The Size of a Revised Knowledge Base*

Marco Cadoli Francesco M. Donini†
Paolo Liberatore Marco Schaerf
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198, Roma, Italy
Email: {cadoli|donini|liberatore|schaerf}@dis.uniroma1.it
WWW: http://www.dis.uniroma1.it/~ai/

Abstract

In this paper we address a specific computational aspect of belief revision: the *size* of the propositional formula obtained by means of the revision of a formula with a new one. In particular, we focus on the size of the smallest formula which is logically equivalent to the revised knowledge base. The main result of this paper is that not all formalizations of belief revision are equal from this point of view. For some of them we show that the revised knowledge base can be represented by a polynomial-size formula (we call these results "compactability" results). On the other hand, for other ones the revised knowledge base does not always admit a polynomial-space representation, unless the polynomial hierarchy collapses at a sufficiently low level ("non-compactability" results). We also show that the time complexity of query answering for the revised knowledge base has definitely an impact on being able to represent the result of the revision compactly. Nevertheless, formalisms with the same complexity may have different compactability properties. We also study compactability properties for a weaker form of equivalence, called *query equivalence,* which allows to introduce new propositional symbols. Moreover, we extend our analysis to the special case in which the new formula has constant size and to the case of sequences of revisions (i.e., *iterated* belief revision). A complete analysis along these four coordinates is shown.

*An extended abstract of this paper appeared as [5].

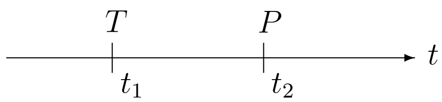
†Current address: Politecnico di Bari, Dipartimento di Elettrotecnica ed Elettronica, Via E. Orabona 4, I-70125 Bari, Italy.

1 Introduction

In many areas of computer science, such as Databases and Artificial Intelligence (AI), we are faced with the problem of constructing and maintaining up-to-date large bodies of information. How can new facts be added to an existing database (or knowledge base)? Intuition might suggest that we simply conjoin the formula P, representing the new information, with our previous information T. However, this would destroy consistency when T and P contradict each other. Moreover, even when T and P are compatible, simply conjoining them might not lead to the desired result, as pointed out by Winslett in [27].

As remarked in [27], the problem of finding general methodologies to update and revise data and knowledge bases has been studied in at least three research communities. In the AI community, the problem of revising a set of beliefs naturally arises when we want to construct an artificial agent that is able to operate in the real world. Since the real world undergoes frequent changes the agent must be able to revise and modify its beliefs accordingly to the new information acquired, without losing an overall consistency. In the Database community problems with updates arose when incompleteness started being introduced in a database via the use of *null* values and the update of views (see, e.g., [2]). Finally, the meaning of belief revision has also been analyzed by philosophers. Recently, Alchourrón, Gärdenfors and Makinson [1, 12] have presented a general framework for belief revision where the basic properties of belief revision are introduced and discussed.

There are two approaches to revision of logical theories, which are known in the literature as *belief revision* and *knowledge base update*. To see intuitively their similarities and differences, we consider the following temporal diagram, representing that 1) our knowledge at time t_1 was T, and 2) at time t_2 we come to know that P is true.



After t_2 there is no doubt we have to assume that P is true, because it represents the most recent observation. As far as T is concerned, things are not so obvious, since $T \wedge P$ might be unsatisfiable. Anyway, a general principle seems to be that we should retain most of the information of T, if possible.

The main assumption in belief revision is that T was (at least partially) wrong at time t_1, because it came from non-valid observations. On the other hand in update we assume that T was true at time t_1, but the world of interest has changed between t_1 and t_2, so T is no longer true after t_2. The following example clarifies the two approaches.

Example. George and Bill share an office near yours, and you wonder whether they are in the office or not. In what follows, propositional letters g and b denote that George is in the office and Bill is in the office, respectively.

Revision. Walking in the corridor, you hear someone talking in the office but you don't recognize his voice. Thus, you suppose that either George or his colleague is in the office, and your knowledge is $T = g \vee b$. Just beyond the corner you see George chatting with someone. This can be formalized as $P = \neg g$, so your conclusion is that the voice you heard was Bill's one (because $T \wedge P \equiv \neg g \wedge b$). A fundamental property of revision is that if $T \wedge P$ is not contradictory then the result of revising T with P is simply $T \wedge P$.

Update. Similarly to the previous case, you are walking in the corridor and hear a voice from the office. Then, George exits the room and you meet him. T and P are exactly the same as in the previous case. Nevertheless, you do not conclude that Bill was in his office, because there is no evidence supporting this conclusion. The only fact you know for sure is P. An interesting property of update is therefore that, even if $T \wedge P$ is consistent, the update of T with P is not necessarily equal to the conjunction of the two formulas. □

More details on the difference between revising a knowledge base and updating it can be found in [19]. Other general references on belief revision and update are [13, 28]. For an account of knowledge update in the context of probabilistic approaches, see [21]. From now on, unless when explicitly stated, we will generically talk about revision of knowledge. In symbols, we denote with $T * P$ the result of the revision of T with P. Both T and P are propositional formulae.

Proposed formalizations of belief revision are very different in spirit, but, following Eiter and Gottlob's [8] presentation, they can be classified according to three orthogonal criteria.

- *(Ir)relevance of syntax.* Some of the approaches perform the revision by adding to T the update formula P and retracting some sub-formula of T, in order to preserve consistency. Such methods (e.g., [10, 15, 23]) are known as *formula-based approaches.* On the other hand, most approaches deal with the models of T, thus not taking into account the syntactic presentation of the knowledge bases. Such methods (e.g., [4, 7, 11, 25, 26, 27]) are known as *model-based approaches.*

- *Measure of closeness.* In model-based approaches the underlying idea is that the models of $T * P$ are the models of P intersected with (a superset of) the models of T. Only the models of T which are *closest* to the models of P are added. Each formalism has its own definition of what "being closer" means. Formula-based approaches are guided by the principle of retracting a minimal set of sub-formulae of T.

- *Revision versus update,* which we explained intuitively above.

Other researchers [8, 23, 27, 22] focused on computational properties of belief revision. As an example, they addressed questions such as: given formulae T, P, Q and a suitable semantics for the revision operator $*$, what is the time complexity of deciding $T * P \models Q$? In which cases polynomial algorithms exist? Same questions have been asked for the problem of deciding $M \models T * P$, where M is an interpretation.

Both aspects of belief revision (semantic and computational) are important from the theoretical as well as the practical point of view.

In this paper we address a new specific computational aspect of belief revision: the *size* of the revised theory $T * P$. An informal description of our work follows. We insist that $T * P$ is represented as a propositional theory, i.e., we want a propositional theory T' such that

$$\{Q \mid T' \models Q\} = \{Q \mid T * P \models Q\}, \tag{1}$$

Q being any formula where only symbols of T or P occur. We call this property *query equivalence*, and we say that a T' satisfying the above criterion is *query equivalent* to $T * P$.

The reason why we insist on $T * P$ being a propositional theory is twofold. From the *epistemological* point of view, it seems reasonable that our set of beliefs does not change the format of its representation after being revised. From the *computational* point of view, it would be nice to split the task of deciding $T * P \models Q$ into two subtasks:

1. compute T' such that (1) holds;

2. decide $T' \models Q$.

There are two positive aspects in such a computational approach: the first subtask can be done off-line, i.e., not necessarily when the query Q arrives. Moreover we could use the same set of algorithms and heuristics both for subtask 2 and for regular query answering.

A question now naturally arises: what is the size of such a T'? If the size of the smallest T' is super-polynomial in the size of T plus the size of P, then the above mentioned approach to query answering is clearly not practical. Moreover –from the cognitive point of view– it is questionable to assume belief revision as the evolutionary model of an agent's mind: An agent would either need an unreasonable amount of storing space, or change the format it uses to represent knowledge. The main result of this paper is that not all formalizations of belief revision are equal from this point of view. For some of them (e.g., Dalal's [7]) we show that T' admits a polynomial-space representation (we call these results "compactability" results). On the other hand we are able to prove that for other ones (e.g., Ginsberg-Fagin-Ullman-Vardi's [15, 10] and Forbus' [11]) T' does not always admit a polynomial-space representation, unless the polynomial hierarchy collapses at a sufficiently low level ("non-compactability" results). The time complexity of answering $T * P \models Q$ on-line (T, P and Q being the input) has definitely an impact on being able to represent T' compactly, although formalisms with the same time complexity may have different compactability properties.

Winslett addresses this problem in [27] for the specific case where the size of P is bounded by a constant, showing several compactability results. We give a complete analysis, proving that some formalisms (e.g., Ginsberg-Fagin-Ullman-Vardi's) are not compactable even in such a restricted case, while other ones (e.g., Forbus') are compactable.

A further aspect we address is the representation of a revised knowledge base using a form of equivalence characterized by the following requirement

$$T' \equiv T * P \tag{2}$$

We call this property *logical equivalence*, and we say that a T' satisfying the above criterion is *logically equivalent* to $T * P$. Notice that a T' satisfying logical equivalence (2) satisfies query equivalence (1) as well, but not the other way around. Basically, query equivalence (1) gives the possibility of introducing new propositional letters, hence it yields formulae T' with less information (e.g., model checking wrt T' gives different results in the two cases). This has definitely an impact on compactability: as an example, Dalal's formalization admits compact representations only according to criterion (1). As we prove in this paper, unless the polynomial hierarchy collapses, Dalal's formalization admits no compact representation w.r.t. criterion (2).

As further information arrives continuously, it may happen that a revised knowledge base needs to be revised once again. In general, we talk about *iterated* belief revision when an unbounded number of revisions occur. As far as the size is concerned, if a formula is compactable after a single revision then it is not guaranteed that it is compactable after several revisions. As an example, we show that Forbus' operator —which admits compact representations for the bounded revision/logical equivalence case— admits polynomially-sized iteratively revised formulae only according to the query equivalence criterion, but not for the logical equivalence criterion (unless the polynomial hierarchy collapses).

Summing up, we provide a complete characterization of the compactability properties for belief revision operators along four coordinates:

1. the formalism;
2. bounded vs. unbounded size of the new formula P;
3. logical equivalence vs. query equivalence;
4. single revision vs. iterated revision.

For what concerns non-compactability results, we use concepts such as Turing machines with advice and non-uniform complexity classes, as well as results relating uniform and non-uniform complexity classes. In fact, our results not only show unlikeliness of *propositional* representations of revised theories, but are valid for a generic data structure, i.e., any structure representing the result of the revision (cf. Section 7 for an exact definition). As for compactability results, we show effective procedures for obtaining compact representations.

The structure of the paper is as follows: Section 2 contains definitions about theories of belief revision and non-uniform complexity classes. Section 3 contains the analysis for the unbounded-size case; we prove compactability as well as non-compactability results. Section 4 contains the analysis for the bounded-size case, while Sections 5 and 6 deal with iterated belief revision, in the unbounded-size and bounded-size case, respectively. The results are summarized in Table 1 and Table 2. Section 7 presents results for generic data structures and Section 8 contains some conclusions.

2 Preliminaries

The *alphabet* of a propositional formula is the set of all propositional letters occurring in it. The special letter $\bot$ denotes falsity, while $\top$ denotes validity. We use $x \not\equiv y$ as a shorthand for $(x \vee y) \wedge (\neg x \vee \neg y)$, and $x \equiv y$ for $(x \wedge y) \vee (\neg x \wedge \neg y)$. Another shorthand we use is $x \rightarrow y$ for $\neg x \vee y$. An *interpretation* of a formula is a truth assignment to the letters of its alphabet. A *theory* T is a finite set of propositional formulae. We denote with $\bigwedge T$ the formula representing the logical "and" of all elements of T. When no confusion arises, we simply write T for $\bigwedge T$. A *model* M of a formula P (theory T) is an interpretation that satisfies P (all formulae in T). This is written $M \models P$ ($M \models T$). Interpretations and models of propositional formulae will be denoted as sets of letters (those which are mapped into *true*). $\bot$ is always mapped into *false* and $\top$ is always mapped into *true*. Given a propositional theory T, we denote with $\mathcal{M}(T)$ the set of its models. The expression $|W|$ denotes the size of W, e.g., the number of distinct occurrences of propositional variables in W, if W is a propositional theory.

Several notational conventions help presenting the revision operators. In particular, the expression $|S|$ denotes the cardinality of a set S, and symmetric difference between two sets S_1, S_2 is denoted by $S_1 \Delta S_2$. Recall that Δ is an associative and commutative operator, with $\emptyset$ as its neutral element.

If S is a set of sets, $\cap S$ denotes the set formed intersecting all sets of S, and analogously $\cup S$ for union; $\min_{\subseteq} S$ denotes the subset of S containing only the minimal (wrt set inclusion) sets in S; $\max_{\subseteq} S$ denotes the subset of S containing only the maximal (wrt set inclusion) sets in S.

Given a propositional theory T we denote with $V(T)$ the letters appearing in T. In the paper we frequently use the notion of substitution of letters in a formula. The notation $P[x/F]$ denotes the formula P where every occurrence of the letter x is replaced by the formula F. This notation is generalized to ordered sets: $P[X/Y]$ denotes the formula P where all occurrences of letters in the set X are simultaneously replaced by the corresponding elements in Y, where X is an ordered set of letters (in general, $X \subseteq V(P)$) and Y is an ordered set of formulae with the same cardinality of X. For example, let Q be the formula $x_1 \wedge (x_2 \vee \neg x_3)$. Let $X = \{x_1, x_3\}$ and $Y = \{y_1, \neg y_3\}$. Then, $Q[X/Y]$ is the formula $y_1 \wedge (x_2 \vee \neg\neg y_3)$.

2.1 Non-uniform complexity classes

As already pointed out, our proofs use the notion of non-uniform computation. We assume the reader is familiar with (uniform) classes of the polynomial hierarchy (P, NP, Σ_2^p, ..., and their complements), and we just briefly introduce non-uniform classes, following Johnson [17].

Definition 2.1 *An* advice-taking Turing machine *is a Turing machine that has associated with it a special "advice oracle" A, which can be any function (not necessarily a recursive one). On input s, a special "advice tape" is automatically loaded with $A(|s|)$ and from then on the computation proceeds as normal, based on the two inputs, x and $A(|s|)$.*

Note that the advice is only function of the *size* of the input, not of the input itself.

