How Hard is it to Revise a Belief Base?

Report 83

Bernhard Nebel
Albert-Ludwigs-Universiät Freiburg
Institut für Informatik
Am Flughafen 17
D-79110 Freiburg, Germany
e-mail: `nebel@informatik.uni-freiburg.de`
WWW: `http://www.informatik.uni-freiburg.de/~nebel`

August 1996

Abstract

If a new piece of information contradicts our previously held beliefs, we have to *revise* our beliefs. This problem of *belief revision* arises in a number of areas in Computer Science and Artificial Intelligence, e.g., in updating logical database, in hypothetical reasoning, and in machine learning. Most of the research in this area is influenced by work in philosophical logic, in particular by Gärdenfors and his colleagues, who developed the *theory of belief revision*. Here we will focus on the computational aspects of this theory, surveying results that address the issue of the computational complexity of belief revision.

Contents

1 Introduction **1**

2 The Theory of Belief Revision **2**
2.1 Preliminaries 3
2.2 Rationality Postulates 3
2.3 Partial Meet Revision Scheme 5
2.4 Cut Revision Scheme 6
2.5 Safe Revision Scheme 8

3 Computational Complexity Theory **8**
3.1 Polynomial vs. Exponential Runtime Requirements 9
3.2 Nondeterministic Computations and the Complexity Class NP . . 10
3.3 NP-Hardness and NP-Completeness 11
3.4 Complexity Classes above NP 12
3.5 Nonuniform Complexity Classes 14

4 Representationally Feasible Revision Schemes **15**
4.1 Model-Based Revision Schemes 16
4.2 Syntax-Based Revision Schemes 17
4.3 Analyzing the Computational Complexity of Base Revision Schemes 19
4.4 A Lower Bound for Representationally Feasible Revision Schemes 19
4.5 Full Meet Revision Scheme 20

5 Meet Base Revision Schemes **21**
5.1 Full Meet Base Revision 22
5.2 Prioritized Base Revision 23
5.3 Linear Base Revision 24
5.4 Cardinality-Maximizing Base Revision 26
5.5 Lexicographic Base Revision 27
5.6 Base Revision Operations: When in Doubt Throw it Out 28

6 Cut Base Revision Schemes **30**
6.1 Priority-Consistent Orderings and Ensconcements 30
6.2 Generating Ensconcement Orderings from Arbitrary Priorities . . 33
6.3 Cut vs. Linear Base Revision 34

7 Safe Base Revision Scheme **36**
7.1 Safe Base Revisions and WIDTIO-Revisions 36
7.2 Computational Complexity of Safe Base Revision 37

8 Generating Revised Belief Bases **37**
8.1 Do Compact Representations for Revised Belief Bases Exist? . . . 38
8.2 How Hard is it to Generate a Revised Base? 40

9 Related Work **41**
9.1 Belief Updates 41
9.2 Evaluating Counterfactual Conditionals 42
9.3 Nonmonotonic Logics 42

10 Summary and Discussion **43**
10.1 Summary of Complexity Results 43
10.2 The Degree of Syntax-Sensitivity of Base Revision Schemes 45
10.3 Discussion 47

Appendix A: Proofs **47**

Appendix B: Symbol Index **61**

Bibliography **63**

1 Introduction

If a new piece of information contradicts our previously held beliefs, we have to *revise* our beliefs. This problem of *belief revision*[1] arises in a number of areas in Computer Science and Artificial Intelligence, e.g., in updating logical database [Fagin *et al.*, 1983], in hypothetical reasoning [Ginsberg, 1986], and in machine learning [Wrobel, 1994]. Most of the research in this area is influenced by work in philosophical logic, in particular by Gärdenfors and his colleagues [Alchourrón *et al.*, 1985; Gärdenfors, 1988; Gärdenfors, 1992a], who developed the *theory of belief revision*. Here we will focus on the computational aspects of this theory, surveying results that address the issue of the computational complexity of belief revision.

The theory of belief revision gives a quite accurate characterization of the space of possible revision operations and analyzes how such revision operations can be generated by *revision schemes* that map a belief state and some extra-logical preference information to a revision operation. However, the theory does not give any indication of how to put belief revision to work in a computational setting. If one wants to apply this theory in a computer science or artificial intelligence application, there are two severe problems. First of all, it is assumed that belief states are modelled by logically closed sets of formulae, which seems to be representationally infeasible since logically closed sets are always infinite. In a computational framework, however, it is common practice to represent infinite entities by finite means, provided this is possible at all. In the following, we will assume that belief states can always be represented by finite sets of formulae in propositional logic, which are called *belief bases*.

Secondly, there is the problem that the extra-logical preference information is usually assumed to be a relation over the set of all formulae in a logically closed theory, a relation over all models of the theory, or a relation over all subsets of the theory. Even assuming that the logically closed theory is finite modulo logical equivalence, the necessary amount of information is still huge, usually exponential or double exponential in the size of the belief base. Hence for applications one is interested in revision schemes that require an amount of preference information that is feasible in the size of the belief base.

Some revision schemes, such as the *model-based revision schemes* [Katsuno and Mendelzon, 1991], do not need any explicit preference information, which is an advantage from a representational point of view, but also means that these schemes are quite *inflexible*. The revision operation is always uniquely determined by the logical contents (the set of models) of the belief base. Flexibility can be achieved when we start with the particular syntactic realization of a belief base and some extra information in order to generate a revision operation. Revi-

[1] The related problem of how to accommodate new information that is the result of a change in the world has been analyzed by Katsuno and Mendelzon [Katsuno and Mendelzon, 1992], who call the corresponding belief change operation *belief update*.

sion schemes defined following this idea are called *syntax-based revision schemes* [Nebel, 1994].

Having a representationally feasible revision scheme does not imply that it is *computationally feasible*, however. To the contrary, belief revision is computationally infeasible for most representationally feasible revision schemes—provided one accepts the assumption of *computational complexity theory* that any problem that is at least as hard as propositional satisfiability is computationally infeasible. Revision is, of course, as hard as deciding propositional satisfiability because satisfiability is a subproblem of all revision schemes. Whether revision is strictly harder than satisfiability and whether there are interesting special cases that may be easier, is not immediately evident, however. Further, it is not obvious, how large a revised belief base can become and how hard it is to generate a revised base. All these questions are addressed in this paper, giving a survey on previous results as well as presenting some new results.[2]

The structure of the rest of the paper is as follows. The next section introduces terminology and notation and sketches some relevant results of the theory of belief revision. Section 3 introduces the basic notions of complexity theory that are necessary to present the results in the rest of the paper. In Section 4, we discuss the two main approaches to representationally feasible revision schemes, describe how the complexity of revision schemes can be evaluated, establish a lower bound for representationally feasible revision schemes, and analyze the simplest possible revision scheme. In Sections 5–7 we consider the three main approaches to revising a belief base using syntax-based revision schemes and analyze the approaches from a computational complexity point of view. In addition to the general case, we also look at restrictions such as bounding the size of the revision formula and restricting the base logic to Horn logic. In Section 8, we study the question of how large a revised base can become and how hard it is to generate a revised base. In Section 9, some pointers to related work are provided. Finally, in Section 10, we summarize the results, evaluate the syntax-sensitivity of syntax-based revision schemes, and discuss further research directions.

2 The Theory of Belief Revision

Before we sketch the theory of belief revision, first some terminological and notational conventions are introduced in the next subsection. We then proceed by presenting rationality postulates for belief revision. Based on that, three belief revision schemes are introduced.

[2] All unpublished proofs are given in the appendix.

2.1 Preliminaries

Throughout this paper, a propositional language $\mathcal{L}$ over a *finite* alphabet Σ of propositional variables $p, q, r \ldots$ with the usual logical connectives ($\neg$, $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$) is assumed. Propositional formulae are denoted by $\varphi, \psi, \chi, \omega, \ldots$, constant truth by $\top$, and its negation by $\bot$. An **atom** or **atomic formula** is a formula consisting of a propositional variable only. A **literal** is an atom or a negated atom. A **clause** is a disjunction of literals. A formula in **conjunctive normal-form** is a formula that is a conjunction of clauses. **Truth assignments**, denoted by α and β, assign one of the values *true* and *false* to the variables in Σ. A truth assignment **satisfies** a formula iff the formula evaluates to *true* under the assignment. A set A of formulae is satisfied by α iff all formulae in A are satisfied by α. A satisfying truth-assignment is also called **model** of the formula or the set of formulae. A formula φ is said to be **logically implied** by a set of formulae A, symbolically $A \models \varphi$, iff all models of A are models of φ as well.

Sets of propositional formulae are denoted by $K, L, M, \ldots$ and $A, B, C, \ldots$ The **size of a formula** φ (counting all symbols in the formula) is denoted by $|\varphi|$ and the **size of a set of formulae** A by $|A|$, where the size of a set of formulae is defined as the sum over the sizes of the formulae in the belief base. This is to be distinguished from the **cardinality** of a set A, denoted by $\|A\|$, where only the number of elements in A is relevant.

The **logical closure** of a set of formulae A, symbolically $Cn(A)$ is defined as follows:

$$Cn(A) \stackrel{\text{def}}{=} \{\varphi \in \mathcal{L} |\ A \models \varphi\}. \tag{1}$$

Instead of $Cn(\{\varphi\})$, we will also write $Cn(\varphi)$. Logically closed sets of propositional formulae, i.e., $K = Cn(K)$, are denoted by the capital letters $K, L, M \ldots$ and are called **belief sets**. The monotonic addition of a propositional formula φ to a belief set K, i.e., $Cn(K \cup \{\varphi\})$, is denoted by $K + \varphi$ and called **expansion** of K by φ. Arbitrary sets of formulae are called **belief bases** and are denoted by capital letters from the beginning of the alphabet. Systems of belief bases and belief sets are denoted by S. Finite belief bases C are often identified with the conjunction of all formulae $\bigwedge C$. If $S = \{A_1, \ldots, A_n\}$ is a finite family of finite belief bases, then $\bigvee S$ shall denote a formula logically equivalent to $(\bigwedge A_1) \vee \ldots \vee (\bigwedge A_n)$. As usual, we set $\bigvee \emptyset = \bot$.

2.2 Rationality Postulates

Gärdenfors and his colleagues [Alchourrón *et al.*, 1985; Gärdenfors, 1988] considered mainly two change operations on belief sets, namely, *contraction* and *revision*. **Contraction** is the removal of a formula φ from a belief set K resulting in a new belief set, denoted by $K \dot{-} \varphi$, that does not contain φ (if φ is not a tautology), and **revision** is the addition of a formula φ to K, denoted by $K \dot{+} \varphi$, such that $Cn(\bot) \neq K \dotplus \varphi$ whenever $\not\models \neg\varphi$. Although contraction and revision are not uniquely determined operations—the only commonly agreed criterion is that the changes to the original belief sets have to be *minimal*—it is possible to constrain the space of reasonable belief change operations. Gärdenfors proposed the following set of **rationality postulates for revision operations**:

($\dotplus$1) $K \dotplus \varphi$ is a belief set;

($\dotplus$2) $\varphi \in K \dotplus \varphi$;

($\dotplus$3) $K \dotplus \varphi \subseteq K + \varphi$;

($\dotplus$4) If $\neg\varphi \notin K$, then $K + \varphi \subseteq K \dotplus \varphi$;

($\dotplus$5) $K \dotplus \varphi = Cn(\bot)$ only if $\models \neg\varphi$;

($\dotplus$6) If $\models \varphi \leftrightarrow \psi$ then $K \dotplus \varphi = K \dotplus \psi$;

($\dotplus$7) $K \dotplus (\varphi \wedge \psi) \subseteq (K \dotplus \varphi) + \psi$;

($\dotplus$8) If $\neg\psi \notin K \dotplus \varphi$, then $(K \dotplus \varphi) + \psi \subseteq K \dotplus (\varphi \wedge \psi)$.

These postulates intend to capture the intuitive meaning of minimal change—from a logical point of view [Alchourrón *et al.*, 1985; Gärdenfors, 1988; Gärdenfors, 1992b; Gärdenfors and Rott, 1995]. The first six postulates, which are straightforward, are called **basic postulates**. The postulates ($\dotplus$7)–($\dotplus$8), which are called **supplementary postulates**, are less obvious. They capture the idea that a revision of K by a conjunction $(\varphi \wedge \psi)$ should be achieved through a revision by φ and an expansion by ψ, if this is possible at all, i.e., if ψ is consistent with $K \dotplus \varphi$.

In the context of belief base revision, additional postulates turn out to be important [Rott, 1993; Gärdenfors and Rott, 1995; Katsuno and Mendelzon, 1991]:

($\dotplus$8r) $K \dotplus (\varphi \vee \psi) \subseteq Cn(K \dotplus \varphi \cup K \dotplus \psi)$;

($\dotplus$8c) If $\psi \in K \dotplus \varphi$ then $K \dotplus \varphi \subseteq K \dotplus (\varphi \wedge \psi)$;

($\dotplus$8i) If $\psi \in K \dotplus \varphi$ and $\varphi \in K \dotplus \psi$ then $K \dotplus \varphi = K \dotplus \psi$.

Postulate ($\dotplus$8r) (called R8 in [Katsuno and Mendelzon, 1991]) constrains the revisions by disjunctions. In the presence of the basic postulates, ($\dotplus$8r) can be derived from ($\dotplus$8) but is not equivalent to it. ($\dotplus$8c) is a weakened form of ($\dotplus$8), and ($\dotplus$8i) (called R7 in [Katsuno and Mendelzon, 1991]) is a condition on the equality of revised belief bases. In the presence of ($\dotplus$1)–($\dotplus$7), the two latter postulates are equivalent, so we will only consider ($\dotplus$8c) in the following.

There exists a similar set of postulates for contraction operations, which is equivalent to this set using the fact that revision and contraction are interdefinable by the **Harper** identity [Gärdenfors, 1981]

$$K \dot{-} \varphi \quad = \quad (K \dotplus \neg\varphi) \cap K \tag{2}$$

and the **Levi** [1977] identity

$$K \dotplus \varphi \;=\; (K \dot{-} \neg\varphi) + \varphi, \tag{3}$$

provided the basic postulates are satisfied.

One interesting point to note is that the postulates do not constrain the operators with respect to varying belief sets. In other words, when we talk about a revision operator, we may restrict ourselves to a given belief set and consider the mapping from $\mathcal{L}$ (the new information) to $2^{\mathcal{L}}$ (the revised belief set)—called **local revision** in [Hansson, 1996]. In fact, in the rest of the paper we only consider such local revisions.

Based on this framework, one can consider different methods to construct revision operations. Such methods are usually specified as operations that map a belief set with associated preference information and a revision formula to a new belief set. Assuming that the preference information is expressed using an element from the set $\mathcal{P}$, such an operator, which we call **revision scheme**, is a mapping from $2^{\mathcal{L}} \times \mathcal{P} \times \mathcal{L}$ to $2^{\mathcal{L}}$. In other words, such a revision scheme can be viewed as a recipe for generating a belief revision operation on a given belief set with given preference information. If $\mathcal{S}$ is a revision scheme, then the generated belief revision operations are called $\mathcal{S}$ **revisions**.

2.3 Partial Meet Revision Scheme

In [Alchourrón *et al.*, 1985], so-called partial meet revisions are investigated. This notion is based on systems of maximal (w.r.t. to set-inclusion) subsets of a given belief set K that do not imply φ, called the **remainders of** K **by** φ and written as $K \perp \varphi$:

$$K \perp \varphi \;\stackrel{\text{def}}{=}\; \{L \subseteq K \mid L \not\models \varphi, \forall M \colon L \subset M \subseteq K \Rightarrow M \models \varphi\}. \tag{4}$$

A **partial meet revision** (on K for all φ) is defined by a **selection function** γ that selects a nonempty subset of $K \perp \neg\varphi$ (provided $K \perp \neg\varphi$ is nonempty, $\{K\}$ otherwise) in the following way:[3]

$$K \dotplus \varphi \;\stackrel{\text{def}}{=}\; \Big(\bigcap \gamma(K \perp \neg\varphi)\Big) + \varphi. \tag{5}$$

Such partial meet revisions satisfy unconditionally the first six postulates. Furthermore, it is possible to show that all revision operations satisfying the basic postulates are partial meet revisions [Alchourrón *et al.*, 1985, Observation 2.5]. Actually, this and the other results cited below were proven for contraction. However, as mentioned above, the postulates for revision and contraction are equivalent if one employs the Harper and Levi identities.

[3]Note that all elements of $K \perp \neg\varphi$ are belief sets and that the intersection of belief sets is a belief set again.

If some constraints are placed on the behavior of the selection function, it is possible to show that ($\dot{+}7$) and ($\dot{+}8$) are also satisfied. The key notion in this context is *relationality* of the selection function, which means that there must exist a so-called **"marking off" relation** $\leq\cdot$ over all subsets of a belief set K independent of φ such that the selection functions always "marks off" the "best" sets, i.e., for all φ,

$$\gamma(K\perp\varphi) \quad = \quad \{L \in (K\perp\varphi) \,|\, \forall M \in (K\perp\varphi) \colon M \leq\cdot L\}, \tag{6}$$

in which case the revision operation is called **relational partial meet revision**. Any such revision satisfies the postulates ($\dot{+}1$)–($\dot{+}7$) [Alchourrón *et al.*, 1985, Observations 3.1, 4.2, and 4.3]. If the relation is additionally transitive, then the revision operation is called **transitively relational** and ($\dot{+}1$)–($\dot{+}8$) are satisfied. Furthermore, it can be shown that any revision operation satisfying ($\dot{+}1$)–($\dot{+}8$) is a transitively relational partial meet revision [Alchourrón *et al.*, 1985, Corollary 4.5].

Rott [1993] showed additional representation theorems for belief sets that are *finite* (modulo Cn). He demonstrated that relational partial meet revisions correspond to revisions satisfying ($\dot{+}1$)–($\dot{+}7$) and ($\dot{+}8$r). Further, using the notion of **negatively transitive relation**, which is a relation such that $a \not\leq b$ and $b \not\leq c$ implies $a \not\leq c$, he showed that **negatively transitive revisions** coincide with the revisions satisfying ($\dot{+}1$)–($\dot{+}7$), ($\dot{+}8$r), and ($\dot{+}8$c).[4]

It should be noted that some special cases of partial meet revisions are unreasonable. If γ always selects all of the elements of $K\perp\neg\varphi$, leading to the so-called **full meet revision**, denoted by $\overset{F}{+}$, then $K \overset{F}{+} \varphi = Cn(\varphi)$ if $\neg\varphi \in K$. This means we throw away all the old beliefs if the new formula is inconsistent with the belief set, which is clearly unreasonable. Although unreasonable, full meet revision is "fully rational" in the sense that it satisfies all the rationality postulates, as is easy to verify. Another unreasonable partial meet revision is **maxichoice revision**, in which the selection function always selects a singleton set. This revision leads to complete belief sets (i.e., belief sets such for each formula φ either $\varphi \in K$ or $\neg\varphi \in K$), even if the original belief set was not complete.

2.4 Cut Revision Scheme

Instead of providing the preference information by a selection function, one may also think of relations over formulae. **Epistemic entrenchment** orderings, written as $\varphi \preceq \psi$, are defined over the entire set of formulae $\mathcal{L}$ and have to satisfy the following properties:[5]

[4] A similar result has been proven by Katsuno and Mendelzon [1991] in a model-theoretic framework.

[5] As shown by Dubois and Prade [Dubois and Prade, 1991], epistemic entrenchments are the qualitative counterparts to necessity measures in possibilistic logic [Dubois *et al.*, 1994].

($\preceq$1) If $\varphi \preceq \psi$ and $\psi \preceq \chi$, then $\varphi \preceq \chi$.

($\preceq$2) If $\varphi \models \psi$, then $\varphi \preceq \psi$.

($\preceq$3) For any φ, ψ, $\varphi \preceq (\varphi \wedge \psi)$ or $\psi \preceq (\varphi \wedge \psi)$.

($\preceq$4) When $K \neq Cn(\bot)$, then $\varphi \notin K$ iff $\varphi \preceq \psi$ for all $\psi \in \mathcal{L}$.

($\preceq$5) If $\psi \preceq \varphi$ for all $\psi \in \mathcal{L}$, then $\models \varphi$.

From ($\preceq$1)–($\preceq$3), it follows immediately that $\preceq$ is a total preorder[6] over $\mathcal{L}$ respecting logical equivalence. The strict part of $\preceq$ is denoted by $\prec$.

Using such relations, Gärdenfors and Makinson define **belief contraction generated by epistemic entrenchments**, written $K \stackrel{\epsilon}{-} \varphi$, by

$$\psi \in K \stackrel{\epsilon}{-} \varphi \text{ iff } \psi \in K \text{ and } (\varphi \prec (\varphi \vee \psi) \text{ or } \models \varphi) \tag{7}$$

and show that such a belief contraction operation satisfies all rationality postulates for contraction as well as the following condition [Gärdenfors and Makinson, 1988, Theorem 4]:

$$\varphi \preceq \psi \quad \text{iff} \quad \varphi \notin K \stackrel{\epsilon}{-} (\varphi \wedge \psi) \text{ or } \models (\varphi \wedge \psi). \tag{8}$$

Further, they show that any belief contraction operation satisfying all of the rationality postulates is generated by some epistemic entrenchment ordering [Gärdenfors and Makinson, 1988, Theorem 5].

Using the Levi identity, it is possible to define a belief revision operation using an epistemic entrenchment ordering. In this case, however, the condition in (7) can be slightly simplified. Instead of testing $\varphi \prec (\varphi \vee \psi)$ it suffices to check for $\varphi \prec \psi$ [Rott, 1991b]. More formally, let $cut_{\prec}$ be defined as

$$cut_{\prec}(\varphi) \stackrel{\text{def}}{=} \{\psi \in \mathcal{L} |\, \varphi \prec \psi\}, \tag{9}$$

then a revision operation based on $\preceq$ can be defined by

$$K \stackrel{\epsilon}{+} \varphi \stackrel{\text{def}}{=} cut_{\prec}(\neg\varphi) + \varphi. \tag{10}$$

This operation is identical to a revision operation obtained by first contracting K using $\stackrel{\epsilon}{-}$ and then applying the Levi identity.

Since revisions based on epistemic entrenchment amount to *cutting away* all formulae with an epistemic entrenchment lower than the negation of the formula to be added, we call this kind of revision scheme **cut revision scheme**.

[6] A **preorder** is a transitive and reflexive relation. A preorder $\leq$ is **total** if for all pairs of elements a, b we have $a \leq b$ or $b \leq a$.

