB') = MaxScen(KB)$ *and* $Nogood(KB') = Nogood(KB)$. *Nevertheless, we do* **not** *have* $Ext(KB' * bad(1)) = Ext(KB * bad(1))$ *since* $Ext(KB' * bad(1)) = \{Cn(\{\neg a, b, c\})\}$.

Accordingly, while knowing only nogoods or maximal scenarios is sufficient to determine whether KB, or the revision of KB by a sequence of validations and cancellations, has only one extension, this is no longer the case when default invalidations are possible. This calls for a more informative notion, like the one of *generalized nogood*:

Definition 5 (generalized nogood) *A* generalized nogood *of* KB *is a minimal (w.r.t.* $\subseteq$*) subset* G *of* $\Delta \cup \{\neg\phi_i \mid \phi_i \in \Delta\}$ *such that* $G \cup W$ *is not satisfiable.* $GNogood(KB)$ *is the set of all generalized nogoods of* KB.

Clearly enough, we have
$GNogood(\langle W, \Delta \rangle) = Nogood(\langle W, \Delta \cup \{\neg\phi_i \mid \phi_i \in \Delta\} \rangle)$
and
$Nogood(\langle W, \Delta \rangle) = \{G \in GNogood(\langle W, \Delta \rangle) \mid G \subseteq \Delta\}$.
Accordingly, $Nogood(KB)$ can be computed in polynomial time from $GNogood(KB)$ by removing from it every element that is not a subset of Δ. Additionally:

Proposition 3 (computing $GNogood(KB * r_i)$**)**

1. $GNogood(KB * ok(i)) = \{G \setminus \{\phi_i\} | G \in GNogood(KB)\}_{min\subset}$.
2. $GNogood(KB * forget(i)) = \{G \in GNogood(KB) | \phi_i \notin G\}$.
3. $GNogood(KB * bad(i)) = \{G \setminus \{\neg\phi_i\} | G \in GNogood(KB)\}_{min\subset}$.

Accordingly, computing $GNogood(KB * R)$ from $GNogood(KB)$ and a consistent state R can be achieved in polynomial time.

4 Purifying KB

In this section, we focus on the problem consisting of the identification of *the* right extension E_I of KB. To this end, we look for sets of tests which discriminate among all extensions, so as to find E_I. This calls for the notion of *purification* of KB by a set of tests X.

4.1 N-purification vs. C-purification

As seen before, revising KB with a state may eventually generate extensions that are not in $Ext(KB)$, hence questioning the (meta)assumption (D). As a consequence, two different notions of purification must be considered, related to the fact that we are ready to defeat (D) or not. If (D) can be relaxed at the light of a new evidence, the objective of the revision process is that the revised KB has only one extension. If (D) cannot be questioned, a valuable revision process can lead to a revised KB with more than one extension, but only one of them must be among the initial extensions of KB.

Definition 6 (purifying states)
Given a verification problem $P = \langle KB, T\rangle$ and a consistent state R:

- *R* N-purifies *KB iff $KB * R$ has only one extension.*
- *R* C-purifies *KB iff $KB * R$ has only one extension E s.t. $E \in Ext(KB)$.*

Certainly, the uniqueness requirement in both definitions is quite idealistic and in practice one would rather try to do as good as possible (identifying as many inconsistencies as possible, even if it cannot be done totally). For the sake of simplicity, we ignore this issue in this paper.

It is important to see why both definitions of purification do not coincide (and none of them implies or excludes the other one):

Example 3 *Let $KB = \langle \emptyset, \{\neg a \wedge \neg b, a, b\}\rangle$. We have $Nogood(KB) = \{\{\neg a \wedge \neg b, a\}, \{\neg a \wedge \neg b, b\}\}$ and $MaxScen(KB) = \{\{\neg a \wedge \neg b\}, \{a, b\}\}$; thus, KB has two extensions, namely $E = Cn(\{\neg a \wedge \neg b\})$ and $E' = Cn(\{a, b\})$. Let $\phi_1 = \neg a \wedge \neg b$, $\phi_2 = a$, and $\phi_3 = b$. Let $R = \{bad(2)\}$. $KB * R = \langle \{\neg a\}, \{\neg a \wedge \neg b, b\}\rangle$ has* two *extensions, namely $Cn(\{\neg a \wedge \neg b\})$ and $Cn(\{\neg a, b\})$, and therefore R does not N-purify KB. Contrastingly, since $Ext(KB) \cap Ext(KB * R) = \{Cn(\{\neg a \wedge \neg b\})\}$ is a singleton, R C-purifies KB. Now, let $R' = \{forget(1), bad(2)\}$; $KB * R' = \langle \{\neg a\}, \{b\}\rangle$ has only one extension $Cn(\{\neg a, b\})$ and thus R' N-purifies KB, while R' does not C-purify KB. Lastly, $R'' = \{ok(3)\}$ both N-purifies and C-purifies KB.*

