



Mixed Iterated Revisions: Rationale, Algorithms, and Complexity

PAOLO LIBERATORE, Sapienza University of Rome, Italy

Several forms of iterable belief change exist, differing in the kind of change and its strength: some operators introduce formulae, others remove them; some add formulae unconditionally, others only as additions to the previous beliefs; some only relative to the current situation, others in all possible cases. A sequence of changes may involve several of them: for example, the first step is a revision, the second a contraction and the third a refinement of the previous beliefs. The ten operators considered in this article are shown to be all reducible to three: lexicographic revision, refinement, and severe withdrawal. In turn, these three can be expressed in terms of lexicographic revision at the cost of restructuring the sequence. This restructuring needs not to be done explicitly: an algorithm that works on the original sequence is shown. The complexity of mixed sequences of belief change operators is also analyzed. Most of them require only a polynomial number of calls to a satisfiability checker, some are even easier.

CCS Concepts: • **Computing methodologies** → **Nonmonotonic, default reasoning and belief revision**;

Additional Key Words and Phrases: Iterated belief revision, translation algorithm, computational complexity

ACM Reference format:

Paolo Liberatore. 2023. Mixed Iterated Revisions: Rationale, Algorithms, and Complexity. *ACM Trans. Comput. Logic* 24, 3, Article 27 (May 2023), 49 pages.

<https://doi.org/10.1145/3583071>

1 INTRODUCTION

New information comes in different forms. At the one end of the spectrum, it is believed to be always true (lexicographic revision [60, 76]); at the far opposite, it is unknown (contraction [1, 73]). Middle cases exist: it may be believed only as long as the current situation is concerned (natural revision [12, 47]), or it may be believed only as long as it does not contradict the previous beliefs (refinement [10, 63]). Sequences of changes are hardly all of the same form, like someone who embraces every single new theory with all of his heart or instead is so skeptical to refuse every piece of information that contradicts what is known.

Example 1. An example against natural revision [12, 47] is that of the red bird [43]: An animal looks like a bird (b) and upon coming close turns out to be red (r); finding out that the image is not a bird ($\neg b$) makes natural revision discard it being red in spite of no evidence to the contrary.

Why does “natural” revision make such a gross mistake? When revising by r , the most likely scenario is that b and r are both true because of the previous revision by b . Natural revision increases

Author’s address: P. Liberatore, Sapienza University of Rome, DIAG, Via Ariosto 25, 00185, Rome, Italy; email: liberato@diag.uniroma1.it.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1529-3785/2023/05-ART27 \$15.00

<https://doi.org/10.1145/3583071>

the credibility of this scenario, but leaves “ b is false and r is true” unlikely because of the previous revision by b . Instead, lexicographically revising by r increases the credibility of all scenarios that are consistent with r , including “ b is false and r is true.”

This is a situation where natural revision is not to be used. Yet, a small variant of the conditions turns the very same formulae, with the very same meaning of variables, into a case for it.

A hunter meets a peasant friend in the countryside, who told him having seen a strange bird in the thicket a couple of miles away (b). Lured by the unique trophy he could make out of it, and by bragging about its hunting at the village fête that evening, the hunter rushes to there.

Midway, he encounters the village postman. In a hurry, the hunter explains he could not stop and why. The postman answers he understands, as he actually saw something red in the thicket (r). This would explain why the peasant friend told the bird was strange, because no red bird has ever been seen around there.

Arrived at the thicket, the hunter checks for anything red but the bushes and trees are too thick to see anything inside. Entering is out of discussion, as any bird would fly away upon hearing the noise. Keeping ready with his rifle, the hunter throws a stone in the thicket, but nothing happens. A second and third stone confirm that no bird is there. The peasant friend and the postman made fun of him by having him run with no reason. No bird is there ($\neg b$), nothing indicates something red (r is no longer believed).

While the sequence of formulae is exactly the same (first b , then r , finally $\neg b$), with the same meaning of the variables (b =bird, r =red), natural revision gives exactly the expected result: Since the first piece of information was made up, there is no reason to believe the second.

Should hunters always use natural revisions?

Of course not.

Seeing the hunter throwing stones for no clear reason, the village policeman approaches the hunter to ask why. Concealing how he was made a fool is useless, as the peasant friend and the postman will tell the story to everyone at the village fête that evening; everyone will know even before the parade. The policeman laughs, but to the hunter’s surprise it is not about the practical joke. A red animal would be unique in the area. Even if one was there, and the hunter managed to shot it, he would have been fined and confiscated the trophy. He could not have shown it in his living room, and certainly not brag about it at the village fête. If it is red, then it cannot be hunted ($r \rightarrow \neg h$).

As it comes from a police officer while on duty, this information is totally reliable. It is also a general rule, not specific to the current situation. No matter if it was a bird (b or $\neg b$), if it is red, then it cannot be hunted ($r \rightarrow \neg h$). The situation where a red bird that can be hunted was there (b , r and h) is less likely than one where it cannot (b , r , $\neg h$), and the same if it is red but not a bird ($\neg b$, r and h) is less likely than $\neg b$, r and $\neg h$). That hunting is forbidden is more likely in every situation, even if a red animal that is not a bird later turns out to be there.

The hunter is better revising lexicographically (by $\neg h$) this time if he wants to avoid being fined, after previously having revised naturally (by $\neg b$). A mixed sequence of revisions is the best course of action.

A related question is why to apply a sequence of revisions in the first place. Why not just coming back to where the hunter started, with no information about the thicket, rather than revising by b , then r and then by $\neg b$? Once the bird is nowhere to be found, everything could be just canceled. The epistemic state is the same as the beginning.

While talking with the policeman, the hunter hears a sound of feathers from the thicket. Feathers mean bird (b). Maybe one that does not fly, or is wounded and unable to fly. The peasant and the postman might have been truthful, after all. Nothing indicates they joked any longer. None of

them, including the postman. The bird might be red (r) after all. This would not be the case if the hunter just forgot everything he was told.

Natural revision, then lexicographic revision. Not two lexicographic revisions, not two natural revisions. A mixed sequence of revisions is what to do in this example. As Booth and Meyer [8] put it: “It is clear that an agent need not, and in most cases, ought not to stick to the same revision operator every time that it has to perform a revision.”

Other forms of revisions exist. They may mix in any order. A lexicographic revision may be followed by a natural revision, a restrained revision and a radical revision. Such a sequence is used as a running example:

$$\emptyset \text{lex}(y) \text{nat}(\neg x) \text{res}(x \wedge z) \text{rad}(\neg z).$$

The sequence begins with \emptyset , the complete lack of knowledge. The first information acquired is y , and is considered valid in all possible situations; it is a lexicographic revision. The next is $\neg x$, but is deemed valid only in the current conditions; this is a natural revision. The next is $x \wedge z$, but is only accepted as long as it does not contradict what is currently known; it is a refinement. Finally, $\neg z$ is so firmly believed that all cases where it does not hold are excluded; this is a radical revision.

No kind of revision is better than the others. Natural revision has its place. As well as lexicographic revision. As well as refinement, radical revision, severe antiwithdrawals and severe revision. None of them is the best. Each is right in the appropriate conditions and wrong in the others. For example, a sequence of severe revisions is problematic, because it coarsens the plausibility degree of different scenarios [28, 67]; yet, it is a common experience that sometimes new information makes unlikely conditions as likely as others. A truck full of parrots to be sold as pets crushed nearby, freeing thousands of exotic animals; a red bird is now as likely to be stumbled upon as a local wild animal. New information may raise the plausibility of a situation to the level of another, making them equally likely. This is a consequence of the new information, not a fault of the revision operator. The problem comes when using severe revisions only, without other revisions that separate the likeliness of different conditions [28, 67].

The solution is not a new framework that encompasses all possible cases, but to deal with mixes of different revision kinds.

Which form of revision is used at each step is assumed known in this article. As in the example, it comes together with the revising formula itself: that there is a red bird is only believed in the current situation, making it a natural revision; that hunting red animals is forbidden is always the case, calling for lexicographic revision. In other scenarios, the form of revision to use may depend on the time point or the sequence of previous revisions. The first approach is exemplified by the alternating sequences of actions and observations by Hunter and Delgrande [40, 41]. The second by reconstructing the epistemic states from the outcome of past revisions [9, 56]. The issue is outside the scope of the present article.

The problem considered here is to determine the outcome of a sequence of mixed revisions. Ten belief change operators are considered: lexicographic revision, refinement, severe withdrawal, natural revision, restrained revision, plain severe revision, severe revision, moderate severe revision, very radical revision, and full meet revision. They all change the plausibility of the possible situations, each in its own way. The possible situations are formalized by propositional models, their plausibility by an ordering among propositional models. Other approaches not considered here involve action histories [9, 56], fixed orderings [2], simultaneous revisions [45], revisions by orderings [60], and bidimensional revisions [69].

The ten considered change operators modify the order of plausibility of the models. Each does it in its own way. For example, natural revision promotes some models to the maximal plausibility;

lexicographic revision makes some models being more plausible than others. Keeping in memory a complete description of these orderings is unfeasible even in the propositional case, as the number of models is exponential in the number of variables. Several operators such as lexicographic revision, refinement, and restrained revision can generate orderings that compare equal no two models, making exponentially large every representation that is linear in the number of equivalence classes of the ordering, such as the ordered sequence of formulae by Rott [68] and the partitions by Meyer, Ghose, and Chopra [59]. Many distance-based one-step revisions suffer from a similar problem: the result of even a single revision may be exponential in the size of the involved formulae [14]. Iterated revisions typically do not employ distances, and the problem can be overcome:

- the ten belief change operators considered in this article (lexicographic revision, refinement, severe withdrawal, natural revision, restrained revision, plain severe revision, severe revision, moderate severe revision, very radical revision, full meet revision) can be reduced to three: lexicographic revision, refinement, and severe withdrawal; these reductions are local: they replace an operator without changing the rest of the sequence before and after it;
- refinement and severe withdrawal can be reduced to lexicographic revision; this however requires structuring the sequence of belief change operators; however, the result is a sequence that behave like the original on subsequent changes;
- this restructuring needs not to be done explicitly; an algorithm that works on the original sequence is shown; it does not change the sequence, but behaves as if it were restructured; apart from the calls to a satisfiability checker, the running time is polynomial.

This mechanism determines the result of an arbitrary sequence of these revisions from an initial ordering that reflects a total lack of information. This is not a limitation when using the ten considered revision forms, as an arbitrary ordering can be created by a sequence of lexicographic revisions [9].

During its execution, the algorithm calculates some partial results called underformulae, which can be used when the sequence is extended by further revisions. The need for a satisfiability checker is unavoidable, given that belief change operates on propositional formulae. However, efficient solvers have been developed [3, 5]. Restricting to a less expressive language [16] such as Horn logic may also reduce the complexity of the problem, as it is generally the case for one-step revisions [21, 52, 58, 62], since satisfiability in this case can be solved efficiently.