2.5 Safe Revision Scheme

Yet another revision scheme is the **safe revision scheme** introduced by Alchourrón and Makinson [1985]. It is in some sense dual to partial meet revision because we do not consider the inclusion-maximal subsets consistent with a formula but the inclusion minimal sets that are inconsistent with a formula. Formally, a subset B of a belief set K is called an **entailment set for** φ iff $B \models \varphi$ and for all subsets $C \subset B$: $C \not\models \varphi$. Clearly, removing one element from each entailment set leads to consistency with the formula to be added. However, which formulae should be removed?

Alchourrón and Makinson [1985] postulate a *strict partial order*[7] $\prec$ over a belief base K that respects logical equivalence, i.e., if $\models \varphi \leftrightarrow \varphi'$ and $\models \psi \leftrightarrow \psi'$ then $\varphi \prec \psi$ iff $\varphi' \prec \psi'$. Such a relation is called **hierarchy** over K. Given such a hierarchy, an element $\varphi \in K$ is called **safe with respect to** ψ iff φ is not a minimal element under $\prec$ in any entailment set for ψ.

The set of all elements of a belief set K that are safe with respect to φ are denoted by K/φ. **Safe revision** is then defined by

$$K \dot{+} \varphi \stackrel{\text{def}}{=} Cn(K/\neg\varphi) + \varphi. \tag{11}$$

Revision operations constructed in this way always satisfy the basic rationality postulates [Alchourrón and Makinson, 1985, Observation 3.2]. Further properties of safe revisions can be shown if some constraints are placed on the hierarchy. Since these results are not relevant for the rest of the paper, we only note that another paper by Alchourrón and Makinson [1986] and a paper by Rott [1992] contain interesting results on this issue.

3 Computational Complexity Theory

In computational complexity theory[8] one tries to classify problems according to their requirements on computational resources (time, memory) depending on the size of the input. In this context, the term **problem** means, contrary to the ordinary meaning, "generic question," which has different **instances**. For example, a particular crossword puzzle is not a problem, but an instance of the problem of crossword puzzle solving. In our context, deciding propositional implication (i.e., $A \models \varphi$) is a *problem*, while finding out whether $\{p \vee q, \neg p\} \models q$ is an *instance* of this problem.

[7] A **strict partial order** is a transitive and irreflexive relation.

[8] Good text books on this topic are [Balcázar *et al.*, 1988; Balcázar *et al.*, 1990; Garey and Johnson, 1979; Papadimitriou, 1994; Wagner and Wechsung, 1988].

3.1 Polynomial vs. Exponential Runtime Requirements

When we denote the size of an instance I by $|I|$, and if f is some monotonically increasing function from positive integers to positive integers, we say that the **computation time of a problem is bounded by** f (on a particular machine T) if for all instances no more than $f(|I|)$ computation steps are necessary. Since in most cases we are not interested in constant factors, usually the **big-O notation** is used. Let f and g be functions and c be a positive constant, then $O(f)$ is defined to be the class of functions g such that

$$g(n) \leq c \times f(n) \tag{12}$$

for all but finitely many positive integers n. By abusing notation, one also says that a computation needs $O(f)$ time if the time is bounded by a function g such that $g \in O(f)$. More generally, one defines the computation time to be **polynomially bounded** if there is a fixed k such that the computation time is bounded by $O(n^k)$.

A problem is considered to be **efficiently solvable** if for all instances of the problem an algorithm can find the correct answer in a number of computation steps that is *polynomially bounded*—assuming a "traditional" model of computation, i.e., a *sequential, deterministic* computation model, e.g., Turing machines or random access machines. It should be noted that the definition of polynomially bounded computation is independent from the particular machine model since all sequential, deterministic machines can be simulated on each other in polynomial time.

In computational complexity theory, one usually restricts the attention to so-called **decision problems**, i.e., problems that have only "yes" and "no" as possible answers. These problems are also viewed as *formal languages*, denoted by $P, Q, \ldots$, formed by the "yes" instances. While this might seem to be a serious restriction at first sight, it turns out that in most cases a polynomial algorithm for a decision problem can be easily transformed into a polynomial algorithm for the corresponding general problem.

The class of decision problems that can be characterized by being solvable in polynomial time is denoted by P. Decision problems not belonging to this class, e.g., problems that need exponential time, are considered to be not efficiently solvable. The reason for this judgement is that the growth rate of exponential functions leads to astronomical runtime requirements even for moderately sized instances.

The distinction between polynomial and exponential runtime requirements becomes even more vivid if one considers the effects of further advances in computer technology. Assuming we have a problem P_1 that needs n^2 steps and that can be solved in reasonable time, e.g. one minute, up to an instance size of m_1, then an increase of speed by 10^6 leads to the effect that instances up to a size of $1000 \times m_1$ can be solved in one minute. For problem P_2 that needs 2^n steps the picture is quite different, however. Assuming that instances up to a size m_2 can be solved with the traditional technology in reasonable time, an increase by 10^6 in speed leads only to $20 + m_2$ as the maximal instance size.

Although the distinction between "efficiently" and "not efficiently" solvable problems according to whether the number of computation steps can be polynomially bounded or not seems to be reasonable, one should always keep in mind that it is only a mathematical abstraction. It is based on the assumption that in case of polynomial runtime the exponent is small, and that in the case of exponential runtime the worst case occurs significantly often. As experience shows, however, these assumptions are valid for most naturally occurring problems.

In order to find out whether a problem is efficiently solvable or not, we only have to find a polynomial algorithm or to prove that such an algorithm is impossible. However, there exist a large number of problems for which no polynomial algorithms are known, but at the same time it seems to be impossible to prove that super-polynomial time is necessary to solve these problems. The formal classification of these problems is one of the challenges of complexity theory.

3.2 Nondeterministic Computations and the Complexity Class NP

One formal tool to characterize these problems is the **nondeterministic Turing machine**. Such a machine can choose nondeterministically among different successor states during its computation and it accepts an input (answers "yes") if, and only if, there exists a sequence of nondeterministic choices that leads to an accepting state. A nondeterministic Turing machine accepts a language P in polynomial time if, and only if, all words of P are accepted using only a polynomial number of computation steps on the nondeterministic machine. The class of languages (or decision problems) that are accepted by nondeterministic Turing machines using polynomial time is called NP.

Another perspective on problems in NP is that these are the problems for which *short proofs* (i.e., of polynomial size) exist, where a proof in this context is something that allows us to verify easily (i.e., in polynomial time) that a given instance belongs to the "yes" instances. For example, the problem of deciding satisfiability of a propositional formula, denoted by SAT, is a problem in NP because truth-assignments are proofs in this context. They are short and it only takes polynomial time to verify that a truth-assignment satisfies a formula.

Since all deterministic machines can be viewed as nondeterministic machines, it follows that $\mathsf{P} \subseteq \mathsf{NP}$. Whether the converse inclusion holds is an open problem, however. This is the famous $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ problem.

Although we do not know whether $\mathsf{P} \neq \mathsf{NP}$, it is nevertheless possible to identify the "hardest" problems in NP. The formal tool for doing so is the *resource-limited reduction* between problems. A problem P can be **polynomially many-**

one reduced to problem Q, symbolically $P \leq_m^p Q$, iff there exists a function f from strings to strings that can be computed in polynomial time and that has the property that $w \in P$ if and only if $f(w) \in Q$. Intuitively, an algorithm for problem Q can be used to solve problem P with only polynomial overhead. In other words, Q must be at least as hard as P with respect to solvability in polynomial time.

3.3 NP-Hardness and NP-Completeness

A problem that has the property that all problems in NP can be polynomially reduced to it is obviously at least as hard as all problems in NP, i.e., it is **NP-hard** with respect to polynomial many-one reductions.

Since it is very difficult to prove something by quantifying over the entire class of languages in NP, it would be convenient to identify a problem P that is in NP and NP-hard. Because of the transitivity of $\leq_m^p$ it then suffices to reduce P to Q in order to show NP-hardness of Q.

Problems that are NP-hard and in NP are called **NP-complete**. More generally, a problem is called **X-complete** for a complexity class X if it is in X and X-hard. Although it is not obvious that NP-complete problems exist, it turns out that a large number of natural problem for which we do not know polynomial algorithms are NP-complete.[9] NP-complete problems have the interesting property that a polynomial, deterministic algorithm for one of these problems implies that all problems in NP can be solved in polynomial time on a deterministic machine, i.e., it implies that P = NP. Since all attempts of finding polynomial algorithms for NP-complete problems have failed so far, the proof that a problem is NP-hard implies that no efficient algorithm is known for this problem according to the current state of the art. Further, because of the large number of unsuccessful attempts to find efficient methods for solving NP-complete problems in polynomial time it is nowadays *believed* that it is impossible to solve such problems in polynomial time.

The prototypical example of an NP-complete problem is SAT. As mentioned above, this problem is in NP because there exist short proofs for demonstrating that a given instance is a "yes" instance—a satisfiable formula. Showing that SAT is also NP-hard is much more difficult. The proof is based on a *generic reduction* that assigns to each pair formed by a nondeterministic Turing machine T and an instance I a boolean formula φ_I that is satisfiable if, and only if, I is accepted by T in polynomial time.

The problem that is complementary to SAT, i.e., the problem to decide whether a boolean formula is unsatisfiable, is called UNSAT. Interestingly, UNSAT is not necessarily in NP, since no nondeterministic algorithm is known that accepts the corresponding language in polynomial time. Alternatively, we know

[9] Garey and Johnson [1979] provide a list of approximately 300 NP-complete problems.

of no proof system that guarantees short proofs for demonstrating that a formula is unsatisfiable.

UNSAT and all other problems complementary to problems in NP are assigned to the class coNP defined as

$$\mathsf{coNP} \stackrel{\text{def}}{=} \{P \mid \overline{P} \in \mathsf{NP}\}, \tag{13}$$

and it is conjectured that $\mathsf{NP} \neq \mathsf{coNP}$. Obviously, UNSAT is a coNP-complete problem under the definitions given above. Similarly, propositional implication (IMPL) and propositional tautology (TAUT) are also coNP-complete because UNSAT, IMPL, and TAUT can be polynomially many-one reduced to each other.

3.4 Complexity Classes above NP

In addition to P and NP, there exist a number of other so-called complexity classes,[10] some of them below P and others above NP. PSPACE, for instance, is the class of problems that can be solved in *polynomial space* on deterministic sequential machines. While it is relatively easy to show that $\mathsf{NP} \subseteq \mathsf{PSPACE}$, it is unknown whether NP is a proper subset of PSPACE. As in the P versus NP case, however, it is believed that NP is indeed a proper subset of PSPACE.

Between NP and PSPACE there exists an infinite hierarchy of complexity classes, called the **polynomial hierarchy**, denoted by PH. Because the computational problems considered in this paper fall into complexity classes located at the lower end of this hierarchy, the hierarchy is briefly sketched [Garey and Johnson, 1979, Sect. 7.2].

Let X be a class of decision problems. Then P^{X} denotes the class of decision problems P that can be decided in polynomial time by a deterministic Turing machine T that is allowed to use a procedure (a so-called **oracle**) for deciding a problem $Q \in \mathsf{X}$, whereby executing the procedure does only cost constant time. Similarly, NP^{X} denotes the class of decision problems P such that there is a nondeterministic Turing-machine that solves all instances of P in polynomial time using an oracle for $Q \in \mathsf{X}$. Based on these notions, the sets Δ_k^p, Σ_k^p, and Π_k^p are defined as follows:[11]

$$\begin{aligned}
\Delta_0^p &\stackrel{\text{def}}{=} \Sigma_0^p = \Pi_0^p = \mathsf{P}, &(14)\\
\Delta_{k+1}^p &\stackrel{\text{def}}{=} \mathsf{P}^{\Sigma_k^p}, &(15)\\
\Sigma_{k+1}^p &\stackrel{\text{def}}{=} \mathsf{NP}^{\Sigma_k^p}, &(16)\\
\Pi_{k+1}^p &\stackrel{\text{def}}{=} \mathsf{co}\Sigma_{k+1}^p. &(17)
\end{aligned}$$

[10] Johnson [1990] gives a survey of the state of the art.

[11] The superscript p is only used to distinguish these sets from the analogous sets in the Kleene hierarchy.

Thus, $\Sigma_1^p = \mathsf{NP}$ and $\Pi_1^p = \mathsf{coNP}$. Further note that $\mathsf{PH} = \bigcup_{k\geq 0} \Delta_k^p = \bigcup_{k\geq 0} \Sigma_k^p = \bigcup_{k\geq 0} \Pi_k^p \subseteq \mathsf{PSPACE}$. As with other classes, it is unknown whether the inclusions between the classes are proper. However, it is strongly believed that this is the case, i.e., that the hierarchy is truly infinite.

The role of the "canonical" complete problem (w.r.t. polynomial many-one reducibility), which is played by SAT for Σ_1^p, is played by k-QBF for Σ_k^p. k-QBF is the problem of deciding whether the following quantified boolean formula is true:

$$\underbrace{\exists \vec{p}\,\forall \vec{q} \ldots}_{k \text{ alternating quantifiers starting with } \exists} \varphi(\vec{p}, \vec{q}, \ldots). \tag{18}$$

The complementary problem, denoted by $\overline{k{-}\mathsf{QBF}}$, is complete for Π_k^p.

For problems in Δ_2^p, it is often difficult to determine their exact complexity, i.e., they cannot be shown to be complete for Δ_2^p. Restricting the number of oracle calls and the way the results of these calls can be used are, however, a useful tool for defining complexity classes inside Δ_2^p, which turn out to be helpful for determining the exact complexity of problems in Δ_2^p. Most notably, the class of problems that can be decided in polynomial time by using only logarithmically many oracle calls $\Delta_2^p[O(\log n)]$ [Wagner, 1987] (also denoted by $\mathsf{P}^{\mathsf{NP}[O(\log n)]}$ and Θ_2^p) plays an important role. Further, inside $\Delta_2^p[O(\log n)]$ are the classes of the so-called *boolean hierarchy* [Cai *et al.*, 1988]. The classes $\mathsf{NP}(k)$ and $\mathsf{coNP}(k)$ (also denoted by BH_k and coBH_k, respectively) of the **boolean hierarchy** BH can be defined as follows [Cai *et al.*, 1988]:

$$\mathsf{NP}(0) \stackrel{\text{def}}{=} \mathsf{P}, \tag{19}$$

$$\mathsf{NP}(2k+1) \stackrel{\text{def}}{=} \{P \cup Q \mid P \in \mathsf{NP}(2k), Q \in \mathsf{NP}\}, \tag{20}$$

$$\mathsf{NP}(2k+2) \stackrel{\text{def}}{=} \{P \cap Q \mid P \in \mathsf{NP}(2k+1), Q \in \mathsf{NP}\}, \tag{21}$$

$$\mathsf{coNP}(k) \stackrel{\text{def}}{=} \{P \mid \overline{P} \in \mathsf{NP}(k)\}. \tag{22}$$

The boolean hierarchy BH is the union over all sets $\mathsf{NP}(k)$ and turns out to be equivalent to the class of problems that can be solved in deterministic polynomial time using a *constant* number of oracle calls.

In Figure 1, all the complexity classes introduced in this section are displayed with their obvious inclusion relationships depicted by arrows. As mentioned above, it is not known whether the inclusion relationships are strict, but it is strongly believed.

From a purely practical point of view, it may not seem to be necessary to distinguish between complexity classes above NP, since all the problems that are complete for those classes are NP-hard and not efficiently solvable for this reason. In particular, all problems in the boolean and polynomial hierarchy have the same property as the NP-complete problems, namely, that they can be solved in polynomial time if and only if $\mathsf{P} = \mathsf{NP}$. Further, all these problems can be solved by an exhaustive search that takes exponential time. This means the

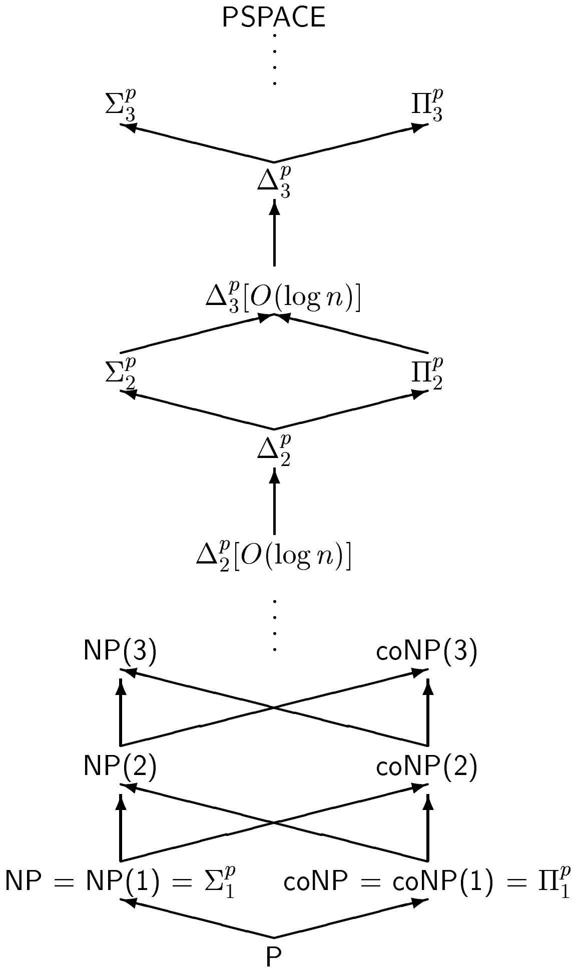

Figure 1: Hierarchy of complexity classes

worst-case behavior of any algorithm for a problem that is complete for some level of the boolean or polynomial hierarchy is most probably not better or worse than the worst-case behavior of any propositional proof method. Nevertheless, a precise classification of a problem in the hierarchy can give us hints about the relative difficulty of a problem, about the sources of combinatorics it contains, about possible ways to restrict the problem in order to make it polynomial, and about whether there are ways to solve aspects of the problem algorithmically.

3.5 Nonuniform Complexity Classes

If we are interested in how much space we need to represent something, so-called *non-uniform complexity classes* can help to prove lower bounds. For example, if we want to know how large a revised belief base could be in the worst case, it turns out that the naive representation can be exponential in some cases. In order to answer the question how much space the most compact representation needs, nonuniform classes are useful, which are defined using *advice-taking machines*.

An **advice-taking Turing machine** is a Turing machine with an **advice oracle**, which is a (not necessarily recursive) function a from positive integers to bit strings. On input I, the machine loads the bit string $a(|I|)$ and then continues as usual. Note that the oracle derives its bit string only from the length of the input and not from the contents of the input. An advice is said to be polynomial if the oracle string is polynomially bounded by the instance size. Further, if X is a complexity class defined in terms of resource-bounded machines, e.g., P or NP, then $\mathsf{X/poly}$ (also called **non-uniform** X) is the class of problems that can be decided on machines with the same resource bounds and polynomial advice.

Because of the advice oracle, the class $\mathsf{P/poly}$ appears to be much more powerful than P. However, it seems to be unlikely that $\mathsf{P/poly}$ contains all of NP. More precisely, it has been shown that $\mathsf{NP} \subseteq \mathsf{P/poly}$ would imply that the polynomial hierarchy collapses at Σ_2^p [Karp and Lipton, 1980], which is considered to be quite unlikely. In addition, Yap [1993] showed that the polynomial hierarchy collapses at Σ_{k+2}^p if $\Sigma_k^p \subseteq \Pi_k^p/\mathsf{poly}$ or if $\Pi_k^p \subseteq \Sigma_k^p/\mathsf{poly}$.

In Section 8, these conditional results are used to demonstrate that assuming that a revised belief base can always be represented in a compact way leads to the consequence that the polynomial hierarchy collapses [Cadoli *et al.*, 1995]. In other words, it appears to be unlikely that it is possible to find a compact form for revised belief bases in general.

4 Representationally Feasible Revision Schemes

In Section 1, we stated the requirement that a revision scheme should be *representationally feasible*. This means that the revision scheme should be satisfied with an amount of preference information that has a size bounded polynomially in the size of the belief base. The belief revision schemes described in Section 2 obviously do not satisfy this requirement. An epistemic entrenchment ordering, for example, has a size that is double exponential in the size of the belief base. Even if we limit the orderings to all maximal disjunctions in the language—from which the entire ordering can be computed [Gärdenfors and Makinson, 1988]—the size would be still exponential.

In general, two methods have been used to construct representationally feasible revision schemes. The first one is to assume that no preference information at all is given, i.e., the revision operation is solely determined by the logical contents of the belief base. One example for such a scheme is full meet revision. Other, more reasonable schemes, are based on identifying models of the revision formula that are "close" to the models of the belief base, so-called *model-based revision schemes* [Katsuno and Mendelzon, 1991]. The second method to construct representationally feasible revision schemes is to attach preference information to the formulae in the base by giving a preference relation over the formulae or by partitioning the base into priority classes. Since in this case the generated revision operation depends not only on the logical contents of the belief base but also on the syntactic form of the base, such schemes are called *syntax-based revision schemes* [Nebel, 1994].

4.1 Model-Based Revision Schemes

The main idea for model-based revision schemes, which are generally denoted by $\circ$, possibly with a subscript, is that in order to revise a belief base A by a formula φ, we select those models from φ that are "closest" or "most similar" to models in A and regard the resulting set of models as the revised base.

There are a number of proposals that differ according to how distance between models is measured and whether we take for each model α of A all the closest models from φ, or if we select only the models α from φ that are closest to all models of A, where the distance between one model of φ and all models of A is the minimal distance between α and some model of A.

As an example, let us consider Dalal's [1988] revision operation. Let $mod(C)$ and $mod(\varphi)$ denote the set of all models of C and φ, respectively. Further, if $\mathcal{M}$ is a set of truth assignments, then $form(\mathcal{M})$ denotes a formula such that $mod(form(\mathcal{M})) = \mathcal{M}$. The function $\delta(\alpha, \beta)$ denotes the number of propositional variables such that α and β map them to different truth-values. The distance $\Delta(A, \varphi)$ between A and φ is then defined to be

$$\Delta(A, \varphi) \stackrel{\text{def}}{=} \min(\{\delta(\alpha, \beta) \mid \alpha \in mod(A), \beta \in mod(\varphi)\}). \tag{23}$$

Now, Dalal's scheme, written $C \circ_D \varphi$ is defined by:

$$A \circ_D \varphi \stackrel{\text{def}}{=} \begin{cases} form(\{\alpha \in mod(\varphi) \mid & \text{if } \not\models \neg A \text{ and } \not\models \neg\varphi \\ \qquad \exists \beta \in mod(A) \text{ s.t.} & \\ \qquad \delta(\alpha, \beta) = \Delta(A, \varphi)\}) & \\ \{\varphi\} & \text{otherwise.} \end{cases} \tag{24}$$

This revision scheme (which can actually be regarded as one global revision operation) satisfies all rationality postulates.