In N- and C-purification, N and C stand respectively for *no completion* and *completion*, which is justified by the following result where "forget-free" states are considered, only:

Proposition 4 (C- and N-purification)
Let $P = \langle KB, T\rangle$ be a verification problem, with $KB = \langle W, \Delta\rangle$, and let R be a consistent "forget-free" state (i.e., $Forget(R) = \emptyset$). The two statements below are equivalent:

1. *R C-purifies KB.*
2. *R N-purifies $\langle Compl_\Delta(W), \Delta\rangle$, where $Compl_\Delta(W) = W \cup \{\bigvee \{S \mid S \in MaxScen(KB)\}\}$ is the* completion *of W w.r.t. Δ.*

Clearly enough, replacing W by $Compl_\Delta(W)$ in KB amounts to considering that (D) cannot be questioned; indeed, the deductive closure of $Compl_\Delta(W)$ is equal to the deductive closure of W interpreted under (D).

Which one of both definitions should be used in practice, i.e., whether we are ready to defeat (D) or not, is a nontrivial problem (which is to be considered as well in model-based diagnosis). Stated otherwise, this question becomes: when taking account of an observation for discriminating among extensions (or diagnoses), should we start from the initial set of extensions and filter out those that are incompatible with the observation (C-purification) or expand the certain knowledge with the observation and then recomputing the extensions (N-purification)?

4.2 Computing purifying sets

In the following, we focus on N-purification, only. We are interested in determining sets of tests that enable purifying KB whatever their outcomes. We are specifically interested in the minimal ones w.r.t. $\subseteq$, called purification bases.

Definition 7 (purifying sets of tests, bases)
Given a verification problem $P = \langle KB, T\rangle$, and a set $X \subseteq T$ of tests:

- *X* N-purifies *KB iff for any consistent state $R \in State(X)$, R N-purifies KB. X is said a* purifying set *for KB.*
- *X is a* N-purification base *(or base for short) for KB iff X N-purifies KB and no proper subset of it does it. $Base(KB)$ is the set of all N-purification bases for KB.*
- *KB is* N-purifiable *iff there exists a N-purification base (or, equivalently, if there exists a purifying set) for it.*

In order to figure out the complexity of computing purifying sets, we investigate the complexity of the two following decision problems (we assume the reader familiar with complexity theory, in particular with the polynomial hierarchy (see for instance [15]):

- PURIFYING STATE: does a given state R N-purify KB?
- PURIFYING SET: does a given set X of tests N-purify KB?

Proposition 5 (PURIFYING STATE)
PURIFYING STATE *is in* $\Delta_2^p[\mathcal{O}(n)]$, *and is both* NP*-hard and* coNP*-hard.*

Accordingly, checking whether a given assumption-based theory has a unique extension is not in NP $\cup$ coNP, unless NP = coNP (which is considered very unlikely).

Proposition 6 (PURIFYING SET)
PURIFYING SET *is* Π_2^p*-complete.*

These complexity results show the computation of purifying sets highly intractable. Indeed, checking whether a given set X is a purifying set for KB is already Π_2^p-hard. This is why it is important to identify restrictions for which the computational task can prove easier.

Accordingly, in the following, we assume that the available set T of tests contains only source verification tests, i.e., tests t_i of the form $\{ok(i), forget(i)\}$. Stated otherwise, we assume that performing any set X of tests can only lead to "bad-free" states. This assumption is called `(bad-free)` in the following. A common situation where the `(bad-free)` assumption holds is when the only test considered are *source verification tests.*

Interestingly, purifying sets for KB can be characterized from $Nogood(KB)$ under this assumption:

Proposition 7 (purification under `(bad-free)`)
Let $P = \langle KB, T\rangle$ a verification problem, where T consists of source verification tests, only. For any nogood N of KB, let $Index(N) = \{i \mid \phi_i \in N\}$.

1. *A consistent state R N-purifies KB if and only if for every nogood N of KB, at least one of the following three statements is true:*
   - *(i)* $Index(N) \cap Forget(R) \neq \emptyset$;
   - *(ii)* $Index(N) \setminus Ok(R)$ *is a singleton;*
   - *(iii)* $\exists N' \in Nogood(KB)$ *s.t.* $Index(N') \setminus Ok(R)$ *is a singleton* $\{i\}$ *with* $i \in Index(N)$.
2. *A set X of tests N-purifies KB if and only if for every nogood N of KB, we have* $card(Index(N) \setminus \{i | t_i \in X\}) \leq 1$.