Some complexity results are proved: some imply the ones announced without proofs in a previous article [55], but extend them to the case of mixed sequences of revisions. Entailment from a sequence of lexicographic, natural, restrained, very radical and severe revisions, refinements, and severe antiwithdrawals is Δ_2^P -complete. Two groups of belief change operators are relevant to complexity. The first is called lexicographic-finding and comprises the ones that behave like lexicographic revision on consistent sequences of formulae; lexicographic and moderate severe revisions are in this group. The second is called bottom-refining, as it includes the revisions that separate the most likely scenarios when some are consistent with the new information; natural revision, restrained revision, and severe revision are in this group. Entailment from a sequence of operators all of the first kind or all of the second is Δ_2^P -complete. Three revision operators require a separate analysis. Entailment from a sequence of very radical revision is $\Delta_2^P[\log n]$ -complete. The same complexity comes from sequences of plain severe and full meet revisions only. Inter-mixing these operators with the others is $\Delta_2^P[\log n]$ -complete in one case and Δ_2^P -complete in the others.

The rest of this article is organized as follows: Section 2 introduces the main concepts of total preorder, lexicographic revision, refinement, and severe withdrawal; Section 3 shows how to reduce the other change operators to these three; Section 4 shows an algorithm for computing

the result of a sequence of revisions; Section 5 presents the computational complexity results; Section 6 investigates the properties of severe antiwithdrawal, especially in light of its central role in iterated belief changes; Section 7 discusses the results in this article, compares them with other work in the literature, and presents the open problems.

2 PRELIMINARIES

A propositional language over a finite alphabet is assumed. Given a formula F , its set of models is denoted $Mod(F)$, while a formula having S as its set of models is denoted $Form(S)$. The symbol \top denotes a tautology, a formula satisfied by all models over the given alphabet.

A *base* is a propositional formula denoting what is believed in certain moment. Historically, revision was defined as an operator that modifies a base in front of new information; an ordering was employed to take choices when this integration may be done in multiple ways, which is usually the case. Assuming this ordering as fixed or depending on the base only is the AGM model or revision [1, 30]. Iterated revision is problematic using this approach; the solution is to reverse the role of the base and the ordering. Instead of being a supporting element, the ordering becomes the protagonist. The base derives from it as the set of most plausible formulae [18, 50]. Such plausibility information can be formalized in several equivalent ways: epistemic entrenchments [26, 31], systems of spheres [32, 35], rankings [46, 77, 80], and KM preorders [44, 64].

2.1 Total Preorders

Katsuno and Mendelzon [44] proved that AGM revision can be reformulated in terms of a total preorder over the set of models, where the models of the base are exactly the minimal ones according to the ordering. Iterated revision can be defined by demoting the base from primary information to derived one. Instead of revising a base using the ordering as a guide, the ordering itself is modified. The base is taken to be just the set of formulae implied by all most plausible models.

Definition 1. A *total preorder* C is a partition of the models into a finite sequence of classes $[C(0), C(1), C(2), \dots, C(m)]$.

Such an ordering can be depicted as a stack, the top boxes containing the most plausible models. This is equivalent to a reflexive, transitive, and total relation, but it makes for simpler definitions and proofs about iterated revisions.

$C(0)$
$C(1)$
$C(2)$
$C(3)$
$C(4)$
$C(5)$
$C(6)$
$C(7)$

A KM total preorder is the same as a partition by Mayer, Ghose, and Chopra [59], who use a formula for each class in place of its set of models. In turns, such a partition is similar to the system for expressing such orderings in possibilistic logic [4], and correspond to a sequence of formulae by Rott [68], $h_1 < \dots < h_m$ via $C(0) \cup \dots \cup C(i) = Mod(h_{i+1} \wedge \dots \wedge h_m)$, and to an epistemic entrenchment [1].

Being a partition, $C = [C(0), \dots, C(m)]$ contains all models. As a result, every model is in a class. No model is “inaccessible,” or excluded from consideration when performing revision. Revisions

producing such models could still be formalized by giving a special status to the last class $C(m)$, as the set of such inaccessible models, but they are not studied in this article. Their analysis is left as an open problem.

Classes are allowed to be empty, even class zero $C(0)$. The base represented by a total preorder $C = [C(0), \dots, C(m)]$ cannot therefore being defined as $Form(C(0))$ but as the minimal models according to C , denoted by $Mod(C, \top)$.

More generally, given a formula P the notation $\min(C, P)$ indicates the set of minimal models of P according to the ordering C . Formally, if i is the lowest index such that $C(i) \cap Mod(P)$ is not empty, then $\min(C, P) = C(i) \cap Mod(P)$. Several iterated revision depends on such an index i and its corresponding set of models $C(i) \cap Mod(P)$.

Another consequence of allowing empty classes is that two total preorder may be different yet comparing models in the same way. For example, $[Mod(\top)]$ and $[\emptyset, Mod(\top)]$ both place all models in the same class, which is class zero for the former and class one for the latter. They are in this sense equivalent. They coincide when removing the empty classes. The minimal models of every formula are the same [63].

Definition 2. Two total preorder C and C' are revision-equivalent if $\min(C, P) = \min(C', P)$ for every formula P .

Since no other kind of equivalence between preorders is considered in this article, revision-equivalence is shortened to equivalence.

Revising by the same formula changes revision-equivalent orderings into revision-equivalent orderings. This holds for all revision semantics considered in this article.

The amount of information an ordering carries can be informally identified with its ability of telling the relative plausibility of two models. Ideally, an ordering should have a single minimal model, representing what is believed to be the state of the world, and a single model in each class, allowing to unambiguously decide which among two possible states of the world is the most likely. Most revision indeed refine the ordering by splitting its classes. At the other end of the spectrum, the total order $\emptyset = [Mod(\top)]$ carries no information: not only its base comprises all models and is therefore tautological, but all models are also considered equally plausible. Studies on the practical use of revision [9, 55] assume an initial empty ordering that is then revised to obtain a more discriminating one. Equivalently, an ordering can be expressed as a suitable sequence of revisions applied to the empty total preorder.

Not all operators considered in this article are revisions, only the ones that produce an ordering whose base implies the revising formula. Some other operators just split classes (like refinement) or merge them (like severe withdrawal). The result of an operator ope modifying a total preorder C by a formula P is defined by the infix notation $Cope(P)$. This is a new total preorder whose base entails P if ope is a revision operator. More specifically, AGM revisions produce a base out of the minimal models of P in C :

$$\min(Cope(P), \top) = \min(C, P).$$

2.2 Iterated Revisions

Several iterated belief revision operators are considered. These can be all expressed in terms of three of them: lexicographic revision, refinement, and severe withdrawal. Intuitively, this is because each of these three includes a basic operation that can be performed over an ordering: moving, splitting, and merging classes. The correspondence is not exact, as the lexicographic revision perform both moving and splitting but can be made to move a single class from a position of the sequence to another.

These three operators are defined in this section. The others will be then introduced in the next, and immediately proved to be reducible to these three. This allows to concentrate on the computational aspects only on the three basic ones.

2.2.1 Lexicographic Revision. Lexicographic revision is one of the two earliest iterated belief revision operator [76]. While its authors initially rejected it, later research have reconsidered it [55, 60, 61, 66]. The tenant of this operator can be summarized as: revising by P means that P is true no matter of everything else. Technically, all models satisfying P are more plausible that every other one.

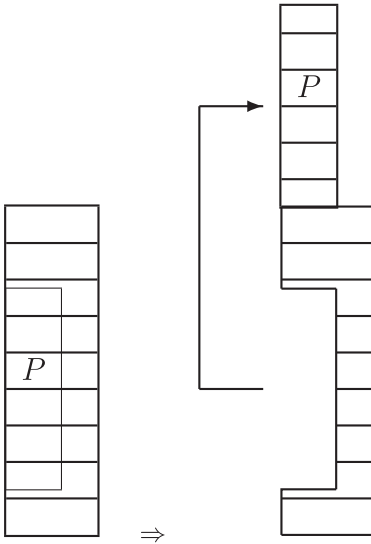
Definition 3. The lexicographic revision of a total preorder C by a formula P is defined as the following total preorder, where i and j are, respectively, the indexes of the minimal and maximal classes of C containing models of P :

$$C_{\text{lex}(P)}(k) = \begin{cases} C(k+i) \cap \text{Mod}(P) & \text{if } k \leq j-i, \\ C(k-j+i-1) \setminus \text{Mod}(P) & \text{otherwise.} \end{cases}$$

Alternatively, a formula directly based on sequences can be taken as the definition of lexicographic revision:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{lex}(P)} \\ &= [C(0) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, C(m) \setminus \text{Mod}(P)]. \end{aligned}$$

This definition does not exactly coincide with the previous one because of some empty classes, which means that the two produce equivalent total preorders. A graphical representation of revising a total preorder by a formula P is the following one:



In words, the models of P are “cut out” from the ordering and shifted together to the top. Their relative ordering is not changed, but they are made more plausible than every model of $\neg P$. By construction, $\min(C_{\text{lex}(P)}, \top)$ is equal to $\min(C, P)$, making this operator a revision.

2.2.2 Refinement. Contrary to lexicographic revision, refinement [10, 63] is not a revision. It is still a basic form of belief revision in which belief in a formula P is strengthened, but never so much

to contradict previous information. Technically, the models of every class are split depending on whether they satisfy P or not. This way, two models are separated only if they were previously considered equally plausible, and only if one satisfies P and the other does not.

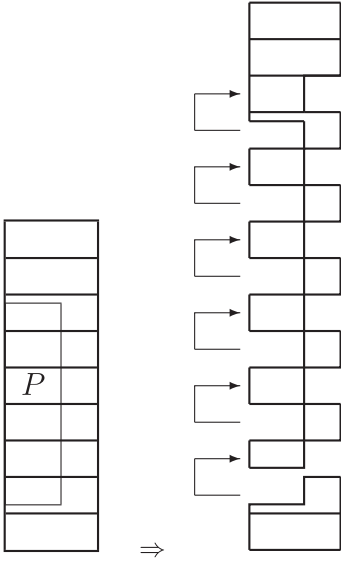
Definition 4. The refinement of a total preorder C by formula P is the following total preorder:

$$C \text{ ref}(P)(k) = \begin{cases} C(0) \cap \text{Mod}(P) & \text{if } k = 0 \text{ and } C(0) \cap \text{Mod}(P) \neq \emptyset, \\ C(0) \setminus \text{Mod}(P) & \text{if } k = 0 \text{ and } C(0) \cap \text{Mod}(P) = \emptyset, \\ C(k/2) \cap \text{Mod}(P) & \text{if } k > 0 \text{ even,} \\ C(k/2) \setminus C \text{ ref}(P)(k-1) & \text{if } k > 0 \text{ odd.} \end{cases}$$

Alternatively, refinement can be defined directly on partitions:

$$\begin{aligned} [C(0), \dots, C(m)] \text{ ref}(P) \\ = [C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)]. \end{aligned}$$

Some of these classes may be empty, and can therefore be removed respecting preorder equivalence. Graphically, refining a total preorder C by a formula P can be seen as follows:



2.2.3 Severe Antiwithdrawal. While this operator was defined [27, 70] as a form of contraction, it is technically cleaner to use it in reverse, with the negated formula. Removing a formula $\neg P$ is the same of creating the consistency with P , but the second definition has been advocated has a most direct formalization of the actual process of belief change [34].