Definition 2.2 *An advice-taking Turing machine uses* polynomial advice *if its advice oracle A satisfies $|A(n)| \leq p(n)$ for some fixed polynomial p and all nonnegative integers n.*

Definition 2.3 *If $\mathcal{C}$ is a class of languages defined in terms of resource-bounded Turing machines, then $\mathcal{C}$/poly is the class of languages defined by Turing machines with the same resource bounds but augmented by polynomial advice.*

Any class $\mathcal{C}$/poly is also known as *non-uniform* $\mathcal{C}$, where non-uniformity is due to the presence of the advice. Non-uniform and uniform complexity classes are related in [18, 29]. In particular, Karp and Lipton proved in [18] that if NP $\subseteq$ P/poly then $\Pi_2^p = \Sigma_2^p =$ PH, i.e., the polynomial hierarchy collapses at the second level, while Yap in [29, pg. 292 and Theorem 2] generalized their results showing that if NP $\subseteq$ coNP/poly then $\Pi_3^p = \Sigma_3^p =$ PH, i.e., the polynomial hierarchy collapses at the third level. Such a collapse is considered very unlikely by most researchers in structural complexity.

2.2 Revision operators

We now recall the different approaches to revision and update, classifying them into *formula-based* and *model-based* ones. A more extensive exposition can be found in [8]. The boldface name prefixed to each approach will be used for further reference.

2.2.1 Formula-based approaches

Formula-based revisions operate on the formulae syntactically appearing in the theory T. Let $W(T,P)$ be the set of maximal subsets of T which are consistent with the revising formula P:

$$W(T,P) \quad = \quad \max_{\subseteq}\{T' \subseteq T \;\mid\; T' \cup \{P\} \not\models \bot\}$$

GFUV. In [10] and in [15], the revised theory is defined as a set of theories: $T *_{GFUV} P \doteq \{T' \cup \{P\} \mid T' \in W(T,P)\}$. Each theory of $W(T,P)$ has been called "possible world" by Ginsberg, with no reference to possible worlds in modal logics. Logical consequence in the revised theory is defined as logical consequence in each of the theories, i.e., $T *_{GFUV} P \models Q$ iff for all $T' \in W(T,P)$, $T' \cup \{P\} \models Q$. If a theory T' is also viewed as the conjunction $\wedge T'$ of its formulae, this consequence relation corresponds to consequence from the disjunction of all theories. Hence, as far as logical equivalence is concerned, we consider $T *_{GFUV} P$ as being equivalent to $(\bigvee_{T' \in W(T,P)}(\wedge T')) \wedge P$.

Nebel. The operator $*_N$, proposed in [23], is an extension of Ginsberg-Fagin-Ullman-Vardi's one, in which the new theories are built based on a prioritized partition of the formulae in T.

WIDTIO. Since there may be exponentially many new theories in $T *_{GFUV} P$, a simpler (but somewhat drastic) approach is the so-called WIDTIO (When In Doubt Throw It Out), which is defined as $T *_{Wid} P \doteq (\cap W(T,P)) \cup \{P\}$.

Note that formula-based approaches are sensitive to the syntactic form of the theory. That is, the revision with the same formula P of two logically equivalent theories T_1 and T_2 may yield different results. We illustrate this fact through an example.

Example. Consider $T_1 = \{a, b\}$, $T_2 = \{a, a \rightarrow b\}$ and $P = \neg b$. Clearly, T_1 is logically equivalent to T_2. The only maximal subset of T_1 consistent with P is $\{a\}$, while there are two maximal consistent subsets of T_2, namely $\{a\}$ and $\{a \rightarrow b\}$.

Thus, $T_1 *_{GFUV} P = \{a \wedge \neg b\}$ while $T_2 *_{GFUV} P = \{(a \vee (a \rightarrow b)) \wedge \neg b\} = \{\neg b\}$. The WIDTIO revision gives the same results. $\square$

2.2.2 Model-based approaches

Model-based revisions obey the principle of "irrelevance of syntax" [7]. They operate by selecting the models of P on the basis of some notion of proximity to the models of T. We distinguish between pointwise proximity and global proximity.

We assume both T and P to be satisfiable. As far as compactness is concerned, the cases in which either T or P are unsatisfiable are not of interest. In fact, in the various semantics, the result of revision is either T, or P, or the unsatisfiable theory, or undefined, hence clearly compactable.

Approaches in which proximity between models of P and models of T is computed pointwise wrt each model of T were proposed as suitable for knowledge update [19].

Let M be a model of T; we define $\mu(M,P)$ as the set containing the minimal (wrt set inclusion) symmetric differences Δ between each model of P and the given M; more formally, $\mu(M,P) \doteq \min_{\subseteq}\{M \Delta N \mid N \in \mathcal{M}(P)\}$ (we remind that models are identified with the set of letters they map into *true*).

Winslett. In [27], Winslett defines the models of the updated theory as $\mathcal{M}(T *_{Win} P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : M \Delta N \in \mu(M,P)\}$.

Borgida. Borgida's operator $*_B$ is the same as Winslett's, except for the case when P is consistent with T, in which case Borgida's revised theory is just $T \cup \{P\}$.

Forbus. This approach [11] takes into account cardinality: let $k_{M,P}$ be the minimum cardinality of sets in $\mu(M,P)$. The models of Forbus' updated theory are $\mathcal{M}(T *_F P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : |M \Delta N| = k_{M,P}\}$. Note that by means of cardinality, Forbus can compare (and discard) models which are incomparable in Winslett's approach.

We now recall approaches where proximity between models of P and models of T is defined considering globally *all* models of T. Let $\delta(T,P) \doteq \min_{\subseteq} \bigcup_{M \in \mathcal{M}(T)} \mu(M,P)$.

Satoh. In [25], the models of a revised theory are defined as $\mathcal{M}(T *_S P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : N \Delta M \in \delta(T,P)\}$.

Dalal. This approach is similar to Forbus', but global. Let $k_{T,P}$ be the minimum *cardinality* of sets in $\delta(T,P)$; in [7], Dalal defines the models of a revised theory as $\mathcal{M}(T *_D P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : |N \Delta M| = k_{T,P}\}$.

Weber. Let $\Omega \doteq \cup\delta(T,P)$, i.e., Ω contains every letter appearing in at least one minimal difference between a model of T and a model of P. Following [8, p. 238], the models of Weber's revised theory [26] are defined as $\mathcal{M}(T *_{Web} P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : N \Delta M \subseteq \Omega\}$.

From the definitions it follows that the set of models relative to the model-based approaches are related as described in Figure 1, where each arrow denotes set containment.

To illustrate the differences between different model-based approaches, consider the following example.

Example. Let T and P be two formulae on the alphabet $\{a,b,c,d\}$ defined as:

$$\begin{aligned} T &= a \wedge b \wedge c \\ P &= (\neg a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge b \wedge (a \not\equiv d)) \end{aligned}$$

T has only two models, which are:

$$\begin{aligned} M_1 &= \{a,b,c,d\} \\ M_2 &= \{a,b,c\} \end{aligned}$$

while P has four models:

$$N_1 = \{a,b\}$$

$$\begin{tikzcd}
 & \mathcal{M}(T *_{Win} P) & \mathcal{M}(T *_B P) & \mathcal{M}(T *_{Web} P) \\
\mathcal{M}(T *_F P) \arrow[ur] & & \mathcal{M}(T *_S P) \arrow[ul] \arrow[u] \arrow[ur] & \\
 & \mathcal{M}(T *_D P) \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Figure 1: Containment between sets of models for the various operators.

$$
\begin{aligned}
N_2 &= \{c\} \\
N_3 &= \{b, d\} \\
N_4 &= \emptyset
\end{aligned}
$$

The symmetric differences between each model of T and each model of P are:

Δ	$N_1 = \{a, b\}$	$N_2 = \{c\}$	$N_3 = \{b, d\}$	$N_4 = \emptyset$
$M_1 = \{a, b, c, d\}$	$\{c, d\}$	$\{a, b, d\}$	$\{a, c\}$	$\{a, b, c, d\}$
$M_2 = \{a, b, c\}$	$\{c\}$	$\{a, b\}$	$\{a, c, d\}$	$\{a, b, c\}$

Hence, the minimal differences between M_1 and models of P are $\mu(M_1, P) = \{\{c, d\}, \{a, b, d\}, \{a, c\}\}$; the minimal differences between M_2 and models of P are $\mu(M_2, P) = \{\{c\}, \{a, b\}\}$.

The cardinalities of set differences between each model of T and each model of P are:

$\lvert\Delta\rvert$	$N_1 = \{a, b\}$	$N_2 = \{c\}$	$N_3 = \{b, d\}$	$N_4 = \emptyset$
$M_1 = \{a, b, c, d\}$	2	3	2	4
$M_2 = \{a, b, c\}$	1	2	3	3

Winslett. The minimal differences in $\mu(M_1, P)$ correspond to the models N_1, N_2, N_3 of P, while those in $\mu(M_2, P)$ correspond to the models N_1, N_2 of P. Therefore, the models of $T *_{Win} P$ are N_1, N_2, N_3. The same result holds for Borgida's revision, since T and P are inconsistent.

Forbus. From the table with cardinalities: the minimum cardinality of differences between M_1 and each model of P is $k_{M_1,P} = 2$, corresponding to models N_1 and N_3; while $k_{M_2,P} = 1$, corresponding to N_1. Therefore, $T *_F P$ has models N_1 and N_3.

We now turn to global proximity approaches, where also entries in different rows of the above tables are compared for minimality.

Satoh. The minimal differences between any model of T and any model of P are $\delta(T, P) = \{\{c\}, \{a, b\}\}$. These minimal differences correspond to models N_1 and N_2 of P, which therefore are the models of $T *_S P$.

Dalal. The minimum cardinality of all set differences is $k_{T,P} = 1$, corresponding to N_1. As a result, $T *_D P$ selects only the model N_1.

Weber. Consider the union of all minimal global differences, that is $\Omega = \cup\delta(T, P) = \{a, b, c\}$. In Weber's revision, one selects the models of P for which there exists a model of T whose symmetric difference is included in Ω. Since all models of P have this property (cf. table with symmetric differences), they are all selected. Thus, the revision coincides with P in this case. □

There is an important property of model-based revision operators that we frequently use:

Proposition 2.1 *Let M be a model of T and $*$ one of the revision operators $\{*_B, *_D, *_F, *_S, *_{Web}, *_{Win}\}$. Then, there exists a model N of $T * P$ such that $M \Delta N \subseteq V(P)$.*

This property states that for every model of T there exists a model of $T * P$ whose distance is bounded by the letters of P. This is sometimes crucial in showing the existence of compact representations since it allows to focus our attention only on the letters of P. This proposition is mentioned by Eiter and Gottlob in [8, proof of Lemma 6.1].

2.2.3 Iterated Revision

All the operators proposed in the literature aim at formalizing the process of revising a formula with a single update. Nevertheless, it is more realistic to formalize the result of applying a *series* of revisions or updates to an existing knowledge base. The iterated application of an update or revision operator is defined as follows: given a theory T and a sequence of updates $P^1, \ldots, P^m$, the m-times repeated application of $*$ is $(\cdots(T * P^1) \cdots * P^m)$. That is, the result of revising T with P^1 is revised with P^2 and so on. In order to simplify the notation, we assume that the revision operator $*$ is *left associative*. Thus, the result of revising the theory T with a series of updates $P^1, \ldots, P^m$ will be denoted as $T * P^1 * \cdots * P^m$.

There are at least two possible computational approaches to deal with iterated revision. The first approach incorporates each revision one-by-one into the knowledge base, while the second approach stores the initial knowledge base and the sequence of update formulae and computes the result in a single step. As we show in Sections 5 and 6, the second approach leads to compact representations in a larger set of cases.

2.2.4 Complexity of Revision

As shown in what follows, the existence of compact representations for the revised knowledge base is related to the complexity of model checking and inference for the various revision operators. The complexity of these decision problems have been analyzed in various papers. The complexity of deciding $T * P \models Q$ (T, P and Q being the input) was studied in [8]. The results are that in Dalal's approach the problem is $\Delta_2^p[\log n]$-complete, while in all other approaches it is Π_2^p-hard (in some cases, Π_2^p-complete).

The complexity of model checking, that is, the complexity of deciding whether $M \models T * P$ (where M is an interpretation and M, T and P are the input) is analyzed in [22]. Noticeably, formalisms with the same complexity w.r.t. inference have different complexity w.r.t. model checking (e.g., Satoh's and Ginsberg-Fagin-Ullman-Vardi's operators).

The complexity of inference for iterated revision was analyzed by Eiter and Gottlob [9].

2.3 Compact Representations

In this paper we investigate the possibility of compactly representing, with a propositional formula, the result of an update or revision. We point out that the result of all revision and update methods presented in Section 2.2 can be represented as a propositional formula. However, it is difficult to know what is the size of the shortest formula representing the result.

We call an operator *query-compactable* if there exists a propositional formula of polynomial size which has the same theorems of the result of the revision. More precisely:

Definition 2.4 (query-compactable operator) *An update or revision operator $*$ is* query-compactable *if and only if there exists a polynomial p such that for any pair of propositional formulae T and P there exists a propositional formula T' with the following properties:*

1. $|T'| \leq p(|T| + |P|)$;
2. *T' is query equivalent (cf. (1)) to $T * P$.*

In order to prove that some belief revision operators are not query-compactable we show that this would imply some very unlikely consequences on complexity classes. In fact, as we show in Section 7, reference to to complexity classes is necessary, because in some cases proving logical-compactability is equivalent to prove long-standing open questions in complexity theory. We now prove a general property that will be helpful in the proofs of the following sections.

A decision problem, seen as the infinite set of its instances, can be partitioned according to the size of the instances for a reasonable encoding (in the sense of [17]) of the instances. In particular, we focus on the NP-complete problem 3-SAT [14, Problem LO2].

Definition 2.5 *We partition the set of instances as $3\text{-SAT} = \bigcup_{n=1}^{\infty} 3\text{-SAT}_n$, according to their size n. For each n, we know that the number of distinct propositional letters occurring in each instance of 3-SAT_n is at most n. Without loss of generality, we assume that all formulae of 3-SAT_n are built on the same set of atoms $B_n = \{b_1, \ldots, b_n\}$. For each n, π_n^{max} is defined as the set of all the three-literal clauses on $\{b_1, \ldots, b_n\}$, and m_n^{max} is the number of clauses in π_n^{max}.*

Note that π_n^{max} has size polynomial in n; in particular it has $\Theta(n^3)$ clauses, and each literal can be represented with $\Theta(\log n)$ bits. Moreover, each instance $\pi \in 3\text{-SAT}_n$, considered as a set of clauses, is a subset of $\pi_n^{max} \in$ 3-SAT. In Theorems 2.2 and 2.3 of this section, and in Theorems 3.1, 3.3, 3.6, 6.5, and 6.5 of the following ones, we refer to such a partition of 3-SAT, and the related notation.