The first point means that R purifies KB if (i) every nogood N is broken by the cancellation of a default $\phi_i \in N$, or if (ii) all defaults of N except one have been validated, which enables to deduce that the remaining one has to be cancelled, or (iii) if it has been possible to deduce by reasoning as in (ii) about another nogood N' that a given default ϕ_i of N has to be cancelled. The second point means that X purifies KB if and only if every nogood contains at most a default whose associated test t_i is not in X.

The above proposition shows that the complexity of both PURIFYING STATE and PURIFYING SET decreases to P when the input is $Nogood(KB)$ and X. As an additional corollary, we have:

Corollary 1 *Let $P = \langle KB, T\rangle$ a verification problem, where T consists of source verification tests, only.*

- *If X and X' are two sets of tests such that $X \subseteq X'$ and X N-purifies KB, then X' N-purifies KB as well.*
- *KB is N-purifiable iff T N-purifies KB.*

Clearly enough, this corollary would not hold in the general case, if the `(bad-free)` assumption were not made (see Example 3 with $X = \emptyset$ and $X' = \{\{ok(1), bad(1)\}\}$ for a counterexample).

Therefore, while the `(bad-free)` assumption does not lower the complexity of PURIFYING SET in the general case, it enables determining whether a KB is N-purifiable in polynomial time whenever the input is $Nogood(KB)$ and T. Moreover, in the case where KB is N-purifiable, a base for KB can be computed in polynomial time from $Nogood(KB)$ and T thanks to the following greedy algorithm: start with $B = T$, for every element t of B, remove t from B if $B \setminus \{t\}$ still N-purifies KB. Once all the tests of T have been processed, the resulting set B is, by construction, a base for KB. This algorithm can also be used to determine in polynomial time whether or not a given set X of tests is a base for KB (just start the algorithm with $B = X$: if the resulting set of tests is X, then X is a base for KB, otherwise it is not). Accordingly, the computation of $Base(KB)$ can be achieved through a simple generate-and-test procedure using the previous greedy algorithm as a routine[5].

Example 4 *Let $P = \langle\langle W, \Delta\rangle, T\rangle$ a verification problem, where $\Delta = \{\phi_1, \ldots, \phi_7\}$, T consists of source verification tests only, and each default ϕ_i ($i \in 1...7$) is testable. Assume that*

$$Nogood(KB) = \{ \quad \{\phi_1, \phi_2, \phi_5\}, \{\phi_2, \phi_6\}, \{\phi_3, \phi_6\}, \{\phi_4, \phi_5, \phi_6\}, \{\phi_7\}\}.$$

[5] The search can be organized using the SE-tree framework given in [19].

We have

$$Base(KB) = \{ \quad \{t_1,t_2,t_4,t_6\},\{t_1,t_3,t_4,t_6\}, \\ \{t_1,t_5,t_6\},\{t_2,t_3,t_4,t_5\},\{t_2,t_5,t_6\}\}.$$

The tractability of PURIFYING SET suggests that the search for *preferred bases*, i.e., bases of minimal cost, could be envisioned under the `(bad-free)` assumption when the input is $Nogood(KB)$ and T. Here, several aggregation policies can be considered in order to define the *cost* $k(X)$ of a set X of tests:

- $k(X) = \sum_{i \in X} k_i$ (SUMCOST policy);
- $k(X) = \max_{i \in X} k_i$ (MAXCOST policy).

The MAXCOST policy is particularly well-suited to the situation where the individual costs k_i represent running times and all tests are performed in parallel (in this case, the overall running time corresponds to the duration of the longest task). The SUMCOST policy must be considered in many of the remaining cases (especially, when the k_i represent monetary costs, or running times when the verification tasks are performed in a pure sequential way).

We have identified the following complexity results:

Proposition 8 (MINIMAL COST BASE)
Let $P = \langle KB, T\rangle$ a verification problem, where T consists of source verification tests, only. Given $Nogood(KB)$, T and an integer α, determining whether there exists a base $X \subseteq T$ for KB such that $k(X) \leq \alpha$ is in P *under the* MAXCOST *policy and is* NP*-complete under the* SUMCOST *policy.*

Let us now show how minimal cost bases for KB can be computed.

Under the MAXCOST policy.