In the specific case of severe antiwithdrawal, creating consistency with P is obtained by merging all classes of the ordering that are in the same class or one of lower index with the minimal models of P . This is motivated by the principle of equal-treated-equally when applied to the plausibility of models: To make P consistent some models of P have to become minimal; but the models of $\neg P$ that are in lower classes have the same plausibility or greater, so they should not be excluded.

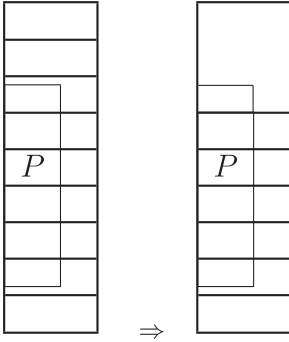
Definition 5. The severe antiwithdrawal of the total preorder C by formula P is the following total preorder, where i is the minimal index such that $C(i) \cap \text{Mod}(P) \neq \emptyset$:

$$C_{\text{sev}}(P)(k) = \begin{cases} C(0) \cup \dots \cup C(i) & \text{if } k = 0, \\ C(k+i) & \text{if } k > 0. \end{cases}$$

Lexicographic antiwithdrawal can also be defined in terms of sequences. If i is the lowest index such that $C(i) \cap \text{Mod}(P) \neq \emptyset$, then

$$\begin{aligned} [C(0), \dots, C(i), C(i+1), \dots, C(m)]_{\text{sev}}(P) \\ = [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)]. \end{aligned}$$

Graphically, severe antiwithdrawal merges all classes of index lower or equal to the minimal class intersecting $\text{Mod}(P)$:



This way, the base of the revised preorder $\min(C_{\text{sev}}(P), \top)$ is guaranteed to contain some models of P , which means that it has been made consistent with P . At the same time, the relative plausibility of two models is never reversed: a model that is more plausible than another according to C is never made less plausible than that according to $C_{\text{sev}}(P)$.

3 REDUCTIONS

Many belief change operators exist. Many of them are expressible in terms of the three presented in the previous section: lexicographic revision, refinement, and severe antiwithdrawal. The reductions do not affect what is before or after then replaced operator applications, which is not the case for the transformations shown in the next section.

Example 2. The following sequence of revisions is used as a running example. The following sections show how to make it into a sequence that only contains lex, ref, and sev:

$$\emptyset \text{lex}(y) \text{nat}(\neg x) \text{res}(x \wedge z) \text{rad}(\neg z).$$

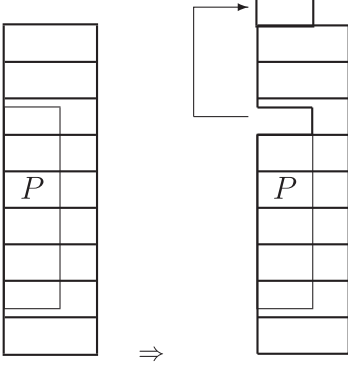
3.1 Natural Revision

This revision was first considered and discarded by Spohn [76], and later independently reintroduced by Boutilier [12]. Among revisions, it can be considered at further opposite to lexicographic revision, in that a formula P is made true by a minimal change to the ordering. This amounts to making the minimal models of P the new class zero of the ordering and changing nothing else.

Formally, if $\min(C, P) = C(i) \cap \text{Mod}(P)$, then

$$\begin{aligned} [C(0), \dots, C(m)]_{\text{nat}}(P) \\ = [C(i) \cap \text{Mod}(P), C(0), \dots, C(i-1), C(i) \setminus \text{Mod}(P), C(i+1), \dots, C(m)]. \end{aligned}$$

Graphically, $\min(C, P)$ is “cut out” from the total preorder C and moved to the beginning of the sequence, making it the new class zero $\text{Cnat}(P)(0)$.



Since by definition $\min(C, P)$ is not empty, it holds $\min(\text{Cnat}(P), \top) = \text{Cnat}(P)(0) = \min(C, P)$, meaning that it is an AGM revision operator.

THEOREM 1. *For every total preorder C and formula P , it holds $\text{Cnat}(P) \equiv \text{Clex}(K)$ where $K = \text{Form}(\min(C, P))$.*

PROOF. Let $K = \text{Form}(\min(C, P))$ and i the index such that $\min(C, P) = C(i) \cap \text{Mod}(P)$. By the properties of set difference, it holds $C(i) \setminus \text{Mod}(P) = C(i) \setminus (C(i) \cap \text{Mod}(P)) = C(i) \setminus \text{Mod}(K)$. Since classes do not share variable and $\text{Mod}(K) \subseteq C(i)$, it holds $C(j) = C(j) \setminus \text{Mod}(K)$ for every $j \neq i$. Natural revision can therefore be recast as

$$\begin{aligned}
 & [C(0), \dots, C(m)]_{\text{nat}(P)} \\
 &= [C(i) \cap \text{Mod}(M), C(0), \dots, C(i-1), C(i) \setminus \text{Mod}(P), C(i+1), \dots, C(m)] \\
 &\equiv [\text{Mod}(K), \\
 &\quad C(0) \setminus \text{Mod}(K), \dots, C(i-1) \setminus \text{Mod}(K), C(i) \setminus \text{Mod}(K), \\
 &\quad C(i+1) \setminus \text{Mod}(K), \dots, C(m) \setminus \text{Mod}(K)] \\
 &\equiv [\text{Mod}(K), \emptyset, \dots, \emptyset, \\
 &\quad C(0) \setminus \text{Mod}(K), \dots, C(m) \setminus \text{Mod}(K)] \\
 &\equiv [C(0) \cap \text{Mod}(K), C(1) \cap \text{Mod}(K), \dots, C(m) \cap \text{Mod}(K), \\
 &\quad C(0) \setminus \text{Mod}(K), \dots, C(m) \setminus \text{Mod}(K)].
 \end{aligned}$$

The equivalences are correct, because: first, $C(i) \cap \text{Mod}(K) = \text{Mod}(K)$, since $\text{Mod}(K) \subseteq C(i)$; second, empty classes can be introduced at every point of every ordering, and $C(j) \cap \text{Mod}(K) = \emptyset$ for every $j \neq i$. The resulting total preorder is $\text{Clex}(K)$. \square

This transformation does not just tell how to compute the propositional result of natural revision, that is, the base $\text{Form}(\text{Cnat}(P)(0))$ of the revised ordering. To the contrary, it requires it, as $\text{Mod}(K) = \min(C, P) = \text{Cnat}(P)(0)$. After K has been calculated, $\text{lex}(K)$ produces the same exact preorder as $\text{nat}(P)$ when applied to the same preorder, not just two preorders having the same base. This means that all subsequent revisions are unaffected by the replacement. In other words, for every initial preorder C and every sequence of previous and future belief changes, it

holds

$$\begin{aligned} & \text{Cope}_1(P_1) \dots \text{ope}_{n-1}(P_{n-1}) \text{nat}(P) \text{ope}_{n+1}(P_{n+1}) \dots \text{ope}_{n'}(P_{n'}) \\ & \equiv \text{Cope}_1(P_1) \dots \text{ope}_{n-1}(P_{n-1}) \text{lex}(K) \text{ope}_{n+1}(P_{n+1}) \dots \text{ope}_{n'}(P_{n'}). \end{aligned}$$

The other reductions in this section have all this property, that an operator application in whichever position of a sequence can be replaced without affecting the final ordering. Natural revision requires the minimal models of P to be calculated, some other operators do not. Natural revision is replaced by a single lexicographic revision, the others may require some lexicographic revisions, refinements and severe antiwithdrawals.

Example 3. The second operation in the running example is a natural revision:

$$\emptyset \text{lex}(y) \text{nat}(\neg x) \text{res}(x \wedge z) \text{rad}(\neg z).$$

Since $\emptyset \text{lex}(y)$ is $[\text{Mod}(y), \text{Mod}(\neg y)]$, and class zero of this ordering contains models of $\neg x$, then $\min(\emptyset, \neg x) = \text{Mod}(\neg x \wedge y)$. As a result, the sequence can be simplified into

$$\emptyset \text{lex}(y) \text{lex}(\neg x \wedge y) \text{res}(x \wedge z) \text{rad}(\neg z).$$

Some other operators are reduced to natural revision, which can in turn be reduced to lexicographic revision. For example, restrained revision is a refinement followed by natural revision (or vice versa). The above theorem shows that it can be further reformulated as a refinement and a lexicographic revision.

3.2 Restrained Revision

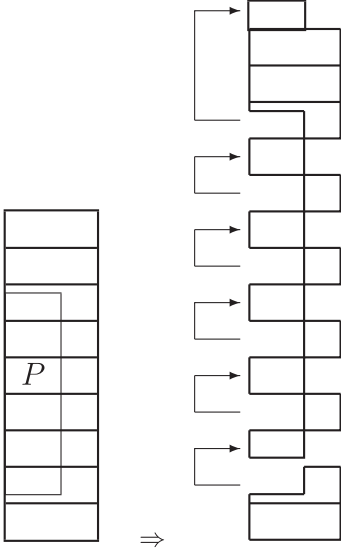
Restrained revision [8] can be seen as a minimal modification of refinement to turn it into a form of revision. Indeed, refining a total preorder by a formula P does not generally makes P entailed by the refined total preorder. This is indeed the case only if $\min(C, \top)$ contains some models of P .

Restrained revision can be seen as an intermediate form of revision: while natural revision changes the preorder in a minimal way to make the revising formula entailed and lexicographic revision makes the formula to be preferred in all possible cases, restrained revision makes it to be preferred only when this is consistent with previous beliefs, and makes it entailed by a minimal change in the ordering.

Restrained revision is defined as follows, where $\min(C, P) = C(i) \cap \text{Mod}(P)$:

$$\begin{aligned} & [C(0), \dots, C(m)] \text{res}(P) \\ & = [C(i) \cap \text{Mod}(P), \\ & \quad C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, C(i-1) \cap \text{Mod}(P), C(i-1) \setminus \text{Mod}(P), \\ & \quad C(i) \setminus \text{Mod}(P), \\ & \quad C(i+1) \cap \text{Mod}(P), C(i+1) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)]. \end{aligned}$$

Graphically, each intersection $C(i) \cap \text{Mod}(P)$ is lifted just above $C(i) \setminus C(i)$, except for $i = 0$, where it is lifted to the top.



The following quite obvious theorem is proved only for the sake of completeness, its statement being almost a direct consequence of the definition.