Theorem 2.2 *Let $*$ be a revision operator. Assume there exists a polynomial p such that, for each $n > 0$, there exists a pair of formulae T_n, P_n with the following properties:*

1. $|T_n| + |P_n| \leq p(n)$;
2. *for all $\pi \in$ 3-SAT$_n$, there exists a formula Q_π such that:*
   (a) *Q_π can be computed from π in polynomial time;*
   (b) *$T_n * P_n \models Q_\pi$ iff π is satisfiable.*

With the above hypothesis, if $*$ is query-compactable, then NP $\subseteq$ coNP/poly.

Proof. Assume there exists a polynomial p such that, for each n, there exists a pair of formulae T_n, P_n with the properties stated. Now, assume $*$ is query-compactable. Hence, for each pair T_n, P_n there exists a T'_n with the properties stated in Definition 2.4. Recall that the problem of deciding whether $T'_n \models Q_\pi$ (the input being T'_n, Q_π) is in coNP. Then we can define an advice-taking non-deterministic Turing machine in the following way.

First, we define the advice oracle as $A(n) = T'_n$. Observe that $|A(n)| = |T'_n| \leq p_1(|T_n| + |P_n|) \leq p_1(p(n))$, where p is the polynomial mentioned in the hypothesis 1 of the theorem, and p_1 is the polynomial mentioned in point 1 of Definition 2.4. Hence, the size of the advice is bounded by a polynomial which is the composition of p_1 and p.

Secondly, the machine operates as specified by the following pseudo-code:

(1) let π be an instance of 3-SAT;
(2) $n := |\pi|$;
(3) load $A(n)$; (* $= T'_n$ *)
(4) use π to compute Q_π; (* in time polynomial in $|\pi|$ *)
(5) if $T'_n \models Q_\pi$
    then return "true"
    else return "false";

By Definition 2.4, "true" is returned if and only if $T_n * P_n \models Q_\pi$. Since $*$ is query-compactable, the problem of checking $T'_n \models Q_\pi$ is in coNP (the input being T'_n and Q_π). Therefore, the problem "is π unsatisfiable?" can be solved by an advice-taking non-deterministic polynomial-time machine whose input has size polynomial in $|\pi|$. Since 3-SAT is an NP-complete problem, and co(NP/poly) = coNP/poly, this would imply NP $\subseteq$ coNP/poly. □

Compactability w.r.t. logical equivalence is a property of belief revision operators stronger than query-compactability.

Definition 2.6 (logically-compactable operator) *An update or revision operator* $*$ *is* logically-compactable *if and only if there exists a polynomial* p *such that for any pair of propositional formulae* T *and* P *there exists a propositional formula* T' *with the following properties:*

1. $|T'| \leq p(|T| + |P|)$;
2. T' *is logically equivalent (cf. (2)) to* $T * P$.

Note that if a belief revision operator is logically-compactable, then it is also query-compactable. Similarly to query-compactability, in order to prove that some belief revision operator is not logically compactable we show that this would imply some very unlikely consequences on complexity classes. Moreover, as we show in Section 7 in some cases proving query-compactability is equivalent to prove long-standing open questions in complexity theory. We show a general property analogous to Theorem 2.2 that will be helpful in the proofs of the next sections.

Theorem 2.3 *Let* $*$, p, T_n, *and* P_n *be as in Theorem 2.2, and let the hypothesis of Theorem 2.2 hold. If for each* $n > 0$,

2'. for all $\pi \in$ 3-SAT$_n$, *there exists an interpretation* M_π *of* $T_n * P_n$ *such that:*

  (a) M_π *can be computed from* π *in polynomial time;*

  (b) $M_\pi \models T_n * P_n$ *iff* π *is satisfiable.*

then, if $*$ *is logically-compactable, then NP* $\subseteq$ *P/poly.*

Proof. The proof has a structure similar to the proof of Theorem 2.2. With the hypothesis stated, and assuming that $*$ is logically-compactable, for each pair T_n, P_n we can define a T'_n with the properties stated in Definition 2.6. Recall that checking whether $M \models T'_n$ (the input being M, T'_n) is a polynomial-time problem. The advice-taking Turing machine is defined similarly to the previous theorem.

In particular, the advice oracle is again defined as $A(n) = T'_n$, and its size is less than or equal to $p'_1(|T_n| + |P_n|) \leq p'_1(p(n))$, where p'_1 is the polynomial mentioned in point 1 of Definition 2.6.

The machine operates in a similar way, in particular, the following pseudo-code substitutes the corresponding lines:

(4) use π to compute M_π; (* in time polynomial in $|\pi|$ *)
(5) if $M_\pi \models T'_n$
    then return "true"
    else return "false";

By Definition 2.6, "true" is returned if and only if $M_\pi \models T_n * P_n$. Since $*$ is logically-compactable, the last decision can be made in time less than or equal to $O(p_1(|T_n| + |P_n|) + |M_\pi|)$. Therefore, the advice-taking Turing machine would globally work in time polynomial in $|\pi|$. Since 3-SAT is an NP-complete problem, this would imply NP $\subseteq$ P/poly. $\square$

Theorems 2.2 and 2.3 provide a general schema for the proofs of non-existence of compact representations that we present in the paper. All our proofs use the same technique, even though with major differences in the details.

We note that usage in AI of non-uniform complexity classes for proving unlikeliness of existence of compact representations has been proposed for the first time by Kautz and Selman in [20]. In particular, they showed that existence of a polynomial-size representation of the *Horn upper bound* of a propositional formula implies that NP $\subseteq$ P/poly.

Horn upper bounds have been studied in the context of *approximate knowledge compilation*, an approach to intractability of reasoning in AI that tries to move off-line a significant part of the computational burden of reasoning, at the cost of losing either soundness or completeness. In [16], Gogic, Papadimitriou, and Sideri have considered approximate knowledge compilation of the revision of a formula, showing that in some cases such an approximation can be computed in linear time. In this paper we do not deal with approximation of formulae, but only with representations preserving equivalence.

3 Single unbounded revision

The purpose of this section is to show an analysis of logical- and query-compactability of revision operators. We consider revision operators mentioned in Section 2, and show both compactability and non-compactability results. There is no assumption on the size of the formula P; the bounded-size case will be addressed in the next section. Moreover, we consider only a single revision. Notice that WIDTIO semantics is logically compactable (hence query compactable) since it immediately follows from its definition that the size of $T *_{Wid} P$ is always less than or equal to $|T| + |P|$.

3.1 Query compactability

In this subsection we investigate query compactability. We begin our investigation focusing on Ginsberg-Fagin-Ullman-Vardi's operator. Other researchers have already noticed that the explicit representation of the result of revising a knowledge base under Ginsberg-Fagin-Ullman-Vardi's semantics might have exponential size.

We introduce this problem with an example presented by Nebel [24]. Let

$$T_1 = \{x_1, \ldots, x_m, y_1, \ldots, y_m\}$$
$$P_1 = \bigwedge_{i=1}^m (x_i \not\equiv y_i)$$

The set $W(T_1, P_1)$ contains 2^m distinct theories, each one containing, for each $i (1 \leq i \leq m)$, exactly one of x_i and y_i. If we represent $T_1 *_{GFUV} P_1$ as the disjunction of all theories in $W(T_1, P_1)$, conjoined with P_1, the size of this representation is exponential in $|T_1| + |P_1|$.

The problem of the explosion of the size of the revised knowledge base was also pointed out by Winslett [27] with another example:

$$\begin{array}{rcllr}
T_2 & = & \{\ x_1, & y_1, & z_1 \equiv (\neg x_1 \vee \neg y_1), \\
 & & \ \ \vdots & \vdots & \vdots \\
 & & \ \ x_i, & y_i, & z_i \equiv (z_{i-1} \wedge (\neg x_i \vee \neg y_i)), \\
 & & \ \ \vdots & \vdots & \vdots \\
 & & \ \ x_m, & y_m, & z_m \equiv (z_{m-1} \wedge (\neg x_m \vee \neg y_m))\ \} \\
P_2 & = & \ \ z_m & &
\end{array}$$

Again, the cardinality of the set $W(T_2, P_2)$ is exponential in m (note that in this example the size of P_2 does not depend on m). These two examples show that, in general, given T and P, the "obvious" representation of $T *_{GFUV} P$ might have size exponential in $|T| + |P|$. However, they do not rule out the existence of a different representation of polynomial size. As Winslett notes [27, pg. 34] the exponential increase in the size is proven if we "assume a completely naive storage organization, where the theories are written out as we would write them down on paper". Later on she also conjectures that "these bounds hold even for clever storage schemes".

We now show that her conjecture on Ginsberg-Fagin-Ullman-Vardi's revision is indeed true for the more general case of query equivalence. This result is later generalized to other revision operators. We recall that the symbol B_n denotes the set of letters $\{b_1, \ldots, b_n\}$ used to build all formulae in 3-SAT$_n$.

Theorem 3.1 *Unless NP $\subseteq$ coNP/poly, the revision operator $*_{GFUV}$ is not query-compactable.*

Proof. We apply the general schema of Theorem 2.2 and the notation of Definition 2.5. We show that, for any integer n, there exists a formula P_n and a set of atomic facts T_n, both depending only on n, of polynomial size w.r.t. n, with the following property: given any $\pi \in$ 3-SAT$_n$, there exists a query Q_π such that π is satisfiable if and only if $T_n *_{GFUV} P_n \models Q_\pi$.

Let L be the alphabet $B_n \cup C \cup D \cup \{r\}$, where r is a new distinct atom, while C and D are sets of new atoms one-to-one with the elements of π_n^{max}. In other words, $C = \{c_i \mid \gamma_i \in \pi_n^{max}\}$ and $D = \{d_i \mid \gamma_i \in \pi_n^{max}\}$.

We define T_n and P_n on the alphabet L according to the following rules:

$$T_n = C \cup D \cup B_n \cup \{r\} \tag{3}$$

$$P_n = \left[\left(\bigwedge_{i=1}^{n} \{\neg b_i \mid b_i \in B_n\} \wedge \neg r\right) \vee \bigwedge_{j=1}^{m_n^{max}} (c_j \to \gamma_j) \right] \wedge \bigwedge_{j=1}^{m_n^{max}} (c_j \not\equiv d_j) \tag{4}$$

We define W_π and Q_π as follows:

$$\begin{aligned} W_\pi &= \{c_i \mid \gamma_i \in \pi\} \cup \{d_i \mid \gamma_i \notin \pi\} \\ Q_\pi &= W_\pi \to r \end{aligned}$$

The interpretation on $C \cup D$ which is the unique model of W_π will be denoted as I_π.

The following remarks on $W_\pi \cup P_n$ are in order:

- No c_i or d_i can be added consistently to $W_\pi \cup P_n$, because P_n imposes that $c_i \not\equiv d_i$.
- The theory $W_\pi \cup P_n$ is consistent, because the interpretation I'_π that extends I_π by making false all atoms in $B \cup \{r\}$ satisfies both W_π and P_n.

Now we are ready to prove our claim, i.e., that π is satisfiable if and only if $T_n *_{GFUV} P_n \models Q_\pi$.

If: suppose π unsatisfiable. We show that $W_\pi \cup P_n$ does not entail Q_π and that W_π is a world from $W(T_n, P_n)$, i.e., that 1) $W_\pi \cup P_n \not\models Q_\pi$, and 2) $W_\pi \in W(T_n, P_n)$.

1. Since π is unsatisfiable, for each interpretation H of the atoms of B_n, there exists a $\gamma \in \pi$ such that $H \not\models \gamma$. Let $J = I_\pi$ be an interpretation on L, i.e., all variables of $B_n \cup \{r\}$ are set to false. We have that $J \not\models \bigwedge_{j=1}^{m_n^{max}} (c_j \to \gamma_j)$. Note that J is a model of P_n, because it satisfies $\bigwedge_{i=1}^{n} \{\neg b_i \mid b_i \in B_n\} \wedge \neg r$ and $\bigwedge_{j=1}^{m_n^{max}} (c_j \not\equiv d_j)$. Moreover, J does not satisfy Q_π, because it satisfies W_π and $\neg r$. As a consequence, $W_\pi \cup P_n \not\models Q_\pi$.
2. Adding to W_π any other element of T_n would make W_π inconsistent with P_n, therefore, $W_\pi \in W(T_n, P_n)$.

Only-If: Suppose π satisfiable. We show that all worlds in $W(T_n, P_n)$ imply Q_π. Note that all worlds not including W_π imply Q_π, hence we concentrate on those containing W_π.

Note that every $W \in W(T_n, P_n)$ such that $W_\pi \subseteq W$ must contain r since $W_\pi \cup \{\bigwedge_{j=1}^{m_n^{max}} (c_i \to \gamma_j), \bigwedge_{j=1}^{m_n^{max}} (c_j \not\equiv d_j)\}$ is satisfiable. Moreover, $W \backslash W_\pi \subseteq B \cup \{r\}$.

Since T_n is a set of literals, each such world W is uniquely characterized by its intersection with B_n. For each interpretation H of the atoms in B_n, let $W_{H,\pi}$ be defined as follows:

$$W_{H,\pi} = W_\pi \cup \{x_i \mid x_i \in H\} \cup \{r\}$$

If H does not satisfy π then $W_{H,\pi} \wedge P_n$ is inconsistent because $W_{H,\pi} \not\models \neg r$ and $W_{H,\pi} \not\models \bigwedge_{j=1}^{m_n^{max}} (c_j \to \gamma_j)$, therefore, $W_{H,\pi} \notin W(T_n, P_n)$. On the other hand, if H satisfies π, the interpretation $J = H \cup I_\pi \cup \{r\}$ of L satisfies $W_{H,\pi}$, P_n and Q_π. Since there exists at least one interpretation H satisfying π, the corresponding $W_{H,\pi}$ is in $W(T_n, P_n)$, and therefore $W_\pi \notin W(T_n, P_n)$. As a consequence, for all elements W of $W(T_n, P_n)$ we have that $W \cup P_n \models Q_\pi$. □

We now consider the model-based revision operators. Even though their semantics is very different from Ginsberg-Fagin-Ullman-Vardi's one, we obtain a similar result for most of them:

Theorem 3.2 *Unless NP ⊆ coNP/poly, the revision operators $*_B, *_S$ and $*_{Win}$ are not query-compactable.*

Proof. Eiter and Gottlob's result [8, Lemma 6.1, point (2)] implies that $T *_{GFUV} P \models Q$ iff $T *_B P \models Q$ iff $T *_S P \models Q$ iff $T *_{Win} P \models Q$ when T is a maximal consistent set of literals, i.e., it has exactly one model, and $V(P) \subseteq V(T)$. Note that these conditions hold for the T_n defined in (3) and P_n defined in (4). □

Using a proof similar to the one of Theorem 3.1, we can show that the same result also holds for Forbus' revision operator.