Clearly enough, there exists a base X for KB s.t. $k(X) \leq \alpha$ iff $\{t_i \in T | k_i \leq \alpha\}$ N-purifies KB, which can be tested in polynomial time. Interestingly, it is possible to derive in polynomial time the smallest α such that $\{t_i \in T | k_i \leq \alpha\}$ N-purifies KB. We note k^* this smallest α. To determine the value of k^*, let us consider the mapping `Secondmax` that associates to every finite multi-set of positive integers the second element of the multi-set sorted in decreasing order, and 0 if the multi-set contains less than two elements. For instance, `Secondmax`$(\{4,6,6,3,9,5,5\}) = 6$, `Secondmax`$(\{4,7,1,7\}) = 7$, `Secondmax`$(\{4\}) = 0$. We have:

Proposition 9 (computing k^*)
$k^* = max_{N \in Nogood(KB)}$ `Secondmax`$(\{k_i\}_{\phi_i \in N})$.

As a direct corollary, we have:

Corollary 2 *$\{t_i \in T \mid k_i \leq k^*\}$ is a purifying set for KB that is of minimal cost under the* MAXCOST *policy.*

Such a set is not necessarily a minimal cost base for KB since it is not necessarily minimal w.r.t. $\subseteq$. Nevertheless, applying the greedy algorithm sketched above to this set enables deriving in polynomial time a minimal cost base for KB.

Example 4 (continued) Assuming that each test t_i has cost i, we have $k^* = 5$, hence $\{t_i \in T \mid k_i \leq k^*\} = \{t_1,t_2,t_3,t_4,t_5\}$ is a purifying set for KB of minimal cost, and the minimal cost base $\{t_2,t_3,t_4,t_5\}$ for it can be derived from the latter set in polynomial time.

Under the SUMCOST policy.

Since the decision problem associated to the optimisation problem we are interested in is NP-complete, there is little hope that a (deterministic) polynomial time algorithm for computing a minimal cost base for KB exists, under the SUMCOST policy. Nevertheless, this does not prevent us from designing algorithms that can exhibit a satisfying computational behaviour for many instances. Interestingly enough, as the next proposition shows it, we can take advantage of numerous algorithms for $0-1$ integer programming in order to achieve our goal:

Proposition 10 (0-1 linear programming)
Let $P = \langle KB, T\rangle$ a verification problem, where T consists of source verification tests, only. $X \subseteq T$ is a minimal cost base for KB under the SUMCOST *policy iff the vector $(x_1,...,x_n)$ such that $x_i = 1$ if $t_i \in X$ and $x_i = 0$ otherwise, is a solution of the following* 0-1 *linear program:*

$$min(\sum_{i=1}^{n} k_i x_i)$$

$$s.t. \begin{cases} \forall i = 1...n, x_i \in \{0,1\} \\ \forall N \in Nogood(KB), \sum_{\phi_i \in N} x_i \geq card(N) - 1. \end{cases}$$

Example 4 (continued) The $0-1$ linear program associated to the previous example when every default has a unary cost is:

$$min(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)$$

$$\text{s.t.} \begin{cases} x_1 + x_2 + x_5 \geq 2 \\ x_2 + x_6 \geq 1 \\ x_3 + x_6 \geq 1 \\ x_4 + x_5 + x_6 \geq 2 \\ x_7 \geq 0 \end{cases}$$

The following vectors are optimal solutions: $(1,0,0,0,1,1,0)$ and $(0,1,0,0,1,1,0)$. They cor-

respond to the minimal cost bases $\{t_1, t_5, t_6\}$ et $\{t_2, t_5, t_6\}$ for KB.

4.3 τ-transformation

From the practical side, in order to compute purifying sets of tests by taking advantage of algorithms for assumption-based reasoning, it can prove convenient to normalize the KB under consideration through the following τ-transformation:

Definition 8 (τ-transformation)
Let $KB = \langle W, \Delta \rangle$ an assumption-based theory where $\Delta = \{\phi_1, \ldots, \phi_n\}$. The τ-transformation of KB is the assumption-based theory $\tau(KB) = \langle \tau(W), \tau(\Delta) \rangle$ with $\tau(W) = W \cup \{ok(i) \Rightarrow \phi_i, bad(i) \Rightarrow \neg\phi_i, forget(i) \Leftrightarrow (\neg ok(i) \wedge \neg bad(i)) \mid i \in 1 \,..\, n\}$ where $ok(1), \ldots, ok(n)$, $bad(1), \ldots, bad(n)$, $forget(1), \ldots, forget(n)$ are new propositional symbols, not appearing in KB, and $\tau(\Delta) = \{ok(1), \ldots, ok(n)\}$.

Slightly abusing notations, we note $\tau(\phi_i) = ok(i)$ for every $\phi_i \in \Delta$, and $\tau(S) = \{\tau(\phi) \mid \phi \in S\}$ for every $S \subseteq \Delta$.