THEOREM 2. *For every total preorder C and formula P , it holds $C_{res}(P) \equiv C \text{ ref}(P) \text{ nat}(P)$*

PROOF. Let $\min(C, P) = C(i) \cap \text{Mod}(P)$. The ordering $C \text{ ref}(P) \text{ nat}(P)$ is

$$\begin{aligned}
 & [C(0), \dots, C(m)] \text{ ref}(P) \text{ nat}(P) \\
 &= [C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)] \text{ nat}(P) \\
 &= [C(i) \cap \text{Mod}(P), \\
 &\quad C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, C(i-1) \cap \text{Mod}(P), C(i-1) \setminus \text{Mod}(P), \\
 &\quad \emptyset, C(i) \setminus \text{Mod}(P), \\
 &\quad C(i+1) \cap \text{Mod}(P), C(i+1) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)] \\
 &= [\min(C \text{ ref}(P), P), C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, \emptyset, \\
 &\quad C(i) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)].
 \end{aligned}$$

By assumption, the minimal class of C containing models of P is $C(i)$. As a result, the minimal class of $C \text{ ref}(P)$ containing models of P is $C(i) \cap \text{Mod}(P)$. As a result, $\min(C \text{ ref}(P), P) = \min(C, P)$. The total preorder above is therefore equivalent to $C \text{ ref}(P)$, since empty classes do not affect equivalence. \square

This reduction is applied to the running example.

Example 4. Restrained revision can be replaced by a refinement followed by natural revision:

$$\emptyset \text{ lex}(y) \text{ lex}(\neg x \wedge y) \text{ res}(x \wedge z) \text{ rad}(\neg z).$$

This operation results into the following sequence:

$$\emptyset \text{ lex}(y) \text{ lex}(\neg x \wedge y) \text{ ref}(x \wedge z) \text{ nat}(x \wedge z) \text{ rad}(\neg z).$$

The resulting natural revision can be then replaced by lexicographic revision. It can be seen that the minimal models of $x \wedge z$ in the ordering just before the natural revision are these of $x \wedge y \wedge z$:

$$\emptyset \text{ lex}(y) \text{ lex}(\neg x \wedge y) \text{ ref}(x \wedge z) \text{ lex}(x \wedge y \wedge z) \text{ rad}(\neg z).$$

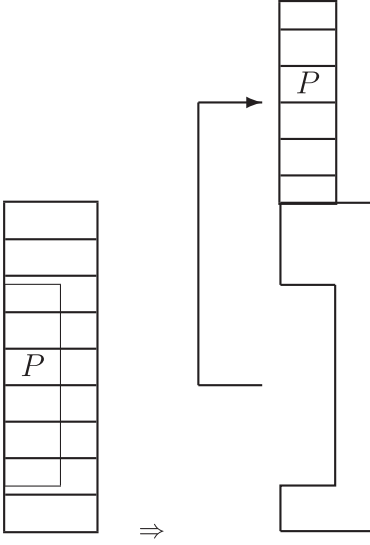
3.3 Very Radical Revision

Irrevocable revision [75] formalizes hypothetical reasoning by excluding from consideration all models that do not satisfy the assumption. Formally, these models are made inaccessible to revision, which cannot therefore recover them (hence the name). The scope of this article is limited to revisions that consider all models. While irrevocable revision excludes some model, the very radical revision variant by Rott [68] does not. Formally, it is defined as follows:

$$\begin{aligned} & [C(0), \dots, C(m)]\text{rad}(P) \\ &= [C(0) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), (C(0) \cup \dots \cup C(m)) \setminus \text{Mod}(P)]. \end{aligned}$$

The original definition has the first part $C(0) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P)$ only for the classes that intersect $\text{Mod}(P)$. The difference is inessential, since the other classes are empty and empty classes are irrelevant.

Graphically, the models of P maintain the same relative order, the others are merged in a single class at the end of the sequence.



Very radical revision can be expressed in terms of a sequence of a lexicographic revision, a severe antiwithdrawal and a second lexicographic revision. Intuitively, this is because very radical revision merges the classes not satisfying P , which is equivalent to make them minimal by a lexicographic revision by $\neg P$ and then by a severe antiwithdrawal by P ; a further lexicographic revision is needed to restore the correct ordering.

THEOREM 3. *For every total preorder C and formula P , it holds $C\text{rad}(P) \equiv C\text{lex}(\neg P)\text{sev}(P)\text{lex}(P)$.*

PROOF. By definition, $C\text{lex}(\neg P)\text{sev}(P)\text{lex}(P)$ is the following total preorder:

$$\begin{aligned} & [C(0), \dots, C(m)]\text{lex}(\neg P)\text{sev}(P)\text{lex}(P) \\ &= [C(0) \cap \text{Mod}(\neg P), \dots, C(m) \cap \text{Mod}(\neg P), \\ & \quad C(0) \setminus \text{Mod}(\neg P), \dots, C(m) \setminus \text{Mod}(\neg P)]\text{sev}(P)\text{lex}(P) \\ &= [C(0) \setminus \text{Mod}(P), \dots, C(m) \setminus \text{Mod}(P), \\ & \quad C(0) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P)]\text{sev}(P)\text{lex}(P) \end{aligned}$$

$$\begin{aligned}
&= [C(0) \setminus \text{Mod}(P) \cup \dots \cup C(m) \setminus \text{Mod}(P) \cup C(0) \cap \text{Mod}(P), \\
&\quad C(1) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P)]_{\text{lex}(P)} \\
&= [(C(0) \cup \dots \cup C(m)) \setminus \text{Mod}(P) \cup C(0) \cap \text{Mod}(P), \\
&\quad C(1) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P)]_{\text{lex}(P)} \\
&= [((C(0) \cup \dots \cup C(m)) \setminus \text{Mod}(P) \cup C(0) \cap \text{Mod}(P)) \cap \text{Mod}(P), \\
&\quad C(1) \cap \text{Mod}(P) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P) \cap \text{Mod}(P), \\
&\quad ((C(0) \cup \dots \cup C(m)) \setminus \text{Mod}(P) \cup C(0) \cap \text{Mod}(P)) \setminus \text{Mod}(P), \\
&\quad C(1) \cap \text{Mod}(P) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P) \setminus \text{Mod}(P)] \\
&= [C(0) \cap \text{Mod}(P), \\
&\quad C(1) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), \\
&\quad (C(0) \cup \dots \cup C(m) \setminus \text{Mod}(P), \emptyset, \dots, \emptyset)].
\end{aligned}$$

Apart from the empty classes, this total preorder is $\text{Crad}(P)$. □

The reduction is applied to the running example.

Example 5. The previous replacements turned the sequence of revisions of the running example into the following:

$$\emptyset \text{lex}(y) \text{lex}(\neg x \wedge y) \text{ref}(x \wedge z) \text{lex}(x \wedge y \wedge z) \text{rad}(\neg z).$$

The last revision of the sequence $\text{rad}(\neg z)$ is replaced by $\text{lex}(\neg \neg z) \text{sev}(\neg z) \text{lex}(\neg z)$:

$$\emptyset \text{lex}(y) \text{lex}(\neg x \wedge y) \text{ref}(x \wedge z) \text{lex}(x \wedge y \wedge z) \text{lex}(z) \text{sev}(\neg z) \text{lex}(\neg z).$$

This sequence contains only lexicographic revisions, a refinement and a severe antiwithdrawal.

3.4 Severe Revisions

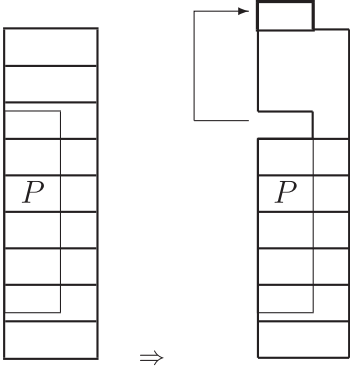
The Levi identity [36] allows constructing a revision operator from a contraction. This can be applied to severe withdrawal, leading to the definition of severe revision. However, the Levi identity only specifies the base of the revised ordering, the set of its minimal models. The rest of the ordering can be obtained in at least three different ways, leading to different revision operators [68].

The first definition is called just “severe revision.” Since the symbol sev is already taken for severe antiwithdrawal, sevr is used for this revision.

Definition 6. If $\min(C, P) = C(i) \cap \text{Mod}(P)$, then the severe revision sevr revises the total preorder C by formula P as follows:

$$\begin{aligned}
&[C(0), \dots, C(m)]_{\text{sevr}(P)} \\
&= [C(i) \cap \text{Mod}(P), (C(0) \cup \dots \cup C(i)) \setminus \text{Mod}(P), C(i+1), \dots, C(m)].
\end{aligned}$$

Graphically, the part of the first class that intersects $\text{Mod}(P)$ is moved at the beginning of the sequence; the rest of that class is merged with the previous ones.



This operator can be shown to be reducible to a severe antiwithdrawal followed by a natural revision, the latter being reducible to lexicographic revision as proved above.

THEOREM 4. *For every total preorder C and formula P , it holds $C_{\text{sevr}}(P) \equiv C_{\text{sev}}(P)\text{nat}(P)$.*

PROOF. Let $C = [C(0), \dots, C(m)]$ and i be the minimal index such that $C(i) \cap \text{Mod}(P) \neq \emptyset$. Revising C by $\text{sev}(P)$ and then $\text{nat}(P)$ produces:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{sev}(P)\text{nat}(P)} \\ &= [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)]_{\text{nat}(P)}. \end{aligned}$$

Since class zero of this ordering is $C(0) \cup \dots \cup C(i)$ and $C(i)$ intersects $\text{Mod}(P)$, it follows that the minimal index of a class of this ordering intersecting $\text{Mod}(P)$ is zero. As a result, natural revision produces:

$$\begin{aligned} & [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)]_{\text{nat}(P)} \\ &= [(C(0) \cup \dots \cup C(i)) \cap \text{Mod}(P), (C(0) \cup \dots \cup C(i)) \setminus \text{Mod}(P), \\ & \quad C(i+1), \dots, C(m)]_{\text{nat}(P)} \\ &= [C(i) \cap \text{Mod}(P), (C(0) \cup \dots \cup C(i)) \setminus \text{Mod}(P), C(i+1), \dots, C(m)]_{\text{nat}(P)} \end{aligned}$$

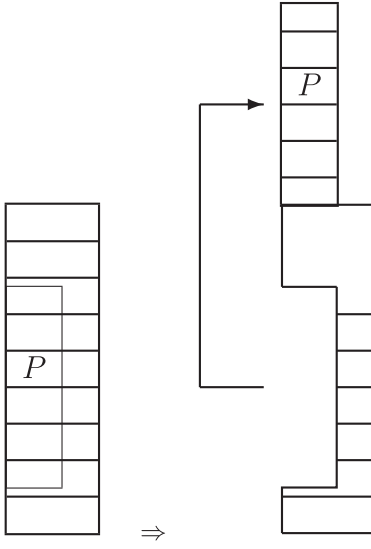
The last step follows from $C(i)$ being the minimal-index class of C intersecting $\text{Mod}(P)$, which implies $C(j) \cap \text{Mod}(P) = \emptyset$ for every $j < i$. The last total preorder is $C_{\text{sevr}}(P)$. \square

Moderate severe revision mixes a severe withdrawal with the changes lexicographic revision makes to a preorder. It will indeed be proved to be equivalent as a sequence of a severe antiwithdrawal and a lexicographic revision.