Theorem 3.3 *Unless NP ⊆ coNP/poly, the revision operator $*_F$ is not query-compactable.*

Proof. We adopt the notation of Definition 2.5. We show that, for any integer n, there exist two formulae P_n and T_n, both depending only on n, of polynomial size w.r.t. n, with the following property: given any $\pi \in$ 3-SAT$_n$, there exists a query Q_π such that π is satisfiable if and only if $T_n *_F P_n \models Q_\pi$.

The proof is similar to that of Theorem 3.1, but we have now to consider closeness between models because the revision operator $*_F$ is model-based. We still use a set C of atoms which serve as "enabling guards" to select, among the clauses of π_n^{max}, those belonging to π. For each clause of π_n^{max} we now use $n+2$ guards, which are always forced to have the same truth value. T_n is defined in such a way that:

- for every two different models of π the corresponding models of T_n have distance at most $n+1$,

- given two different instances π_1, π_2 of 3-$\textsc{sat}_n$ a model in T_n corresponding to a model of π_1 has distance at least $n+2$ from a model of T_n corresponding to a model of π_2.

For any given n, let $C = \{c_i^j \mid 1 \le i \le n+2, 1 \le j \le m_n^{max}\}$ be an $(n+2) \times m_n^{max}$ matrix of atoms (recall that m_n^{max} is the number of clauses in π_n^{max}). Let L be the alphabet $B_n \cup C \cup \{r\}$, where r is a new distinct atom.

Let

$$U_n = \bigwedge_{j=1}^{m_n^{max}} \bigwedge_{i=2}^{n+2} (c_i^j \equiv c_1^j)$$

This formula forces all rows of the matrix C to be equal. Let T_n and P_n be defined as follows:

$$\begin{aligned} T_n &= \{U_n\} \cup B_n \cup \{r\} \\ P_n &= \left[(\bigwedge_{i=1}^{n} \{\neg b_i \mid b_i \in B_n\} \wedge \neg r) \vee \bigwedge_{j=1}^{m_n^{max}} (c_1^j \rightarrow \gamma_j) \right] \wedge U_n \end{aligned}$$

Given $\pi \in$ 3-$\textsc{sat}_n$, we define:

$$Q_\pi = \bigvee_{i=1}^{n+2} \bigvee_{\gamma_j \notin \pi} c_i^j \vee \bigvee_{i=1}^{n+2} \bigvee_{\gamma_j \in \pi} \neg c_i^j \vee \bigvee_{i=1}^{n} b_i \vee r$$

We show that $T_n *_F P_n \models Q_\pi$ if and only if π is satisfiable. Note that every interpretation of L, but $M_\pi = \bigcup_{i=1}^{n+2} \{c_i^j \mid \gamma_j \in \pi\}$, satisfies Q_π. Thus, we only need to show that $M_\pi \models T_n *_F P_n$ if and only if π is unsatisfiable.

First of all, note that for each $\pi \subseteq \pi_n^{max}$ the interpretation I_π defined as $\bigcup_{i=1}^{n+2} \{c_i^j \mid \gamma_j \in \pi\} \cup B_n \cup \{r\}$ is a model of T_n.

The distance between M_π and I_π is exactly $n+1$, while the distance from I_π to any other $N \in \mathcal{M}(P_n)$, with a different valuation of atoms in C, is at least $n+2$. Hence, the models of P_n that are closest to I_π must have the same valuation of atoms in C.

Suppose π unsatisfiable. The model of P_n closest to I_π is now M_π, since there is no other model of P_n with the same valuation of C. Hence $M_\pi \models T_n *_F P_n$.

On the converse, suppose π satisfiable. Let ϕ be the interpretation of the atoms in B_n satisfying π. The model $N_\pi = I_\pi - \{b \mid b \notin \phi\}$ is now in $\mathcal{M}(P_n)$, and its distance from I_π is at most n letters. Hence, M_π is not a model of $T_n *_F P_n$. □

We now turn our attention to operators that are query-compactable. In particular, we focus on Dalal's and Weber's operators (WIDTIO has already been discussed).

Let $X = \{x_1, \ldots, x_n\}$ be the alphabet of the initial knowledge base T and the revising formula P, and $Y = \{y_1, \ldots, y_n\}$ be another set of (distinct) letters. In order to show that $*_D$ is query-compactable, we make use of a propositional formula $EXA(k, X, Y, W)$, of size polynomial in n, containing letters of X and Y, and possibly other letters W, which is true iff the Hamming distance between the values assigned to X and Y is exactly k. We first show that such a polynomial size formula exists, and then (cf. Theorem 3.4 below) that the revised theory $T *_D P$ can be represented as $T[X/Y] \wedge P \wedge EXA(k, X, Y, W)$, where k is the minimum distance between the models of P and T, denoted as $k_{T,P}$ in the definition of Dalal's revision.

First of all, deciding whether the Hamming distance between two models is exactly k requires time linear in the number of the atoms. Secondly, it has been proved (cf. e.g., [3, Theorem 2.1]) that if a problem can be solved in time $O(f(m))$, m being the size of the input, then there exists a circuit determining the solution of such a problem using $O(f(m) \cdot \log f(m))$ gates. In our case, the input is composed of $\log n + 2n$ bits representing k and truth assignments to atoms in $X \cup Y$. Therefore there exists a circuit of size $O(n \cdot \log n)$ determining whether the Hamming distance between two models is exactly k. Such a circuit can be represented, with routine methods, as a polynomial size propositional formula using literals from $X \cup Y$, $\log n$ literals representing k, and a polynomial number of new atoms W representing the internal nodes of the circuit. $EXA(k, X, Y, W)$ is such a formula. An explicit representation of $EXA(k, X, Y, W)$ is shown in [5].

Theorem 3.4 *Let X be the alphabet of T and P and let k be the the minimum distance between models of T and models of P, i.e., $k = k_{T,P}$. Then $T[X/Y] \wedge P \wedge EXA(k, X, Y, W)$ is query equivalent (1) to $T *_D P$.*

Proof. Let Q be a query on the alphabet X and k be the minimal distance between the models of P and T. We prove the theorem by showing that $T[X/Y] \wedge P \wedge EXA(k, X, Y, W) \models Q$ iff $T *_D P \models Q$.

If. We show that $T[X/Y] \wedge P \wedge EXA(k, X, Y, W) \not\models Q$ implies $T *_D P \not\models Q$. Let M be a model of $T[X/Y] \wedge P \wedge EXA(k, X, Y, W)$ such that $M \not\models Q$. We have:

- $M \cap X$ satisfies P;
- since M satisfies $EXA(k, X, Y, W)$, $M \cap Y$ satisfies $T[X/Y]$; and
- $M \cap X$ has distance k from $(M \cap Y)[X/Y]$ (this is the model obtained from M by intersecting it with Y and then replacing any y_i in the resulting set with the corresponding x_i).

Since $M \models P$ and k is by definition the minimal distance between models of T and P, $M \cap X$ is also a model of $T *_D P$. Therefore, $T *_D P \not\models Q$.

Only If. We show that $T *_D P \not\models Q$ implies $T[X/Y] \wedge P \wedge EXA(k, X, Y, W) \not\models Q$. Let M be a model of $T *_D P$ such that $M \not\models Q$. This model satisfies P. Let M_T be a model of T having distance k from M. Define M' as $M \cup \{y_i | x_i \in M_T\}$. Obviously, M' satisfies both P and $T[X/Y]$. By definition of M_T, M can be extended to an assignment to W so that it also satisfies $EXA(k, X, Y, W)$. Therefore, $T[X/Y] \wedge P \wedge EXA(k, X, Y, W) \not\models Q$. $\square$

Now we show how Weber's revision can be compactly represented. Winslett [27] gives a similar proof, but only if the new formula has a size bounded by a constant. Our result, instead, does not need such a restriction. Let $\Omega = \{\omega_1, \ldots, \omega_k\}$ be the set of letters in the definition of Weber's revision operator, and $Z = \{z_1, \ldots, z_k\}$ be a new set of letters one-to-one with Ω.

Theorem 3.5 *$T[\Omega/Z] \wedge P$ is query equivalent (1) to $T *_{Web} P$.*

Proof. Let $X = V(T) \cup V(P)$ and Q be a formula on the alphabet X. We show that $T *_{Web} P \models Q$ iff $T[\Omega/Z] \wedge P \models Q$.

If. Assume that $T[\Omega/Z] \wedge P \not\models Q$: we prove that $T *_{Web} P \not\models Q$. By hypothesis, there exists a model M of $T[\Omega/Z] \wedge P$ on the alphabet $X \cup Z$ such that $M \not\models Q$. From M we build two models over the alphabet X as follows:

$$
\begin{aligned}
M' &= (M \cap (X \backslash \Omega)) \cup \{\omega_i \mid z_i \in M\} \\
M'' &= M \cap X
\end{aligned}
$$

We prove that M'' is a model of $T *_{Web} P$ and $M'' \not\models Q$. First, M' is a model of T, since M is a model of $T[\Omega/Z]$ and M' is obtained by replacing the value of the variables in Ω with the corresponding values in Z. Similarly, M'' is a model of P. Second, M' and M'' differ only on Ω. Thus, M'' is a model of $T *_{Web} P$. Since M and M'' evaluate atoms in X in the same way, Q is a formula over X, and $M \not\models Q$, it follows that $M'' \not\models Q$. As a result, there is a model of $T *_{Web} P$ which is not a model of Q, which implies that $T *_{Web} P \not\models Q$.

Only if. Assume that $T *_{Web} P \not\models Q$. We prove that Q is not implied by $T[\Omega/Z] \wedge P$. By hypothesis, there exists a model N of $T *_{Web} P$ such that $N \not\models Q$. This means that $N \models P$ and there exists model $M \models T$ such that M and N differ only for the atoms in Ω. Models M and N are interpretations over the alphabet X. From them, we build a single model H over the extended alphabet $X \cup Z$, as follows:

$$H = N \cup \{z_i \mid \omega_i \in M\}$$

It holds $H \models P \wedge T[\Omega/Z]$ and $H \not\models Q$. It follows that $T[\Omega/Z] \wedge P \not\models Q$. $\square$

We note that this representation of $T *_{Web} P$ increases the size of T only by —at most— the length of P whereas the less "compact" representation of $T *_D P$ requires a formula whose size is quadratic in the number of the letters.

3.2 Logical compactability

In this section we investigate logical compactability. In particular, we show that Dalal's and Weber's operators, which are query-compactable, are probably not logically-compactable.

Theorem 3.6 *Unless NP $\subseteq$ P/poly, the revision operators $*_D$ and $*_{Web}$ are not logically-compactable.*

Proof. We apply the general schema of Theorem 2.3 and the notation of Definition 2.5. We show that, for any integer n, there exists a formula P_n and a knowledge base T_n, both depending only on n, of polynomial size w.r.t. n, having the following property: given $\pi \in 3\text{-}\textsc{sat}_n$, there exists an interpretation M_π such that π is satisfiable iff M_π is a model of $T_n *_D P_n$ iff M_π is a model of $T_n *_{Web} P_n$.

Let $Y = \{y_1, \ldots, y_n\}$ be a set of new letters in one-to-one correspondence with the letters of B_n, and let $C = \{c_i \mid \gamma_i \in \pi_n^{max}\}$. Finally, let L be the set $B_n \cup Y \cup C$. Notice that L has size $O(n^3)$. We define T_n as the conjunction of two formulae:

$$T_n = \Phi_n \wedge \Gamma_n$$

The 2CNF formula Φ_n states non-equivalence between atoms in B_n and their correspondents in Y:

$$\Phi_n = \bigwedge_{i=1}^{n} (b_i \not\equiv y_i).$$

Γ_n codes every possible set of clauses in π_n^{max}, using the atoms in C as "enabling guards". The formula is defined as:

$$\Gamma_n = \bigwedge_{i=1}^{m_n^{max}} \gamma_i \vee \neg c_i.$$

Γ_n is a 4CNF formula and it contains $O(n^3)$ clauses. We define P_n as:

$$P_n = \bigwedge_{i=1}^{n} (\neg b_i \wedge \neg y_i)$$

Note that the size of T_n and P_n is $O(n^3)$. Moreover, T_n and P_n depend only on the size n. We define $C_\pi = \{c_i \in C \mid \gamma_i \text{ is a clause of } \pi\}$. We divide the proof into two parts. We show that:

(a) π is satisfiable implies that C_π is a model of $T_n *_D P_n$

(b) π is unsatisfiable implies that C_π is not a model of $T_n *_{Web} P_n$

Combining these results with the fact that all models of Dalal's revision are also models of Weber's revision (cf. Figure 1), we obtain that π is satisfiable iff C_π is a model of $T_n *_D P_n$ iff C_π is a model of $T_n *_{Web} P_n$.

Proof of (a) In every model of T_n exactly n atoms from $B_n \cup Y$ are true, while in every model of $T_n *_D P_n$ all atoms from $B_n \cup Y$ are false. Hence, $k_{T_n,P_n} \geq n$. Moreover, $k_{T_n,P_n} = n$ since B_n is a model of T_n and $\emptyset$ is a model of P_n, and the cardinality of their difference is n. Now, C_π is a model of P_n, hence it is also a model of $T_n *_D P_n$ iff there exists a model of T_n such that $|C_\pi \Delta M| = k_{T_n,P_n} = n$.

Assume π is satisfiable. Let B_π be a model of π, $\overline{Y_\pi} = \{y_i \mid b_i \notin B_\pi\}$ and $M = C_\pi \cup B_\pi \cup \overline{Y_\pi}$. We show that M is a model of T_n. In fact, M satisfies Φ_n by construction of $\overline{Y_\pi}$. M satisfies Γ_n as well, because for each clause $\gamma_i \vee \neg c_i$ of Γ_n, either $c_i \notin C_\pi$ or γ_i is satisfied by B_π. Now observe that $|C_\pi \Delta M| = |B_\pi \cup \overline{Y_\pi}| = n$. Hence, C_π is a model of $T_n *_D P_n$. Since the models of Dalal's revision are also models of Weber's revision, we also have that C_π is a model of $T_n *_{Web} P_n$.