Example 1 (continued) $\tau(KB) = \langle \tau(W), \tau(\Delta) \rangle$ with

$\tau(W) =$ $(ok(1) \Rightarrow (a \wedge b)) \wedge (bad(1) \Rightarrow \neg(a \wedge b))$
$\wedge\ (forget(1) \Leftrightarrow (\neg ok(1) \wedge \neg bad(1)))$
$\wedge\ (ok(2) \Rightarrow \neg a) \wedge (bad(2) \Rightarrow a)$
$\wedge\ (forget(2) \Leftrightarrow (\neg ok(2) \wedge \neg bad(2)))$
$\wedge\ (ok(3) \Rightarrow \neg b \wedge c) \wedge (bad(3) \Rightarrow \neg(\neg b \wedge c))$
$\wedge\ (forget(3) \Leftrightarrow (\neg ok(3) \wedge \neg bad(3)))$
$\wedge\ (ok(4) \Rightarrow d) \wedge (bad(4) \Rightarrow \neg d)$
$\wedge\ (forget(4) \Leftrightarrow (\neg ok(4) \wedge \neg bad(4)))$ and

$\tau(\Delta) =$ $\{ok(1), ok(2), ok(3), ok(4)\}$.

The τ-transformation is a linear-time transformation that can be considered as a generalization of the notion of σ-transformation given in [8], better suited to the default revision situation. τ-transforming a KB offers many advantages. First, the set of assumptions of a τ-transformed KB is (by construction) a set of variables. This enables representing nogoods and maximal scenarios as terms. Second, it is just a naming trick, so it does not deeply change the nogoods, maximal scenarios and extensions. To be more precise, we need the notion of V-equivalence of two assumption-based theories.

Definition 9 (V-equivalence of two KBs)
*Let $KB_1 = \langle W_1, \Delta_1 \rangle$ and $KB_2 = \langle W_2, \Delta_2 \rangle$ be two assumption-based theories, and let $V \subseteq PS$. KB_1 and KB_2 are V-*equivalent*, noted $KB_1 \equiv_V KB_2$, iff there exists a bijective mapping M from $Ext(KB_1)$ to $Ext(KB_2)$ such that for every $E \in Ext(KB_1)$, $E \cap PROP_V = M(E) \cap PROP_V$.*

Proposition 11
Let $KB = \langle W, \Delta \rangle$ with $\Delta = \{\phi_1, \ldots, \phi_n\}$. Let $V = PS \setminus \cup_{i=1}^{n} \{ok(i), bad(i), forget(i)\}$. We have $KB \equiv_V \tau(KB)$. To be more precise, we have:

- *S is a nogood of KB iff $\tau(S)$ is a nogood of $\tau(KB)$.*
- *S is a maximal scenario of KB iff $\tau(S)$ is a maximal scenario of $\tau(KB)$.*
- *Let ψ be any formula not containing any variable from $\{ok(i), bad(i), forget(i)\}_{i \in 1..n}$ and let E_S be the extension of KB generated by the maximal scenario S. $\psi \in E_S$ iff $\psi \in E_{\tau(S)}$, where $E_{\tau(S)}$ is the extension of $\tau(KB)$ generated by $\tau(S)$.*

Additionally, while $\tau(KB)$ can have exponentially many nogoods and exponentially many maximal scenarios, there are situations in which computing $Nogood(\tau(KB))$ or $MaxScen(\tau(KB))$ is easy. Indeed, as the next proposition shows it, computing $MaxScen(\tau(KB))$ is tractable when $\tau(W)$ is a DNF formula. Moreover, when $\tau(W)$ is given by its prime implicates, computing $Nogood(\tau(KB))$ is tractable.

Proposition 12
Let $KB = \langle W, \Delta \rangle$ be an assumption-based theory.

- *If $\tau(W) = \psi_1 \vee \ldots \vee \psi_m$ is a DNF formula, then $MaxScen(\tau(KB)) = \{\{ok(i) \in \tau(\Delta) \mid \psi_j \wedge ok(i)$ satisfiable$\} \mid j \in 1..m\}_{\min \supseteq}$.*
- *If $\tau(W) = \psi_1 \wedge \ldots \wedge \psi_p$ is given by the conjunction of its prime implicates, then $Nogood(\tau(KB)) = \{S \subseteq \tau(\Delta) \mid \neg S$ (disjunctively interpreted) is a prime implicate of $\tau(W)\}$.*

Finally, contrariwise to σ-transformed KB, the default revision process for τ-transformed KB can be achieved through a simple expansion of $\tau(W)$.