Other similar schemes have been defined in the literature. Satoh's [1988] scheme uses set-inclusion over sets of propositional variables with different truth values instead of cardinality to determine distance. This means $\delta(\alpha, \beta)$ is the set of propositional atoms p_i such that $\alpha(p_i) \neq \beta(p_i)$ and $\Delta(A, \varphi)$ contains the set-inclusion minimal sets $\delta(\alpha, \beta)$, where α ranges over $\mathcal{M}(A)$ and β over $\mathcal{M}(\varphi)$. Borgida's [1985] scheme also uses set-inclusion for measuring distance, but collects for each model of A the closest models of φ—provided φ is not consistent with A. Otherwise, it simply uses the intersection over the models, i.e., it expands the base by the revision formula. Finally, Weber's [1986] scheme computes a revised base by collecting all models from φ that are identical to models of A when ignoring the atoms that appear in the $\Delta(A, \varphi)$ function of Satoh's scheme.

An extensive semantical analysis of these schemes was performed by Katsuno and Mendelzon [1991]. The computational complexity of those schemes (and others) has been thoroughly analyzed by Eiter and Gottlob [1992].

4.2 Syntax-Based Revision Schemes

The schemes described in the previous subsection have the disadvantage that they are *inflexible*. From a more formal point of view, **flexibility** of a revision scheme may be defined as its ability to generate a reasonable large class of revision operations, for instance, the entire class of revision operations that satisfy all rationality postulates. Since in all model-based revision schemes the result of a revision depends only on the logical contents of the belief base and the new information, it is clear that the class of generated revisions is quite limited. From a more pragmatic point of view, *flexibility* could mean that one has a more fine-grained control over what formulae are discarded and what formulae are going to stay. Although this sense of flexibility of a revision scheme is captured by the formal definition only in an imperfect way, we will stick to the formal notion.

Flexibility can be achieved by giving preference information that is used by the revision scheme. Since it is representationally infeasible to use preference information over all formulae in the logical closure of a belief base, it seems reasonable to represent preference information only with regard to the formulae explicitly mentioned in the belief base.

This line of research has been pursued by many researchers [Alchourrón and Makinson, 1982; Dubois and Prade, 1991; Fagin *et al.*, 1983; Fuhrmann, 1991; Hansson, 1996; Hannson, 1991; Hansson, 1994b; Nayak, 1994; Nebel, 1989; Nebel, 1991; Nebel, 1992; Rott, 1993]. Further, similar approaches have been studied in the context of evaluating conditionals [Ginsberg, 1986; Kratzer, 1981; Pollock, 1976; Veltman, 1976], hypothetical reasoning [Rescher, 1964], and default reasoning [Benferhat *et al.*, 1995; Brewka, 1989; Poole, 1988; Reiter, 1987].

The main idea in all these approaches is to start with a belief base and possibly a preference ordering over the formulae in the base or other means to express preferences and to generate a result by operations on the base. However, there are subtle differences in what the objects of interest are and what particular operations on the base are permitted.

First of all, one may consider the belief base itself as representing a belief state, implying that the particular syntactic representation of the base is an essential part of the belief state (see Section 5.6). This view has been adopted by Hansson [1991; 1994b; 1996]. A revision operation is then an operation on the belief base. We call such operations **base revision operations**. This implies in particular that the result of a base revision operation should again be a belief base and it should have some connections to the original belief base. Such connections could be formalized by rationality postulates similar to those given in Section 2.2. For example, one such postulate could state that a formula in the revised belief base should either be a member of the original belief base or be identical with the revision formula. Such base revision operations can be *lifted* to belief revision operations by considering the belief revision operation that corresponds to an operation on the base that generates the belief set to be revised. In this case, we speak about **base-generated belief revisions** [Fuhrmann, 1991; Hansson, 1993b; Hansson, 1996]. One of the interesting questions in this context is which of the rationality postulates described in Section 2.2 are satisfied by the base-generated belief revisions.

Secondly, one may consider *revision schemes* that produce belief revision operations on a belief set by taking into account information about how the belief set is represented syntactically and some preference information attached to the syntactic representation of the belief set [del Val, 1994; Nebel, 1994]. Such schemes are similar to the belief revision schemes as discussed in Section 2 in that they provide a recipe how belief revision operations are to be generated. The only difference is that the preference information is provided in form of a belief base (that generates the belief set to be revised) and some extra information. Such revision schemes are called **base revision schemes** and if $\mathcal{S}$ is a base-revision scheme the generated belief revision operations are called **$\mathcal{S}$ base revisions**.

The difference between *base-generated revision* and *base revision schemes* may appear to be marginal. However, the crucial difference is that a base revision scheme does not generate a belief base but a belief set, which may not be easily representable as a belief base. Since we assumed a finite propositional language by restricting the set Σ of variables to be finite, it is always possible to find a finite belief base for a belief set. Nevertheless, such a base will most probably not satisfy the above mentioned postulates for base revisions. Further such a base may turn out to be difficult to represent, as will be demonstrated in Section 8.

Regardless of which approach we adopt, the revision will be dependent on the syntactic form of the belief base that represents the belief set. This means, for two logically equivalent bases A and B, i.e., $Cn(A) = Cn(B)$, we will in general not have that $Cn(A)$ revised by φ is identical to $Cn(B)$ revised by φ. This sensitivity to the syntactic form has been extensively criticized in the literature [Dalal, 1988; Winslett, 1988; Katsuno and Mendelzon, 1991]. However, from a pragmatic and application-oriented point of view, syntax-based revision seems to be reasonable in situations where the formulae in the belief base have a special status, such as in the case of logical databases [Fagin *et al.*, 1983; Gabbay *et al.*, 1994] or when diagnosing possibly defect artifacts [Reiter, 1987].

Further, the underlying assumptions in the two approaches sketched above are, of course, reasonable. In the first case, we view belief bases as representing a belief state, and for this reason, the syntactic form should play an important role. In the second case, we view the syntactic representation of a belief set as one ingredient for generating a belief revision operation on the generated belief set. Finally, it should be noted that in all syntax-based revision schemes there is no arbitrary dependence on the syntactic representation [Benferhat *et al.*, 1995], a point we will return to in Section 10.2.

4.3 Analyzing the Computational Complexity of Base Revision Schemes

For analyzing the computational complexity of base revision schemes, we could consider the problem of generating a belief base resulting from a revision scheme. However, in this case the complexity may be dominated by producing the revised base—which might be very large. Further, the problem of generating a revised base is not a decision problem, which leads to some unnecessary formal problems. Instead, we will consider the problem of deciding whether a formula ψ is implied by a base A revised by φ. Denoting a base revision scheme by $\odot$, we want to decide the following problem, which we call **revision problem**:

$$A \odot \varphi \models \psi \tag{25}$$

The instance size is $|A|+|\varphi|+|\psi|$, where we assume that A includes the preference information on the base (or, if this is not the case, that the size of this information is bounded polynomially in $|A|$).

Since this problem will in general not turn out to be computationally feasible, some simplifications are considered. One way to simplify the revision problem is to assume that the size of the revision formula is bounded. This restriction is reasonable when the revision formula is small compared with the size of the belief base—an assumption that is usually true in a database context [Eiter and Gottlob, 1992; Winslett, 1990].

Another possible way to simplify the revision problem is to restrict the base logic to a fragment such that satisfiability can be decided in polynomial time. For instance, for a conjunctive formula consisting only of **Horn clauses**, i.e., disjunctions with at most one positive literal, satisfiability is polynomial. We will call such formulae **Horn formulae** and use the term **Horn logic** when all formulae are Horn formulae.

4.4 A Lower Bound for Representationally Feasible Revision Schemes

Assuming a representationally feasible revision scheme, this scheme can be easily used to decide NP-problems and coNP-problems. According to ($\dot{+}$4) and ($\dot{+}$5), satisfiability of a formula φ can be decided by deciding

$$\{\varphi\} \odot \top \models \varphi. \tag{26}$$

Additionally, using the same postulates and the fact that $\varphi \rightarrow p$ is always satisfiable when p does not appear in φ, the validity of a formula φ can be decided by testing

$$\{\varphi \rightarrow p\} \odot \top \models p, \tag{27}$$

In other words, revision is NP-hard and coNP-hard. From that it follows that representationally feasible revision schemes are most likely not problems in $\mathsf{NP} \cup \mathsf{coNP}$, because this would imply $\mathsf{NP} = \mathsf{coNP}$, which is considered unlikely.

Proposition 4.1 ([Nebel, 1992]) *Any representationally feasible revision scheme is* NP*-hard and* coNP*-hard, and it is not a member of* $\mathsf{NP} \cup \mathsf{coNP}$ *(provided* $\mathsf{NP} \neq \mathsf{coNP}$*).*

This means that revision is harder than propositional satisfiability and propositional implication regardless of what representationally feasible scheme we consider. However, on one hand this result does not tell us how hard it is for particular schemes and, on the other hand, it does not rule out polynomial algorithms for simplifications.

Restricting the size of the new formula that is to be incorporated into the belief case does not help. Proposition 4.1 holds even if we assume that the size of the new formula is constant, which follows from the generic problem instances (26) and (27), where the revision formula is just "$\top$".[12]

Proposition 4.2 *Any representationally feasible revision scheme is* NP*-hard and* coNP*-hard, even if the size of the revision formula is bounded by a constant.*

A restriction to Horn logic may, however, lead to a lower complexity. Since the lower complexity bound for belief revision is caused by the problems SAT and TAUT, a restriction to Horn logic leads obviously to a lower bound of P. As we will see below, this does not imply that this restriction reduces the complexity in all cases. However, for some revision schemes the restriction of the base logic has a positive effect.

4.5 Full Meet Revision Scheme

Full meet revision is the simplest way of revising a belief state—and also quite unreasonable because it deletes too much. Although it is "fully rational" in that it satisfies all the rationality postulates (see Section 2.2), it only generates a very small class of revision operations.[13] We will, nevertheless, have a brief look at this revision scheme and analyze its complexity.

[12] At this point, *belief revision* and *belief update* differ. Because of the postulate (U2) [Katsuno and Mendelzon, 1992] an unsatisfiable belief base stays unsatisfiable and (26) cannot be used to decide satisfiability.

[13] From a global perspective quantifying over all belief sets, it actually generates just one *global* belief revision operation.

By the above result, we know that it must be NP-hard and coNP-hard. However, it is not much harder than propositional satisfiability. In particular, it is quite obvious that full meet revision is a problem in Δ_2^p because the following algorithm, which uses SAT-oracle calls, solves the full meet revision problem $\psi \in Cn(A) \overset{F}{+} \varphi$:

$$\begin{array}{l} \text{if } A \not\models \neg\varphi \\ \quad \text{then } A \cup \{\varphi\} \models \psi \\ \quad \text{else } \varphi \models \psi. \end{array} \tag{28}$$

In order to determine the precise complexity of the problem, we use the *boolean hierarchy* introduced in Section 3.4.

Theorem 4.3 *Full meet revision is* coNP(3)*-complete.*

As is obvious from the reduction used in the proof of the theorem above, the hardness result holds even if the revision formula is bounded in size by a constant.

Theorem 4.4 *Full meet revision is* coNP(3)*-complete even if the revision formula is restricted in size by a constant.*

While this result implies that full meet revision is somewhat harder than propositional implication, the simple algorithm (28) suggests that full meet revision is easy provided that all formulae belong to a polynomial fragment of propositional logic.

Proposition 4.5 *Full meet revision is polynomial for Horn logic.*

5 Meet Base Revision Schemes

In analogy to the partial meet revision scheme described in Section 2.3, one can try to identify subsets of a base that are consistent with the revision formula and are most preferred according to some criteria. These most preferred consistent sets are then used to construct the new base, perhaps implicitly. For example, the idea of *changing a belief base minimally* could be formalized by selecting inclusion-maximal subsets of the belief base not implying the revision formula. If there is more than one such maximal subset, the intersection of the *logical closures* of these subsets is used as the result—an approach analyzed in the next subsection. Of course, instead of using the intersection over the logical closures of the inclusion-maximal subsets, one could also use the intersection over the subsets itself (see Section 5.6). In addition to that, other approaches are possible, which are also analyzed below.

5.1 Full Meet Base Revision

Applying the definition of remainder sets to belief bases, the **full meet base revision scheme**, written as $A \textcircled{\scriptsize F} \varphi$, could be defined as follows [Fagin *et al.*, 1983; Ginsberg, 1986; Nebel, 1989; Nebel, 1991]:

$$A \textcircled{\scriptsize F} \varphi \overset{\text{def}}{=} \Big(\bigcap\{Cn(B) |\, B \in (A\bot\,\neg\varphi)\}\Big) + \varphi. \tag{29}$$

As shown in [Nebel, 1991], the belief revision operations generated by the full meet base revision scheme are partial meet revisions. Further, it is easy to verify that the *marking off* relation is the complement of (a restricted) subset relation. i.e., it is *negatively transitive*. This means that all these revision operations satisfy the postulates ($\dotplus$1)–($\dotplus$7) together with ($\dotplus$8r) and ($\dotplus$8c) [Rott, 1993]. More interestingly, as shown independently by Rott [1993] and del Val [1994], all revision operations satisfying these postulates can be generated by the full meet base revision scheme. In other words, this scheme is quite flexible.

Theorem 5.1 ([Rott, 1993; del Val, 1994]) *The class of revision operations generated by the full meet base revision scheme coincides with the class of revision operations satisfying ($\dotplus$1)–($\dotplus$7), ($\dotplus$8r), and ($\dotplus$8c).*

Since the full meet base revision scheme allows for more flexibility than full meet revision by incorporating preference information, it is not surprising that it is harder than the full meet revision scheme. In particular, full meet base revision turns out to be complete for the class Π^p_2, which means that this problem is complementary to problems that can be solved in polynomial time on *non-deterministic* machines using *NP-oracles.*

Theorem 5.2 ([Nebel, 1991; Eiter and Gottlob, 1992]) *Deciding $A \textcircled{\scriptsize F} \varphi \models \psi$ is Π^p_2-complete.*

This result shows that full meet base revision contains two interacting sources of complexity, namely, propositional satisfiability and the selection of an inclusion-maximal consistent set. For this reason, we cannot expect to arrive at a polynomial revision scheme when eliminating only one source, e.g., by restricting the base logic to Horn logic. In fact, as proved by Eiter and Gottlob [1992], the restriction to Horn logic results in a coNP-complete problem.

Theorem 5.3 ([Eiter and Gottlob, 1992]) *Deciding $A \textcircled{\scriptsize F} \varphi \models \psi$ is* coNP-*complete for Horn logic.*

Sometimes, size restrictions on the revision formula can be helpful in reducing the complexity. However, in syntax-based revision schemes this usually does not hold. The reason is that in most cases it is possible to move the revision formula into the belief base using an atomic enabling condition that becomes the new revision formula without changing the outcome of some interesting results (necessary for a reduction). In fact, this holds for the full meet base revision scheme (and all the other schemes we consider).

Theorem 5.4 ([Eiter and Gottlob, 1992]) *The full meet base revision scheme is Π_2^p-complete for general propositional logic and* coNP*-complete for Horn logic even if the revision formula has a size bounded by a constant.*

5.2 Prioritized Base Revision

The operation $\textcircled{F}$ considers all formulae in a base as equally relevant. In most applications, however, we want to distinguish between the importance or relevance of different formulae. Fagin *et al.* [1983], for instance, assign priorities to formulae in a logical database in order to reflect the distinction between facts and integrity rules. Ginsberg [1986] makes a distinction between facts that can change and those that are "protected," and Pollock [1976] makes a distinction between strong and weak subjunctive generalizations and simple propositions.

This idea of assigning different priorities to formulae can be formalized by employing an ordering over the formulae in the belief base. Since we consider only finite bases, this can be done by partitioning a base A into n different **priority classes** A_i, $1 \leq i \leq n$, with the understanding that for $i > j$ the formulae in class A_i are more relevant or important than those in A_j. The associated total preorder $\sqsubseteq$ defined by

$$\varphi \sqsubseteq \psi \text{ iff } \varphi \in A_i, \psi \in A_j, i \leq j, \tag{30}$$

is called **epistemic relevance ordering** [Nebel, 1990; Nebel, 1991].

A belief base together with an epistemic relevance ordering will be called **prioritized base**. Using the priorization, we define a **priority-inclusion preference ordering** "$\ll$" on subsets of a base A as follows:

$$B \ll C \overset{\text{def}}{\Longleftrightarrow} \exists i\, A_i \cap B \subset A_i \cap C \wedge \forall j > i : A_j \cap B = A_j \cap C. \tag{31}$$

The **prioritized remainder set** of A by φ, written $A \downarrow \varphi$ is then defined as the priority-inclusion preferred subsets consistent with $\neg\varphi$:

$$(A \downarrow \varphi) \overset{\text{def}}{=} \{B \subseteq A |\, B \not\models \varphi, \forall C \subseteq A \wedge B \ll C \Rightarrow C \models \varphi\} \tag{32}$$

Intuitively, the elements of $A \downarrow \varphi$ are constructed by selecting a maximal subset not implying φ from A_n, then a maximal subset of A_{n-1} such that φ is not implied by the two subsets, and so on.[14]

[14] Note that this procedure is quite similar to the construction of an extension in Brewka's [1989; 1991] level default theories. In fact, cautious reasoning in such theories is identical to prioritized base revision with a tautology [Nebel, 1991].

As should be obvious, a prioritized remainder set selects a subset of the maximal subsets of a base not implying a given proposition. Thus, it makes sense to use $\downarrow$ instead of $\perp$ in the definition (29). The resulting scheme is called **prioritized base revision scheme**, denoted by $\circledP$:

$$A \circledP \varphi \stackrel{\text{def}}{=} \Big(\bigcap\{Cn(B) \,|\, B \in (A \downarrow \neg\varphi)\}\Big) + \varphi. \tag{33}$$

While the priorization of a base is an advantage from a practical point of view, from a formal point of view, prioritized base revision is almost identical to full meet base revision. First of all, it offers the same *flexibility* in the sense that the same class of revision operations are generated. The main observation is here that the *marking off* relation of the generated partial meet revisions is still negatively transitive, i.e., we do not get more revisions than in the full meet base revision case. Further, since full meet base revision is a special case of prioritized base revision, we get at least all the revision operations generated by full meet base revision, i.e., we get precisely the same class of revisions.

Theorem 5.5 ([Rott, 1993; del Val, 1994]) *The class of revision operations generated by the prioritized base revision scheme coincides with the class of revision operations satisfying ($\dot{+}$1)–($\dot{+}$7), ($\dot{+}$8r), and ($\dot{+}$8c).*

Further, the computational complexity of the prioritized belief revision scheme is the same as the computational complexity of full meet base revision.

Theorem 5.6 *Deciding $A \circledP \varphi \models \psi$ is Π_2^p-complete for general propositional logic and* coNP*-complete for Horn logic. This also holds under the assumption that the size of the revision formula is bounded by a constant.*

Although prioritized base revision does not appear to be much of an improvement, it contains an interesting special case. If we assume that the epistemic relevance ordering is a total linear ordering, i.e., if all priority classes are singleton sets, we get a revision scheme that is quite well-behaved.

5.3 Linear Base Revision

The prioritized base revision scheme specialized to *total linear* epistemic relevance orderings will be called **linear base revision scheme** [Nebel, 1994]. The corresponding operation on the base is denoted by $\circledL$. Instead of requiring that the ordering is is a total linear ordering or that each priority class is a singleton set, we may as well allow total preorders or arbitrary priority classes, respectively, but discard a priority class entirely if one of the formulae has to go under the prioritized base revision scheme. The outcome is evidently identical and we will not distinguish between these cases below.

As is easy to see, the set $(A \downarrow \varphi)$ contains at most one subset of A [Nebel, 1989]. For this reason, this scheme can be viewed as a *maxichoice base revision scheme*. In contrast to a *maxichoice belief revision scheme*, however, we do not get the unwanted property that a revised base is a complete theory. Further, it can be shown that all revision operations generated by this scheme are fully rational in the sense that they satisfy all rationality postulates for revision.

Theorem 5.7 ([Nebel, 1992]) *Revision operations generated by the linear base revision scheme satisfy ($\dot{+}$1)–($\dot{+}$8).*

As a matter of fact, all fully rational revision operations can be generated by this scheme.

Theorem 5.8 ([Nebel, 1994; del Val, 1994]) *The class of revision operations generated by the linear base revision scheme coincides with the class of revision operations satisfying ($\dot{+}$1)–($\dot{+}$8).*

In addition, also the computational properties are quite appealing. In general, we get a reduction of complexity by one level in the polynomial hierarchy. The obvious reason is that the selection of a consistent subbase is very easy since there is at most one such subbase.

Theorem 5.9 ([Nebel, 1994]) *The linear base revision scheme is Δ_2^p-complete. This holds even when the revision formula is bounded in size by a constant.*

From the algorithm in the proof of the previous Theorem it follows straightforwardly that a restriction to Horn logic leads immediately to an efficiently solvable problem.

Theorem 5.10 ([Nebel, 1994]) *For Horn logic, $A \,\textcircled{\scriptsize L}\, \varphi \models \psi$ can be decided in $O(n^2)$ time, where $n = |A| + |\varphi| + |\psi|$.*

While these results appear to be quite appealing from a theoretical point of view, there is the question for the practical relevance. Although it might be considered to be very unrealistic to require that all formulae in a base are linearly ordered, there are applications where such an order appears to be natural. Gabbay *et al.* [1994] considered, for instance, hypothetical reasoning in a logic programming context, where the clauses are tagged with the time they were entered into the system. These tags are used to assign priorities to the clauses such that clauses entered later have precedence over the clauses entered earlier. In evaluating a hypothetical goal that leads to inconsistencies, the prioritized base revision scheme is used.

One remaining question is whether the positive complexity results could not be extended to the prioritized base revision scheme with a constant number of elements in each priority class. However, with at most two elements in each priority class, the hardness results of the previous subsection apply.

Theorem 5.11 *Theorem 5.6 also holds under the assumption that the cardinality of the priority classes is bounded by a constant $k \geq 2$.*

5.4 Cardinality-Maximizing Base Revision

Instead of selecting all inclusion-maximal sets that are consistent with the revision formula, we might as well consider the *cardinality-maximal* sets. Such a strategy might be reasonable, for instance, if we are in a *diagnosis* context, where we assume that it is more likely that only a few components are faulty instead of many [Ginsberg, 1986; de Kleer, 1990].