Definition 7. If $\min(C, P) = C(i) \cap \text{Mod}(P)$, then the moderate severe revision msev revises the total preorder C by formula P as follows:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{msev}(P)} \\ &= [C(0) \cap \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), \\ & \quad (C(0) \cup \dots \cup C(i)) \setminus \text{Mod}(P), C(i+1) \setminus \text{Mod}(P), \dots, C(m) \setminus \text{Mod}(P)]. \end{aligned}$$

The difference with severe revision is that all of $\text{Mod}(P)$ is shifted at the beginning of the sequence.



Moderate severe revision can be proved to be equivalent to a severe antiwithdrawal followed by a lexicographic revision.

THEOREM 5. *For every total preorder C and formula P , it holds $C_{msev}(P) \equiv sev(P)lex(P)$.*

PROOF. Let $C = [C(0), \dots, C(m)]$ and i be the index such that $\min(C, P) = C(i) \cap Mod(P)$. Revising C by $sev(P)$ and then $lex(P)$ produces:

$$\begin{aligned}
& [C(0), \dots, C(m)]_{sev(P)}lex(P) \\
&= [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)]lex(P) \\
&= [(C(0) \cup \dots \cup C(i)) \cap Mod(P), C(i+1) \cap Mod(P), \dots, C(m) \cap Mod(P), \\
&\quad (C(0) \cup \dots \cup C(i)) \setminus Mod(P), C(i+1) \setminus Mod(P), \dots, C(m) \setminus Mod(P))] \\
&= [C(i) \cap Mod(P), C(i+1) \cap Mod(P), \dots, C(m) \cap Mod(P), \\
&\quad (C(0) \cup \dots \cup C(i)) \setminus Mod(P), C(i+1) \setminus Mod(P), \dots, C(m) \setminus Mod(P))] \\
&\equiv [C(0) \cap Mod(P), \dots, C(i-1) \cap Mod(P), \\
&\quad C(i) \cap Mod(P), C(i+1) \cap Mod(P), \dots, C(m) \cap Mod(P), \\
&\quad (C(0) \cup \dots \cup C(i)) \setminus Mod(P), C(i+1) \setminus Mod(P), \dots, C(m) \setminus Mod(P)].
\end{aligned}$$

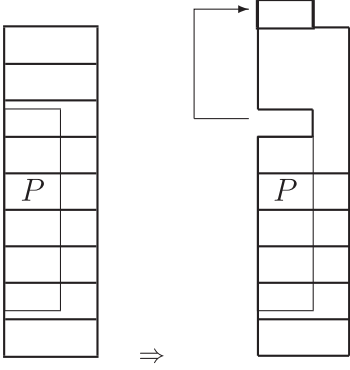
Equivalence $(C(0) \cup \dots \cup C(i)) \cap Mod(P) = C(i) \cap Mod(P)$ holds, because $C(i)$ is by assumption the lowest-index class intersecting $Mod(P)$. For the same reason, $C(0) \cap Mod(P), \dots, C(i-1) \cap Mod(P)$ are all empty; therefore, their introduction leads to an equivalent preorder. The preorder obtained this way is $C_{msev}(P)$. \square

The last variant of severe revision is plain severe revision.

Definition 8. If $\min(C, P) = C(i) \cap Mod(P)$ and j is the minimal index such that $i < j$ and $C(j) \neq \emptyset$ if any, otherwise $j = i + 1$, then plain severe revision $psev$ revises the total preorder C by formula P as follows:

$$\begin{aligned}
& [C(0), \dots, C(m)]_{psev}(P) \\
&= [C(i) \cap Mod(P), C(0) \cup \dots \cup (C(i) \setminus Mod(P)) \cup C(j), C(j+1), \dots, C(m)].
\end{aligned}$$

Moderate severe revision differs from severe revision in that it also merges the first non-empty class following the first that intersects $\text{Mod}(P)$.



Plain severe revision can be reformulated in terms of severe antiwithdrawal and lexicographic revision.

THEOREM 6. *For every total preorder C and formula P , it holds $C_{\text{psev}}(P) \equiv C_{\text{sev}}(\neg K') \text{lex}(K)$ where $K = \text{Form}(\min(C, P))$ and $K' = \text{Form}(\min(C_{\text{sev}}(P), \top)$.*

$$\begin{aligned} & [C(0), \dots, C(i), C(i+1), \dots, C(m)]_{\text{sev}}(P) \\ &= [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)]. \end{aligned}$$

PROOF. Let i and j be the indexes as in the definition of plain severe revision. The models of K' are the first non-empty class of the ordering $C_{\text{sev}}(P)$, where $C = [C(0), \dots, C(m)]$:

$$\begin{aligned} & [C(0), \dots, C(i), C(i+1), \dots, C(m)]_{\text{sev}}(P) \\ &= [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)]. \end{aligned}$$

By definition i is such that $C(i) \cap \text{Mod}(P) \neq \emptyset$. As a result, $C(0) \cup \dots \cup C(i)$ is not empty and therefore defines the set of models of K' . The models of $\neg K'$ are the other ones:

$$\text{Mod}(\neg K') = C(i+1) \cup \dots \cup C(m).$$

The ordering $C_{\text{sev}}(\neg K') \text{lex}(K)$ can now be determined. By construction, none of the classes $C(0), \dots, C(i)$ intersect $\text{Mod}(\neg K')$. The next class $C(i+1)$ may, but only if it is not empty. In particular, the lowest index class intersecting $\text{Mod}(K')$ is $C(j)$:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{sev}}(\neg K') \text{lex}(K) \\ &= [C(0) \cup \dots \cup C(i) \cup \dots \cup C(j), C(j+1), \dots, C(m)] \text{lex}(K) \\ &= [C(i) \cap \text{Mod}(P), C(0) \cup \dots \cup (C(i) \setminus \text{Mod}(P)) \cup \dots \cup C(j), C(j+1), \dots, C(m)] \\ &\equiv [C(i) \cap \text{Mod}(P), C(0) \cup \dots \cup (C(i) \setminus \text{Mod}(P)) \cup C(j), C(j+1), \dots, C(m)]. \end{aligned}$$

The last simplification can be done, because all classes between $C(i)$ and $C(j)$ are by definition empty. What results coincides with the definition of $C_{\text{psev}}(P)$. \square

The following theorem shows that plain severe revision is not able to increase the number of levels over two. It also links it with full meet revision, to be defined in the next section.

THEOREM 7. *If C has at most two non-empty classes, then $C_{\text{psev}}(P) \equiv [C(i) \cap \text{Mod}(P), \text{Mod}(\top) \setminus (C(i) \cap \text{Mod}(P))]$ holds for every formula P , where $\min(C, P) = C(i) \cap \text{Mod}(P)$.*

PROOF. Since C has at most two non-empty classes, and removing empty classes produces an equivalent preorder, it can be assumed $C = [C(0), C(1)]$ where $C(0) \neq \emptyset$ while $C(1)$ may be empty. The definition of the plain severe revision depends on the minimal class i intersecting $\text{Mod}(P)$ and the minimal non-empty class of index greater than i . Since C has only two classes, i can only be 0 or 1. In the first case, j can only be 1, regardless of whether $C(1)$ is empty or not. As a result:

$$\begin{aligned} & [C(0), C(1)]\text{psev}(P) \\ &= [C(0) \cap \text{Mod}(P), (C(0) \setminus \text{Mod}(P)) \cup C(1)] \\ &= [C(0) \cap \text{Mod}(P), \text{Mod}(\top) \setminus (C(0) \cap \text{Mod}(P))] \\ &= [C(0) \cap \text{Mod}(P), \text{Mod}(\top) \setminus (C(i) \cap \text{Mod}(P))]. \end{aligned}$$

If $C(0) \cap \text{Mod}(P) = \emptyset$, then $i = 1$ and $j = i + 1 = 2$, since no class of index greater than i exists, therefore none is different from the empty set. As a result:

$$\begin{aligned} & [C(0), C(1)]\text{psev}(P) \\ &= [C(1) \cap \text{Mod}(P), C(0) \cup (C(1) \setminus \text{Mod}(P)) \cup C(2)] \\ &= [C(1) \cap \text{Mod}(P), \text{Mod}(\top) \setminus (C(1) \cap \text{Mod}(P))] \\ &= [C(1) \cap \text{Mod}(P), \text{Mod}(\top) \setminus (C(i) \cap \text{Mod}(P))]. \end{aligned}$$

Since C has only two classes, $C(2)$ is empty and can be removed. Since $C(0) \cap \text{Mod}(P) = \emptyset$, the set $C(0) \cup (C(1) \setminus \text{Mod}(P))$ is equal to $(C(0) \cup C(1)) \setminus \text{Mod}(P)$, in turn equal to $\text{Mod}(\top) \setminus (C(1) \cap \text{Mod}(P))$. \square

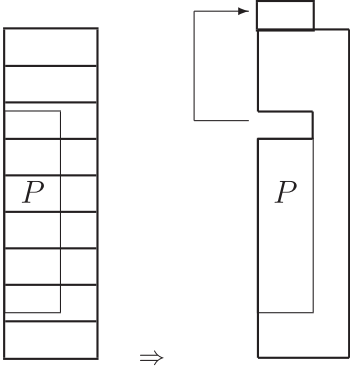
3.5 Full Meet Revision

Full meet revision was created for a single step of revision [1, 30] and then considered in iteration [2, 37, 54]. It was originally defined as the result of disjoining all possible ways of minimally revising a propositional theory, formalizing both the assumption of minimal change and that of a complete lack of knowledge about the plausibility of the various choices. The initial plausibility ordering is not used other than for its set of minimal models. The resulting ordering only distinguish models in two classes: the base and the others.

Definition 9. The full meet revision full revises an ordering C by a formula P as follows, where $\min(C, P) = C(i) \cap \text{Mod}(P)$:

$$\begin{aligned} & [C(0), \dots, C(m)]\text{full}(P) \\ &= [C(i) \cap \text{Mod}(P), \text{Mod}(\top) \setminus (C(i) \cap \text{Mod}(P))]. \end{aligned}$$

The models of the first class that intersects $\text{Mod}(P)$ are moved at the beginning of the sequences; all others are merged in a single, giant class.



Due to its simplicity, full meet revision can be expressed in a number of ways in terms of the other operators. For example, it is equivalent to a sequence made of a lexicographic revision followed by a severe antiwithdrawal and another lexicographic revision.