Proposition 13
Let $KB = \langle W, \Delta \rangle$ with $\Delta = \{\phi_1, \ldots, \phi_n\}$. Let $V = PS \setminus \cup_{i=1}^{n} \{ok(i), bad(i), forget(i)\}$. For any i in $1...n$:

- *$KB * ok(i) \equiv_V \langle \tau(W) \cup \{ok(i)\}, \tau(\Delta) \rangle$.*
- *$KB * bad(i) \equiv_V \langle \tau(W) \cup \{bad(i)\}, \tau(\Delta) \rangle$.*
- *$KB * forget(i) \equiv_V \langle \tau(W) \cup \{forget(i)\}, \tau(\Delta) \rangle$.*

Accordingly, computing $KB * R$ simply consists in expanding W with R. This proves particularly valuable in the situation where W is a DNF formula (resp. the conjunction of its prime implicates). Indeed, in this situation, a DNF (resp. the set of prime implicates) of $W \cup R$ can be easily computed (see, e.g., [10]), hence it is also the case for $MaxScen(KB * R)$ (resp. $Nogood(KB * R)$). As a consequence, determining whether revising KB with R N-purifies KB can be decided tractably in these two restricted cases.

5 Designing default verification tests

In the formal framework developed in this paper, only tests enabling direct default validation, cancellation or invalidation are considered. At a first glance, this restriction may seem unrealistic since, in the general case, there might be actions of verification that do not check the default ϕ_i itself but a relevant formula (maybe weaker than ϕ_i, maybe stronger, maybe neither of both). For instance, Charlie (he's an engineer) wrote a program C that checks whether an integer is divisible by 2, 3, 5 or 7. A positive answer on 348754697 results in a formula which implies (but is not equivalent to) "348754697 is not prime". Moreover, if Charlie wrote as well a program D that tests whether an integer is divisible by an integer strictly greater than 5, then running C and D not only enables us determining whether "348754697 is prime" but gives us as well extra-information which might be relevant for another default of Δ.

In the following, we show how default verification tests can be generated from more general tests.

We propose the following propositional formulation of a test. More complex formulations can be found in the literature [13] [20] [22] [11].

Definition 10 (test)
A test t *is a (finite) set of propositional formulas* $\{\psi_1, \ldots, \psi_m\}$ *where* $m \geq 2$. *Each* ψ_i *represents a possible outcome of the test. To each test* t *is attached a positive integer* k, *the cost of performing* t.

A *test environment* is an ordered pair $\langle W, T\rangle$ where W is a finite set of propositional formulas (facts) and $T = \{t_1, \ldots, t_p\}$ is a finite set of available tests. We make the following assumptions:

- Tests always give a (unique) outcome[6].
- Tests are *reliable*: if the outcome of $t \in T$ is ψ_i, then ψ_i is true in I [7].
- Tests are *non-intrusive*: performing t does not change the actual world I. Subsequently, the order with respect to which tests are performed does not matter: the outcome R of a set $X \subseteq T$ of tests is simply the set of the outcomes of each individual test t of X. $State(X)$ is the set of all possible resulting outcomes of a set X of tests.

[6] This assumption is technical and does not mean that tests do not have preconditions or never fail: simply, when the test is not executable or when it fails, its outcome will be the tautology $\top$.

[7] We can assume without loss of generality that for all $i \in 1 \,..\, m$, $W \cup \{\psi_i\}$ is satisfiable – otherwise, ψ_i will never be reached as an outcome and is thus irrelevant.

Note that we do not require tests to have exclusive outcomes. This enables taking into account tests that may succeed or fail in a graduate way. For instance, $t = \{a, a \vee b, \top\}$ used for determining the truth value of a can either succeed (the outcome is a), fail completely (the outcome is $\top$: no information) or only partially (the outcome $a \vee b$ is relevant to a). The verification actions defined in Section 3 are special cases of tests. Namely, $\{ok(i), bad(i)\}$, $\{ok(i), forget(i)\}$ and $\{ok(i), bad(i), forget(i)\}$ respectively correspond to the tests $\{\phi_i, \neg\phi_i\}$, $\{\phi_i, \top\}$ and $\{\phi_i, \neg\phi_i, \top\}$.

Sets of tests can be characterized according to their ability to help in determining the truth value of a given formula ϕ_i (for our purposes, ϕ_i is a default of Δ):

Definition 11 ((in)validation, discrimination)
Let $\langle W, T\rangle$ *a test environment,* $X \subseteq T$ *a set of tests, and* ϕ_i *a formula from* $PROP_{PS}$:

- X potentially validates ϕ_i *iff at least one of its outcomes* $R \in State(X)$ *is s.t.* $W \cup R \models \phi_i$. X potentially invalidates ϕ_i *iff it potentially validates* $\neg\phi_i$.
- X discriminates ϕ_i *iff for each of its outcomes* $R \in State(X)$, *we have either* $W \cup R \models \phi_i$ *or* $W \cup R \models \neg\phi_i$.