Let $(A \perp\!\!\!\perp \varphi)$ denote the cardinality-maximal subsets of A that are consistent with $\neg\varphi$, i.e.,

$$A \perp\!\!\!\perp \varphi \stackrel{\text{def}}{=} \{B \subseteq A \mid B \not\models \varphi, \forall C\colon C \subseteq A \wedge \|B\| < \|C\| \Rightarrow C \models \varphi\}. \quad (34)$$

Based on this operator, we can define the operation

$$A \circledcirc \varphi \stackrel{\text{def}}{=} \Big(\bigcap\{Cn(B) \mid B \in (A \perp\!\!\!\perp \varphi)\}\Big) + \varphi, \quad (35)$$

which is called **cardinality-maximizing base revision scheme**. It is obvious that this revision scheme can be regarded as a refinement of the full meet base revision scheme in that we have $(A \perp\!\!\!\perp \varphi) \subseteq (A \perp \varphi)$.

Similar to the full meet and prioritized meet base revision scheme, the generated revision operations can be easily associated with relational partial meet revisions. Further the *marking off* relation generated by the selection function is transitive, from which we can conclude—using the results sketched in Section 2.3—that the cardinality-maximizing scheme is “fully rational,” i.e., satisfies all rationality postulates. By linking it to the linear base revision, we can show an even stronger result.

Lemma 5.12 *Any revision generated by the linear base revision scheme can be generated by the cardinality-maximizing base revision scheme.*

Since the linear base revision scheme generates all “fully rational” revision operations (Theorem 5.8), this also holds for the cardinality-maximizing scheme.

Theorem 5.13 *The class of revision operations generated by the cardinality-maximizing base revision scheme coincides with the class of revision operations satisfying ($\dot{+}$1)–($\dot{+}$8).*

Turning now to the computational complexity of the scheme, we notice that it is somewhat easier than full meet and prioritized meet base revision in the general case.

Theorem 5.14 *Deciding* $A \circledcirc \varphi \models \psi$ *is* $\Delta_2^p[O(\log n)]$*-complete, and this holds even if the size of the revision formula is bounded by a constant.*

One might hope that eliminating one source of complexity, namely, reasoning in full propositional logic, leads to a polynomial-time revision problem. However, studying the proof of the theorem above carefully, one notes that the algorithm demonstrating membership in $\Delta_2^p[O(\log n)]$ does not become a deterministic polynomial-time algorithm when satisfiability becomes a polynomial-time problem. The algorithm contains also guesses on subsets of the belief base where only the cardinality of the subset is specified—and this cannot be easily solved deterministically. As a matter of fact, restricting the logic to Horn logic does not help in reducing computational complexity.

Theorem 5.15 *Deciding* $A \circledcirc \varphi \models \psi$ *is* $\Delta_2^p[O(\log n)]$*-complete for Horn logic, and this holds even if the size of the revision formula is bounded by a constant.*

5.5 Lexicographic Base Revision

Similar as the move from full meet to prioritized meet base revision, we can extend the cardinality-maximizing scheme by introducing priority classes. Such a scheme has been proposed for inference from inconsistent belief bases (which is equivalent to revising a base with "$\top$") [Benferhat *et al.*, 1993] and for reasoning in a nonmonotonic logic [Lehmann, 1993]. Defining the **lexicographic preference relation** $\lll$ on subsets of a belief base A as

$$B \lll C \overset{\text{def}}{\iff} \exists i \, \|A_i \cap B\| < \|A_i \cap C\| \wedge \forall j > i : \|A_j \cap B\| = \|A_j \cap C\|, \tag{36}$$

the remainder set for the new scheme is defined as follows:

$$(A \Downarrow \varphi) \stackrel{\text{def}}{=} \{B \subseteq A \mid B \not\models \varphi, \forall C \subseteq A \wedge B \lll C \Rightarrow C \models \varphi\}. \tag{37}$$

Then the **lexicographic base revision scheme** is defined as:

$$A \otimes \varphi \stackrel{\text{def}}{=} \left(\bigcap \{Cn(B) \mid B \in (A \Downarrow \neg\varphi)\}\right) + \varphi. \tag{38}$$

Similar to prioritized base revision, we gain more expressiveness from a practical point of view, but the formal flexibility does not change by introducing priority classes. Since the *marking off* relation of the partial meet revision operations that are generated by our scheme is still transitive, the scheme satisfies all rationality postulates. Further, since the cardinality-maximizing scheme is a special case, the next proposition follows immediately from Theorem 5.13.

Proposition 5.16 *The class of revision operations generated by the lexicographic base revision scheme coincides with the class of revision operations satisfying* $(\dot{+}1)$*–*$(\dot{+}8)$*.*

From a computational perspective, however, cardinality-maximizing and lexicographic base revisions differ. The introduction of priority classes implies a slight increase of complexity, as spelled out in the next two theorems, which are variations of theorems by Cayrol and Lagasquie-Schiex [1993], who analyzed the complexity of making inferences from inconsistent belief bases.

Theorem 5.17 *Deciding $A \circledX \varphi \models \psi$ is Δ_2^p-complete, and this holds even if the size of the revision formula is bounded by a constant.*

Not very surprisingly, we do not get a reduction in complexity when the logic is restricted to Horn logic.

Theorem 5.18 *Deciding $A \circledX \varphi \models \psi$ is Δ_2^p-complete for Horn logic, and this holds even if the size of the revision formula is bounded by a constant.*

5.6 Base Revision Operations: When in Doubt Throw it Out

In almost all of the schemes considered above, the remainder sets contain more than one element, and these elements were combined by using the intersection over the logical closure of the remainders, leading to a belief set instead of a belief base.

Instead one may have the perspective that change operations should directly operate on belief bases, resulting in new belief bases. Adopting this view, Hansson developed postulates for such base change operations that are parallel to the rationality postulates for belief revision [Hansson, 1993a]. One of these postulates, the **inclusion postulate** states that the belief base resulting from a base revision operation should only contain formulae that were already there plus the new formula. Formally, denoting a base revision operator by $\dot{\circ}$, the postulate states:

$$A \mathbin{\dot{\circ}} \varphi \subseteq A \cup \{\varphi\}. \tag{39}$$

The base revision schemes considered in this section so far clearly do not satisfy this postulate since they generate belief sets. However, most of them even do not satisfy this postulate in a weak sense, namely, in the sense that there exists a base representing the result of the revision and satisfying the inclusion postulate. Consider, for example, the base $A = \{p \wedge r, q \wedge r\}$ and the full meet base revision of this base by $\neg p \vee \neg q$. Clearly, we have $A \circledF \varphi \models r$ and $A \circledF \varphi \not\models p, q$. However, there is no subset of $A \cup \{\neg p \vee \neg q\}$ that achieves the same effect.

A straightforward way to satisfy the inclusion postulate would be to produce the result of the base revision operation by taking the intersection over the remainder sets (instead of intersecting the logical closures of the remainders). For example, corresponding to the *full meet base revision scheme*, we could define a **full meet base revision operation** $\overset{F}{\circ}$ by:

$$A \overset{F}{\circ} \varphi \overset{\text{def}}{=} \bigcap(A\bot\neg\varphi) \cup \{\varphi\}. \tag{40}$$

Winslett [1990] coined the phrase *when in doubt, throw it out* (WIDTIO) to describe this approach, because formulae are deleted when they do not appear in all remainders. Obviously, this approach is applicable to all the base revision schemes we have considered so far. Viewing such operations as base revision schemes, we call the resulting operations WIDTIO-revision schemes in order to distinguish them from the base revision schemes considered so far. The above defined base revision, corresponding to the *full meet base revision scheme* would then be called **full meet WIDTIO-revision scheme**. It is obvious that the WIDTIO-revision schemes defined in this way lead to different results than the base revision schemes, save the case of the linear base revision, where the remainder set is a singleton set.

From a computational perspective, the WIDTIO-revision schemes appear to be not easier than base revision schemes. In fact, all reductions used in the proofs above apply to WIDTIO-revisions as well, leading to identical hardness results.

Lemma 5.19 *For full meet, prioritized, linear, cardinality-maximizing, and lexicographic base revision schemes, the corresponding* WIDTIO*-revision schemes are as hard as the base revision schemes.*

As spelled out above, the linear base revision scheme and the linear WIDTIO-revision scheme coincide because we get a singleton remainder set in either case. For this reason, the upper bound is also identical. For the other cases, upper bounds are not immediate, however. The WIDTIO-revision schemes appear to be slightly harder than the base revision schemes because non-implication from one remainder plus the revision formula is not any longer complementary to implication from the revised base. For full meet WIDTIO-revisions Eiter and Gottlob [1992] proved an upper bound of $\Delta_3^p[O(\log n)]$. As the next lemma shows, however, we can do better than that. In fact, we achieve tight upper bounds for all WIDTIO-revision schemes.

Lemma 5.20 *For full meet, prioritized, linear, cardinality-maximizing, and lexicographic base revision schemes, the corresponding* WIDTIO*-revision schemes are in the same complexity class as the base revision schemes.*

Hence, from a computational complexity point of view, it does not make a difference whether we use the base revision schemes or the corresponding WIDTIO-revision schemes because upper and lower bounds are identical according to the above two lemmata.

Theorem 5.21 *For full meet, prioritized, linear, cardinality-maximizing, and lexicographic base revision schemes, the corresponding* WIDTIO*-revision schemes have the same computational complexity as the base revision schemes.*

However, there are clearly some significant differences. First of all, if we view WIDTIO-revisions as revision schemes, they seem to behave odd compared with the original base revision schemes. In particular, postulates ($\dot{+}7$) and ($\dot{+}8$) are in general not satisfied by the generated belief revision operations. It is nevertheless possible to obtain representation results for the generated revision operations, as shown by Hansson [1993b; 1996]. Secondly, there is a difference when we consider the space necessary to represent a revised base. For the base revision schemes considered in 5.1–5.5, it is not obvious that we can obtain a compact form of the revised belief base (see also Section 8). WIDTIO-revisions lead to compact bases by definition, however.

6 Cut Base Revision Schemes

Comparing the three different ways of generating belief revision operations as described in Section 2, it is apparent that the *cut revision scheme* has two advantages over the other schemes. Firstly, it is fully rational and able to generate all fully rational revision operations. Secondly, only one subset of the belief set is generated in computing the revised belief set, while meet and safe revision require the generation of all maximally preferred consistent subsets or of all inclusion-minimal entailment sets.

In particular the second property may lead to computational advantages for *cut revision schemes* applied to belief bases. However, in order to realize a cut *base* revision scheme, we first have to identify orderings on a belief base that correspond to epistemic entrenchment orderings.

6.1 Priority-Consistent Orderings and Ensconcements

It is apparent that the notion of *epistemic entrenchment* cannot be directly applied to belief bases since they are not logically closed – which is required, e.g., by ($\preceq 2$) and ($\preceq 3$). Instead of a genuine epistemic entrenchment ordering on a belief base, we will look for total preorders over the base denoted by $\leq$. The idea is that these orderings can be extended to epistemic entrenchment orderings over the belief set generated by the base. Since such total preorders are equivalent to a prioritized base as defined in Section 5.2, we will use the notion of priority classes here as well.

If such a relation is to be extended to an epistemic entrenchment ordering, it should respect logical relationships. For instance, starting with an arbitrary total preorder $\leq$ on a base A, we may well have the case that $\varphi \models \psi$ but $\psi < \varphi$, contradicting ($\preceq$2). In other words, there is no epistemic entrenchment relation $\preceq$ on $Cn(A)$ that extends the ordering $\leq$ on A.

In the example above, it does not seem to make much sense that φ is "more entrenched than" ψ since φ has to be retracted in any case if ψ is forced to be deleted. More generally, if $C \subseteq A$, $C \models \varphi$ and C is set-inclusion minimal w.r.t. this property, then it does not make much sense that φ is "less entrenched than" the "least entrenched" formula in C. For this reason, let us assume that the ordering satisfies the following **priority consistency condition** (PCC) [Rescher, 1973; Rott, 1991a]:

> For all $\varphi \in A$, if C is a nonempty subset of A such that $C \models \varphi$, then there exists $\chi \in C$ such that $\chi \leq \varphi$.

As has been shown by Rott [1991a], this condition is necessary and sufficient for the extendibility of $\leq$ on A to an epistemic entrenchment ordering on the generated belief set $Cn(A)$. For this reason, Rott called belief bases with a PCC-ordering *E-bases*.

Williams [1994b; 1994a] defined the almost equivalent notion of an **ensconcement ordering** on a belief base A as a total preorder $\leq$ satisfying:[15]

($\leq$1) For all nontautological $\varphi \in A$: $\{\psi \in A \mid \varphi < \psi\} \not\models \varphi$.

($\leq$2) For all $\varphi \in A$: $\psi \leq \varphi$ for all $\psi \in A$ iff $\models \varphi$.[16]

Similar to cut-sets for epistemic entrenchments as defined by Equation (9) one can define cut-sets for ensconcement orderings on a base A:

$$cut_{<}(\varphi) \stackrel{\text{def}}{=} \begin{cases} \{\psi \in A \mid \{\chi \in A \mid \psi \leq \chi\} \not\models \varphi\} & \text{if } \not\models \varphi, \\ \emptyset & \text{otherwise.} \end{cases} \tag{41}$$

In other words, $cut_{<}(\varphi)$ selects all the formulae in all high priority classes such that adding the next lower priority class leads to the implication of φ. Based on this notion, a relation $\preceq_{\leq}$ on $\mathcal{L}$ can be generated:

$$\varphi \preceq_{\leq} \psi \stackrel{\text{def}}{\Longleftrightarrow} cut_{<}(\psi) \subseteq cut_{<}(\varphi). \tag{42}$$

This relation is in fact an epistemic entrenchment ordering on the belief set $Cn(A)$.

Theorem 6.1 ([Nebel, 1994; Williams, 1994a]) *If $\leq$ is a ensconcement ordering on A, then $\preceq_{\leq}$ as defined by Equation (42) extends $\leq$ and is an epistemic entrenchment on $Cn(A)$.*

[15] Williams [1994a] called the base with its ordering an *ensconcement*. We will talk about a base and the ensconcement ordering over this base.

[16] This implies that we have at least one tautology in our base, which is necessary for satisfying ($\preceq$5) in the construction below.

This means that we can consider an ensconcement ordering on a base as a concise representation of an epistemic entrenchment ordering on the corresponding belief set. The interesting question is, whether there is an appropriate base revision operation that corresponds to the revision generated by this entrenchment ordering.

Given a base A with ensconcement ordering $\leq$, we define the following operation $\textcircled{\scriptsize E}$ on a base:

$$A \textcircled{\scriptsize E} \varphi \stackrel{\text{def}}{=} cut_{<}(\neg\varphi) \cup \{\varphi\}, \tag{43}$$

which we call **cut base revision scheme using ensconcements**.

Theorem 6.2 ([Nebel, 1994; Williams, 1994a]) *Let A be a base with ensconcement ordering $\leq$. Let $\preceq_{\leq}$ be the epistemic entrenchment order generated from $\leq$ and $\stackrel{\epsilon}{+}$ be the belief revision operation based on $\preceq_{\leq}$, then*

$$Cn(A) \stackrel{\epsilon}{+} \varphi = Cn(A \textcircled{\scriptsize E} \varphi). \tag{44}$$

In other words, if we interpret an ensconcement ordering on A as a concise representation of an epistemic entrenchment ordering for $Cn(A)$, then there is a very concise and straightforward representation of the revised belief set. Further, it follows that the cut base revision scheme using ensconcements is fully rational, because the cut revision scheme using epistemic entrenchments is fully rational (see Section 2.4).

Corollary 6.3 *The cut base revision scheme using ensconcements satisfies ($\dot{+}$1)–($\dot{+}$8).*

Additionally, every fully rational revision operation on belief sets finite modulo logical implication can be generated by the cut base revision scheme. The main reason is that an epistemic entrenchment ordering on a belief set always satisfies (PCC) and hence can also be interpreted as an ensconcement ordering on the belief set viewed as a belief base.

Proposition 6.4 ([Nebel, 1994]) *The class of revision operations generated by the cut base revision scheme using ensconcements coincides with the class of revision operations satisfying ($\dot{+}$1)–($\dot{+}$8).*

Furthermore, it seems to be the case that deciding the computational problem whether a formula follows from a revised belief base is relatively easy, which is confirmed by the next theorem.[17]

[17] While the upper bound was already known [Nebel, 1994], the lower bound was unknown previously.

Theorem 6.5 *Deciding $A \circledE \varphi \models \psi$ is $\Delta_2^p[O(\log n)]$-complete, and this holds even if the size of the revision formula is bounded by a constant.*

As is evident from the proof of Theorem 6.5, restricting the logical language to a subset such that satisfiability could be decided in polynomial time leads in fact to the situation that the revision could be decided in polynomial time. For Horn logic, we even can come up with an $O(n \log n)$ algorithm.

Theorem 6.6 ([Nebel, 1994]) *Provided A is a set of Horn formulae, φ is a Horn formula and ψ is a Horn clause, $A \circledE \varphi \models \psi$ can be decided in time $O(n \log n)$, where $n = |A| + |\varphi| + |\psi|$.*

6.2 Generating Ensconcement Orderings from Arbitrary Priorities

The only grain of salt in the above results is the condition that an ensconcement ordering must be priority-consistent, i.e., satisfy the condition (PCC). Since the condition involves the problem of deciding propositional implication, it is unlikely that somebody constructing a belief base is able to generate a priority consistent ordering.

We may, however, take the perspective that one specifies priorities on formulae of a belief base that are interpreted as *lower bounds* for the intended priorities.[18] Using (41) and (42) on some arbitrary prioritized base A, the resulting relation $\preceq_\leq$ is again an epistemic entrenchment ordering for the generated belief set $Cn(A)$ and the restriction of $\preceq_\leq$ to A, denoted by $\trianglelefteq$, is an ensconcement ordering.

Theorem 6.7 ([Nebel, 1994]) *Let $\leq$ be an arbitrary total preorder over a belief base A. Then (1) the relation $\preceq_\leq$ generated by (41) and (42) is an epistemic entrenchment over $Cn(A)$ and (2) its restriction to A, denoted by $\trianglelefteq$, is an ensconcement ordering on A.*

Furthermore, the ensconcement ordering $\trianglelefteq$ generated from the arbitrary ordering $\leq$ and $\leq$ itself are very closely related. The deductive closure of the cut-sets for all formulae turn out to be equivalent.

Lemma 6.8 *Let $\leq$ be an arbitrary total preorder on the belief base A and $\trianglelefteq$ the restriction of the epistemic entrenchment ordering generated by $\leq$ to A. Then for all $\varphi \in \mathcal{L}$*

$$Cn(cut_<(\varphi)) = Cn(cut_\triangleleft(\varphi)). \tag{45}$$

[18]Such an interpretation comes quite close to the semantics of weighted formulae in a possibilistic knowledge base [Dubois and Prade, 1991; Dubois and Prade, 1992], and, in fact, the construction we will use corresponds to the generation of a "closure of weighted formulae" [Dubois and Prade, 1992, p. 165]; see also [Dubois *et al.*, 1994].

From that we can conclude that an arbitrary prioritized base has indeed the property that priorities specify lower bounds. It may be the case that a formula φ is in priority class A_i but also implied by the classes $A_j \cup \ldots \cup A_n$ with $j > i$, in which case the "real" priority is j. In fact, removing φ from A_i and adding it to A_j doesn't change anything.

Further, an immediate consequence of Lemma 6.8 is that the cut base revision scheme applied to a base with an arbitrary total preorder $\leq$ computes a revision with respect to the generated epistemic entrenchment ordering $\preceq_{\leq}$.[19]

Theorem 6.9 ([Nebel, 1994]) *Let A be a base with an arbitrary total preorder $\leq$. Let $\preceq_{\leq}$ be the epistemic entrenchment ordering derived from $\leq$ by (41) and (42) and let $\overset{\epsilon}{\dotplus}$ be the revision operation generated from $\preceq_{\leq}$. Then*

$$Cn(A) \overset{\epsilon}{\dotplus} \varphi = Cn(A \circledE \varphi). \tag{46}$$

This means in particular that the representation theorem (Proposition 6.4) and the complexity results (Theorems 6.5 and 6.6) for *cut base revision based on ensconcements* apply in this case as well.

Corollary 6.10 ([Nebel, 1994]) *The class of revisions generated by cut base revision scheme using arbitrary total preorders coincides with the class of revisions satisfying ($\dotplus$1)–($\dotplus$8). Further, the cut base revision scheme using arbitrary total preorders has the same complexity as the cut base revision scheme using ensconcements.*

6.3 Cut vs. Linear Base Revision

The basic intuition in cut (base) revisions is that one simply cuts away all formulae at level p and lower if there is contradiction between the new information and the formulae in level p up to n. In particular, even if a formula in a level below p does not contribute to the contradiction, it is thrown away—a method that appears to be quite drastic.

However, being more liberal has its disadvantages. In fact, trying to keep as many formulae as possible (maximizing high prioritized formulae) results in the prioritized base revision scheme considered in Section 5.2—which led to very bad computational properties. In trying to find a compromise, one may consider the method of deleting an entire priority class if one formula in it leads to a contradiction that cannot be blamed on formulae in lower levels, but keeping as many of the other classes as possible. This, however, is the linear base revision scheme (see Section 5.3).

Dubois and Prade [1991; 1992] also considered a linear base revision scheme on prioritized bases. They used this scheme to revise a possibilistic knowledge

[19] And this revision corresponds exactly to the revision of possibilistic knowledge bases defined in [Dubois *et al.*, 1994, Section 3.10], save the case that $\models \neg\varphi$.

base and noted that this revision process is more parsimonious than a cut base revision that interprets priorities as lower bounds on necessity values (which are the quantitative counter-parts to epistemic entrenchment). However, they also remark that the linear base revision scheme has the disadvantage that "the revision process cannot be expressed at the semantic level" [Dubois and Prade, 1992, p. 167], i.e., it is impossible to describe this process as a change of possibility distributions resulting from conditioning in possibilistic logic.

In fact, it seems to be interesting to relate the linear base scheme to the scheme considered in the previous section. First of all, it is easy to show that any revision generated by the cut base revision scheme can be generated by the linear base revision scheme.

Lemma 6.11 *Any revision generated by the cut base revision scheme can be generated by the linear base revision scheme.*

From that and Proposition 6.4, we get as a direct consequence the representation theorem for linear base revision (Theorem 5.8) spelled out already in Section 5.3.