Proof of (b) We prove the claim by contradiction. Assume π is unsatisfiable and C_π is a model of $T_n *_{Web} P_n$. Then there exists a model M of T_n such that $M \cap C = C_\pi$. We claim that $M \cap B_n$ is a model of π. Indeed, as for Dalal's revision, M satisfies $\Gamma_n = \bigwedge_{c_i \in C} \gamma_i \vee \neg c_i$. Simplifying Γ_n with truth values of $M \cap C = C_\pi$, we conclude that M satisfies $\bigwedge_{c_i \in C_\pi} \gamma_i$, which is exactly formula π. Since the formula contains only atoms from B_n, $M \cap B_n$ satisfies π, hence π is satisfiable and contradiction arises. Thus, C_π is not a model of $T_n *_{Web} P_n$. □

We conclude the section by generalizing the negative results about query compactability to logical compactability.

Theorem 3.7 *Unless NP ⊆ coNP/poly, the revision operators $*_B, *_F, *_{GFUV}, *_N, *_S$, and $*_{Win}$ are not logically-compactable.*

Proof. The claim follows from Theorems 3.1, 3.2, 3.3 and the fact that the non-existence of a representation satisfying criterion (1) implies the non-existence of a representation satisfying criterion (2). □

4 Single bounded revision

In the previous section we investigated the issue of the existence of compact representations of revised knowledge bases. As it turned out, for most of the operators a compact explicit representation of the result of revising a knowledge base with a new formula does most likely not exist. From an analysis of the proofs, it turns out that this behavior depends on the new formula P being very complex. There are some applications in which the size of the new formula is very small w.r.t. the size of the knowledge base. In database theory, for instance, scenarios with very large pieces of information are very common, while modifications only concern a small part of the base. In this section we investigate which impact this assumption has on the existence of compact representations. In particular, throughout this section we assume that the size of the new formula P is bounded by a constant (k from now on).

We first show that Ginsberg-Fagin-Ullman-Vardi's revision remains not compactable even under the above assumption.

Theorem 4.1 *Unless NP$\subseteq$ coNP/poly, the revision operator $*_{GFUV}$ is not query compactable, even if $|P| \leq k$.*

Proof. Let s be a new propositional variable. We define T'_n, P'_n from T_n, P_n in (3) and (4) by $T'_n = \{f \wedge (\neg s \vee P_n) | f \in T_n\} \cup \{\neg s\}$, $P'_n = s$. It immediately follows that, for all formulae Q on the alphabet $V(T_n) \cup V(P_n)$, we have that $T'_n *_{GFUV} P'_n \models Q$ iff $T_n *_{GFUV} P_n \models Q$. Assume that $*_{GFUV}$ is query compactable when $|P| \leq k$. By the above reduction we have that $*_{GFUV}$ is query compactable even when there is no bound on the size of P. By Theorem 3.1 it follows that NP $\subseteq$ coNP/poly. □

The situation for model-based revision operators is more complex. As it turns out all of them admit a compact representation, w.r.t. both query and logical equivalence, when the size of P is bounded.

Without loss of generality, since the size of P is bounded we assume that the alphabet $V(P)$ of P is included in the alphabet $V(T)$ of T (e.g., we can add to T the formula $\bigwedge_{p \in V(P)} (p \vee \neg p)$). Because of the assumption of $|P|$ being bounded by k, it follows that $|V(P)| \leq k$. Without loss of generality, we assume $|V(P)| = k$, and denote letters in $V(P)$ as $\{v_1, \ldots, v_k\}$.

We use the following notation: for every set of letters H, we denote with $\overline{H}$ the set $\{\neg x | x \in H\}$. The formula $F[H/\overline{H}]$, where $H \subseteq V(F)$, is F with each letter in H replaced by the corresponding letter in $\overline{H}$ (that is, its negation).

A useful property that we shall use is the following:

Proposition 4.2 *For each interpretation M of the letters in $V(F)$ and set $H \subseteq V(F)$, $M \models F$ if and only if $M \Delta H \models F[H/\overline{H}]$.*

In other words, if an interpretation M satisfies a formula F then, for any given set of letters H, the model $M\Delta H$, that agrees with M on all letters in $V(F) \backslash H$ and disagrees on all letters in H, satisfies the formula $F[H/\overline{H}]$. For example, let $F = x_1 \wedge (x_2 \vee \neg x_3)$, $M = \{x_1\}$ and $H = \{x_2, x_3\}$. Note that $M \models F$. Applying the definitions, we obtain that $M\Delta H = \{x_1, x_2, x_3\}$ and $F[H/\overline{H}] = x_1 \wedge (\neg x_2 \vee \neg\neg x_3)$. It follows that $\{x_1, x_2, x_3\} \models x_1 \wedge (\neg x_2 \vee \neg\neg x_3)$.

4.1 Compactability of "pointwise" operators

We exhibit a compact representation of $T *_{Win} P$ which uses exactly the same alphabet of T and P. Basically, we exploit the fact that we can explicitly represent all assignments to $V(P)$ in constant space, since $|V(P)| = k$.

Let S be an arbitrary set of letters such that $S \subseteq V(P)$. Let us consider the formula

$$P \wedge \bigvee_{S \subseteq V(P)} (T[S/\overline{S}] \wedge R)$$

where $\bigvee_{S \subseteq V(P)}$ means a disjunction for all possible subsets S of $V(P)$. This formula is satisfied by a model N of P iff there is a model M of T (whose distance from N is given by the set S), satisfying formula R.

In this formula S represents the distance between N and M. We define formula R in such a way it specifies that there is no other model in P whose distance from M is less than S.

$$R = \neg \bigvee_{C \subseteq V(P)\ ,\ C \Delta S \subset S} P[C/\overline{C}]$$

where $\bigvee_{C \subseteq V(P)\ ,\ C \Delta S \subset S}$ means a disjunction for all possible subsets C of $V(P)$, satisfying condition $C \Delta S \subset S$ (an equivalent condition is $C \neq \emptyset, C \subseteq S$). This formula imposes a condition over the distance C between two models of P. Namely, it forbids that C is between M and N.

The whole formula is

$$P \wedge \bigvee_{S \subseteq V(P)} (T[S/\overline{S}] \wedge \neg \bigvee_{C \subseteq V(P)\ ,\ C \Delta S \subset S} P[C/\overline{C}])$$

This formula can be rewritten, applying De Morgan's rule, to

$$P \wedge \bigvee_{S \subseteq V(P)} (T[S/\overline{S}] \wedge \bigwedge_{C \subseteq V(P)\ ,\ C \Delta S \subset S} \neg P[C/\overline{C}]) \tag{5}$$

It is easy to show that the following proposition holds.

Proposition 4.3 *Formula (5) has size linear in $|T|$ and is logically equivalent to $T *_{Win} P$.*

Using Proposition 4.3 and the fact that $T *_B P$ is $T \wedge P$ if consistent and $T *_{Win} P$ otherwise, we obtain:

Corollary 4.4 *There exists a formula of size linear in $|T|$ that is logically equivalent to $T *_B P$.*

We now exhibit a compact representation of $T *_F P$ which uses exactly the same alphabet of T and P. The main difference between Forbus' operator and Winslett's one relies on the fact that the notion of distance between models for the former one is based on set cardinality, while for the latter one on set containment. As a consequence, we obtain a formula very similar to the one obtained for Winslett's operator. In fact, the formula representing $T *_F P$ is the following:

$$P \wedge \bigvee_{S \subseteq V(P)} (T[S/\overline{S}] \wedge \neg \bigvee_{C \subseteq V(P)\ ,\ |C \Delta S| < |S|} P[C/\overline{C}]) \tag{6}$$

(cf. formula (5) and note that here cardinality of sets is considered in the subscript of the last disjunction). The proof of the theorem has been omitted.

Theorem 4.5 *Formula (6) has size linear in $|T|$ and is logically equivalent to $T *_F P$.*

We conclude this section by showing an example of the formulae obtained by applying formula (6).

Example. Let T and P be defined as:

$$\begin{array}{rcl} T & = & a \wedge b \wedge c \wedge d \wedge e \\ P & = & \neg a \vee \neg b \end{array}$$

T has just one model (call it M), while P has $3{\times}2^3$ models (each combination of the models of $\neg a \vee \neg b$ with $2^{\{c,d,e\}}$). $T \wedge P$ has no models. The models of P closest to T are $\{a, c, d, e\}$ and $\{b, c, d, e\}$ (we have $k_{M,P} = 1$). These are the models of $T *_F P$.

$V(P)$ is $\{a, b\}$. Applying (6), we get for $T *_F P$

$$P \quad \wedge \quad \bigvee_{S=\{\},\{a\},\{b\},\{a,b\}} (T[S/\overline{S}] \wedge \neg \bigvee_{C \subseteq V(P)\ ,\ |C \Delta S| < |S|} P[C/\overline{C}])$$

i.e.,

$$\begin{array}{rcl} P & \wedge & (\quad (a \wedge b \wedge c \wedge d \wedge e) \\ & & \vee \quad ((\neg a \wedge b \wedge c \wedge d \wedge e) \wedge \neg \bigvee_{C=\{a\}} P[C/\overline{C}]) \\ & & \vee \quad ((a \wedge \neg b \wedge c \wedge d \wedge e) \wedge \neg \bigvee_{C=\{b\}} P[C/\overline{C}]) \\ & & \vee \quad ((\neg a \wedge \neg b \wedge c \wedge d \wedge e) \wedge \neg \bigvee_{C=\{a\},\{b\},\{a,b\}} P[C/\overline{C}]) \quad) \end{array}$$

i.e.,

$$\begin{array}{rcl} (\neg a \vee \neg b) & \wedge & (\quad (a \wedge b \wedge c \wedge d \wedge e) \\ & & \vee \quad ((\neg a \wedge b \wedge c \wedge d \wedge e) \wedge \neg(a \vee \neg b)) \\ & & \vee \quad ((a \wedge \neg b \wedge c \wedge d \wedge e) \wedge \neg(\neg a \vee b)) \\ & & \vee \quad ((\neg a \wedge \neg b \wedge c \wedge d \wedge e) \wedge \neg((a \vee \neg b) \vee (\neg a \vee b) \vee (a \vee b))) \quad) \end{array}$$

It is easy to verify that this formula has exactly two models: $\{b, c, d, e\}$ (i.e., the unique model of the subformula on the second line), and $\{a, c, d, e\}$ (i.e., the unique model of the subformula on the third line). $\Box$

4.2 Compactability of "global" operators

For all global belief revision operators (Satoh's, Dalal's, and Weber's) the result of the revision of T with P is logically equivalent to a formula whose size is linear in the size of T. This means that its size is polynomial whenever the size of P is bounded by a constant.

Theorem 4.6 *The following equivalences hold:*

$$
\begin{aligned}
T *_S P &\equiv P \wedge \bigvee_{S \in \delta(T,P)} T[S/\overline{S}] && (7)\\
T *_D P &\equiv P \wedge \bigvee_{|S| = k_{T,P}} T[S/\overline{S}] && (8)\\
T *_{Web} P &\equiv P \wedge \bigvee_{S \subseteq \Omega} T[S/\overline{S}] && (9)
\end{aligned}
$$

The right-hand size formulas of those equations have size bounded by a polynomial in the size of T.

Proof. The proof for Weber's revision operator (9) follows immediately from Weber's original definition of $*_{Web}$. We report a skecth for Dalal's revision only, as the proof of the other case is similar, and can be found in [5].

First of all, the considered formula has size polynomial in the size of T. Assuming the size of P to be bounded by a constant, the number of sets S such that $|S| = k_{T,P}$ or $S \subseteq \Omega$ is bounded by a constant, too. This implies that the size of Formula (8) is linear in the size of T.

Formula (8) is a disjunction of subformulas: each one $P \wedge T[S/\overline{S}]$ is used to express a set of models of P whose distance from a model of T is exactly S. Since S can be any set such that its cardinality is $k_{T,P}$, the formula expresses exactly the result of Dalal's revision. $\Box$

Note that all representations can be simplified by omitting in the disjunction all $T[S/\overline{S}]$ which are inconsistent with P.

We close the description of results for single revision with an example:

Example. We continue the previous example. Recall that T and P are defined as:

$$
\begin{aligned}
T &= a \wedge b \wedge c \wedge d \wedge e\\
P &= \neg a \vee \neg b
\end{aligned}
$$

The set of minimal differences between models of T and models of P is $\delta(T,P) = \{\{a\},\{b\}\}$. The models of P whose difference from a model of T is in $\delta(T,P)$ are $\{a,c,d,e\}$ and $\{b,c,d,e\}$. These are the models of $T *_S P$. Since both differences have the minimum cardinality, $k_{T,P} = 1$, these two models are also models for $T *_D P$. Moreover, since $\Omega = \{a,b\} = V(P)$, $T *_{Web} P$ has also the third model $\{b,c,e\}$.

Applying either Formula (7) or Formula (8), we get for $T *_S P = T *_D P$

$$P \quad \wedge \quad (\bigvee_{S=\{a\},\{b\}} T[S/\overline{S}])$$

i.e.,

$$\begin{array}{lll} (\neg a \vee \neg b) & \wedge & (\quad (\neg a \wedge b \wedge c \wedge d \wedge e) \\ & & \vee \quad (a \wedge \neg b \wedge c \wedge d \wedge e) \quad) \end{array}$$

This formula admits indeed the two models $\{b,c,d,e\}$, and $\{a,c,d,e\}$. Although it has exactly the same two models as $T *_F P$, it is syntactically much simpler.