We now briefly show how to generate default verification tests from a set $T = \{t_1, \ldots, t_p\}$ of more "general" tests: if T discriminates ϕ_i (which is the most favorable case), then ϕ_i is associated to the test $\{ok(i), bad(i)\}$; else, if T potentially validates and potentially invalidates ϕ_i, then the associated test is $\{ok(i), bad(i), forget(i)\}$; if T only potentially validates ϕ_i, then the test is $\{ok(i), forget(i)\}$; if T does not potentially validate ϕ_i (even if it potentially invalidates it), then no default verification test is associated to ϕ_i (ϕ_i is not testable)[8].

When tests of T have costs, a default verification test generated from T as described above has an associated cost too. For instance, the cost of a $\{ok(i), bad(i)\}$ default verification test is the cost k of a set $X \subseteq T$ of minimal cost that discriminates ϕ_i (note that k is obtained by aggregating the costs of the individual tests of X). Several aggregation policies and a trade-off between the costs and the discrimination power of tests can be considered.

Example 5 *Let* $\langle W, T\rangle$ *be a test environment s.t.* $W = \{a \vee b, b \Leftrightarrow c, a \wedge b \Rightarrow f, b \vee g, a \vee \neg f, a \vee \neg g\}$

[8] It would not cause any technical difficulty to consider an action $\{bad(i), forget(i)\}$ when T only potentially invalidates ϕ_i; but, as explained in Section 3, the assumption that a default is plausible makes them of little relevance.

and $T = \{t_a, t_b, t_c\}$ *where* $t_a = \{a, \neg a\}$, $t_b = \{b, \neg b\}$ *and* $t_c = \{c, \neg c\}$. *Let* $\phi_1 = f$, $\phi_2 = f \vee g$, $\phi_3 = f \wedge \neg g$.

- T *both potentially validates and potentially invalidates* ϕ_1 *(consider respectively* $R = \{a, b, c\}$ *and* $R' = \{\neg a, b, c\}$*), but does not discriminate* ϕ_1 *(consider* $R = \{a, \neg b, \neg c\}$*).*
- T *discriminates* ϕ_2*. Since* $W \models (\phi_2 \Leftrightarrow a)$, $\{t_a\}$ *discriminates* ϕ_2 *(hence* T *discriminates* ϕ_2 *as well).*
- T *does not potentially validate* ϕ_3 *(but it potentially invalidates it).*

Clearly enough, the complexity of designing default verification tests from more general tests merely depends on the complexity of determining whether T potentially validates ϕ_i (resp. discriminates it). We call the corresponding decision problems VALIDABILITY (resp. DISCRIMINABILITY)[9].

Proposition 14

- VALIDABILITY *is* Π_2^p*-complete.*
- DISCRIMINABILITY *is* coNP*-complete.*

It is particularly interesting to note that discriminability (which is a stronger, and perhaps more useful notion than validability and invalidability) is also computationally easier. Introducing a name for each test outcome (under the form of a new variable) and binarizing tests with more than two possible outcomes enables characterizing the sets of tests we are looking for using prime implicates. The validability (resp. discriminability) issue is strongly related to logic-based abduction (resp. definability), and it is possible to take advantage of the complexity results from [5] (resp. [9]) to derive the results above.

6 Related work and conclusion

This paper is (to our knowledge) the first approach aiming at "debugging" default theories by performing appropriate default verification actions, but is is closely connected (and often complementary) to the following areas: tests in diagnosis, revision of default theories and inference from default theories.

The model-based diagnosis community has been interested for long in test selection strategies for isolating faulty components. Discriminating among several extensions and discriminating among plausible diagnoses are two almost identical tasks: the parallel between the complements of maximal scenarios (often called candidates) and preferred (consistency-based) diagnoses is well-known [18] [6]. A first class of approaches, along the works on GDE [4], consists in selecting the next best test to perform (*one-step lookahead* strategy) so as to maximize the expected gain of information. Much closer to our formalization of tests is the work of McIlraith and Reiter [12] [11] where tests are classified in function of their potentiality to confirm or discard one or several hypotheses; the search for "good" tests is characterized in abductive terms and consequently, test generation can be automated using results and algorithms for the computation of abductive explanations. This line of work (focusing on test generation) can be thought of as upstream whereas ours (rather concerned with what can be done with tests once they are fixed) is downstream.