However, does it also work in the other direction? As it turns out, it is possible to specify a transformation π from bases with an associated linear epistemic relevance ordering to a prioritized base such that the linear base revision on the original base gives a result that is logically equivalent to the result achieved by the cut base revision on the transformed base. Assume a linearly ordered prioritized base $A = \langle A_i \rangle$ with n priority classes. Then we define $B = \pi(A)$ with 2^n priority classes. The priority classes of B are again singletons, and the elements of these classes are disjunctions of formulae over the formulae in A. In particular, the priority classes B_l with $0 \leq l \leq 2^n$ are defined as follows

$$B_l = \Big\{ \bigvee_{i=j_1}^{j_k} A_i \Big\}, \text{where} \tag{47}$$

$$l = \sum_{m=1}^{k} 2^{j_m - 1}. \tag{48}$$

This means that $\pi(A)$ has a size that is exponential in the size of A. Ignoring this for the moment, we will analyze the effect of the transformation.

Theorem 6.12 ([Nebel, 1994]) *Let A be a prioritized base with a linear epistemic relevance ordering and let π be transformation defined by Equations (47) and (48). Then*

$$Cn(A \circledL \varphi) = Cn(\pi(A) \circledE \varphi) \tag{49}$$

for all $\varphi \in \mathcal{L}$.

Translating this result to the revision processes employed for possibilistic knowledge bases [Dubois and Prade, 1991; Dubois and Prade, 1992] shows that in order to model the "parsimonious" revision, we would need exponentially many different necessity values. However, the problem of justifying the "parsimonious" revision semantically has been solved by this transformation.

The only remaining question is whether it is possible to specify a transformation that is less expensive. Comparing the complexity results for the two different revision schemes reveals, however, that a polynomial-time transformation would only be possible if $\Delta_2^p[O(\log n)] = \Delta_2^p$, something which is quite unlikely [Johnson, 1990].

7 Safe Base Revision Scheme

Corresponding to the *safe revision scheme* sketched in Section 2.5, it is, of course, possible to define a **safe base revision scheme**, which uses a *hierarchy* over the formulae in the belief base [Alchourrón and Makinson, 1985; Fuhrmann, 1991; Nayak, 1994]. This scheme will be denoted by $\circledS$ in the following and is defined as follows:

$$A \circledS \varphi \stackrel{\text{def}}{=} Cn(A/\neg\varphi) + \varphi. \tag{50}$$

In general, this scheme satisfies the basic postulates, but fails on ($\dot{+}7$) and ($\dot{+}8$). However, it is possible to restore the latter two postulates by modifying safe revisions in a way such that it becomes identical to linear base revision [Nayak, 1994].

Interestingly, the computational complexity of the safe base revision scheme (and simplifications of it) or of the similar *kernel-revision scheme* [Hansson, 1994a; Hansson, 1996] has never been analyzed.

7.1 Safe Base Revisions and WIDTIO-Revisions

In order to achieve a lower bound, we will relate safe base revisions to WIDTIO-revisions. As it turns out, safe base revision is identical to full meet WIDTIO-revision, provided the hierarchy is empty. This is no surprise considering the duality between remainder sets and entailment sets.

Theorem 7.1 *Let A be a base with an empty hierarchy. Then*

$$A \overset{F}{\circ} \varphi = A/\neg\varphi \cup \{\varphi\} \tag{51}$$

for all $\varphi \in \mathcal{L}$.

However, this equivalence does not extend to prioritized and linear base revision. The reason is that safe revision may remove more formulae than "necessary."

Consider, e.g., the following base:

$$
\begin{aligned}
A &= \langle A_1, A_2, A_3 \rangle, & (52)\\
A_3 &= \{t \rightarrow p\}, & (53)\\
A_2 &= \{t \rightarrow (\neg q \wedge \neg p)\}, & (54)\\
A_1 &= \{t \rightarrow q\}. & (55)
\end{aligned}
$$

Now revising this base by t under the prioritized base revision scheme gives us:

$$A \, ⓟ \, t = Cn(\{t \rightarrow p, t \rightarrow q, t\}). \tag{56}$$

Revising the base under the safe base revision scheme interpreting the epistemic relevance ordering induced by the priority classes as a hierarchy, we get:

$$A \, ⓢ \, t = Cn(\{t \rightarrow p, t\}). \tag{57}$$

The class A_3 is not included because it is the minimal element of an entailment set. Since the the priority classes are all singletons, it is a counter-example for prioritized as well as linear base revision.

7.2 Computational Complexity of Safe Base Revision

From Theorem 7.1 it follows straightforwardly, that the safe base revision scheme is as hard as full meet WIDTIO-revision, even for Horn logic and size-bounded revision formulae. As it turns out, safe base revision is not harder than full meet WIDTIO-revision, as well.

Theorem 7.2 *The safe base revision scheme is Π_2^p-complete for general propositional logic and* coNP*-complete for Horn logic even if the revision formula has a size bounded by a constant.*

Restricting the hierarchy can, of course, lead to computationally better behaved revision schemes. For instance, requiring that the hierarchy is a linear order leads obviously to membership in Δ_2^p. However, it should be noted that the safe base revision scheme on a linear hierarchy is not identical to the linear base revision scheme, as demonstrated by the example (52)–(57) above.

8 Generating Revised Belief Bases

So far, we have only considered the complexity of deciding whether a formula is implied by a revised belief base. The questions of how such a revised base can be constructed and what the size of such a base is has been deliberately ignored, however.

In some cases, there are immediate answers to these questions. All meet base revisions schemes that lead to a singleton remainder set, WIDTIO-revisions, as well as cut and safe base revisions produce by definition a revised belief set that can be represented by a base that contains only formulae from the original belief base plus the revisions formula. For this reason, the results of these base revision schemes can be represented in a compact way and the bases can be generated in time only polynomially longer than the time needed to decide implication from the revised base.[20]

Proposition 8.1 *The result of linear base revisions, cut base revisions, safe base revisions, and of all* WIDTIO*-revisions can be represented in space polynomially bounded by the original belief base and the revision formula, and it can be generated with only polynomial overhead with respect to the time needed to decide the revision problem.*

For the other cases, answers are not obvious, however. If we consider, for example, the following belief base and revision formula

$$A = \{p_1, \ldots, p_n, q_1, \ldots, q_n\} \quad (58)$$

$$\varphi = \bigwedge_i p_i \leftrightarrow q_i \quad (59)$$

then $(A \bot \neg\varphi)$ clearly contains exponentially many remainders. Nevertheless, in this case it is possible to find a compact base representing the generated belief set, since the base has to contain only the revision formula. Whether a compact form of the revised base can always be identified is not clear, however.

8.1 Do Compact Representations for Revised Belief Bases Exist?

Identifying a compact representation of the result of a full meet base revision, i.e., a representation that has size polynomial in the size of the original belief base and the revision formula, will most probably be very costly from a computational point of view. For instance, assuming that identifying a compact base can be done in deterministic polynomial time with polynomially many SAT-oracle calls leads to the conclusion that $\Delta_2^p = \Pi_2^p$, because $A \circledF \varphi \models \psi$ could then be decided by the procedure that computes a compact representation plus one extra SAT-call. So, finding a compact base is most probably more expensive. However, how much time should we allow?

In order to answer the question whether a compact form can always be found, we will abstract from the time necessary to compute such a base and consider the question of whether such compact representations always exist. This question

[20]Usually, one can do better than that, however (see Section 8.2).

was first posed by Winslett [1990] and answered for a considerable number of base revision schemes and belief update operators by Cadoli *et al.* [1995].

A negative answer can be given by using nonuniform complexity classes (see Section 3.5). One can try to show that assuming compact representations, these could be used as advice strings for an advice-taking Turing machine, leading to the conclusion that some relations between uniform and nonuniform complexity classes are implied, which in turn imply the collapse of the polynomial hierarchy. Using such an argument, it seems very unlikely that compact representations of belief bases revised by the full meet base revision can be found.

Theorem 8.2 ([Cadoli *et al.*, 1995]) *Unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, *there exists no polynomial p such that for $A \circledF \varphi$ there exists a belief base B with size $p(|A|+|\varphi|)$ and the property that $A \circledF \varphi \models \psi$ iff $B \models \psi$ for all formulae ψ that use only variables appearing in A and φ.*

As mentioned in Section 3.5, $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ implies $\Sigma_3^p = \Pi_3^p$ by the results of Yap [1993], i.e., it is very unlikely that compact forms of revised bases can be found—even allowing an unlimited amount of computation. As a corollary, it follows that the theorem also applies to prioritized base revision, since the full meet base revision scheme is a special case of the prioritized base revision scheme.

Corollary 8.3 ([Cadoli *et al.*, 1995]) *Unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, *there exists no polynomial p such that for $A \circledP \varphi$ there exists a belief base B with size $p(|A|+|\varphi|)$ and the property that $A \circledP \varphi \models \psi$ iff $B \models \psi$ for all formulae ψ that use only variables appearing in A and φ.*

Further, by inspecting the proofs of the above theorem and the proofs of the theorems for proving Π_2^p-completeness of full meet and prioritized base revision, one notes that the restriction of the size of the revision formula does not make a difference.

It should be noted that the above theorem is quite general in that it also rules out compact representations which uses new variables.[21] Cadoli *et al.* coined the term *query equivalence* and *logical equivalence* in order to distinguish between the general notion used in the theorem and a stricter notion of equivalence, respectively. $A \odot \varphi$ and B are **query-equivalent** iff

$$\{\psi \in \mathcal{L} \mid A \odot \varphi \models \psi\} = \{\psi \in \mathcal{L} \mid B \models \psi\}, \tag{60}$$

where ψ should contain only variables from A and φ. **Logical equivalence** is defined by the stronger requirement that

$$Cn(A \odot \varphi) = Cn(B). \tag{61}$$

[21] In fact the original theorem by Cadoli *et al.* [1995] is even more general.

That this distinction is relevant is demonstrated by the result that one can find compact representations for belief bases revised using Dalal's model-based scheme if only query-equivalence is required. For logical equivalence, however, such representations are ruled out, unless $\mathsf{NP} \subseteq \mathsf{P/poly}$ [Cadoli *et al.*, 1995]. In addition to these results, Cadoli *et al.* [1995] analyzed the effect of size restrictions on the revision formula, which led to positive results for all model-based schemes, even for logical equivalence. However, a restriction on the base logic and an analysis of the other syntax-based schemes considered here is still to be done.

8.2 How Hard is it to Generate a Revised Base?

The question of how hard it is to actually generate a base has been addressed by Gogic *et al.* [1994]. One first discouraging but expected result is a result on generating a revised base under the full meet WIDTIO-scheme if only Horn logic is allowed.

Theorem 8.4 ([Gogic *et al.*, 1994]) *Generating a revised base under the full meet* WIDTIO*-scheme for Horn logic is* $\mathrm{F}\Delta_2^p[O(\log n)]$*-complete.*

The prefix "F" in $\mathrm{F}\Delta_2^p[O(\log n)]$ stands for *function* and is intended to turn a complexity class for decision problem into one for *search problems*, i.e., problems that have answers more informative than "yes" or "no" [Johnson, 1990]. One may wonder why $O(\log n)$ oracle calls are enough to generate the base since Proposition 8.1 stated a polynomial overhead.

For generating a base, however, it is enough to determine the number of "deleted" formulae, which can be computed by binary search using $O(\log n)$ oracle calls to an oracle that decides non-membership in the revised base. Then we can guess a subset of the original belief base and a truth assignment and verify in polynomial time that the set has the right size and together with the revision formula is satisfied by the guessed truth-assignment (an NP-problem).

Let us now focus on how hard it is to generate a base under the full meet base revision scheme for Horn logic. The first formal problem is that this base might be very large. However, we might now be satisfied with an **output-polynomial algorithm**, i.e., an algorithm that needs time polynomial in the output of the algorithm. Unfortunately, even this possibility can be ruled out (conditionally).

Many enumeration problems arising in Artificial Intelligence and Computer Science have the property that no output-polynomial algorithm is known for them. A prototypical example is the *hypergraph transversal problem* [Eiter and Gottlob, 1995]. As it turns out, a considerable number of these enumeration problems can be classified as being $\mathsf{TRANSVERSAL}$-hard, which means that any output-polynomial algorithm for those problems would lead to an output-polynomial problem of the *hypergraph transversal problem* [Eiter and Gottlob, 1995]. Now, generating a revised base under the full meet base revision scheme for Horn logic is one of those problems.

Theorem 8.5 ([Gogic *et al.*, 1994]) *Generating a revised base under the full meet base revision scheme for Horn logic is* TRANSVERSAL*-hard.*

As a positive result, Gogic *et al.* [1994] pointed out that Winslett's [1988] scheme (see Section 9.1) allows for reasonably behaved updates, when one wants to compute iterated Horn approximations [Selman and Kautz, 1991]. It is possible to do what Gogic *et al.* [1994] call *incremental knowledge compilation* in polynomial time if the size of the update formula is bounded in size by a constant.

9 Related Work

Related to the problem of belief base revision are the problems of *updating* a belief base, *evaluating conditionals*, and *reasoning in nonmonotonic logics.* These problems and their computational properties are briefly sketched below.

9.1 Belief Updates

As already mentioned, the problem of changing a belief base in order to keep track of the results of events in the world is called *belief update* problem [Katsuno and Mendelzon, 1992]. One main difference to belief revision is that even in the case that the new information does not contradict the original belief base, it may nevertheless be changed. Winslett's [1988; 1990] and Forbus' [1989] schemes are prototypical examples of belief updates schemes. They are defined in a similar way as the model-based revision schemes described in Section 4.1. In difference to most model-based revision schemes, however, they use a pointwise combination of models, collecting for all models of the belief base the closest models of the update formula, where distance is measured by cardinality or set-inclusion of the set of variables with differing truth values.

Although these schemes are inherently model-based, it is possible to give a proof-theoretic characterization. Del Val [1992] gave one such characterization for Winslett's scheme, which still has a model-based flavor, however. Fariñas del Cerro and Herzog [1993] provided a proof-theoretic account of Winslett's scheme by specifying a translation to classical propositional logic—which is, of course, leads to an exponential blowup. However, this translation is very efficient if the update formula is a clause or a conjunction of literals in which case the translation needs only quadratic space.

The complexity of update schemes has been investigated by Eiter and Gottlob [1992]. As it turns out, they have a similar complexity as the base revision schemes, i.e., they are Π_2^p-complete and, similarly to the model-based revision schemes, decidable in polynomial time if the logic is restricted to Horn logic and the update formula is bounded in size (cf. Table 3).

9.2 Evaluating Counterfactual Conditionals

Evaluating conditional statements such as "if φ, then ψ," where φ is false in the current context, was one of the motivations behind studying the belief revision problem in the first place [Gärdenfors, 1986; Ginsberg, 1986]. Conditionals, which are written $\varphi > \psi$, can be evaluated by using the *Ramsey Test*. This test can be roughly described as follows. $\varphi > \psi$ is accepted in a belief base describing the current context if a minimal change to the belief base to accept φ leads necessarily to the acceptance of ψ.

Based on this idea, the complexity of evaluating conditionals using different revision and update schemes has been analyzed [Eiter and Gottlob, 1992; Eiter and Gottlob, 1993; Grahne and Mendelzon, 1995]. Grahne and Mendelzon [1995] approached the problem by assuming a model-checking framework in that the belief base is represented as a set of models. Based on this assumption, they derive polynomial algorithms for evaluating conditionals under Winslett's update scheme – provided that the formula is fixed. Evaluating arbitrary (also nested) conditionals and testing for equivalence of conditionals is PSPACE-complete, however.

Eiter and Gottlob [1993] analyzed the complexity of nested conditionals assuming a variant of the full meet base revision scheme. A revised base is represented by a set of belief bases (also called a *flock* of bases [Fagin *et al.*, 1986]) consisting of all remainders extended by the revision formula. In evaluating iterated revisions – which is necessary for evaluating nested conditionals – they apply the scheme to all bases in the set. Using this approach, they show that right-nested conditionals that correspond to *iterated base revisions* have the same complexity as single revisions.[22] Arbitrarily nested conditionals can become as hard as PSPACE-complete, however.

9.3 Nonmonotonic Logics

Belief revision and nonmonotonic logics are also quite closely related. Gärdenfors [1990], for example, called belief revision and nonmonotonic logics "two sides of the same coin." In fact, a number of desirable properties of nonmonotonic logics [Kraus *et al.*, 1990; Lehman and Magidor, 1992] have direct counter-parts as postulates in belief revision and *vice versa* [Makinson and Gärdenfors, 1991]. This also translates to concrete instances of revision schemes and nonmonotonic logics. For example, cautious reasoning in Brewka's level default theories [1989; 1991] is identical to revising a belief base under the prioritized base scheme by $\top$ [Nebel, 1991]. Also inference from classically inconsistent possibilistic theories [Dubois *et al.*, 1994] and other syntax-based schemes for inference from classically inconsistent theories [Benferhat *et al.*, 1995] are identical to revising a belief base with $\top$ under one of the syntax-base schemes described above.

[22] Note that in these cases it is not necessary to compute the revised bases explicitly!

The complexity of nonmonotonic logics has been extensively analyzed. A survey is given by Cadoli and Schaerf [1993]. One interesting result in this context is the one by Gottlob [1992], who showed that the "standard" nonmonotonic logics have all the same complexity as some of the belief revision schemes described in this paper, namely, they are Π_2^p-complete.

10 Summary and Discussion

Belief revision is an important problem in Artificial Intelligence and in Computer Science. However, the theory of belief revision as developed in philosophical logic does not provide us with solutions to this problem because computational aspects have been abstracted away.

Assuming that a belief revision scheme has to be *representationally feasible*, we focussed on methods that revise a belief base using an amount of preference information with size bounded polynomially in the size of the belief base to be revised. Further, we required a certain flexibility of a revision scheme, which led us to focus on *syntax-based revision schemes.* As it turns out, most of these schemes are considerably harder than reasoning in classical propositional logic. However, it is sometimes possible to achieve computational tractability, i.e., polynomial-time decidability, by restricting the problem.

In this section, the complexity results are summarized and contrasted with complexity results for model-based schemes. Further, we try to determine the degree of syntax-sensitivity of the different schemes. Finally, we discuss the results achieved and point out further research directions.

10.1 Summary of Complexity Results

The computational complexity results from Sections 4–7 are summarized Table 1. As in the following tables, previously unpublished results are marked by a "$\star$". In addition to the complexity we also show the flexibility of a scheme by displaying which of the postulates are satisfied and whether all of the belief revision operations satisfying those postulates can be generated.

It is interesting to note that only the linear and cut base revision scheme become polynomial when the logic is restricted to Horn logic. Further, one should note that the restriction of the size of the revision formula does obviously not have an effect on the complexity. The reason for that is that we can always "hide" the revision formula in the belief base—in syntax-based revision schemes.

Table 2 gives the results for the WIDTIO-revision schemes analyzed in Section 5.6. These results are all original since either tight bounds were unknown before or the WIDTIO-scheme was not analyzed at all. As already spelled out in Section 5.6, the WIDTIO-schemes are identical to the corresponding ordinary base revision schemes concerning computational complexity. However, from a seman-

$A \odot \varphi \models \psi$?	General case		Horn logic		Postulates	
	any φ	$\lvert\varphi\rvert \leq k$	any φ	$\lvert\varphi\rvert \leq k$	satisfied	gen.
lower bound	NP- & coNP-hard		in P		$(\dot{+}1)$–$(\dot{+}6)$	?
full meet revision	coNP(3)-complete*		in P		$(\dot{+}1)$–$(\dot{+}8)$	No
full meet base	Π^p_2-complete		coNP-complete		$(\dot{+}1)$–$(\dot{+}7)$, $(\dot{+}8r)$, $(\dot{+}8c)$	Yes
prioritized						
linear	Δ^p_2-complete		in P		$(\dot{+}1)$–$(\dot{+}8)$	Yes
cardinality-max.	$\Delta^p_2[O(\log n)]$-complete*					
lexicographic	Δ^p_2-complete					
cut	$\Delta^p_2[O(\log n)]$-comp.*		in P			
safe	Π^p_2-complete*		coNP-complete*		$(\dot{+}1)$–$(\dot{+}6)$	?

Table 1: Complexity and flexibility of base revision schemes

tical point of view, there is a definite difference as is obvious from the columns about postulates.

$A \mathbin{\dot{\circ}} \varphi \models \psi$?	General case		Horn logic		Postulates	
	any φ	$\lvert\varphi\rvert \leq k$	any φ	$\lvert\varphi\rvert \leq k$	satisfied	generated
full meet base	Π^p_2-complete*		coNP-complete*		$(\dot{+}1)$–$(\dot{+}6)$	?
prioritized						
linear	Δ^p_2-complete*		in P*		$(\dot{+}1)$–$(\dot{+}8)$	Yes
cardinality-max.	$\Delta^p_2[O(\log n)]$-complete*				$(\dot{+}1)$–$(\dot{+}6)$	?
lexicographic	Δ^p_2-complete*					

Table 2: Complexity and flexibility of WIDTIO-revisions

Having now a good picture of what the complexity of syntax-based revision schemes is, it may be interesting to compare them with model-based schemes. As Table 3 [Eiter and Gottlob, 1992] shows, there are some similarities and some differences.

First of all, one notes that in contrast to syntax-based schemes, size restrictions on the revision formula have an effect. The reason for the drop in complexity when restricting the size of revision formula in model-based revision schemes is that we only have to consider a constant number of models for the revision formula.

A further interesting observation is that cardinality-based revision schemes do not lead to a drop in complexity when the logic is restricted, regardless of whether we consider syntax-based or model-based revision schemes.

$A \circ \varphi \models \psi$?	General case any φ	General case $\lvert\varphi\rvert \leq k$	Horn logic any φ	Horn logic $\lvert\varphi\rvert \leq k$	Postulates satisfied[a]
Dalal	$\Delta_2^p[O(\log n)]$-c.	NP-hard	$\Delta_2^p[O(\log n)]$-c.	in P	$(\dot{+}1)$–$(\dot{+}8)$
Satoh	Π_2^p-complete	coNP-hard & in BH	coNP-complete		$(\dot{+}1)$–$(\dot{+}7)$
Borgida					$(\dot{+}1)$–$(\dot{+}7)$, $(\dot{+}8c)$
Weber	Π_2^p-complete[b,⋆]		coNP-complete[b,⋆]		$(\dot{+}1)$–$(\dot{+}6)$

[a] In fact, the postulates hold only if A and φ are satisfiable.

[b] These results that are tighter than in [Eiter and Gottlob, 1992] can be achieved using the same technique as in Lemma 5.20.

Table 3: Complexity of model-based revision schemes

10.2 The Degree of Syntax-Sensitivity of Base Revision Schemes

The *pros* and *cons* of subscribing to a syntax-based view on revision have already been discussed in Section 4.2. On one hand, it gives us more flexibility from a pragmatic and formal point of view. On the other hand, it makes the result dependent on the syntactic realization. However, not all pieces of the syntax contribute to the revision and some revision schemes are more sensitive to syntax than other. For example, the cardinality-based schemes are all sensitive to the addition of logically equivalent formulae, while the other schemes are insensitive to that.