THEOREM 8. *For every total preorder C and formula P , it holds $C_{\text{full}}(P) \equiv \text{Clex}(\neg K)\text{sev}(K)\text{lex}(K)$, where $K = \text{Form}(\min(C, P))$.*

PROOF. Let $C = [C(0), \dots, C(m)]$ and i be the lowest index such that $C(i) \cap \text{Mod}(P) \neq \emptyset$. By definition, $\text{Mod}(K) = C(i) \cap \text{Mod}(P)$. Since $\text{Mod}(K) \subseteq C(i)$, it holds $\text{Mod}(K) \cap C(j) = \emptyset$ for every $j \neq i$:

$$\begin{aligned}
 & [C(0), \dots, C(m)]\text{lex}(\neg K)\text{sev}(K)\text{lex}(K) \\
 &= [C(0) \cap \text{Mod}(\neg K), \dots, C(m) \cap \text{Mod}(\neg K), \\
 &\quad C(0) \setminus \text{Mod}(\neg K), \dots, C(m) \setminus \text{Mod}(\neg K)]\text{sev}(K)\text{lex}(K) \\
 &= [C(0) \setminus \text{Mod}(K), \dots, C(m) \setminus \text{Mod}(K), \\
 &\quad C(0) \cap \text{Mod}(K), \dots, C(m) \cap \text{Mod}(K)]\text{sev}(K)\text{lex}(K) \\
 &= [C(0) \setminus \text{Mod}(K), \dots, C(m) \setminus \text{Mod}(K), \text{Mod}(K)]\text{sev}(K)\text{lex}(K), \\
 &\quad \text{since } \text{Mod}(K) \cap C(j) = \emptyset \text{ if } j \neq i \text{ and } \text{Mod}(K) \cap C(i) = \text{Mod}(K) \\
 &= [\text{Mod}(\top), \emptyset]\text{lex}(K) \\
 &\equiv [\text{Mod}(\top)]\text{lex}(K) \\
 &= [\text{Mod}(K), \text{Mod}(\neg K)].
 \end{aligned}$$

Since $\text{Mod}(K) = C(i) \cap \text{Mod}(P)$, the final total preorder is $C_{\text{full}}(P)$. □

An alternative reduction is $C_{\text{full}}(P) = \text{Clex}(\neg \text{Form}(\{I\}))\text{sev}(\text{Form}(\{I\}))\text{lex}(K)$, where I is an arbitrary propositional interpretation. Indeed, the proof relies on $\text{Clex}(\neg F)\text{sev}(F) = [\text{Mod}(\top)]$, which holds for every formula F such that $\text{Mod}(F)$ is contained in a single class of C . This is the case for $\min(C, P)$ but also for every formula having only one model.

Yet another reduction is $C_{\text{full}}(P) = \emptyset\text{lex}(K)$ dove $K = \text{Form}(\min(C, P))$. This is, however, not used, because contrary to the other reductions it affects the previous sequence of revisions. For example, $C_{\text{nat}}(N)\text{full}(P)$ is turned into $\emptyset\text{lex}(K)$, therefore making the initial natural revision disappear.

The previous revisions are instead preserved by the reduction $C_{\text{full}}(P) \equiv \text{Crad}(K)$ where $K = \text{Form}(\min(C, P))$, which however requires the calculation of the minimal models of P in the total preorder C before being applied. The very radical revision can be then expressed in terms of two lexicographic revisions and a severe antiwithdrawal.

Starting from an empty ordering, full meet revision and plain severe revision behave in exactly the same way. Formally, for every sequence of formulae P_1, \dots, P_n , it holds

$$\emptyset \text{full}(P_1) \dots \text{full}(P_n) \equiv \emptyset \text{psev}(P_1) \dots \text{psev}(P_n).$$

This means that a sequence of mixed plain severe and full meet revisions can be turned into one containing only one type of revisions. This fact is a consequence of how they change an ordering comprising one or two classes: they both produce an ordering containing the class $\min(C, P)$ and the class containing all other models. For full meet revision, this is the definition and holds in all cases. For plain severe revision this is proved in Theorem 7.

4 THE ALGORITHM

The previous section shows that every considered belief change operator can be reduced to a sequence of lexicographic revisions, refinements, and severe antiwithdrawals. As a result, every sequence of operators can be turned into one made of these three only. This section presents an algorithm for computing the base of the ordering at every time step of such a sequence.

This is done by first proving that refinements and severe antiwithdrawals can be removed by suitably modifying the sequence. This is done differently than the reductions in the previous section, which only modify the sequence locally: Nothing is changed before or after the operator that is replaced. Removing refinements instead requires introducing lexicographic revisions in other points of the sequence, and removing severe antiwithdrawals requires changing the previous lexicographic revisions.

The algorithm that computes the bases of a sequence of lexicographic revisions is then modified to work on the original sequence. The detour to the sequence of lexicographic revisions is necessary to prove that the final algorithm works. In particular, it is shown to do the same as the original algorithm on the simplified sequence.

All sequences are assumed to start with the empty ordering \emptyset . Every other ordering $C = [C(0), \dots, C(m)]$ is the result of a sequence of lexicographic revision applied to the empty ordering: $\emptyset \text{lex}(\text{Form}(C(m))) \dots \text{lex}(\text{Form}(C(0)))$.

4.1 Simplification

A sequence of lexicographic revisions ending in either a refinement or a severe antiwithdrawal can be turned into a sequence of lexicographic revisions that has exactly the same final ordering when applied to the same original ordering. As a result, a sequence containing every of these three operators can be scanned from the beginning until the first operator that is not a lexicographic revising is found. The initial part of the sequence is then turned into a sequence containing only lexicographic revisions, and the process restarted.

Removal of the refinements is done thanks to the following theorems, which proves that a refinement can be moved at the beginning of a sequence of lexicographic revisions, and then turned into a lexicographic revision itself.

THEOREM 9. *For every ordering C and two formulae R and L , it holds $C \text{lex}(L) \text{ref}(R) = C \text{ref}(R) \text{lex}(L)$.*

PROOF. According to the definitions of lex and ref , the ordering $C \text{lex}(L) \text{ref}(R)$ is

$$\begin{aligned} & [C(0), \dots, C(m)] \text{lex}(L) \text{ref}(R) \\ &= [C(0) \cap \text{Mod}(L), \dots, C(m) \cap \text{Mod}(L), C(0) \setminus \text{Mod}(L), \dots, C(m) \setminus \text{Mod}(L)] \text{ref}(R) \\ &= [C(0) \cap \text{Mod}(L) \cap \text{Mod}(R), C(0) \cap \text{Mod}(L) \setminus \text{Mod}(R) \end{aligned}$$

$$\begin{aligned}
& \dots C(m) \cap \text{Mod}(L) \cap \text{Mod}(R), C(m) \cap \text{Mod}(L) \setminus \text{Mod}(R), \\
& C(0) \setminus \text{Mod}(L) \cap \text{Mod}(R), C(0) \setminus \text{Mod}(L) \setminus \text{Mod}(R) \\
& \dots C(m) \setminus \text{Mod}(L) \cap \text{Mod}(R) C(m) \setminus \text{Mod}(L) \setminus \text{Mod}(R)].
\end{aligned}$$

The ordering resulting from the opposite application $C \text{ ref}(R) \text{ lex}(L)$ is

$$\begin{aligned}
& [C(0), \dots, C(m)] \text{ ref}(R) \text{ lex}(L) \\
& = [C(0) \cap \text{Mod}(R), C(0) \setminus \text{Mod}(R) \dots C(m) \cap \text{Mod}(R), C(m) \setminus \text{Mod}(R)] \text{ lex}(L) \\
& = [C(0) \cap \text{Mod}(R) \cap \text{Mod}(L), C(0) \setminus \text{Mod}(R) \cap \text{Mod}(L) \\
& \quad \dots C(m) \cap \text{Mod}(R) \cap \text{Mod}(L), C(m) \setminus \text{Mod}(R) \cap \text{Mod}(L), \\
& \quad C(0) \cap \text{Mod}(R) \setminus \text{Mod}(L), C(0) \setminus \text{Mod}(R) \setminus \text{Mod}(L) \\
& \quad \dots C(m) \cap \text{Mod}(R) \setminus \text{Mod}(L), C(m) \setminus \text{Mod}(R) \setminus \text{Mod}(L)].
\end{aligned}$$

Since \cap and \setminus commute, these two sequences are the same. \square

This proves that $\emptyset \text{ lex}(L_1) \dots \text{lex}(L_{n-1}) \text{ lex}(L_n) \text{ ref}(R)$ is equal to $\emptyset \text{ lex}(L_1) \dots \text{lex}(L_{n-1}) \text{ ref}(R) \text{ lex}(L_n)$. Iteratively applying commutativity produces $\emptyset \text{ ref}(R) \text{ lex}(L_1) \dots \text{lex}(L_n)$. Since $\emptyset = [\text{Mod}(\top)]$, by definition $\emptyset \text{ ref}(R) = [\text{Mod}(R), \text{Mod}(\neg R)]$, and this is also the total preorder $\emptyset \text{ lex}(R)$. As a result, the whole sequence is equivalent to $\emptyset \text{ lex}(R) \text{ lex}(L_1) \dots \text{lex}(L_n)$.

COROLLARY 1. *For every formulae L_1, \dots, L_n and R , it holds*

$$\emptyset \text{ lex}(L_1) \dots \text{lex}(L_{n-1}) \text{ lex}(L_n) \text{ ref}(R) \equiv \emptyset \text{ lex}(R) \text{ lex}(L_1) \dots \text{lex}(L_{n-1}) \text{ lex}(L_n).$$

If a sequence contains lex and ref , then every $\text{ref}(R)$ in order can be moved to the beginning of the sequence and then replaced by $\text{lex}(L)$. What results is a sequence containing only lexicographic revisions.

Example 6. The sequence in the running example was changed to comprise lexicographic revisions, refinements and severe antiwithdrawals only:

$$\emptyset \text{ lex}(y) \text{ lex}(\neg x \wedge y) \text{ ref}(x \wedge z) \text{ lex}(x \wedge y \wedge z) \text{ lex}(z) \text{ sev}(\neg z) \text{ lex}(\neg z).$$

The corollary above shows that $\text{ref}(x \wedge z)$ can be moved to the beginning of the sequence and there turned into a lexicographic revision:

$$\emptyset \text{ lex}(x \wedge z) \text{ lex}(y) \text{ lex}(\neg x \wedge y) \text{ lex}(x \wedge y \wedge z) \text{ lex}(z) \text{ sev}(\neg z) \text{ lex}(\neg z).$$

This transformation can be used to prove the folklore theorem linking lexicographic revision with maxcon , usually defined for sets of formulae and extended to sequences by Booth and Nittka [9]:

$$\begin{aligned}
\text{maxcon}(F) & \equiv F, \\
\text{maxcon}(F_1, F_2) & \equiv \begin{cases} F_1 \wedge F_2 & \text{if consistent,} \\ F_1 & \text{otherwise,} \end{cases} \\
\text{maxcon}(F_1, \dots, F_n) & \equiv \text{maxcon}(\text{maxcon}(F_1, \dots, F_{n-1}), F_n).
\end{aligned}$$

The theorem establishes that maxcon can be used to determine the minimal models of a formula in the ordering resulting from a sequence of lexicographic revisions. The proof is included here for the sake of completeness.