Williams and Antoniou [23] propose a methodology for revising default theories. They consider four types of revision, namely (a) contraction (removal of a fact), (b) denial (ensuring that a given formula appears in no extension), (c) incorporation of information consistent with W and (d) incorporation of information inconsistent with W. Their revisions lead to change W while ours lead to change both Δ and W. Our validations, consisting of "moving" a formula from Δ to W, are a special case of (c) – no matter whether the default remains in Δ or not. Our invalidations are also a special case of (c) (note that we never question the truth of formulas of W, neither the outcomes of tests, which guarantees that test outcomes are consistent with W). Lastly, our cancellations are a different type of contraction than (a) and they induce a smaller amount of change than fact contraction.

There is a large amount of work on inference from default theories [2], and, relatedly, on belief base revision [14] and on reasoning from inconsistent knowledge bases [3]. There is much to say, first, on the effects of tests outcomes on inference: to be short, default validations strengthen skeptical inference and weaken credulous inference, cancellations weaken credulous inference and have various kinds of effects on skeptical inference (depending on the "role" played by the cancelled default before its cancellation); things are more complex with invalidations. Now, there is also much to say and to do about the *cooperation* between both tasks (inference and discrimination). Inference enables pointing on relevant tests: to give an example, it can be shown that a test $\{ok(i), forget(i)\}$ is useful for discrimination iff the corresponding ϕ_i belongs to at least one extension but not all, i.e., if it is credulously inferred but not skeptically. Other questions of interest are: until when should we discriminate among extensions so as to make the inference informa-

[9] Obviously, the complexity of INVALIDABILITY and of VALIDABILITY coincide.

tive enough and/or sound enough? Which verification actions should we focus on when we are no more interested in the whole purification task but rather in the inference (or not) of a particular formula? Some of our results can be considered as a first step towards these issues.

Acknowledgements

The second author has been partly supported by a "Contrat d'objectifs de la Région Nord/Pas-de-Calais" and by the IUT de Lens.

References

[1] G. Antoniou. On the dynamics of default reasoning. *Proc. of ECSQARU'99*, 1-10.

[2] G. Brewka. Preferred subtheories: an extended logical framework for default reasoning. *Proc. of IJCAI'89*, 1043-1048.

[3] S. Benferhat, C.Cayrol, D. Dubois, J. Lang and H. Prade. Inconsistency management and prioritized syntax-based entailment. *Proc. of IJCAI'93*, 640-645.

[4] J. de Kleer and B. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32, 97-130, 1987.

[5] Th. Eiter and G. Gottlob. The complexity of logic-based abduction. *JACM*, 42(1), 3-42, 1995.

[6] J. de Kleer, A. Mackworth and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56, 197-222, 1992.

[7] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2, 397-425, 1992.

[8] A. Kean. A formal characterization of a domain independent abductive reasoning system. Ph.D. thesis, University of British Columbia, 1992.

[9] J. Lang and P. Marquis. Complexity results for independence and definability in propositional logic. *Proc. of KR'98*, 356-367.

[10] P. Marquis. Consequence finding algorithms. In vol. 5, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, Kluwer Academic, 1999 (to appear).

[11] S. McIlraith. Generating tests using abduction. *Proc. of KR'94*, 449-460.

[12] S. McIlraith and R. Reiter. On tests for hypothetical reasoning. *Readings in model-based diagnosis*, Morgan Kaufman, 89-96, 1992.

[13] R.C. Moore. A formal theory of knowledge and action. *Readings in planning*, Morgan Kaufmann, 480-519, 1990.

[14] B. Nebel. Belief revision and default reasoning: syntax-based approaches. *Proc. of KR'91*, 417-428.

[15] Ch. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[16] D. Poole. A logical framework for default reasoning. *Artificial Intelligence* 36, 27-47, 1987.

[17] D. Poole. Normality and faults in logic-based diagnosis. *Readings in model-based diagnosis*, Morgan Kaufman, 71-77, 1992.

[18] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1), 57-96, 1987.

[19] R. Rymon. Search through systematic set enumeration. *Proc. of KR'92*, 539-550.

[20] R.B. Scherl and H.J. Levesque. The frame problem and knowledge-producing actions. *Proc. of AAAI'93*, 689-695.

[21] J. Stillman. The complexity of propositional default logics. *Proc. of AAAI'92*, 794-799.

[22] B. van Linder, W. van der Hoek and J.-J. Meyer. Tests as epistemic updates. *Proc. of ECAI'94*, 331-335.

[23] M.A. Williams and G. Antoniou. A strategy for revising default theory extensions. *Proc. of KR'98*, 24-33.