As for $T *_{Web} P$, Formula (9) yields

$$P \quad \wedge \quad (\bigvee_{S=\{a\},\{b\},\{a,b\}} T[S/\overline{S}])$$

i.e.,

$$\begin{array}{lll} (\neg a \vee \neg b) & \wedge & (\quad (\neg a \wedge b \wedge c \wedge d \wedge e) \\ & & \vee \quad (a \wedge \neg b \wedge c \wedge d \wedge e) \\ & & \vee \quad (\neg a \wedge \neg b \wedge c \wedge d \wedge e) \quad) \end{array}$$

which admits also the model $\{c,d,e\}$. $\square$

4.3 Summary of results for single revision

In Section 3 and in the present section we have presented several results about the size of a propositional theory T' representing the revision of a knowledge base and satisfying either criterion (1) or (2). We proved that some formalizations of belief revision (e.g., Forbus' and Ginsberg-Fagin-Ullman-Vardi's) lead to propositional theories T' which are intrinsically not representable in polynomial space (unless the polynomial hierarchy collapses). In other cases, e.g., logical equivalence (2) for Dalal's and Weber's operators, we showed polynomial-size representations which are equivalent to the revised theory. The results are summarized in Table 1, where YES stands for compactable, while NO stands for not compactable.

The following comments on the table are in order.

Formalism	General case: Logical equiv. (2)	General case: Query equiv. (1)	Bounded case: Logical equiv. (2)	Bounded case: Query equiv. (1)
GFUV, Nebel	NO Th. 3.7	NO Th. 3.1	NO Th. 4.1	NO Th. 4.1
Winslett	NO Th. 3.7	NO Th. 3.2	YES Prop. 4.3	YES Prop. 4.3, [27]
Borgida	NO Th. 3.7	NO Th. 3.2	YES co. 4.4	YES co. 4.4, [27]
Forbus	NO Th. 3.7	NO Th. 3.3	YES Th. 4.5	YES Th. 4.5
Satoh	NO Th. 3.7	NO Th. 3.2	YES Th. 4.6	YES Th. 4.6
Dalal	NO Th. 3.6	YES Th. 3.4	YES Th. 4.6	YES Th. 3.4, Th. 4.6
Weber	NO Th. 3.6	YES Th. 3.5	YES Th. 4.6	YES Th. 3.5, Th. 4.6, [27]
WIDTIO	YES –	YES –	YES –	YES –

Table 1: Is the revised knowledge base compactable?

- First of all, we remind that:
  - if an entry in a query equivalence column is “NO”, then also the corresponding entry in the logical equivalence column must be “NO”;
  - conversely, if an entry in a logical equivalence column is “YES”, then also the corresponding entry in the query equivalence column must be “YES”;
  - analogously, “NO” in the bounded case implies “NO” in the general case, and “YES” in the general case implies “YES” in the bounded case.
- As for syntax-based operators (with the exception of WIDTIO which is obviously compactable), the size of P is not relevant. Intuitively, this holds because a single literal is able to generate an exponential number of distinct possibilities, cf. example concerning the $*_{GFUV}$ operator in Subsection 3.1.
- As for model-based operators:
  - in the bounded case the revision involves only literals occurring in P, hence compactness is guaranteed, regardless of the equivalence criterion;
  - in the unbounded case, revision according to query equivalence (1) can be computed in two steps:
    1. computation of a “measure of the minimal distance” (e.g., $k_{T,P}$ for $*_D$, Ω for $*_{Web}$, $\delta_{T,P}$ for $*_S$);
    2. using the measure for computing minimal sets.

    Only for $*_D$ and $*_{Web}$ the measure appears to be compactable.

    Note anyway that the above two-steps method does not work for logical equivalence (2), because it uses new letters. Summing up, Dalal's and Weber's formalizations have an interesting behavior: T' has no polynomial-size representation if we insist on logical equivalence, but such a representation does exist if we ask only query equivalence.
- Compactability is not directly related to the selectivity of the operators in choosing the resulting set of models. The two model-based operators that more often admit compact representations are $*_D$ and $*_{Web}$. As shown in Figure 1, $*_D$ is the most selective operator while $*_{Web}$ is one of the least selective ones.

5 Iterated unbounded revision

In the previous sections we have analyzed the size of the smallest propositional representation of the result of a single revision or update. We now turn our attention to the size of the result of a *series* of revisions or updates. Obviously, we only investigate operators that admit a compact representation after a single revision, since operators shown uncompactable for a single revision are also uncompactable for a series of revisions. In the style of Sections 3 and 4, we divide our analysis into two cases: the general one (considered in this section), where no constraints are imposed on the sequence of m revising formulae $\{P^1, \ldots, P^m\}$, and the bounded case (considered in the next section), where we assume that the size of each P^i is bounded by a constant.

In the general case, we only need to investigate the properties of Dalal's and Weber's operators under the query equivalence criterion. In particular, we show that Dalal's and Weber's operators retain their compactability properties even if the revision process is iterated m times, if we go for criterion (1), i.e., the initial language is extended.

For what concerns Dalal's operator, note that the straightforward, m-times repeated, application of Theorem 3.4 yields a formula of size exponential in $|T|+|P|+m$. Therefore, we need to show a different formula.

Let $EXA(r, S_1, S_2, S_3)$ denote the formula defined in Section 3.1 and containing letters of S_1 and S_2, and possibly other letters S_3, which is true iff the Hamming distance between the values assigned to S_1 and S_2 is exactly r.

Now let $P^1, \ldots, P^m$ be the sequence of revising formulae. We denote with k_1 the minimum distance between the models of P^1 and T and with k_i the minimum distance between the models of P^i and $T *_D P^1 *_D \cdots *_D P^{i-1}$. Moreover, we define $X = V(T) \cup V(P^1) \cup \cdots \cup V(P^m)$ and we use a family Y_i of sets of (distinct) letters, where each member Y_i of the family Y is one-to-one with X. Using this notation, the revised theory $T *_D P^1 *_D \cdots *_D P^m$ can be expressed as:

$$\begin{aligned} \Phi_m \;=\;& T[X/Y_1] \wedge P^1[X/Y_2] \wedge \cdots \wedge P^{m-1}[X/Y_m] \wedge P^m \wedge \\ & EXA(k_1, Y_1, Y_2, W_1) \wedge \cdots \wedge EXA(k_m, Y_m, X, W_m) \end{aligned}$$

Note that the formula contains m distinct instances of the formula EXA, each one comparing sets of cardinality at most n. Thus, we have:

Theorem 5.1 *Φ_m has size polynomial in $|T|+|P^1|+\cdots+|P^m|$ and is query equivalent to $T *_D P^1 *_D \cdots *_D P^m$.*

Proof. Let Q be a query on the alphabet X. We prove the theorem by showing that $\Phi_m \models Q$ iff $T *_D P^1 *_D \cdots *_D P^m \models Q$. The proof is by induction on m, and the base case is proven in Theorem 3.4.

If. We show that $\Phi_m \not\models Q$ implies $T *_D P^1 *_D \cdots *_D P^m \not\models Q$. Let M be a model of Φ_m such that $M \not\models Q$, and let $N = M \cap X$. Since Q only uses letters of X, it holds $N \not\models Q$. Since $M \models EXA(k_m, Y_m, X, W_m)$, the distance between the two sets $M \cap X$ and $M \cap Y_m$ is exactly k_m. Let $M' = (M \Delta (X \cup Y_m)) \cup \{x_i | y_n^i \in M\}$. Note that, by construction, $M' \models \Phi_{m-1}$ and, by the inductive hypothesis, $M' \cap X$ is a model of

$T *_D P^1 *_D \cdots *_D P^{m-1}$. Therefore, $M \cap X$ has distance k_m from a model $(M' \cap X)$ of $T *_D P^1 *_D \cdots *_D P^{m-1}$ and, therefore, it is also a model of $T *_D P^1 *_D \cdots *_D P^m$.

Only If. We show that $T *_D P^1 *_D \cdots *_D P^m \not\models Q$ implies $\Phi_m \not\models Q$. Let N_{m+1} be a model of $T *_D P^1 *_D \cdots *_D P^m$ such that $N_{m+1} \not\models Q$. Hence there exists a model N_m of $T *_D P^1 *_D \cdots *_D P^{m-1}$ such that the distance between N_{m+1} and N_m is exactly k_m. Inductively, for each i $(1 \leq i \leq m-1)$, there exists a model M_i of $T *_D P^1 *_D \cdots *_D P^i$ such that the distance between M_{i+1} and M_i is exactly k_i. Now, let M be a model of Φ_m such that M coincides with N_m on X. Since Q only uses letters of X, we have that $M \not\models Q$. $\square$

Now we show how Weber's revision can be compactly represented also for iterated revisions. By Theorem 3.5 we know that the propositional representation of $T *_{Web} P$ only increases the size of $|T| + |P|$ by a linear factor. We denote with Ω_i the set associated to the i-th revision step $(T *_{Web} P^1 *_{Web} \cdots *_{Web} P^{i-1}) *_{Web} P^i$. To each Ω_i we associate a new set Z_i, one-to-one with Ω_i. Therefore, $T *_{Web} P^1 *_{Web} \cdots *_{Web} P^m$ can be represented by the formula:

$$T[\Omega_1/Z_1; \cdots; \Omega_n/Z_m] \wedge P^1[\Omega_2/Z_2; \cdots; \Omega_n/Z_m] \wedge \cdots \wedge P^i[\Omega_{i+1}/Z_{i+1}; \cdots; \Omega_n/Z_m] \wedge \cdots \wedge P^m \tag{10}$$

where the substitutions must be performed in left-to-right order. Thus, we obtain:

Corollary 5.2 *Formula (10) has size linear in $|T| + |P^1| + \cdots + |P^m|$ and is query equivalent to $T *_{Web} P^1 *_{Web} \cdots *_{Web} P^m$.*

We close this section with an example of the application of Formula (10).

Example. We expand the example of the previous section. To highlight the one-to-one correspondence between sets of letters, we use $\{x_1, x_2, x_3, x_4, x_5\}$ instead of $\{a, b, c, d, e\}$. We define T, P^1 and P^2 as:

$$\begin{array}{rcl} T & = & x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \\ P^1 & = & \neg x_1 \vee \neg x_2 \\ P^2 & = & \neg x_5 \end{array}$$

T has just one model (call it M), while P^1 has 3×2^3 models and P^2 has 2^4 models. Recall that $\Omega_1 = \{x_1, x_2\} = V(P^1)$ and, therefore, $T *_{Web} P^1$ has models $\{x_1, x_3, x_4, x_5\}$, $\{x_2, x_3, x_4, x_5\}$ and $\{x_3, x_4, x_5\}$. Now, $\Omega_2 = \{x_5\} = V(P^2)$. As a consequence, $T *_{Web} P^1 *_{Web} P^2$ has models $\{x_1, x_3, x_4\}$, $\{x_2, x_3, x_4\}$ and $\{x_3, x_4\}$. We define two sets $Z_1 = \{z_1^1, z_1^2\}$ and $Z_2 = \{z_2^1\}$.

Applying Formula (10) yields $T[x_1/z_1^1; x_2/z_1^2; x_5/z_2^1] \wedge P^1[x_5/z_2^1] \wedge P^2$ i.e.,

$$(z_1^1 \wedge z_1^2 \wedge x_3 \wedge x_4 \wedge z_2^1) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_5)$$

which admits the three models $\{x_1, x_3, x_4, z_1^1, z_1^2, z_2^1\}$, $\{x_2, x_3, x_4, z_1^1, z_1^2, z_2^1\}$ and $\{x_3, x_4, z_1^1, z_1^2, z_2^1\}$ which – projecting out z_1^1, z_1^2 and z_2^1 – are exactly the models of $T *_{Web} P^1 *_{Web} P^2$. $\square$

6 Iterated bounded revision

In this section we assume that the size of all formulae $\{P^1, \ldots, P^m\}$ of the sequence of revisions is bounded by a constant. However, we assume that the number of revisions is arbitrary, therefore, in order to show the existence of compact representations, we must show formulae whose size is bounded by a polynomial in $|T| + m$.

We first focus on the query equivalence criterion. Note that the propositional formulae presented in Section 4.2 increase exponentially in m their size if the revision operator is iterated m times. As a consequence, we first find new formulae, that only preserve query equivalence, but that can be iterated without exploding the size. Since all the representations are very similar, we only show the representation for Winslett's operator.

Let $V(T)$ be the alphabet of T; without loss of generality, assume that the alphabet $V(P)$ of P is included in $V(T)$. Because of the assumption of $|P|$ being bounded by a constant k, it follows that $|V(P)| \leq k$. We assume $|V(P)| = k$, and rearrange subscripts in such a way that $V(P) = \{x_1, \ldots, x_k\}$. Let $Y = \{y_1, \ldots, y_k\}$ be a set of letters one-to-one with $V(P)$. Since $T \wedge P$ may be inconsistent, we replace in T the letters in $V(P)$ with the new letters Y. This yields the formula

$$T[V(P)/Y] \wedge P \tag{11}$$

This formula is satisfiable – if both T and P are – but it is not query equivalent to $T *_{Win} P$, since any model of P can be suitably extended to a model of (11). Hence we want to impose further constraints to this formula. In the following we use the notation $M_{|S}$ to represent the set of letters that are mapped into *true* by M and that belong to S, i.e., $M \cap S$.

Let M be a model of (11), i.e., an assignment to $V(T) \cup Y$. Let $\overline{V(P)}$ denote the set $V(T) - V(P)$, i.e., letters of T not appearing in P. We partition M as $M_{|V(P)} \cup M_{|\overline{V(P)}} \cup M_{|Y}$. Let $M' = \{x_i \in V(P) \mid y_i \in M_{|Y}\}$, i.e., $M' = M_{|Y}[Y/V(P)]$. Observe that M is a model of (11) if and only if M is a model of P and $M' \cup M_{|\overline{V(P)}}$ is a model of T.

By Proposition 2.1, it follows that the models of P which are closest to a model M of T – i.e., having minimal set difference with M – agree with M on letters in $\overline{V(P)}$.

Therefore, a model $M = M_{|V(P)} \cup M_{|\overline{V(P)}} \cup M_{|Y}$ of (11) should be discarded if there exists another model of P which is closer to the model $M' \cup M_{|\overline{V(P)}}$ of T. Let Z be a set of letters one-to-one with $V(P)$. In order to make the notation more compact, we introduce three formula schemata:

$$
\begin{aligned}
F_P(S) &= P[V(P)/S] \\
F_{\subseteq}(S_1, S_2, S_3, S_4) &= \bigwedge_{j=1}^{k} ((s_1^j \not\equiv s_2^j) \rightarrow (s_3^j \not\equiv s_4^j))
\end{aligned}
$$

where the sets S, S_1, S_2, S_3 and S_4 contain k letters each. The first schema F_P ensures that the assignment to S is a model of P, while the second one $F_{\subseteq}$ states that for each truth assignment to $S_1 \cup S_2 \cup S_3 \cup S_4$ the set of atoms that have a different truth value in S_1 and S_2 is a subset of the set of atoms that have a different truth value in S_3 and S_4.