THEOREM 10. *For every formula P and sequence of formulae L_1, \dots, L_n , it holds*

$$\min(\emptyset \text{ lex}(L_1) \dots \text{lex}(L_n), P) = \text{Mod}(\text{maxcon}(P, L_n, \dots, L_1)).$$

PROOF. Proved by induction on the length of the sequence. With $n = 0$, $\maxcon(P) = P$ and $\min(\emptyset, P) = \text{Mod}(P)$. The claim therefore holds.

If the claim holds for $n - 1$ formulae, then $\min(\emptyset \text{lex}(L_2) \dots \text{lex}(L_n), P) = \text{Mod}(\maxcon(P, L_n, \dots, L_2))$. The same has to be proved with a formula L_1 more.

Let $C = [C(0), \dots, C(m)] = \emptyset \text{lex}(L_2) \dots \text{lex}(L_n)$. By the above theorem, $\emptyset \text{lex}(L_1) \text{lex}(L_2) \dots \text{lex}(L_n) = \emptyset \text{lex}(L_2) \dots \text{lex}(L_n) \text{ref}(L_1) = C \text{ref}(L_1)$.

Let i be the index such that $\min(C, P) = C(i) \cap \text{Mod}(P)$. This implies that $C(0), \dots, C(i - 1)$ do not intersect $\text{Mod}(P)$. By definition, $C \text{ref}(L_1)$ is

$$\begin{aligned} & [C(0), \dots, C(m)] \text{ref}(L_1) \\ &= [C(0) \cap \text{Mod}(L_1), C(0) \setminus \text{Mod}(L_1), \dots, C(i - 1) \cap \text{Mod}(L_1), C(i - 1) \setminus \text{Mod}(L_1), \\ & \quad C(i) \cap \text{Mod}(L_1), C(i) \setminus \text{Mod}(L_1), \dots, C(m) \cap \text{Mod}(L_1), C(m) \setminus \text{Mod}(L_1)]. \end{aligned}$$

Since $C(0), \dots, C(i - 1)$ do not intersect $\text{Mod}(P)$, the minimal class of $\text{Cref}(P)$ doing that is $C(i) \cap \text{Mod}(L_1)$ if not empty and $C(i) \setminus \text{Mod}(L_1)$ otherwise. In the second case, $C(i) \setminus \text{Mod}(L_1) = C(i)$, since $C(i) \cap \text{Mod}(L_1) = \emptyset$. Therefore, $\min(C \text{ref}(L_1), P)$ is $C(i) \cap \text{Mod}(L_1) \cap \text{Mod}(P)$ if not empty and $C(i) \cap \text{Mod}(P)$ otherwise.

By the inductive assumption, $\min(C, P) = \text{Mod}(\maxcon(P, L_n, \dots, L_2))$, and by definition of i it holds $C(i) \cap \text{Mod}(P) = \min(C, P)$. Therefore, $\min(\text{Cref}(L_1), P)$ is $\text{Mod}(\maxcon(P, L_n, \dots, L_2)) \cap \text{Mod}(L_1)$ if not empty, and $\text{Mod}(\maxcon(P, L_n, \dots, L_2))$ otherwise. In terms of formulae, $\min(\text{Cref}(P), P)$ is the set of models of $\maxcon(P, L_n, \dots, L_2) \wedge L_1$ if this formula is consistent and of $\maxcon(P, L_n, \dots, L_2)$ otherwise. By the recursive definition of \maxcon , $\min(C \text{ref}(L_1), P) = \text{Mod}(\maxcon(P, L_n, \dots, L_2, L_1))$. \square

In a sequence of lexicographic revisions, refinements, and severe antiwithdrawals, if the first operator of the sequence that is not a lexicographic revision is a refinement, then it can be turned into a lexicographic revision and moved to the beginning of the sequence. If it is a severe antiwithdrawal, then a more complex change needs to be applied to the sequence.

As the previous refinements can be turned into lexicographic revisions, the previous belief change operators can be all assumed to be lexicographic revisions. In other words, the considered sequence has all lexicographic revisions but the last operator, which is a severe antiwithdrawal. Such a sequence can be modified as follows, where $B = \text{under}(S; L_1, \dots, L_1)$ is a formula defined as follows:

$$\emptyset \text{lex}(L_1) \dots \text{lex}(L_n) \text{sev}(S) = \emptyset \text{lex}(L_1 \vee B) \dots \text{lex}(L_n \vee B) \text{lex}(B).$$

Intuitively, B is constructed so that it collects all models that are in the same class of the minimal ones of S or in lower classes. Disjoining every revising formula with B ensures that these models remain in class zero over each revision. The claim therefore requires two proofs: first, that B actually comprises these models; second, that modifying the lexicographic sequence this way does not change the resulting total preorder.

Definition 10. The *underformula* of a sequence of formulae is

$$\begin{aligned} & \text{under}(S; \epsilon) = \top, \\ & \text{under}(S; L_n, L_{n-1}, \dots, L_1) \\ &= \begin{cases} L_n \wedge \text{under}(S \wedge L_n; L_{n-1}, \dots, L_1) & \text{if } S \wedge L_n \text{ is consistent,} \\ L_n \vee \text{under}(S; L_{n-1}, \dots, L_1) & \text{otherwise.} \end{cases} \end{aligned}$$

Informally, this construction includes as alternatives the formulae that are excluded from $\text{maxcon}(P, L_n, \dots, L_1)$, because they are inconsistent with the partially built $\text{maxcon } M$. Starting from $M = S$, the procedure of maxcon construction adds L_i to M if $M \wedge L_i$ is consistent. Otherwise, L_i is skipped. This procedure results in the minimal models of S . If $M \wedge L_i$ is inconsistent only because of S , then its models are in lower classes than all models of S in the final total preorder. Disjoining M with L_i gathers all such models. This is obtained in the last case of the definition: L_i is disjoined with the underformula but not added to M . This intuition is formalized by the following theorem.

LEMMA 1. *If $C = [C(0), \dots, C(m)] = \emptyset \text{lex}(L_1) \dots \text{lex}(L_n)$ and i is the minimal index such that $C(i) \cap \text{Mod}(S) \neq \emptyset$, then*

$$C(0) \cup \dots \cup C(i) = \text{Mod}(\text{under}(S; L_n, \dots, L_1)).$$

PROOF. The class of a model I is lower or equal than all classes of $\text{Mod}(S)$ if and only if $I \in \min(C, S \vee \text{Form}(I))$. As a result, $C(0) \cup \dots \cup C(i)$ is defined by $\{I \mid I \models \text{maxcon}(S \vee \text{Form}(I), L_n, \dots, L_1)\}$. This set can be inductively proved to be equal to $\text{Mod}(\text{under}(S; L_n, \dots, L_1))$. By induction on n , the following is proved:

$$I \models \text{maxcon}(S \vee \text{Form}(I), L_n, \dots, L_1) \text{ iff } I \models \text{under}(S; L_n, \dots, L_1).$$

With $m = 0$, by definition $\text{under}(S; \epsilon) = \top$, which contains all models; $\text{maxcon}(S \vee \text{Form}(I)) = S \vee \text{Form}(I)$, which always contains I . The base case is therefore proved.

The induction step assumes

$$I \models \text{maxcon}(S \vee \text{Form}(I), L_{n-1}, \dots, L_1) \text{ iff } I \models \text{under}(S; L_{n-1}, \dots, L_1).$$

The claim is the same with L_n added. The underformula of the claim is by definition:

$$\begin{aligned} & \text{under}(S; L_n, L_{n-1}, \dots, L_1) \\ &= \begin{cases} L_n \wedge \text{under}(S \wedge L_n; L_{n-1}, \dots, L_1) & \text{if } S \wedge L_n \text{ is consistent,} \\ L_n \vee \text{under}(S; L_{n-1}, \dots, L_1) & \text{otherwise.} \end{cases} \end{aligned}$$

The maxcon is instead

$$\begin{aligned} & \text{maxcon}(S \vee \text{Form}(I), L_n, L_{n-1}, \dots, L_1) \\ &= \begin{cases} \text{maxcon}((S \vee \text{Form}(I)) \wedge L_n, L_{n-1}, \dots, L_1) & \text{if } (S \vee \text{Form}(I)) \wedge L_n \text{ is consistent,} \\ \text{maxcon}(S \vee \text{Form}(I), L_{n-1}, \dots, L_1) & \text{otherwise.} \end{cases} \end{aligned}$$

The claim is proved if $I \models \text{maxcon}(S \vee \text{Form}(I), L_n, L_{n-1}, \dots, L_1)$ is shown to be the same as $I \models \text{under}(S; L_n, L_{n-1}, \dots, L_1)$. This is done by reformulating the first condition. Since $(S \vee \text{Form}(I)) \wedge L_n$ is consistent if and only if $S \wedge L_n$ is consistent or $I \models L_n$, the first case in the definition of maxcon can be divided into three:

$$\begin{aligned} & \text{maxcon}(S \vee \text{Form}(I), L_n, L_{n-1}, \dots, L_1) \\ &= \begin{cases} \text{maxcon}((S \vee \text{Form}(I)) \wedge L_n, L_{n-1}, \dots, L_1) & \text{if } S \wedge L_n \text{ is consistent and } I \models L_n, \\ \text{maxcon}((S \vee \text{Form}(I)) \wedge L_n, L_{n-1}, \dots, L_1) & \text{if } S \wedge L_n \text{ is consistent and } I \not\models L_n, \\ \text{maxcon}((S \vee \text{Form}(I)) \wedge L_n, L_{n-1}, \dots, L_1) & \text{if } S \wedge L_n \text{ is inconsistent and } I \models L_n, \\ \text{maxcon}(S \vee \text{Form}(I), L_{n-1}, \dots, L_1) & \text{if } S \wedge L_n \text{ is inconsistent and } I \not\models L_n. \end{cases} \end{aligned}$$

In the first case, since $I \models L_n$ it follows that $I \models \text{maxcon}((S \vee \text{Form}(I)) \wedge L_n, L_{n-1}, \dots, L_1)$ is equivalent to $I \models L_n \wedge \text{maxcon}((S \vee \text{Form}(I)) \wedge L_n, L_{n-1}, \dots, L_1)$. The first argument $(S \vee \text{Form}(I)) \wedge L_n$ can be rewritten $(S \wedge L_n) \vee \text{Form}(I)$, since $I \models L_n$. By the induction assumption,

$I \models L_n \wedge \maxcon((S \wedge L_n) \vee \text{Form}(I), L_{n-1}, \dots, L_1)$ is the same as $I \models L_n \wedge \text{under}(S \wedge L_n; L_{n-1}, \dots, L_1)$. Since $S \wedge L_1$ is consistent, the latter is equivalent to $I \models \text{under}(S; L_n, \dots, L_1)$.