Benferhat *et al.* [1995] classified nonmonotonic consequence relations according to different properties and we will adopt that classification here for base revision schemes. We will say that a formula φ implied by a prioritized base A has an **argument of priority** i iff there exists an entailment set for φ such that the minimal element in it has priority i. Further, we say that φ is of **priority** j iff the argument with the highest priority has priority j. Finally, we define a **prioritized expansion** of a belief set by $A +_i \varphi$ with the meaning that φ is added to priority class i.

Now let us consider the following properties:

Formula equivalence insensitivity (FEI): Let A and A' be two bases such that for all priority classes A_i and A'_i we have $\|A_i\| = \|A'_i\|$ and for all $\chi \in A_i$ there is a formula $\chi' \in A'_i$ with $\models \chi \leftrightarrow \chi'$ and *vice versa.* A base revision scheme $\odot$ is then said to be an **FEI-scheme** iff for all $\varphi \in \mathcal{L}$

$$Cn(A \odot \varphi) = Cn(A' \odot \varphi). \tag{62}$$

Class equivalence insensitivity (CEI): Let A and and A' be two bases such that for all priority classes $Cn(A_i) = Cn(A'_i)$. Then a base scheme $\odot$ is said to be a **CEI-scheme** iff for all $\varphi \in \mathcal{L}$

$$Cn(A \odot \varphi) = Cn(A' \odot \varphi). \tag{63}$$

Redundancy insensitivity (RI): Let $\psi \in A_i$ and let $\models \psi \leftrightarrow \chi$. Then a base scheme $\odot$ is said to be an **RI-scheme** iff for all $\varphi \in \mathcal{L}$

$$Cn(A \odot \varphi) = Cn((A +_i \chi) \odot \varphi). \qquad (64)$$

Local consequence insensitivity (LCI): Let $A \models \psi$ with ψ having an argument of priority i and let $\models \psi \leftrightarrow \chi$. Then a base scheme $\odot$ is said to be an **LCI-scheme** iff for all $\varphi \in \mathcal{L}$

$$Cn(A \odot \varphi) = Cn((A +_i \chi) \odot \varphi). \qquad (65)$$

Possibilistic consequence insensitivity (PCI): Let $A \models \psi$ with ψ having priority i and let $\models \psi \leftrightarrow \chi$. Then a base scheme $\odot$ is said to be a **PCI-scheme** iff for all $\varphi \in \mathcal{L}$ such that $A \circledE \varphi \models \psi$:

$$Cn(A \odot \varphi) = Cn((A +_i \chi) \odot \varphi). \qquad (66)$$

Table 4 summarizes the above defined properties for all base revision schemes that incorporate priorities. The "Yes" entries are almost immediate consequences of the definitions of the schemes while the "No" entries can be verified with simple counter-examples (for full proofs see [**?**]).

Scheme	FEI	CEI	RI	LCI	PCI
prioritized	Yes	No	Yes	No	Yes
linear	Yes	Yes	Yes	No	Yes
lexicographic	Yes	No	No	No	Yes
cut	Yes	Yes	Yes	Yes	Yes
safe[a]	Yes	No	Yes	No	Yes

[a]Assuming that the hierarchy corresponds to priority classes.

Table 4: Syntax-sensitivity of base revision schemes

As Table 4 shows, the weakest form of insensitivity FEI is always satisfied. Although this might be regarded as a triviality, one could think of base revision schemes that use weaker logics under which classically equivalent formulae are not equivalent.

Further, we note that there seems to be a continuum of revision schemes regarding syntax sensitivity instead of a sharp distinction between syntax-sensitive and syntax-insensitive schemes [Benferhat *et al.*, 1995]. The most syntax-insensitive scheme according to this classification is the cut-base revision scheme, which for this reason should hardly be called syntax-based.

10.3 Discussion

The extensive complexity analysis of different variants of base revision schemes and related problems has a value in itself, because it provides us with many natural problems located in the lower end of the polynomial hierarchy—something which was thought to be unlikely [Stockmeyer, 1987]. However, it also helps us to relate it to other, similar problems, to identify sources of complexity, and to identify subproblems that are easily solvable [Nebel, 1996]. In summary, we get a much better understanding of the problem and its hard and easy to solve aspects, which helps us to base design decision on a firm ground.

The general revision problem for propositional logic appears to be hopelessly infeasible from a computational point of view because they are located on the second level of the polynomial hierarchy. Interesting research problems in the area of computational approaches to belief revision that have not been solved yet are the identification of methods that attack the revision problems that are in the first level of the polynomial hierarchy, i.e., in Δ_2^p. For instance, it seems to be interesting to apply the GSAT method [Selman *et al.*, 1992; Selman and Kautz, 1993] to the revision problems in Δ_2^p. Another interesting avenue of research is the work by Gogic *et al.* [1994], who combined the ideas of knowledge compilation by Horn approximation [Selman and Kautz, 1991] and belief revision, showing that the combination can be computationally feasible under some conditions. Other forms to approximate the result of a base revision may be perhaps also applicable [Cadoli and Schaerf, 1995]. Finally, it may interesting to evaluate the empirical efficiency of complete (and exponential-time) algorithms for some of the revision problems, which might be reasonable for moderately sized revision instances.

Acknowledgements

I would like to thank Salem Benferhat, Didier Dubois, Dov Gabbay, Henri Prade, and Hans Rott for discussion on the subject of this paper, and Thomas Eiter, Georg Gottlob, and Werner Nutt for some hints on computational complexity issues.

This work has been supported by the European Commission as part of DRUMS-II, the ESPRIT Basic Research Project P6156.

Appendix A: Proofs

Theorem 4.3 *Full meet revision is* $\mathsf{coNP}(3)$*-complete.*

Proof. We use the $\mathsf{UNSAT}(3)$ problem, which can be defined as follows:

$$\mathsf{UNSAT}(3) \stackrel{\text{def}}{=} \{\langle\varphi_1, \varphi_2, \varphi_3\rangle \mid \varphi_1 \in \mathsf{UNSAT} \vee (\varphi_2 \in \mathsf{SAT} \wedge \varphi_3 \in \mathsf{UNSAT})\}. \quad (67)$$

This problem is $\mathsf{coNP}(3)$-complete as follows from the results by Cai et al [1988].

For membership, note that implication of ψ by the revised belief set in (25) for a full meet revision operation can be decided by the following $\mathsf{UNSAT}(3)$ instance:

$$(\varphi \wedge \neg\psi) \in \mathsf{UNSAT} \text{ or } ((\bigwedge A \wedge \varphi) \in \mathsf{SAT} \text{ and } (\bigwedge A \wedge \varphi \wedge \neg\psi) \in \mathsf{UNSAT}). \quad (68)$$

Hardness is proved by reducing $\mathsf{UNSAT}(3)$ to full meet revision. Let $\langle\varphi_1, \varphi_2, \varphi_3\rangle$ be an $\mathsf{UNSAT}(3)$ instance and assume without loss of generality that the formulae do not have propositional variables in common. Further, let

$$\begin{aligned} \psi &= \neg(\varphi_1 \wedge (p \vee \varphi_3)), && (69)\\ A &= \{\varphi_2 \wedge q\}, && (70)\\ \varphi &= \neg p \vee \neg q, && (71) \end{aligned}$$

where p and q are variables not appearing in φ_i. According to (68), the membership relation $\psi \in Cn(A) \overset{F}{+} \varphi$ is equivalent to

$$\begin{aligned} &((\neg p \vee \neg q) \wedge (\varphi_1 \wedge (p \vee \varphi_3))) \in \mathsf{UNSAT} \text{ or} && (72)\\ &\quad \big((\varphi_2 \wedge q \wedge (\neg p \vee \neg q)) \in \mathsf{SAT}) \text{ and} \\ &\quad (\varphi_2 \wedge q \wedge (\neg p \vee \neg q) \wedge \varphi_1 \wedge (p \vee \varphi_3)) \in \mathsf{UNSAT}\big), \end{aligned}$$

which in turn is equivalent to

$$\varphi_1 \in \mathsf{UNSAT} \text{ or } (\varphi_2 \in \mathsf{SAT} \text{ and } \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \in \mathsf{UNSAT}). \quad (73)$$

Finally, condition (73) is true if and only if

$$\varphi_1 \in \mathsf{UNSAT} \text{ or } (\varphi_2 \in \mathsf{SAT} \text{ and } \varphi_3 \in \mathsf{UNSAT}) \quad (74)$$

is true, i.e. iff

$$\langle\varphi_1, \varphi_2, \varphi_3\rangle \in \mathsf{UNSAT}(3). \quad (75)$$

■

Theorem 5.6 *Deciding $A \circledP \varphi \models \psi$ is Π_2^p-complete for general propositional logic and coNP-complete for Horn logic. This also holds under the assumption that the size of the revision formula is bounded by a constant.*

Proof.

The hardness result follows from Theorem 5.4 since full meet base revision is a special case of prioritized base revision.

Membership of $A \circledP \varphi \not\models \psi$ in Σ_2^p follows from the following algorithm that needs nondeterministic polynomial time using an oracle for SAT [Nebel, 1992, Theorem 20]:

1. Guess a set $B \subseteq A$.

2. Verify that $B \in A \downarrow \neg\varphi$:

   (a) Verify consistency of B with φ using one SAT-oracle call: $B \not\models \neg\varphi$.

   (b) Verify that B is $\ll$-preferred using $O(|A|)$ SAT-oracle calls: For all priority classes A_i there is no $\chi \in A_i - B$ such that

$$\bigcup_{j=i}^{n}(B \cap A_j) \cup \{\chi\} \not\models \neg\varphi. \tag{76}$$

3. Verify non-implication of ψ using one SAT-oracle call: $B \cup \{\varphi\} \not\models \psi$.

Hence, it follows that the complementary problem is in Π_2^p. For Horn logic all SAT-oracle calls can be replaced by a procedure that decides satisfiability of Horn logic, which results in an NP-algorithm. ∎

Theorem 5.8 *The class of revision operations generated by the linear base revision scheme coincides with the class of revision operations satisfying ($\dot{+}$1)–($\dot{+}$8).*

Proof. Because of Theorem 5.7 we only have to show that all "fully rational" revisions can be generated. This follows, however, from Lemma 6.11, which states that all revision operations generated by cut base revisions can be generated, which by Proposition 6.4 coincide with the class of "fully rational" revisions. ∎

Theorem 5.9 *The linear base revision scheme is Δ_2^p-complete. This holds even when the revision formula is bounded in size by a constant.*

Proof. Membership of deciding $A \circledL \varphi \models \psi$, where A contains n priority classes, follows from the following algorithm:

1. Initialize $B = \emptyset$ and $i = n$.

2. Test $B \cup A_i \not\models \neg\varphi$. If so, set $B = B \cup A_i$.

3. Decrement i.

4. If $i = 0$ return with the result $(B \cup \{\varphi\} \models \psi)$.

5. Otherwise continue with step 2.

Using an oracle for SAT, this algorithms runs in polynomial time. Thus, linear base revision is in Δ_2^p.

Hardness follows by a reduction from the Δ_2^p-complete problem MAX-SAT-ASG$_{odd}$ [Wagner, 1987], which can be defined as follows:

> Given a propositional formula χ in conjunctive normal-form (or a set of clauses) over the propositional variables $p_1, \ldots, p_n$ and a weight function W over truth assignments $\alpha\colon \{p_1, \ldots, p_n\} \to \{0, 1\}$ defined by $W(\alpha) \stackrel{\text{def}}{=} \Sigma_i \alpha(p_i) \times 2^{i-1}$, has the truth-assignment that satisfies χ with a maximal weight an *odd* weight value?

Given a formula χ over the variables $p_1, \ldots, p_n$, construct the following linear base revision instance using the following new variables q, r:

$$
\begin{aligned}
A &= \langle A_1, \ldots, A_n, A_{n+1} \rangle, && (77)\\
A_i &= \{p_i\} \text{ for } 1 \leq i \leq n, && (78)\\
A_{n+1} &= \{q \to (\chi \wedge r)\}, && (79)\\
\varphi &= q, && (80)\\
\psi &= p_1 \wedge r. && (81)
\end{aligned}
$$

Obviously, $A \textcircled{\scriptsize L} \varphi \models \psi$ iff the satisfying truth-assignment with the maximal weight has an odd weight value. Hence linear base revision is Δ_2^p-hard. ■

Theorem 5.10 *For Horn logic, $A \textcircled{\scriptsize L} \varphi \models \psi$ can be decided in $O(n^2)$ time, where $n = |A| + |\varphi| + |\psi|$.*

Proof. Since satisfiability of Horn formulae can be decided in linear time [Dowling and Gallier, 1984], each of the satisfiability tests in step 2 in the algorithm of Theorem 5.9 require at most $O(n)$ time. Since there are at most n priority classes, we need at most n such satisfiability tests. Deciding implication in step 4 can be done in at most $O((|A| + |\varphi|) \times |\psi|)$ time, i.e., at most $O(n^2)$ time. Thus, overall we need $O(n^2)$ time. ■

Theorem 5.11 *Theorem 5.6 also holds under the assumption that the cardinality of the priority classes is bounded by a constant $k \geq 2$.*

Proof. Membership is obvious. For the hardness part note that the reductions in the proofs of Theorem 19 in [Nebel, 1992] (maybe together with Corollary 15 to obtain a consistent belief base) or Lemma 6.2 in [Eiter and Gottlob, 1992] for the general case and Lemma 7.1 in [Eiter and Gottlob, 1992] for the Horn case can be reused. The only important point under the assumption that $k \geq 2$ formulae can be in one priority class is that the atomic formulae representing different truth values, e.g., x_i and y_i in the latter Lemma, have to be put in the same priority class. The other formulae can be placed in priority classes such that the effect of the reduction is not affected. ■

Lemma 5.12 *Any revision generated by the linear base revision scheme can be generated by the cardinality-maximizing base revision scheme.*

Proof. Given a prioritized base $A = \langle A_1, \ldots, A_n \rangle$, we specify a function π that generates a new base B as follows:

$$B = \{A_1 \times 2^0, \ldots, A_n \times 2^n\}, \tag{82}$$

where $A_i \times 2^{i-1}$ denotes 2^{i-1} syntactically different copies of A_i (by adding, e.g., "$\wedge\top$" 2^{i-1} times).[23] Obviously, this translation leads to

$$Cn(A \circledL \varphi) = Cn(\pi(A) \circledcirc \varphi) \tag{83}$$

for all $\varphi \in \mathcal{L}$. ∎

Theorem 5.14 *Deciding $A \circledcirc \varphi \models \psi$ is $\Delta_2^p[O(\log n)]$-complete, and this holds even if the size of the revision formula is bounded by a constant.*

Proof. In order to prove membership in $\Delta_2^p[O(\log n)]$, we first introduce the MAX-SAT problem:

> Given a set of propositional formulae C, a formula χ, and a positive integer k, is there a subset B of C with at least k elements such that $\bigwedge B \wedge \psi$ is satisfiable?

This is clearly a generalization of SAT, hence NP-hard. It is in NP because by guessing a subset and a truth-assignment, and checking that the truth assignment satisfies $\bigwedge B \wedge \psi$ and that $\|B\| \geq k$, the problem can be decided in non-deterministic polynomial time.

Membership of $A \circledL \varphi \not\models \psi$ in $\Delta_2^p[O(\log n)]$ is now demonstrated by the following algorithm:

1. Determine the largest k such that there is a k-element subset of A that is consistent with φ by using binary search, resulting in $O(\log n)$ calls to a MAX-SAT-oracle.

2. Make one call to the MAX-SAT-oracle to check whether there is a k-element subset $B \subseteq A$ that is consistent with $\varphi \wedge \neg\psi$, i.e., $B \cup \{\varphi\} \not\models \psi$.

Since the class $\Delta_2^p[O(\log n)]$ is closed under complements, the complementary problem of deciding implication of ψ is also in $\Delta_2^p[O(\log n)]$.

Hardness is proved using the $\Delta_2^p[O(\log n)]$-complete problem UOCSAT [Kadin, 1989]:

> Given a set of clauses C, decide if all truth-assignments that satisfy a maximum number of clauses in C always satisfy the same clauses.

[23] This is not a polynomial-time reduction. However, here we are not concerned about resource bounds, but only whether it is possible to generate a revision operation *in principle*.

Given an instance of UOCSAT $C = \{\chi_1, \ldots, \chi_m\}$ over the variables $p_1, \ldots, p_n$, we construct the following revision instance containing copies of the χ_i's, denoted by $\chi'_1, \ldots, \chi'_m$ using the variables $p'_1, \ldots, p'_n$, and the new variables $q_1, \ldots, q_m$, $q'_1, \ldots, q'_m$, r, s:

$$\begin{aligned} A &= \Big\{r \rightarrow \chi_1, \ldots, r \rightarrow \chi_m, r \rightarrow \chi'_1, \ldots, r \rightarrow \chi'_m, \\ &\quad \bigwedge_{i=1}^{m} (\chi_i \leftrightarrow q_i), \bigwedge_{i=1}^{m} (\chi'_i \leftrightarrow q'_i), \Big(\bigwedge_{i=1}^{m} (q_i \leftrightarrow q'_i)\Big) \rightarrow s\Big\} \end{aligned} \tag{84}$$

$$\varphi = r \tag{85}$$

$$\psi = s \tag{86}$$

Obviously, this revision instance can be constructed in time polynomial in $|C|$.

First of all, note that the formulae $\bigwedge_i(\chi_i \leftrightarrow q_i), \bigwedge_i(\chi'_i \leftrightarrow q'_i), (\bigwedge_i(q_i \leftrightarrow q'_i)) \rightarrow s$ are included in all elements of $(A \perp\!\!\!\perp \neg\varphi)$. Secondly, note that the subset of formulae from $\{(r \rightarrow \chi_i)\}$ that remain in $(A \perp\!\!\!\perp \neg\varphi)$ constitutes a subset of C with a maximum number of satisfiable clauses. The same holds, of course, for the "primed" version of the formulae $\{r \rightarrow \chi'_i\}$. Since the primed and the unprimed version are independent of each other, $\bigwedge_i(q_i \leftrightarrow q'_i)$ (and, hence, s) are valid over all sets $(A \perp\!\!\!\perp \neg\varphi)$ joined with φ iff $(A \perp\!\!\!\perp \neg\varphi)$ contains only one element, i.e., if there is only one subset with a maximal number of satisfiable clauses. Hence, $A \perp\!\!\!\perp \varphi \models \psi$ iff $C \in$ UOCSAT, which proves the hardness claim. ■

Theorem 5.15 *Deciding $A \circledcirc \varphi \models \psi$ is $\Delta_2^p[O(\log n)]$-complete for Horn logic, and this holds even if the size of the revision formula is bounded by a constant.*

Proof. Membership in $\Delta_2^p[O(\log n)]$ follows from Theorem 5.14.

For the hardness part, let the set $C = \{\chi_1, \ldots, \chi_m\}$ of clauses be an instance of UOCSAT, where the variables used in C are $\{p_1, \ldots, p_n\}$. Then we construct in polynomial time the following revision instance containing in addition to C a set of clauses $C' = \{\chi'_1, \ldots, \chi'_n\}$, which are copies of the clauses χ_i using new variables $p'_1, \ldots, p'_n$, Further, we use new variables $q_1, \ldots, q_n$, $q'_1, \ldots, q'_n$, $r_1, \ldots, r_m$, $r'_1, \ldots, r'_m$, s. The notation $\chi_i[+p/\neg q]$ means that all positive occurrences of p_j are replaced by $\neg q_j$ in order to turn χ_i into a Horn formula [Eiter and Gottlob, 1992, Lemma 7.1]. Further, as in the proof of Lemma 5.12, we use the notation $\varphi \times n$ to mean n syntactically different copies of φ.

$$\begin{aligned} A = \{&p_1 \times (2m+1), \ldots, p_n \times (2m+1), \\ &p'_1 \times (2m+1), \ldots, p'_n \times (2m+1), \\ &r_1 \times 2, \ldots, r_m \times 2, \\ &r'_1 \times 2, \ldots, r'_m \times 2, s\} \end{aligned} \tag{87}$$

$$\begin{aligned}
\varphi &= \bigwedge_{i=1}^{m} (r_i \rightarrow \chi_i[+p/\neg q]) \wedge \bigwedge_{i=1}^{m} (r_i' \rightarrow \chi_i'[+p/\neg q]) \wedge \\
&\quad \bigwedge_{j=1}^{n} (\neg p_j \vee \neg q_j) \wedge \bigwedge_{j=1}^{n} (\neg p_j' \vee \neg q_j') \wedge \\
&\quad \bigwedge_{i=1}^{m} (((s \wedge r_i) \rightarrow r_i') \wedge ((s \wedge r_i') \rightarrow r_i)) & (88) \\
\psi &= s & (89)
\end{aligned}$$

As is easily seen, all formulae are Horn and the entire reduction can be performed in polynomial time.

In order to see that $A \circledcirc \varphi \models \psi$ iff $C \in \mathsf{UOCSAT}$, note that by $\bigwedge_j (\neg p_j \vee \neg q_j)$ in the revision formula at most one of $p_j \times (2m+1)$ and $q_j \times (2m+1)$ can be valid in each remainder $B \in (A \perp\!\!\!\perp \neg\varphi)$. Further, since removing $p_j \times (2m+1)$ and $q_j \times (2m+1)$ leads to a set with less formulae (even if all r_i's and s stay) than a set containing one of the variables, all remainders $B \in (A \perp\!\!\!\perp \neg\varphi)$ contain exactly one of the variables, which can be used to construct a truth-assignment for the χ_i's. Obviously, each remainder B contains a maximal number of r_i's corresponding to clauses satisfied by the truth-assignment.

All these arguments are, of course, valid for the primed versions of the formulae. Since the primed and unprimed versions are independent of each other, s can stay if, and only if, always the same subsets of formulae from C are selected for the primed and unprimed version, i.e., if $C \in \mathsf{UOCSAT}$.