Using this notation, we impose that an assignment to Z is a model of P with the formula $F_P(Z)$. Since a model of (11) must be considered only if there is not a closer model, the whole revision can be reformulated as the following quantified boolean formula:

$$
T[V(P)/Y] \wedge P \wedge \forall Z.(F_P(Z) \wedge F_{\subseteq}(Z, Y, Y, V(P))) \rightarrow F_{\subseteq}(V(P), Y, Y, Z) \qquad (12)
$$

where $\forall Z$ is a shorthand for $\forall z_1 \cdots \forall z_k$. This formula is now turned into an (unquantified) propositional formula by replacing the universal quantification with a conjunction over all assignments to Z. Since there exist $2^{|P|}$ distinct assignments to Z, the total size of $T *_{Win} P$ is $O(|T| + |P| + 2^{|P|})$. Observe that the assignments to Z which are not models of P can be discarded (they do not satisfy $F_P(Z)$, hence the implication simplifies to $\top$), and the simplified formula is linear in the number of models of P, hence it could be significantly smaller than $2^{|P|}$. Note that this explicit representation introduces new letters, hence it does not preserve logical equivalence (2). A linear-size representation with new letters was also shown by Winslett herself in [27].

Example. As in the last example, $T = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5$ and M is its unique model. Now, we take $P = \neg x_1$. Note that $T \wedge P$ has no models.

$V(P)$ is $\{x_1\}$. Let $Y = \{y_1\}$ and $Z = \{z_1\}$ be two new sets of letters. Applying (12), we get for $T *_{Win} P$

$$
\begin{gathered}
(y_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5) \wedge (\neg x_1) \\
\wedge \\
\forall z_1. \Big(\; ((\neg z_1) \wedge (z_1 \not\equiv y_1) \rightarrow (y_1 \not\equiv x_1)) \rightarrow ((x_1 \not\equiv y_1) \rightarrow (y_1 \not\equiv z_1)) \; \Big)
\end{gathered}
$$

Substituting the universal quantification with a conjunction corresponding to the two possible assignments for z_1 and considering that the assignment mapping z_1 to *true* does not satisfy $F_P(Z)$, hence it can be discarded, we obtain

$$
\begin{gathered}
(y_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5) \wedge (\neg x_1) \\
\wedge \\
\Big(\; ((\neg \bot) \wedge (\bot \not\equiv y_1) \rightarrow (y_1 \not\equiv x_1)) \rightarrow ((x_1 \not\equiv y_1) \rightarrow (y_1 \not\equiv \bot)) \; \Big)
\end{gathered}
$$

Now the formula could be simplified and rearranged; anyway, observe that its single model is $\{x_2, x_3, x_4, x_5, y_1\}$ which – projecting out y_1 – is exactly the model of $T *_{Win} P$. $\Box$

Along the same lines, one could verify that Satoh's revision $T *_S P$ is query-equivalent to the quantified boolean formula

$$T[V(P)/Y] \wedge P \wedge \forall W \forall Z.(F_P(Z) \wedge T[V(P)/W] \wedge F_{\subseteq}(Z, W, Y, V(P)) \rightarrow F_{\subseteq}(V(P), Y, W, Z)) \quad (13)$$

Analogously, Forbus' revision $T *_F P$ is query-equivalent to the quantified boolean formula

$$T[V(P)/Y] \wedge P \wedge \forall Z.(F_P(Z) \rightarrow \neg(\exists W_1 W_2.DIST(Z, Y, W_1) < DIST(V(P), Y, W_2))) \quad (14)$$

where $DIST(\cdot, \cdot, \cdot)$ is a formula that computes the Hamming distance between the assignments to the two sets of letters and W_1 and W_2 are two sets of new letters that represent the intermediate results and must satisfy the appropriate constraints. Moreover, the $<$ operator can be represented with a circuit that compares the binary representations of the two numbers $DIST(Z, Y, W_1)$ and $DIST(V(P), Y, W_2)$.

As far as Borgida's revision is concerned, since it coincides with Winslett's one if $T \wedge P$ is unsatisfiable and it is $T \wedge P$ otherwise, a query-equivalent formula can be directly obtained from formula (12).

Formulae (12), (13) and (14) only apply to the result of a single revision, but, differently from formulae (5), (7) and (6), they can be extended to hold for a series of revisions. We focus on Winslett's operator.

The main idea of the following construction is to define a series of formulae WIN_i that are query equivalent to T after a series of updates with the formulae $P^1, \ldots, P^i$ (i.e., $T *_{Win} P^1 *_{Win} \cdots *_{Win} P^i$). These formulae WIN_i are built inductively, starting with formula (12). In order to make the final formula understandable we introduce some notation.

Without loss of generality, assume that for each i the alphabet $V(P^i)$ of P^i is included in the alphabet $V(T)$ of T. If we assume that for each P^i its size is bounded by a constant k_i, it follows that for each i, $|V(P^i)| \leq k_i$. We assume $|V(P^i)| = k_i$. For each P^i, let Y_i and Z_i be sets of letters one-to-one with $V(P^i)$. Given a formula T and a series of revising formulae $P^1, \ldots, P^m$, we start with formula (12) rewritten with index 1:

$$WIN_1 = T[V(P^1)/Y_1] \wedge P^1 \wedge \forall Z_1.(F_{P^1}(Z_1) \wedge F_{\subseteq}(Z_1, Y_1, Y_1, V(P^1)) \rightarrow F_{\subseteq}(V(P_1), Y_1, Y_1, Z_1) \quad (15)$$

and, inductively, we define:

$$WIN_i = WIN_{i-1}[V(P^i)/Y_i] \wedge P^i \wedge \forall Z_i.(F_{P^i}(Z_i) \wedge F_{\subseteq}(Z_i, Y_i, Y_i, V(P^i)) \rightarrow F_{\subseteq}(V(P^i), Y_i, Y_i, Z_i) \quad (16)$$

It can be easily proved that the above formula is polynomial, and equivalent to Winslett's revision.

Theorem 6.1 *Formula (16) has size polynomial in* $|T|+m$ *and is query equivalent to* $T *_{Win} P^1 *_{Win} \cdots *_{Win} P^m$.

Given the similarities with Borgida's operator, the size of the above formula is also an upper bound for Borgida's operator. A similar construction can also be applied to Satoh's and Forbus' revision operators. As a consequence, we have that:

Theorem 6.2 *The iterated version of formula (13) has size polynomial in* $|T|+m$ *and is query equivalent to* $T *_S P^1 *_S \cdots *_S P^m$. *Furthermore, the iterated version of formula (14) has size polynomial in* $|T|+m$ *and is query equivalent to* $T *_F P^1 *_F \cdots *_F P^m$.

In order to prove that those revision operators are query equivalent to propositional formulas of polynomial size, what is missing is to prove that Formulas 12–16 can be rewritten as propositional formulas (in the way they are currently written, they are quantified boolean formulas). This is done by replacing each universal quantifier with a conjunction over all possible assignments over the quantified variables. Note that there are only polynomially many assignments to the quantified variables.

Theorem 6.3 *Formulas 12–16 can be converted into equivalent propositional formulas with an increase in size that is at most quadratic.*

As a result, we have proved the query compactability of all the above revision operators.

Corollary 6.4 *The iterated version of Winslett's, Borgida's, Forbus', and Satoh's revision operators are query-compactable.*

We remark that the existence of a compact representation for Dalal's and Weber's operators is guaranteed by Theorems 5.1 and 5.2.

6.1 Logical Equivalence

While compact representations still exist if we go for query equivalence, this is not the case for logical equivalence.

Theorem 6.5 *Unless* $NP \subseteq P/poly$*, there is no formula of size polynomial in* $|T|+m$ *that is logically equivalent to* $T * P^1 * \cdots * P^m$*, where* $* \in \{*_B, *_D, *_F, *_S, *_{Web}, *_{Win}\}$.

Proof. The proof relies on (implicitly) showing that an unbounded number of bounded revisions can accomplish the same task of a single unbounded revision. Since we know that all of the above operators are not logically compactable when a single unbounded revision is applied, this proves the result.

We apply an iterated version of the general schema of Theorem 2.3 and the notation of Definition 2.5, so we show that for any integer n, there exists a sequence of n formulae $P_n^1 \dots P_n^n$ and a knowledge base T_n, all depending only on n, of polynomial size w.r.t. n, such that given any $\pi \in$ 3-$\textsc{sat}_n$, there exists an interpretation C_π such that π is satisfiable iff C_π is a model of $T_n * P_n^1 * \dots * P_n^n$.

Let $Y = \{y_1, \dots, y_n\}$ be a set of new letters in one-to-one correspondence with letters of B_n, and C be a set of new letters one for each clause in π_n^{max}, i.e., $C = \{c_i \mid \gamma_i \in \pi_n^{max}\}$. Finally, let L be the set $B_n \cup Y \cup C$. Notice that $|L|$ is $O(n^3)$. We define:

$$T_n = \Phi_n \wedge \Gamma_n,$$

where $\Gamma_n = \bigwedge_{i=1}^{m_n^{max}} c_i \rightarrow \gamma_i$ and the 2CNF formula Φ_n states non-equivalence between atoms in B_n and their correspondent in Y:

$$\Phi_n = \bigwedge_{i=1}^{n} (b_i \not\equiv y_i).$$

We define the set of n formulae $\{P_n^1 \dots P_n^n\}$ as:

$$P_n^i = (\neg b_i \wedge \neg y_i)$$

Note that the size of T_n is $O(n^3)$ and the size of each P_n^i is constant. Moreover, T_n and $\{P_n^1 \dots P_n^n\}$ do not depend on a specific 3CNF formula π, but only on its size n.

We denote $C_\pi = \{c_i \in C \mid \gamma_i \text{ is a clause of } \pi\}$. We show that π is satisfiable iff C_π is a model of $T * P_n^1 * \dots * P_n^n$, where $* \in \{*_B, *_D, *_F, *_S, *_{Web}, *_{Win}\}$.

We first show that the sets of models of $(T * P_n^1 * \dots * P_n^n)$ coincide for all considered model-based operators. Equivalence is shown inductively on n. Let us consider the base case and show equivalence of $T * P_n^1$ for all operators. Let $S = \{N | \exists M \in \mathcal{M}(T) \text{ s.t. } N = M/\{b_1, y_1\}\}$. First of all, note that all models in S satisfy P_n^1 since they contain neither b_1 nor y_1. We want to show that S is the set of models of $T * P_n^1$ for all of the operators. In order to accomplish this, we only need to show that S is the set of models of $T *_D P_n^1$, $T *_{Web} P_n^1$ and $T *_{Win} P_n^1$. The other equivalences follow from Figure 1 and the fact that $T \wedge P_n^1$ is inconsistent, and, therefore $T *_B P_n^1$ coincides with $T *_{Win} P_n^1$.

Dalal We observe that in every model of T for each i there is exactly one of b_i and y_i that is true, while in every model of $T *_D P_n^1$ both b_1 and y_1 are false. Hence, $k_{T,P_n^1} \geq 1$. Moreover, $k_{T,P_n^1} = 1$ since B_n is a model of T and $B_n - \{b_1\}$ is a model of P_n^1. Since S is exactly the set of models of P_n^1 that have distance 1 from a model of T, it follows that $\mathcal{M}(T *_D P_n^1) = S$.

Weber Note that B_n and Y are models of T, $\mu(B_n, P_n^1) = \{b_1\}$ and $\mu(Y, P_n^1) = \{y_1\}$. This implies that both $\{b_1\}$ and $\{y_1\}$ are in $\delta(T, P_n^1) = \bigcup_{M \in \mathcal{M}(T)} \mu(M, P_n^1)$. Recall that $\Omega_1 = \cup\delta(T, P_n^1)$. Hence, $\{b_1, y_1\} \subseteq \Omega_1$. Moreover, for every model M of T, $\mu(M, P_n^1)$ contains no atom from C because for every subset of C there is a model of P_n^i. Hence, $\Omega_1 = \{b_1, y_1\}$. By definition of Weber's operator it follows that $\mathcal{M}(T *_{Web} P_n^1) = S$.

Winslett Given a model M of T, we compute $\mu(M, P_n^1)$. If $b_1 \in M$ then $\mu(M, P_n^1) = b_1$, otherwise if $y_1 \in M$ then $\mu(M, P_n^1) = y_1$. As a consequence, we have $\mathcal{M}(T *_{Win} P_n^1) = A \cup B$, where:

$$\begin{array}{rcl} A & = & \{N \in \mathcal{M}(P_n^1) | \exists M \in \mathcal{M}(T) : ((b_1 \in M) \wedge (M \Delta N = \{b_1\}))\} \\ B & = & \{N \in \mathcal{M}(P_n^1) | \exists M \in \mathcal{M}(T) : ((y_1 \in M) \wedge (M \Delta N = \{y_1\})\} \end{array}$$

Since any model of T contains exactly one of y_1 and b_1, this set is equal to $\{N \in \mathcal{M}(P_n^1) | \exists M \in \mathcal{M}(T) : M \Delta N \subseteq \{b_1, y_1\}\}$. This set clearly coincides with S.

By repeating the same line of reasoning, it follows that for all $1 \leq i \leq m$ we have that $k_{T *_D P_n^1 \ldots *_D P_n^{i-1}, P_n^i} = 1$, $\Omega_i = \{b_i\} \cup \{y_i\}$ and for any model M of $T *_{Win} P_n^1 \cdots *_{Win} P_n^{i-1}$ if $b_i \in M$ then $\mu(M, P_n^i) = b_i$, otherwise if $y_i \in M$ then $\mu(M, P_n^i) = y_i$. The equivalence immediately follows.

Using the equivalence of all of the model-based operators it suffices to show that π is satisfiable if and only if C_π is a model of $T *_D P_n^1 \cdots *_D P_n^n$.

By definition, C_π is a model of $T *_D P_n^1 \ldots *_D P_n^n$ iff there exists a sequence of m models $\{M_0, M_1, \ldots, M_m\}$ such that:

1. $M_0 \models T$;
2. $M_i \models T *_D P_n^1 \ldots *_D P_n^i$;
3. $M_m = C_\pi$;
4. $|M_{i+1} \Delta M_i| = 1$.

If. Let π be satisfiable, B_π be a model of π and $\overline{Y_\pi} = \{y_i \mid b_i \notin B_\pi\}$. We define the sequence of models as follows:

- $M_0 = C_\pi \cup B_\pi \cup \overline{Y_\pi}$;
- $M_{i+1} = M_i - \{b_i, y_i\}$.