In order to show that hardness also holds for an atomic revision formula, let t be a new variable and φ_t the formula φ where each clause φ_l in φ is replaced by $\neg t \vee \varphi_l$ (see [Eiter and Gottlob, 1992, Theorem 8.4]). Given the UOCSAT-instance C, we construct a revision instance using the reduction above modifying it as follows:

$$\begin{aligned}
A' &= A \cup \{\varphi_t \times (\|A\| + 1)\} & (90) \\
\varphi' &= t & (91) \\
\psi' &= \psi & (92)
\end{aligned}$$

We have obviously $A \circledcirc \varphi \models \psi$ iff $A' \circledcirc \varphi' \models \psi'$. ∎

Theorem 5.17 *Deciding $A \otimes \varphi \models \psi$ is Δ_2^p-complete, and this holds even if the size of the revision formula is bounded by a constant.*

Proof. Membership follows by replacing the $\mathsf{MAX\text{-}SAT}$-oracle calls in the membership proof in Theorem 5.14 by $\mathsf{MAX\text{-}LEVEL\text{-}SAT}$-oracle calls, where the latter problem is defined as follows:

> Given a sequence of sets of formulae $C_1, \ldots, C_l$, a sequence of positive integers $k_1, \ldots, k_l$, and a formula χ, does there exist a sequence of subsets $B_1, \ldots, B_l$ such that $B_i \subseteq C_i$ and $\|B_i\| \geq k_i$ for each $1 \leq i \leq l$ and $\bigcup_i B_i \cup \{\chi\}$ is satisfiable?

Further, step 1 in the algorithm in the proof of Theorem 5.14 has to be repeated for each priority class, leading to $O(n \times \log n)$ oracle calls instead of $O(\log n)$ calls.

Hardness follows from Theorem 5.9 because linear base revision is a special case of lexicographic base revision, where each priority class has just one element. ∎

Theorem 5.18 *Deciding $A \otimes \varphi \models \psi$ is Δ_2^p-complete for Horn logic, and this holds even if the size of the revision formula is bounded by a constant.*

Proof. Membership follows from Theorem 5.17. Hardness will be shown using the Δ_2^p-complete problem $\mathsf{MAX\text{-}SAT\text{-}ASG}_{odd}$ introduced in the proof of Theorem 5.9.

Let $C = \{\chi_1, \ldots, \chi_m\}$ be a set of clauses over the variables $p_1, \ldots, p_n$. Let $q_1, \ldots, q_n, r_1, \ldots, r_n, s, t$ be new variables, and let $\chi_i[+p/\neg q]$ be the formula χ_i with all positive occurrences of p_j be replaced by $\neg q_j$, which means that it is a Horn formula. Now we construct the following revision instance:

$$\begin{aligned}
A &= \langle A_1, \ldots, A_{n+2} \rangle && (93)\\
A_{n+2} &= \Big\{ (s \wedge p_1 \wedge r_1 \wedge \ldots \wedge r_n \rightarrow t) \wedge && (94)\\
&\quad \bigwedge_{j=1}^{n} \Big((\neg s \vee \neg p_j \vee \neg q_j) \wedge (p_j \rightarrow r_j) \wedge (q_j \rightarrow r_j) \Big) \wedge \\
&\quad \bigwedge_{i=1}^{m} (s \rightarrow \chi_i[+p/\neg q]) \Big\}, \\
A_{n+1} &= \{p_1, \ldots, p_n, q_1, \ldots, q_n\}, && (95)\\
A_i &= \{p_j\} \text{ for } 1 \leq j \leq n && (96)\\
\varphi &= s && (97)\\
\psi &= t && (98)
\end{aligned}$$

The idea behind this reduction is that the uppermost priority class encodes the clause set and some tests and the next class encodes truth-assignments in a way such that we have the same cardinality for all satisfying assignments. The classes A_1 up to A_n are then used to single out the truth-assignment with the highest weight.

By $\bigwedge_j ((\neg s \vee \neg p_j \vee \neg q_j) \ldots)$, we enforce that all remainders $B \in (A \Downarrow \neg\varphi)$ can contain at most one of p_j and q_j. If at least one of such a pair is in a remainder, r_j is implied in this remainder by $\bigwedge_j ((\ldots (p_j \rightarrow r_j) \wedge (q_j \rightarrow r_j))$. If exactly one of (p_j, q_j) for each j is present in the remainder and p_1 is present, then t—the query formula—is implied, which is enforced by $(s \wedge p_1 \wedge r_1 \wedge \ldots \wedge r_n \rightarrow t)$.

First of all, it is obvious that A_{n+2} is consistent with s, i.e., it is in each remainder $B \in (A \Downarrow \neg\varphi)$. Further, if all χ_i's are simultaneously satisfiable, then each satisfying truth-assignment results in a subset $\{l_1, \ldots, l_n\}$ of A_{n+1} with

$l_j = p_j$ or $l_j = q_j$ that is consistent with the $\chi_i[+p/\neg q]$s. For each of these subsets the classes A_n down to A_1 are added, if consistent. Now, the satisfying truth-assignment with the highest weight value leads obviously to a set that is $\lll$-preferred among all possible consistent subbases of A. If this preferred set contains p_1 (corresponding to an odd weight value), then by the rule $(s \wedge p_1 \wedge r_1 \wedge \ldots \wedge r_n \rightarrow t)$, the query formula is implied. If C is unsatisfiable, some of the r_j's will not be implied, hence the query-formula will not be implied.

Conversely, a $\lll$-preferred consistent subset that implies t is obviously a set that corresponds to a satisfying truth-assignment such that the assignment has the highest possible weight and the weight value is odd. ∎

Lemma 5.19 *For full meet, prioritized, linear, cardinality-maximizing, and lexicographic base revision schemes, the corresponding* WIDTIO*-revision schemes are as hard as the base revision schemes.*

Proof. In the hardness proofs for *full meet base revision* and *prioritized base revision* (cf. [Nebel, 1992, Theorem 19],[Eiter and Gottlob, 1992, Lemma 6.2, Lemma 7.1, Theorem 8.2]) the query-formula is part of the base to be revised. Hence, the query-formula is in the intersection of the logical closures of the remainders iff it is in the intersection of the remainders.

Since $(A \circledL \varphi)$ is always a singleton set, the *linear base revision scheme* is identical to the a linear WIDTIO-revision scheme. Thus, the complexity is identical (Theorems 5.9 and 5.10).

For the *cardinality-maximizing base revision scheme* note that in the first hardness proof (Theorem 5.14) the query-formula is implied iff the remainder set is a singleton set. Hence, the reduction works as well for the corresponding WIDTIO-revision scheme. In the second hardness proof (Theorem 5.15), the query-formula is again part of the belief base, hence the above arguments apply.

The first hardness proof in the case of *lexicographic base revision* (Theorem 5.17) uses the hardness proof of Theorem 5.9, in which the query formula is an element of the belief base, hence the above argument apply. The second hardness proof (Theorem 5.18) is again a reduction that is based on a singleton remainder set. ∎

Lemma 5.20 *For full meet, prioritized, linear, cardinality-maximizing, and lexicographic base revision schemes, the corresponding* WIDTIO*-revision schemes are in the same complexity class as the base revision schemes.*

Proof. Membership of *full meet* WIDTIO*-revision* in Π_2^p is demonstrated by the following algorithm for deciding $A \overset{F}{\circ} \varphi \not\models \psi$:

1. Guess a positive integer $k \leq \|A\|$.

2. Guess k sets $B_1, \ldots, B_k \subseteq A$ and formulae $\chi_1, \ldots, \chi_k$.

3. Verify for each j that $\chi_j \notin B_j$.

4. Verify for each j that $B_j \in (A \perp \neg\varphi)$:

   (a) Verify $B_j \not\models \neg\varphi$ using one SAT-oracle call.

   (b) Verify for each $\omega \in A - B_j$ that $B_j \cup \{\omega\} \models \neg\varphi$ using $\|A - B_j\|$ SAT-oracle calls.

5. Verify that $A - \{\chi_1, \ldots, \chi_k\} \cup \{\varphi\} \not\models \psi$ using a SAT-oracle.

Membership of *prioritized* WIDTIO-*revision* in Π_2^p can be shown in a similar way. In step 4, we have to verify that the guessed sets are in $A \downarrow \neg\varphi$ instead of checking against $A \perp \neg\varphi$. However, this can also be done in deterministic polynomial time using a SAT-oracle (see proof of Theorem 5.6).

For *linear* WIDTIO-*revision*, the claim follows since it is equivalent to the linear base revision scheme.

Membership of *cardinality-maximizing* WIDTIO-*revision* in $\Delta_2^p[O(\log n)]$ is demonstrated by the following algorithm that decides the non-implication problem:

1. Determine the maximal k such that there is a k-element subset B of A that is consistent with φ using $O(\log n)$ calls to a MAX-SAT-oracle.

2. Guess a positive integer $m \leq \|A\|$ and subsets $B_1, \ldots, B_m \subseteq A$, formulae $\chi_1, \ldots, \chi_m \in A$ and truth-assignments $\alpha_1, \ldots, \alpha_m, \beta$.

3. Verify in polynomial time that $\|B_j\| = k$, that $\chi_j \notin B_j$, and that α_j satisfies $B_j \cup \{\varphi\}$.

4. Verify in polynomial time that $A - \{\chi_1, \ldots, \chi_m\} \cup \{\varphi, \neg\psi\}$ is satisfied by β.

Membership of *lexicographic* WIDTIO-*revision* in Δ_2^p follows from the algorithm above and the same argument as used in the proof of Theorem 5.17. ■

Theorem 6.5 *Deciding $A \circledE \varphi \models \psi$ is $\Delta_2^p[O(\log n)]$-complete, and this holds even if the size of the revision formula is bounded by a constant.*

Proof. Membership follows because $cut_<(\neg\varphi)$ can be computed using binary search over the priority classes employing a satisfiability test for each step in the search, resulting in $O(\log n)$ calls to a SAT-oracle. The final problem of deciding $cut_<(\neg\varphi) \cup \{\varphi\} \models \psi$ can be decided using another call to the SAT-oracle.

Hardness will be shown using the following $\Delta_2^p[O(\log n)]$-complete problem from [Wagner, 1990]:

Given a sequence $\chi_1, \ldots, \chi_n$ of propositional formulae such that $\chi_i \notin \mathsf{SAT}$ implies $\chi_{i+1} \notin \mathsf{SAT}$, is the maximum index i such that $\chi_i \in \mathsf{SAT}$ an *odd* number?

Given an instance of this problem such that without loss of generality n is even, construct the following revision instance using $p_1, \ldots, p_n$ and q as new variables:

$$\begin{aligned}
A &= \langle A_1, \ldots, A_n, A_{n+1} \rangle, && (99)\\
A_i &= \{p_i\} \text{ for } 1 \le i \le n, && (100)\\
A_{n+1} &= \Big\{ q \rightarrow \bigwedge_{j=1}^{n} (p_j \rightarrow \chi_j) \Big\}, && (101)\\
\varphi &= q, && (102)\\
\psi &= \bigvee_{k=0}^{(n-1)/2} \Big(\bigwedge_{l=1}^{2k+1} p_l \wedge \bigwedge_{l=2k+2}^{n} \neg p_l \Big). && (103)
\end{aligned}$$

Obviously, $A \circledE \varphi \models \psi$ iff the maximum index of $\chi_i \in \mathsf{SAT}$ is odd for instances of the given problem, i.e., revisions based on ensconcements are $\Delta_2^p[O(\log n)]$-hard. ■

Theorem 6.6 *Provided A is a set of Horn formulae, φ is a Horn formula and ψ is a Horn clause, $A \circledE \varphi \models \psi$ can be decided in time $O(n \log n)$, where $n = |A| + |\varphi| + |\psi|$.*

Proof. Using the algorithm sketched in the proof of Theorem 6.5, computing $cut_<(\neg\varphi)$ can be evidently done in $O(n \log n)$ time. Additionally, the problem of deciding $cut_<(\neg\varphi) \cup \{\varphi\} \models \psi$ can be done in $O(n)$ time, if ψ is a Horn clause.[24] ■

Theorem 6.7 *Let $\le$ be an arbitrary total preorder over a belief base A. Then (1) the relation $\preceq_\le$ generated by (41) and (42) is an epistemic entrenchment over $Cn(A)$ and (2) its restriction to A, denoted by $\trianglelefteq$, is an ensconcement ordering on A.*

Proof. First we prove (1).

($\preceq$1): Assume $\varphi \preceq_\le \psi$ and $\psi \preceq_\le \chi$. By (42) we must have $cut_<(\varphi) \supseteq cut_<(\psi)$ and $cut_<(\psi) \supseteq cut_<(\chi)$, hence $cut_<(\varphi) \supseteq cut_<(\chi)$, hence $\varphi \preceq_\le \chi$.

($\preceq$2): Assume $\varphi \models \psi$ and suppose for contradiction that $cut_<(\varphi) \subset cut_<(\psi)$. By the definition of $cut_<$, we must have $cut_<(\psi) \models \varphi$ and by applying our assumption $cut_<(\psi) \models \psi$, which is a contradiction. So, $cut_<(\varphi) \supseteq cut_<(\psi)$ and by (42) $\varphi \preceq_\le \psi$, as desired.

($\preceq$3): Suppose for contraction that $cut_<(\varphi) \subset cut_<(\varphi \wedge \psi)$ and $cut_<(\psi) \subset cut_<(\varphi \wedge \psi)$. This, however, implies by the definition of $cut_<$ that $cut_<(\varphi \wedge \psi) \models \varphi$

[24] For the last step, we would need $O((|A| + |\varphi|) \times |\psi|)$ time, if ψ were a Horn *formula*.

and $cut_<(\varphi \wedge \psi) \models \psi$, hence $cut_<(\varphi \wedge \psi) \models \varphi \wedge \psi$, which is a contradiction. Hence, we must have $cut_<(\varphi) \supseteq cut_<(\varphi \wedge \psi)$ or $cut_<(\psi) \supseteq cut_<(\varphi \wedge \psi)$, and by the definition of $\preceq_\leq$, $\varphi \preceq_\leq (\varphi \wedge \psi)$ or $\psi \preceq_\leq (\varphi \wedge \psi)$, as desired.

($\preceq$4): Assume $A \not\models \bot$. If $A \not\models \varphi$ then $cut_<(\varphi) = A$, hence φ is minimal under $\preceq_\leq$. Conversely, if φ is a minimal element under $\preceq_\leq$, then $cut_<(\varphi)$ must be maximal, i.e, equal to A, and so $A \not\models \varphi$.

($\preceq$5): Follows immediately from Definition (41).

For proving (2), we show that for the generated ensconcement ordering $\trianglelefteq$ (PCC) is satisfied for arbitrary $\varphi \in A$ and $B \subseteq A$ such that $B \models \varphi$. Since $\bigwedge B \models \varphi$, it follows by ($\preceq$2) that $\bigwedge B \trianglelefteq \varphi$ and by ($\preceq$3) there must be $\psi \in B$ such that $\psi \trianglelefteq \bigwedge B$, so by transitivity ($\preceq$1) there must be a formula $\psi \in B$ such that $\psi \trianglelefteq \varphi$, hence (PCC) is satisfied. ■

Lemma 6.8 *Let $\leq$ be an arbitrary total preorder on the belief base A and $\trianglelefteq$ the restriction of the epistemic entrenchment ordering generated by $\leq$ to A. Then for all $\varphi \in \mathcal{L}$*

$$Cn(cut_<(\varphi)) = Cn(cut_\triangleleft(\varphi)). \tag{104}$$

Proof. "$\subseteq$": Let $K = Cn(cut_<(\varphi))$. By the construction of $\preceq_\leq$, it holds that for all $\psi \in K$: $\varphi \prec_\leq \psi$, so for all $\psi \in (K \cap A)$: $\varphi \triangleleft \psi$. Clearly, we have $Cn(K \cap A) = K$, which implies $K \subseteq Cn(cut_\triangleleft(\varphi))$, which is the desired conclusion.

"$\supseteq$": Let $L = Cn(cut_\triangleleft(\varphi))$. Because $\trianglelefteq$ is the restriction of the epistemic entrenchment ordering $\preceq_\leq$ to A, we know that for all $\psi \in L : \varphi \prec_\leq \psi$. Because of the construction of $\preceq_\leq$, it follows that $cut_<(\varphi) \models \psi$ for all $\psi \in L$, i.e., $Cn(cut_<(\varphi)) \supseteq Cn(cut_\triangleleft(\varphi))$. ■

Lemma 6.11 *Any revision generated by the cut base revision scheme can be generated by the linear base revision scheme.*

Proof. Given a prioritized base $A = \langle A_1, \ldots, A_n \rangle$, we specify a function π that generates a new base $B = \langle B_1, \ldots, B_n \rangle$ as follows:

$$B_i = \Big\{ \bigwedge_{j=i}^{n} A_i \Big\}. \tag{105}$$

Obviously, this translation leads to

$$Cn(A \circledE \varphi) = Cn(\pi(A) \circledL \varphi) \tag{106}$$

for all $\varphi \in \mathcal{L}$. ■

Theorem 6.12 *Let A be a prioritized base with a linear epistemic relevance ordering and let π be transformation defined by Equations (47) and (48). Then*

$$Cn(A \circledL \varphi) \;=\; Cn(\pi(A) \circledE \varphi) \tag{107}$$

for all $\varphi \in \mathcal{L}$.

Proof. If $\models \neg\varphi$, then Equality (107) clearly holds. So assume $\not\models \neg\varphi$.

We will prove the claim (107) by induction on the number of priority classes in A.

For $n = 1$ we have $A = \langle A_1 \rangle$ and $B = \pi(A) = \langle B_1, \bot \rangle$. In this case, the claim (107) is obviously true.

Now assume that A has $n + 1$ priority classes. Let A' be the base without priority class A_1 (i.e, $A'_1 = A_2$ etc.). Further let $B = \pi(A)$ and $B' = \pi(A')$ and let $\leq$ and $\leq'$ be the orderings associated with the priority classes in B and B', respectively. Let B'_q be the highest class in B' such that B'_q is not included in $cut_{<'}(\neg\varphi)$, i.e.,

$$C_1 = \bigcup_{i=q+1}^{2^n} B'_i \not\models \neg\varphi, \tag{108}$$

$$C_2 = \bigcup_{i=q+1}^{2^n} B'_i \cup B'_q \models \neg\varphi. \tag{109}$$

Now we distinguish the two cases (1) $A \circledL \varphi \models A_1$ and (2) $A \circledL \varphi \models \neg A_1$.

(1): Since A_1 is consistent with $cut_{<'}(\neg\varphi) \cup \{\varphi\}$ according to the assumption (1) and the induction hypothesis, we can add any disjunction with A_1 as a disjunct to C_1 without implying $\neg\varphi$. When adding $B'_q = B_{2q}$ $\neg\varphi$ is, of course, implied. In particular, we have

$$C_3 = \bigcup_{i=2q+2}^{2^n} B_i \not\models \neg\varphi, \tag{110}$$

$$C_4 = \bigcup_{i=2q+1}^{2^n} B_i \not\models \neg\varphi, \tag{111}$$

$$C_5 = \bigcup_{i=2q}^{2^n} B_i \models \neg\varphi. \tag{112}$$

So, the highest class not included in $cut_{<}(\neg\varphi)$ is B_{2q}. Further, $Cn(C_3 \cup \{\varphi\}) = Cn(C_1 \cup \{\varphi\})$. So, $Cn(B \circledE \varphi) = Cn((B' \circledE \varphi) \cup \{B'_q \vee A_1\})$. Since B'_q is inconsistent with $(B' \circledE \varphi)$, this means $Cn(B \circledE \varphi) = Cn((B' \circledE \varphi) \cup \{A_1\})$. Now, we have obviously $Cn(A \circledL \varphi) = Cn((A' \circledL \varphi) \cup A_1)$. Hence, $Cn(A \circledL \varphi) = Cn(B \circledE \varphi)$, which proves the induction step for case (1).

(2): Since C_1 does not imply $\neg\varphi$, we can add any of the components possibly with another disjunct to C_1 without implying $\neg\varphi$. Because $A \circledL \varphi \models \neg A_1$, we must already have $A' \circledL \varphi \models \neg A_1$, so by the induction hypothesis we have $C_1 \cup \{\varphi\} \models \neg A_1$. For this reason, we cannot add $B'_q \vee A_1$ without implying $\neg\varphi$.

$$C_3 = \bigcup_{i=2q+2}^{2^n} B_i \not\models \neg\varphi, \tag{113}$$

$$C_4 = \bigcup_{i=2q+1}^{2^n} B_i \models \neg\varphi, \tag{114}$$

leading to the fact that $Cn(B \circledE \varphi) = Cn(B' \circledE \varphi)$. Since, as mentioned above, $A' \circledL \varphi \models \neg A_1$, we also have $Cn(A \circledL \varphi) = Cn(A' \circledL \varphi)$. Hence, also in case (2), the induction step is valid. ∎

Theorem 7.1 *Let A be a base with an empty hierarchy. Then*

$$A \overset{F}{\circ} \varphi = A/\neg\varphi \cup \{\varphi\} \tag{115}$$

for all $\varphi \in \mathcal{L}$.

Proof. "$\subseteq$": Assume that $\psi \in A \overset{F}{\circ} \varphi$. If $\psi = \varphi$ then the claim clearly holds. So assume $\psi \neq \varphi$. This means, ψ is in all remainders $(A \bot \neg\varphi)$. Now suppose for contradiction that $\psi \notin A/\neg\varphi$. This means that ψ is in one entailment set E. Let E_j, $1 \leq j \leq n$, be the remaining entailment sets. Choose a set of k formulae χ_i, $1 \leq i \leq k \leq n$ from these entailment sets that are not identical with formulae from E such that for each entailment set E_j there is a formula $\chi_i \in E_j$. This is possible because all entailment sets involving formulae from E contain more than one formula. Now, the set $A - \{\chi_1, \ldots, \chi_k\} - \{\psi\}$ does obviously not imply $\neg\varphi$ and adding ψ leads to the implication of $\neg\varphi$. From that it follows that there exists a remainder that does not contain ψ, which is a contradiction to our assumption. So, we must have $\psi \in A/\neg\varphi$.

"$\supseteq$": If $\psi = \varphi$, then the claim is obviously true. So assume that this is not that case and that $\psi \in A/\neg\varphi$. This means that ψ is in no entailment set. Suppose for contradiction that $\psi \notin A \overset{F}{\circ} \varphi$. This means there exists a remainder $B \in (A \bot \neg\varphi)$ such that $\psi \notin B$. Because ψ is by assumption in no entailment set, there exists no subset C such that $C \not\models \neg\varphi$ and $C \cup \{\psi\} \models \neg\varphi$. In particular, this applies to the remainder B, so it must contain ψ, which is a contradiction. So, we must have $\psi \in A \overset{F}{\circ} \varphi$. ∎

Theorem 7.2 *The safe base revision scheme is Π_2^p-complete for general propositional logic and* coNP*-complete for Horn logic even if the revision formula has a size bounded by a constant.*

Proof. Hardness follows from Theorem 7.1, Lemma 5.19, and Theorem 5.4.

Membership follows for general propositional logic from the following algorithm that decides $A \circledS \varphi \not\models \psi$:

1. Guess a positive integer $k \leq \|A\|$.

2. Guess k sets $B_i \subseteq A$ and k formulae ψ_i.

3. Verify that $B_i \models \neg\varphi$ using a SAT-oracle.

4. Verify that for all $\chi \in B_i$: $B_i - \{\chi\} \not\models \neg\varphi$ using a SAT-oracle.

5. Verify that ψ_i is a minimal element of the hierarchy restricted to B_i.

6. Verify that $A - \bigcup_i \{\psi_i\} \not\models \psi$ using a SAT-oracle.

This algorithms also demonstrates membership of $A \textcircled{\scriptsize S} \varphi \not\models \psi$ in NP if only Horn logic is allowed, because in this case all verification steps can be performed in deterministic polynomial time. ∎

Appendix B: Symbol Index

Symbol	Definition	Explanation
$>$	p. 42	conditional
$\leq$	p. 30, p. 31	arbitrary total preorder over a base or ensconcement ordering
$<$	p. 30	strict part of $\leq$
$\leq^p_m$	p. 11	polynomial many-one reducibility
$\underline{\lessdot}$	p. 6	marking off relation
$\ll$	p. 23, Eq. (31)	priority-inclusion ordering
$\unlhd$	p. 33	ensconcement ordering generated from $\leq$
$\prec\!\!\!\mid$	p. 8	hierarchy for safe revision
$\preceq$	p. 6	epistemic entrenchment ordering
$\prec$	p. 7	strict part of entrenchment ordering $\preceq$
$\preceq_{\leq}$	p. 31, Eq. (42)	epistemic entrenchment generated from $\leq$
$\prec\!\!\prec$	p. 27, Eq. (36)	lexicographic preference ordering
$\sqsubseteq$	p. 23, Eq. (30)	epistemic relevance ordering
$\dot{-}$	p. 3	belief contraction
$\stackrel{\epsilon}{-}$	p. 7, Eq. (7)	contraction based on epistemic entrenchment ordering
$+$	p. 3	belief expansion
$+_i$	p. 45	prioritized expansion
$\dot{+}$	p. 3	belief revision
$\stackrel{\epsilon}{+}$	p. 7, Eq. (10)	revision based on epistemic entrenchment ordering
$\stackrel{F}{+}$	p. 6	full meet revision
$\circ$	p. 16	model-based revision scheme
$\circ_D$	p. 16, Eq. (24)	Dalal's revision scheme
$\dot{\circ}$	p. 28	base revision operation or WIDTIO-scheme
$\stackrel{F}{\circ}$	p. 29, Eq. (40)	full meet WIDTIO-scheme
$\odot$	p. 19	base revision scheme
$\textcircled{\scriptsize C}$	p. 26, Eq. (35)	cardinality-based base revision scheme
$\textcircled{\scriptsize E}$	p. 32, Eq. (43)	cut base revision scheme

$\textcircled{F}$	p. 22, Eq. (29)	full meet base revision scheme
$\textcircled{L}$	p. 24	linear base revision scheme
$\textcircled{P}$	p. 24	prioritized base revision scheme
$\textcircled{S}$	p. 36, Eq. (50)	safe base revision scheme
$\textcircled{X}$	p. 27, Eq. (38)	lexicographic base revision scheme
$/$	p. 8	safe elements under hierarchy $\prec\!\!\mid$
$\Downarrow$	p. 27, Eq. (37)	lexicographic remainder set
$\downarrow$	p. 23, Eq. (32)	priority-inclusion remainder set
$\perp\!\!\!\perp$	p. 26, Eq. (34)	cardinality-based remainder set
$\perp$	p. 5, Eq. (4)	set-inclusion remainder set
$\perp$	p. 3	falsity
$\top$	p. 3	truth
$\models$	p. 3	logical implication
$\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$	p. 3	propositional connectives
$\lvert \cdot \rvert$	p. 3	size of a formula, set of formulae, or instance
$\lVert \cdot \rVert$	p. 3	cardinality of a set
$A, B, C, \ldots$	p. 3	belief bases
$A_i, B_i, C_i, \ldots$	p. 23	priority class i of a belief base
$a(\cdot)$	p. 15	advice function
α, β	p. 3	truth assignment
coNP	p. 12, Eq. (13)	problems complementary to NP-problems
$\mathsf{coNP}(k)$	p. 13, Eq. (22)	problems complementary to NP(k)-problems
BH	p. 13	boolean hierarchy
$Cn(\cdot)$	p. 3, Eq. (1)	logical closure
$cut_{\prec}(\cdot)$	p. 7, Eq. (9)	cut-set based on epistemic entrenchment $\preceq$
$cut_{<}(\cdot)$	p. 31, Eq. (41)	cut-set based on ensconcement ordering $\leq$
$cut_{\lhd}(\cdot)$	p. 33	cut-set based on ensconcement ordering $\unlhd$
$\Delta(\cdot,\cdot)$	p. 16, Eq. (23)	global distance between a set of formulae and a formula
$\delta(\cdot,\cdot)$	p. 16	distance between two models
Δ_k^p	p. 12, Eq. (15)	complexity class in the polynomial hierarchy
$\Delta_2^p[O(\log n)]$	p. 13	Δ_2^p with only $O(\log n)$ oracle calls
$\mathrm{F}\Delta_2^p[O(\log n)]$	p. 40	class of search problems solvable in deterministic polynomial time using $O(\log n)$ NP-oracle calls
$form(\cdot)$	p. 16	formula equivalent to a finite set of models
$\gamma(\cdot)$	p. 5	selection function
I	p. 9	instance of a problem
$\mathcal{L}$	p. 3	logical language
$K, L, M, \ldots$	p. 3	belief sets
k-QBF	p. 13	problem of deciding truth for a quantified boolean

		formula with k alternating quantifiers
$\overline{k-\mathsf{QBF}}$	p. 13	problem complementary to k-QBF
MAX-SAT	p. 51	problem of deciding if k formulae from a given set are simultaneously satisfiable
MAX-SAT-ASG$_{odd}$	p. 49	a Δ_2^p-complete problem
$mod(\cdot)$	p. 16	all models of a formula or a set of formulae
NP	p. 10	class of problems decidable in nondeterministic polynomial time
NP^{X}	p. 12	class of problems decidable in nondeterministic polynomial time using an oracle for a problem in X
NP(k)	p. 13, Eq. (19)–(21)	complexity class in the boolean hierarchy
$O(\cdot)$	p. 9	"big O" notation for runtime requirements
$p, q, r, \ldots$	p. 3	propositional variables
$P, Q, \ldots$	p. 9	formal languages or decision problems
P	p. 9	class of problems decidable in deterministic polynomial time
P/poly	p. 15	nonuniform P
P^{X}	p. 12	class of problems decidable in deterministic polynomial time using an oracle for a problem in X
PH	p. 12	polynomial hierarchy
PSPACE	p. 12	class of problems decidable in polynomial space
Π_k^p	p. 12, Eq. (17)	complexity class in the polynomial hierarchy
Π_k^p/poly	p. 15	nonuniform Π_k^p
$\varphi, \psi, \chi, \omega$	p. 3	propositional formulae
π	p. 35, p. 51	translation between belief bases
S	p. 3	family of belief bases of belief sets
SAT	p. 10	satisfiability problem for propositional formulae
Σ	p. 3	finite alphabet of propositional variables
Σ_k^p	p. 12, Eq. (16)	complexity class in the polynomial hierarchy
Σ_k^p/poly	p. 15	nonuniform Σ_k^p
T	p. 9	Turing-machine or other sequential machine
TRANSVERSAL	p. 40	hypergraph transversal problem
TAUT	p. 12	tautology problem for propositional formulae
UNSAT	p. 11	unsatisfiability problem for propositional formulae
UNSAT(3)	p. 47, Eq. (67)	a coNP(3)-complete problem
UOCSAT	p. 51	a $\Delta_2^p[O(\log n)]$-complete problem
X	p. 11	some complexity class
X/poly	p. 15	nonuniform complexity class based on X

References

[Alchourrón and Makinson, 1982] Carlos E. Alchourrón and David Makinson.

On the logic of theory change: contraction functions and their associated revision functions. *Theoria*, 48:14–37, 1982.

[Alchourrón and Makinson, 1985] Carlos E. Alchourrón and David Makinson. On the logic of theory change: Safe contractions. *Studia Logica*, 44:405–422, 1985.

[Alchourrón and Makinson, 1986] Carlos E. Alchourrón and David Makinson. Maps between some different kinds of contraction function: The finite case. *Studia Logica*, 45(2):187–198, 1986.

[Alchourrón *et al.*, 1985] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, June 1985.

[Balcázar *et al.*, 1988] José Luis Balcázar, Josep Diaz, and Joaquim Gabarró. *Structural Complexity I*. Springer-Verlag, Berlin, Heidelberg, New York, 1988.

[Balcázar *et al.*, 1990] José Luis Balcázar, Josep Diaz, and Joaquim Gabarró. *Structural Complexity II*. Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[Benferhat *et al.*, 1993] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang, and Henri Prade. Inconsistency management and prioritized syntax-based revision. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 640–647, Chambery, France, August 1993.

[Benferhat *et al.*, 1995] Salem Benferhat, Didier Dubois, and Henri Prade. How to infer from inconsistent beliefs without revising. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1449–1455, Montreal, Canada, August 1995.

[Borgida, 1985] Alexander Borgida. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems*, 10(4):565–603, December 1985.

[Brewka, 1989] Gerhard Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1043–1048, Detroit, MI, August 1989. Morgan Kaufmann.

[Brewka, 1991] Gerhard Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, Cambridge, UK, 1991.

[Cadoli and Schaerf, 1993] Marco Cadoli and Marco Schaerf. A survey of complexity results for non-monotonic logics. *The Journal of Logic Programming*, 17:127–160, 1993.

[Cadoli and Schaerf, 1995] Marco Cadoli and Marco Schaerf. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, 1995.

[Cadoli *et al.*, 1995] Marco Cadoli, Francesco Donini, Paolo Liberatore, and Marco Schaerf. The size of a revised knowledge base. In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database-Systems (PODS-95)*, pages 151–162, 1995.

[Cai *et al.*, 1988] Jin-Yi Cai, Thomas Gundermann, Juris Hartmanis, Lane A. Hemachandra, Vivian Sewelson, Klaus W. Wagner, and Gerd Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, December 1988.

[Cayrol and Lagasquie-Schiex, 1993] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Comparaison de relations d'inférence nonmonotone: Etude de complexité. Research Report IRIT/93-23-R, IRIT, Toulouse, France, 1993.

[Dalal, 1988] Mukesh Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence (AAAI-88)*, pages 475–479, Saint Paul, MI, August 1988.

[de Kleer, 1990] Johan de Kleer. Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45:381–391, 1990.

[del Val, 1992] Alvaro del Val. Computing knowledge base updates. In B. Nebel, W. Swartout, and C. Rich, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference (KR-92)*, pages 740–752, Cambridge, MA, October 1992. Morgan Kaufmann.

[del Val, 1994] Alvaro del Val. On the relation between the coherence and foundations theories of belief revision. In *Proceedings of the 12th National Conference of the American Association for Artificial Intelligence (AAAI-94)*, pages 909–914, Seattle, WA, July 1994. MIT Press.

[Dowling and Gallier, 1984] William F. Dowling and Jean H. Gallier. Linear time algorithms for testing the satisfiability of propositional Horn formula. *The Journal of Logic Programming*, 3:267–284, 1984.

[Dubois and Prade, 1991] Didier Dubois and Henri Prade. Epistemic entrenchment and possibilistic logic. *Artificial Intelligence*, 50:223–239, 1991.

[Dubois and Prade, 1992] Didier Dubois and Henri Prade. Belief change and possibility theory. In Gärdenfors [1992a], pages 142–182.

[Dubois *et al.*, 1994] Didier Dubois, Jérôme Lang, and Henri Prade. Possibilistic logic. In D. M. Gabbay, C. J. Hogger, J. A. Robinson, and D. Nute, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming – Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 439–513. Oxford University Press, Oxford, UK, 1994.

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

[Eiter and Gottlob, 1993] Thomas Eiter and Georg Gottlob. The complexity of nested counterfactuals and iterated knowledge base revisions. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 526–533, Chambery, France, August 1993.

[Eiter and Gottlob, 1995] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):421–457, 1995.

[Fagin *et al.*, 1983] Ronald Fagin, Jeffrey D. Ullman, and Moshe Y. Vardi. On the semantics of updates in databases. In *2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 352–365, Atlanta, Ga., 1983.

[Fagin *et al.*, 1986] Ronald Fagin, Gabriel M. Kuper, Jeffrey D. Ullman, and Moshe Y. Vardi. Updating logical databases. *Advances in Computing Research*, 3:1–18, 1986.

[Fariñas del Cerro and Herzig, 1993] Luis Fariñas del Cerro and Andreas Herzig. A conditional logic for updating in the possible models approach. In B. Nebel and L. Dreschler-Fischer, editors, *KI-94: Advances in Artificial Intelligence*, pages 237–247, Saarbrücken, Germany, 1993. Springer-Verlag.

[Forbus, 1989] Kenneth D. Forbus. Introducing actions into qualitative simulation. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1273–1278, Detroit, MI, August 1989. Morgan Kaufmann.

[Fuhrmann, 1991] André Fuhrmann. Theory contraction through base contraction. *Journal of Philosophical Logic*, 20:175–203, 1991.

[Gabbay *et al.*, 1994] Dov M. Gabbay, Laura Giordano, Alberto Martelli, and Nicola Olivetti. Conditional logic programming. In P. Van Hentenryck, editor, *Proceedings of the Eleventh International Conference on Logic Programming (ICLP-94)*, pages 272–289, Santa Marherita Ligure, Italy, 1994. MIT Press.

[Gärdenfors and Makinson, 1988] Peter Gärdenfors and David Makinson. Revision of knowledge systems using epistemic entrenchment. In *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the Second Conference (TARK-88)*. Morgan Kaufmann, Asilomar, CA, 1988.

[Gärdenfors and Rott, 1995] Peter Gärdenfors and Hans Rott. Belief revision. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming – Vol. 4*, chapter 4.2. Oxford University Press, Oxford, UK, 1995.

[Gärdenfors, 1981] Peter Gärdenfors. An epistemic approach to conditionals. *American Philosophical Quaterly*, 18:203–211, 1981.

[Gärdenfors, 1986] Peter Gärdenfors. Belief revision and the Ramsey test for conditionals. *The Philosophical Review*, XCV(1):81–93, January 1986.

[Gärdenfors, 1988] Peter Gärdenfors. *Knowledge in Flux—Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, MA, 1988.

[Gärdenfors, 1990] Peter Gärdenfors. Belief revision and nonmonotonic logic: Two sides of the same coin? In *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90)*, pages 768–773, Stockholm, Sweden, August 1990. Pitman.

[Gärdenfors, 1992a] P. Gärdenfors, editor. *Belief Revision*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1992.

[Gärdenfors, 1992b] Peter Gärdenfors. Belief revision: An introduction. In Gärdenfors [1992a], pages 1–28.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[Ginsberg, 1986] Matthew L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30(1):35–79, October 1986.

[Gogic *et al.*, 1994] Goran Gogic, Christos H. Papadimitriou, and Martha Sideri. Incremental recompilation of knowledge. In *Proceedings of the 12th National Conference of the American Association for Artificial Intelligence (AAAI-94)*, pages 922–927, Seattle, WA, July 1994. MIT Press.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal for Logic and Computation*, 2(3):397–425, 1992.

[Grahne and Mendelzon, 1995] Gösta Grahne and Alberto O. Mendelzon. Updates and subjunctive queries. *Information and Computation*, 116(2):183–198, 1995.

[Hannson, 1991] Sven O. Hannson. *Belief base dynamics*. Doctoral dissertation, Uppsala University, Sweden, 1991.

[Hansson, 1993a] Sven O. Hansson. Reversing the Levi identity. *Journal of Philosophical Logic*, 22:637–669, 1993.

[Hansson, 1993b] Sven O. Hansson. Theory and base contraction unified. *Journal of Symbolic Logic*, 58:602–625, 1993.

[Hansson, 1994a] Sven O. Hansson. Kernel contraction. *Journal of Symbolic Logic*, 59:845–859, 1994.

[Hansson, 1994b] Sven O. Hansson. Taking belief bases seriously. In Prawitz, editor, *Logic and Philosophy of Science in Uppsala*, pages 13–28. Kluwer, Dordrecht, Holland, 1994.

[Hansson, 1996] Sven O. Hansson. Revision of belief sets and belief bases. In *this volume*. 1996.

[Johnson, 1990] David S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. A*, pages 67–161. MIT Press, 1990.

[Kadin, 1989] Jim Kadin. $P^{NP[O(\log n)]}$ and sparse Turing-complete sets for NP. *Journal of Computer and System Sciences*, 39(3):282–298, December 1989.

[Karp and Lipton, 1980] Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Conference Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing (STOC-80)*, pages 302–309, Los Angeles, California, 28–30 April 1980.

[Katsuno and Mendelzon, 1991] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

[Katsuno and Mendelzon, 1992] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In Gärdenfors [1992a], pages 183–203.

[Kratzer, 1981] A. Kratzer. Partition and revision: The semantics of counterfactuals. *Journal of Philosophical Logic*, 10:201–216, 1981.

[Kraus *et al.*, 1990] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Non-monotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[Lehman and Magidor, 1992] Daniel Lehman and Menachem Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55:1–60, 1992.

[Lehmann, 1993] Daniel Lehmann. Another perspective on default reasoning. Technical report, Hebrew University, Jerusalem, Israel, 1993.

[Levi, 1977] Isaac Levi. Subjunctives, dispositions and chances. *Synthese*, 34:423–455, 1977.

[Makinson and Gärdenfors, 1991] David Makinson and Peter Gärdenfors. Relations between the logic of theory change and nonmonotonic logic. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change*, volume 465 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York, 1991.

[Nayak, 1994] Abhaya C. Nayak. Foundational belief change. *Journal of Philosophical Logic*, 23:495–533, 1994.

[Nebel, 1989] Bernhard Nebel. A knowledge level analysis of belief revision. In R. Brachman, H. J. Levesque, and R. Reiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 1st International Conference (KR-89)*, pages 301–311, Toronto, ON, May 1989. Morgan Kaufmann.

[Nebel, 1990] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[Nebel, 1991] Bernhard Nebel. Belief revision and default reasoning: Syntax-based approaches. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference (KR-91)*, pages 417–428, Cambridge, MA, April 1991. Morgan Kaufmann.

[Nebel, 1992] Bernhard Nebel. Syntax-based approaches to belief revision. In Gärdenfors [1992a], pages 52–88.

[Nebel, 1994] Bernhard Nebel. Base revision operations and schemes: Semantics, representation, and complexity. In *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI-94)*, pages 341–345, Amsterdam, The Netherlands, August 1994. Wiley.

[Nebel, 1996] Bernhard Nebel. Artificial intelligence: A computational perspective. In G. Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 237–266. CSLI Publications, Stanford, CA, 1996.

[Papadimitriou, 1994] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley, Reading, MA, 1994.

[Pollock, 1976] J. Pollock. *Subjunctive Reasoning.* Reidel, Dordrecht, Holland, 1976.

[Poole, 1988] David Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Reiter, 1987] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.

[Rescher, 1964] Nicholas Rescher. *Hypothetical Reasoning.* North-Holland, Amsterdam, Holland, 1964.

[Rescher, 1973] Nicholas Rescher. *The Coherence Theory of Truth.* Oxford University Press, Oxford, UK, 1973.

[Rott, 1991a] Hans Rott. A nonmonotonic conditional logic for belief revision I. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change*, volume 465 of *Lecture Notes in Artificial Intelligence*, pages 135–183. Springer-Verlag, Berlin, Heidelberg, New York, 1991.

[Rott, 1991b] Hans Rott. Two methods of constructing contractions and revisions of knowledge systems. *Journal of Philosophical Logic*, 20:149–173, 1991.

[Rott, 1992] Hans Rott. On the logic of theory change: More maps between different kinds of contraction functions. In Gärdenfors [1992a], pages 122–141.

[Rott, 1993] Hans Rott. Belief contraction in the context of the general theory of rational choice. *Journal of Symbolic Logic*, 58, December 1993.

[Satoh, 1988] Ken Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 455–462, Tokyo, Japan, 1988. Springer-Verlag.

[Selman and Kautz, 1991] Bart Selman and Henry A. Kautz. Knowledge compilation using Horn approximations. In *Proceedings of the 9th National Conference of the American Association for Artificial Intelligence (AAAI-91)*, pages 904–909, Anaheim, CA, July 1991. MIT Press.

[Selman and Kautz, 1993] Bart Selman and Henry A. Kautz. Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 290–295, Chambery, France, August 1993.

[Selman *et al.*, 1992] Bart Selman, Hector J. Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the 10th National Conference of the American Association for Artificial Intelligence (AAAI-92)*, pages 440–446, San Jose, CA, July 1992. MIT Press.

[Stockmeyer, 1987] Larry Stockmeyer. Classifying the computational complexity of problems. *Journal of Symbolic Logic*, 52(1):1–43, 1987.

[Veltman, 1976] Frank Veltman. Prejudices, presuppositions, and the theory of counterfactuals. In J. Groenendijk and M. Stokhof, editors, *Amsterdam Papers of Formal Grammar*, volume I, pages 248–281. Centrale Interfaculteit, Universiteit Amsterdam, Amsterdam, The Netherlands, 1976.

[Wagner and Wechsung, 1988] Klaus W. Wagner and Gerd Wechsung. *Computational Complexity.* Reidel, Dordrecht, Holland, 1988.

[Wagner, 1987] Klaus W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.

[Wagner, 1990] Klaus W. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[Weber, 1986] Andreas Weber. Updating propositional formulas. In L. Kerschberg, editor, *Expert Database Systems—Proceedings From the 1st International Conference*, pages 487–500. Benjamin/Cummings, 1986.

[Williams, 1994a] Mary-Anne Williams. On the logic of theory base change. In C. MacNish, D. Pearce, and L.M. Pereira, editors, *Logics in Artificial Intelligence – European Workshop JELIA'94*, pages 86–105, York, UK, 1994. Springer-Verlag.

[Williams, 1994b] Mary-Anne Williams. *Transmutations of Knowledge Systems.* PhD thesis, University of Sydney, Australia, 1994.

[Winslett, 1988] Marianne S. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence (AAAI-88)*, pages 89–93, Saint Paul, MI, August 1988.

[Winslett, 1990] Marianne S. Winslett. *Updating Logical Databases.* Cambridge University Press, Cambridge, UK, 1990.

[Wrobel, 1994] Stefan Wrobel. *Concept Formation and Knowledge Revision.* Kluwer, Dordrecht, Holland, 1994.

[Yap, 1993] Chee K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1993.