Note that conditions (3) and (4) are trivially satisfied, while condition (1) is satisfied since M_0 satisfies Φ_n by construction of $\overline{Y_\pi}$, and also M_0 satisfies Γ_n, because for each clause $c_i \rightarrow \gamma_i$ of Γ_n, either $c_i \notin C_\pi$ or γ_i is satisfied by B_π. We show that also condition (2) is satisfied by induction on m. The base case is proven by condition (1); Assume that $M_i \models T *_D P_n^1 \dots *_D P_n^i$. Since $M_{i+1} \models P_n^{i+1}$ and $|M_{i+1} \Delta M_i| = 1$ it follows that $M_{i+1} \models T *_D P_n^1 \dots *_D P_n^{i+1}$.

Only if. Suppose C_π is a model of $T *_D P_n^1 \dots *_D P_n^n$. Then there exists a sequence of models $\{M_0, M_1, \dots, M_m\}$ such that

1. $M_0 \models T$;
2. $M_i \models T *_D P_n^1 \dots *_D P_n^i$;
3. $M_m = C_\pi$;
4. $|M_{i+1} \Delta M_i| = 1$.

Since $|M_{i+1} \Delta M_i| = 1$, the difference $M_{i+1} \Delta M_i$ contains exactly 1 atom from $B_n \cup Y$. Hence, all M_i agree on the truth assignment to atoms of C, that is, $M_i \cap C = C_\pi$. We claim that $M_0 \cap B_n$ is a model of π. In fact, M_0 satisfies $\Gamma_n = \bigwedge_{i=1}^{m_n^{max}} c_i \rightarrow \gamma_i$. Simplifying Γ_n with truth values of $M_0 \cap C = C_\pi$, we conclude that M_0 satisfies all clauses in $\{\gamma_i | c_i \in C_\pi$, which is exactly formula π. Since the formula contains only atoms from B_n, the interpretation $M_0 \cap B_n$ satisfies π, hence π is satisfiable. □

6.2 Summary of results for iterated revision

In Section 5 and in the present section we have shown several results about the size of a propositional theory representing the iterated revision of a knowledge base and satisfying either query equivalence or logical equivalence. In general, all formalizations of belief revision considered (with the exception of WIDTIO) lead to logically equivalent propositional theories which are intrinsically not representable in polynomial space (unless the polynomial hierarchy collapses). Restricting our attention to query equivalence, we found situations where compact representations exist. Results are summarized in Table 2, where YES stands for compactable, while NO stands for not compactable.

The following comments on the table are in order.

- First of all, if an entry is "NO" in Table 1, then it is "NO" also in this table.
- As for the general case, the "YES" entries are the same as in Table 1, because the methods for compacting Dalal's and Weber's operators can be iterated without leading to explosion of space.

Formalism	Iterated General case Logical equiv. (2)	Iterated General case Query equiv. (1)	Iterated Bounded case Logical equiv. (2)	Iterated Bounded case Query equiv. (1)
GFUV, Nebel	NO Th. 3.7	NO Th. 3.1	NO Th. 4.1	NO Th. 4.1
Winslett, Borgida	NO Th. 3.7	NO Th. 3.2	NO Th. 6.5	YES Cor. 6.4
Forbus	NO Th. 3.7	NO Th. 3.3	NO Th. 6.5	YES Cor. 6.4
Satoh	NO Th. 3.7	NO Th. 3.2	NO Th. 6.5	YES Cor. 6.4
Dalal	NO Th. 3.6	YES Th. 5.1	NO Th. 6.5	YES Th. 5.1
Weber	NO Th. 3.6	YES Cor. 5.2	NO Th. 6.5	YES Cor. 5.2
WIDTIO	YES –	YES –	YES –	YES –

Table 2: Is the iteratively revised knowledge base compactable?

- As for the bounded case:
  - when logical equivalence is considered, this case is similar to the single revision for the unbounded case, and non-compactability holds in all cases;
  - when query equivalence is considered, although the proofs of compactness for the single revision cannot be directly used, the methods have been adapted.

Tables 1 and 2 give also some indications on implementations and practical systems. Note that exponential space is needed to store the result of a revision $T * P$ in almost all cases. Hence, a reasonable strategy (as suggested by Winslett [27]) seems to be to delay revisions $P^1, \ldots, P^m$ and incorporate them when $T * P^1 * \cdots * P^m$ is accessed. Moreover, it is helpful to save the formulae $P^1, \ldots, P^m$ even after incorporation, for possible further revisions. In fact polynomiality in Table 2 is guaranteed only if all formulae are available.

7 Generalization and strengthening of results

Our results can be easily generalized in several directions. First of all, we can withdraw the assumption that the result of a revision must be a propositional formula. Such an assumption, which we made in the introduction, is motivated from both an epistemological and a practical point of view. However, suppose the result of revision can be a generic data structure which admits a polynomial-time algorithm for model checking; Definition 2.6 can be rephrased as follows:

Definition 7.1 (logically-compactable (data structure) operator) *An update or revision operator $*$ is* logically-compactable with a data structure *if and only if there exist two polynomials p_1 and p_2 and an algorithm ASK such that for any pair of propositional formulae T and P there exists a data structure D with the following properties:*

1. $|D| \leq p_1(|T| + |P|)$*;*
2. *for all interpretations M of $V(T) \cup V(P)$ the call $ASK(D, M)$ returns* `yes` *iff $M \models T * P$;*
3. *$ASK(D, M)$ requires time $\leq p_2(|M| + |D|)$.*

In the above definition the algorithm ASK ensures that the data structure D correctly represents the set of models of $T * P$, and can be used to perform model checking.

Theorem 7.1 *Let $*$ be a revision operator. Assume there exists a polynomial p such that, for each $n > 0$, there exists a pair of formulae T_n, P_n with the following properties:*

1. $|T_n| + |P_n| \leq p(n)$*;*
2. *for all $\pi \in$ 3-SAT$_n$, there exists an interpretation M_π of $T_n * P_n$ such that:*
   (a) *M_π can be computed from π in polynomial time;*
   (b) *$M_\pi \models T_n * P_n$ iff π is satisfiable.*

With the above hypothesis, if $*$ is logically-compactable **with a data structure***, then NP $\subseteq$ P/poly.*

As a consequence, our negative results also apply to all representations where the equivalent of model checking can be decided in polynomial time.

A similar modification can also be applied to Definition 2.4 of query-compactability and to Theorem 2.2.

In the previous sections we showed that, for many revision operators, it is very unlikely that the result can be expressed with a compact propositional formula. In particular, the existence of these representations would imply a collapse in the polynomial hierarchy, which most researchers in computational complexity consider to be very unlikely. However, *necessary and sufficient* conditions for the existence of compact representations can be obtained quite easily by applying the technique used in [6, Proof of Theorems 6 and 7]. In fact, Theorems 2.3 and 2.2 can be, more precisely, formulated as follows:

Theorem 7.2 *Let $*$ be a revision operator. Assume there exists a polynomial p such that, for each $n > 0$, there exists a pair of formulae T_n, P_n with the following properties:*

1. $|T_n| + |P_n| \leq p(n)$;
2. *for all $\pi \in$ 3-SAT$_n$, there exists an interpretation M_π of $T_n * P_n$ such that:*
   (a) *M_π can be computed from π in polynomial time;*
   (b) *$M_\pi \models T_n * P_n$ iff π is satisfiable.*

With the above hypothesis, $*$ **is logically-compactable if and only if NP $\subseteq$ NC^1/poly**.

We remind (cf. [17]) that the class NC^1 consists of all languages recognizable by log-space uniform families of Boolean circuits having polynomial size and depth $O(\log n)$. Since NC^1 $\subseteq$ NC^1/poly, NP $\not\subseteq$ NC^1/poly implies NP$\neq$ NC^1. Hence proving the unconditioned impossibility of logical compactability of a revision operator would be, by Theorem 7.2, at least as strong a result as proving NP$\neq$ NC^1.

Theorem 7.3 *Let $*$ be a revision operator. Assume there exists a polynomial p such that, for each $n > 0$, there exists a pair of formulae T_n, P_n with the following properties:*

1. $|T_n| + |P_n| \leq p(n)$;
2. *for all $\pi \in$ 3-SAT$_n$, there exists a formula Q_π such that:*
   (a) *Q_π can be computed from π in polynomial time;*
   (b) *$T_n * P_n \models Q_\pi$ iff π is satisfiable.*

With the above hypothesis, $*$ **is query-compactable if and only if NP $\subseteq$ coNP/poly**.

Using the above reasoning schema, Theorem 7.3 implies that if we are able to prove that a revision operator is unconditionally not query-compactable, then NP$\neq$ coNP.

8 Conclusions

When we are faced with the problem of representing and updating a large body of information, we must choose the most appropriate representation formalism and revision operator. Our analysis suggests that important aspects in the choice of a revision operator are its *compactability* properties. We presented several results about the size of a propositional theory T' representing the revision $T * P$ of a knowledge base. We considered both the query-equivalence criterion (1) and the more restrictive logical equivalence criterion (2). In particular, we proved that some formalizations of belief revision (e.g., Forbus' and Ginsberg-Fagin-Ullman-Vardi's) lead to propositional theories T' which are intrinsically not representable in polynomial space. Dalal's and Weber's formalizations have an interesting behavior: T' has no polynomial-sized representation if we insist on logical equivalence (2), but such a representation does exist if we ask only for query equivalence (1).

Furthermore, we investigated the impact of bounding the size of the revising formula P. Several operators (e.g., Winslett's and Satoh's) are logically compactable only in such a restricted case (cf. Table 1). We made another analysis about the impact of iterating the revision process an unbounded number of times. The analysis showed that many revision operators (e.g., Winslett's and Satoh's) become not logically compactable, although they remain query compactable (cf. Table 2).

We proved non-existence of polynomial-sized representations subject to non-collapse of the polynomial hierarchy. Anyway, proving unconditional non-existence of such representations is equivalent to solving some long-standing questions in computational complexity (cf. Section 7). Our results can also be generalized from propositional formulae to general data structures.

There are several lessons to be learned from this analysis.

- WIDTIO is a very drastical approach, but always results in a logically compactable formula.

- On the other hand, most belief revision operators have the undesired property that, in the worst case, it is not feasible to explicitly store the result of revising an existing knowledge base with a new formula.

- In particular, the syntax-based $*_{GFUV}$ operator has the worst behavior: it is uncompactable even for query equivalence and in the bounded case (a single literal is able to generate an exponential number of distinct possibilities).

- As for model-based operators, boundedness of P is a significant restriction: the revision involves only literals occurring in P, hence compactness is guaranteed, regardless of the equivalence criterion.

- Compactability and computational complexity of inference are different, although somehow related. In fact, from Eiter and Gottlob's work [8] it follows that two of the computationally more difficult operators are $*_{Web}$ and $*_{Wid}$; nevertheless, these operators admit compact representations in more cases than computationally simpler operators, such as $*_S$ and $*_{GFUV}$.

- Compactability is also not directly related to the selectivity of the operators in choosing the resulting set of models. As a matter of fact, the two model-based operators that more often admit compact representations are $*_D$ and $*_{Web}$. As shown in Figure 1, $*_D$ is the most selective operator while $*_{Web}$ is one of the least selective ones.

- The iterated bounded case is often similar to the single revision for the unbounded case.

- Delaying revisions $P^1, \ldots, P^k$ and incorporating them when $T * P^1 * \cdots * P^k$ is accessed seems to be a reasonable strategy. Moreover, it is helpful to save the formulae $P^1, \ldots, P^k$ even after incorporation, for possible further revisions.

References

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Transactions on Database Systems*, 6(4):557–575, 1981.

[3] R. Boppana and M. Sipser. The complexity of finite functions. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 14. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.

[4] A. Borgida. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems*, 10:563–603, 1985.

[5] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf. The size of a revised knowledge base. In *Proceedings of the Fourteenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'95)*, pages 151–162, 1995. Extended version available as Technical Report DIS 34-96, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", November 1996.

[6] M. Cadoli, F. M. Donini, M. Schaerf, and R. Silvestri. On compact representations of propositional circumscription. *Theoretical Computer Science*, 182:183–202, 1997.

[7] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988.

[8] T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.

[9] T. Eiter and G. Gottlob. The complexity of nested counterfactuals and iterated knowledge base revisions. *Journal of Computer and System Sciences*, 53(3):497–512, 1996.

[10] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *Proceedings of the Second ACM SIGACT SIGMOD Symposium on Principles of Database Systems (PODS'83)*, pages 352–365, 1983.

[11] K. D. Forbus. Introducing actions into qualitative simulation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1273–1278, 1989.

[12] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* Bradford Books, MIT Press, Cambridge, MA, 1988.

[13] P. Gärdenfors and H. Rott. Belief revision. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4*, pages 35–132. Oxford University Press, 1995.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, San Francisco, Ca, 1979.

[15] M. L. Ginsberg. Conterfactuals. *Artificial Intelligence*, 30:35–79, 1986.

[16] G. Gogic, C. Papadimitriou, and M. Sideri. Incremental recompilation of knowledge. *Journal of Artificial Intelligence Research*, 8:23–37, 1998.

[17] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.

[18] R. M. Karp and R. J. Lipton. Some connections between non-uniform and uniform complexity classes. In *Proceedings of the Twelfth ACM Symposium on Theory of Computing (STOC'80)*, pages 302–309, 1980.

[19] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, 1991.

[20] H. A. Kautz and B. Selman. Forming concepts for fast inference. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, pages 786–793, 1992.

[21] H. E. Kyburg Jr. Uncertainty logics. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3*, pages 397–438. Oxford University Press, 1994.

[22] P. Liberatore and M. Schaerf. The complexity of model checking for belief revision and update. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, pages 556–561, 1996.

[23] B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 417–428, 1991.

[24] B. Nebel. Base revision operations and schemes: Semantics, representation and complexity. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI'94)*, pages 341–345, 1994.

[25] K. Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88)*, pages 455–462, 1988.

[26] A. Weber. Updating propositional formulas. In *Proc. of First Conf. on Expert Database Systems*, pages 487–500, 1986.

[27] M. Winslett. *Updating Logical Databases.* Cambridge University Press, 1990.

[28] M. Winslett. Epistemic aspects of databases. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4*, pages 133–174. Oxford University Press, 1995.

[29] C. K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983.