In the second case, the claim is proved by showing that I satisfies neither $\maxcon((S \vee \text{Form}(I)), L_n, L_{n-1}, \dots, L_1)$ nor $\text{under}(S; L_n, L_{n-1}, \dots, L_1)$. Since $S \wedge L_n$ is consistent, $(S \vee \text{Form}(I)) \wedge L_n$ is also consistent. As a result, $\maxcon((S \vee \text{Form}(I)), L_n, L_{n-1}, \dots, L_1)$ implies $(S \vee \text{Form}(I)) \wedge L_n$, which implies L_n . Since $I \not\models L_n$, it follows $I \not\models \maxcon((S \vee \text{Form}(I)), L_n, L_{n-1}, \dots, L_1)$. This model does not satisfy $\text{under}(S; L_n, L_{n-1}, \dots, L_1)$ either, because this formula is equal to $L_n \wedge \text{under}(S \wedge L_n; L_{n-1}, \dots, L_1)$ when $S \wedge L_n$ is consistent.

In the third case, $(S \vee \text{Form}(I)) \wedge L_n \equiv \text{Form}(I)$, since $S \wedge L_n$ is inconsistent and $I \models L_n$. As a result, I is the only model of $\maxcon(S \vee \text{Form}(I), L_n, L_{n-1}, \dots, L_1)$.

Together with the fourth case, this means that if $S \wedge L_n$ is inconsistent then $I \models \maxcon(S \vee \text{Form}(I), L_n, L_{n-1}, \dots, L_1)$ if and only if $I \models L_n$ or $I \models \maxcon(S \vee \text{Form}(I), L_{n-1}, \dots, L_1)$. By the induction assumption, the latter is equivalent to $I \models \text{under}(S; L_{n-1}, \dots, L_1)$. Therefore, the condition can be rewritten as $I \models L_n \vee \text{under}(S; L_{n-1}, \dots, L_1)$. Since $S \wedge L_n$ is inconsistent, $L_n \vee \text{under}(S; L_{n-1}, \dots, L_1) = \text{under}(S; L_n, L_{n-1}, \dots, L_1)$. \square

It is now shown that $\text{under}(S, L_n, \dots, L_1)$ allows rewriting the sequence without affecting the final total preorder.

THEOREM 11. *If $B = \text{under}(S; L_n, \dots, L_1)$, then*

$$\emptyset \text{lex}(L_1) \dots \text{lex}(L_n) \text{sev}(S) \equiv \emptyset \text{lex}(L_1 \vee B) \dots \text{lex}(L_n \vee B) \text{lex}(B).$$

PROOF. The claim is proved by showing that for every consistent formula P , its minimal models according to the two total preorders are the same.

By the previous theorem, if $B = \text{under}(S; L_n, \dots, L_1)$, then $\text{Mod}(B) = C(0) \cup \dots \cup C(i)$, where $C = [C(0), \dots, C(m)] = \emptyset \text{lex}(L_1) \dots \text{lex}(L_n)$ and i is the minimal index such that $C(i) \cap \text{Mod}(S) \neq \emptyset$. Since $C \text{sev}(S) = [C(0) \cup \dots \cup C(i), C(i+1), \dots, C(m)] = [\text{Mod}(B), C(i+1), \dots, C(m)]$, it follows that

$$\min(P, C \text{sev}(S)) = \begin{cases} \text{Mod}(P) \cap \text{Mod}(B) & \text{if not empty,} \\ \text{Mod}(P) \cap C(j) & \text{otherwise, for } j \text{ minimal index} \\ & \text{such that this set is not empty.} \end{cases}$$

Since $C = \emptyset \text{lex}(L_1) \dots \text{lex}(L_n)$, $\text{Mod}(P) \cap C(j)$ for the minimal j for which this set is not empty is the set of models of $\maxcon(P, L_n, \dots, L_1)$. As a result, $\min(P, C \text{sev}(S))$ can be rewritten as

$$\min(P, C \text{sev}(S)) = \begin{cases} \text{Mod}(P \wedge B) & \text{if consistent,} \\ \text{Mod}(\maxcon(P, L_n, \dots, L_1)) & \text{otherwise.} \end{cases}$$

Let $C' = \emptyset \text{lex}(L_1 \vee B) \dots \text{lex}(L_n \vee B) \text{lex}(B)$. It is now shown that $\min(C \text{sev}(S), P) = \min(C', P)$. Since C' results from applying a number of lexicographic revisions to an empty total preorder, it holds

$$\min(P, C') = \maxcon(P, B, L_n \vee B, \dots, L_1 \vee B).$$

If $P \wedge B$ is consistent, then it is consistent with all formulae $L_i \vee B$ but also entails all of them. As a result,

$$\begin{aligned} \min(P, C') &= \maxcon(P, B, L_n \vee B, \dots, L_1 \vee B) \\ &= \maxcon(P \wedge B, L_n \vee B, \dots, L_1 \vee B) \\ &= P \wedge B \wedge (L_n \vee B) \wedge \dots \wedge (L_1 \vee B) \\ &= P \wedge B. \end{aligned}$$

If $P \wedge B$ is inconsistent, then $P \models \neg B$; therefore,

$$\begin{aligned} \min(P, C') &= \maxcon(P, B, L_n \vee B, \dots, L_1 \vee B) \\ &= \maxcon(P, L_n \vee B, \dots, L_1 \vee B). \end{aligned}$$

Since $P \models \neg B$, then $P \models \neg B$. As a result, $P \wedge (L_n \vee B)$ is equivalent to $P \wedge L_i$. This implies that $\maxcon(P, L_n \vee B, \dots, L_1 \vee B)$ is $\maxcon(P \wedge L_n, L_{n-1} \dots, L_1 \vee B)$ if $P \wedge L_n$ is consistent, otherwise it is $\maxcon(P, L_{m-1} \dots, L_1 \vee B)$. This argument can be repeated for every $P \wedge L$ with $L \subseteq \{L_n, \dots, L_i\}$, proving that the result is $\maxcon(P, L_n, \dots, L_1)$. \square

These theorems tell how to modify a sequence into one that only contains lexicographic revisions: starting from the beginning, the first operator that is not lex can be $\text{ref}(R)$ or $\text{sev}(S)$; in the first case, it is turned into $\text{lex}(R)$ and moved at the beginning of the sequence; in the second case, the underformula B of the previous revisions (which are all lexicographic by assumption) is used to replace $\text{sev}(S)$ with $\text{lex}(B)$ and is disjoined to all previous revisions. This part of the sequence now contains only lexicographic revisions, and the process can therefore be repeated for the next ref or sev operator. The final result is a sequence of lexicographic revisions applied to the empty preorder.

Such a sequence not only has the correct vale $\min(C, P)$ at each step but also the same final preorder of the original sequence. This implies that it is equivalent to it even regarding subsequent revisions.

Example 7. The sequence in the running example has been shown to be equivalent to the following one, which only contains lexicographic revisions and a severe antiwithdrawal:

$$\emptyset \text{lex}(x \wedge z) \text{lex}(y) \text{lex}(\neg x \wedge y) \text{lex}(x \wedge y \wedge z) \text{lex}(z) \text{sev}(\neg z) \text{lex}(\neg z).$$

The severe antiwithdrawal can be turned into a lexicographic revision by first calculating its underformula:

$$\begin{aligned} &\text{under}(\neg z; z, x \wedge y \wedge z, \neg x \wedge y, y, x \wedge z) \\ &= z \vee \text{under}(\neg z; x \wedge y \wedge z, \neg x \wedge y, y, x \wedge z) \\ &= z \vee (x \wedge y \wedge z) \vee \text{under}(\neg z; \neg x \wedge y, y, x \wedge z) \\ &= z \vee (x \wedge y \wedge z) \vee (\neg x \wedge y \wedge \text{under}(\neg z \wedge \neg x \wedge y, y, x \wedge z)) \\ &= z \vee (x \wedge y \wedge z) \vee (\neg x \wedge y \wedge y \wedge \text{under}(\neg z \wedge \neg x \wedge y \wedge y, x \wedge z)) \\ &= z \vee (x \wedge y \wedge z) \vee (\neg x \wedge y \wedge y \wedge ((x \wedge z) \vee \text{under}(\neg z \wedge \neg x \wedge y \wedge y))) \\ &= z \vee (x \wedge y \wedge z) \vee (\neg x \wedge y \wedge y \wedge ((x \wedge z) \vee \top)). \end{aligned}$$

This formula is equivalent to $B = z \vee (\neg x \wedge y)$. The sequence is therefore turned into

$$\emptyset \text{lex}((x \wedge z) \vee B) \text{lex}(y \vee B) \text{lex}((\neg x \wedge y) \vee B) \text{lex}((x \wedge y \wedge z) \vee B) \text{lex}(z \vee B) \text{lex}(B) \text{lex}(\neg z).$$

Some simplifications can be then applied. For example, $(x \wedge z) \vee B = (x \wedge z) \vee (z \vee (\neg x \wedge y)) \equiv z \vee (\neg x \wedge y)$ and $y \vee (z \vee (\neg x \wedge y)) \equiv y \vee z$.

In this particular case, only one severe antiwithdrawal occurs. More generally, they are transformed into lexicographic revisions starting from the first.

The only apparent drawback of this procedure is that every $\text{sev}(S)$ requires the underformula B to be disjoined to all previous formulae. This makes B to be included in the underformula of the next $\text{sev}(S')$. This problem is solved by leaving the sequence as it is and processing it as if the transformation has been done.

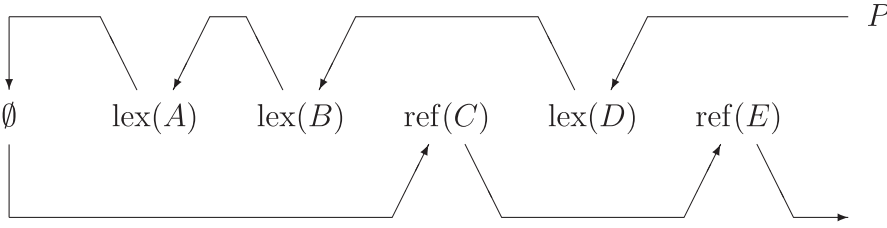
4.2 Algorithm

A sequence contains only lexicographic revisions and refinements applied to the empty ordering can be turned into a sequence of lexicographic revisions by moving all refinements to the beginning. After this change, the minimal models of a formula P can be calculated using the maxcon construction. Since the refinements are moved to the start of the sequence in the order in which they are encountered, they end up there in reverse order. As a result, the maxcon can be calculated from the original sequence following the order that would result from the simplification:

- (1) start with $M = P$;
- (2) proceeding from the end to the start of the sequence, for every $\text{lex}(L_i)$ turn M into $M \wedge L_i$ if this formula is consistent;
- (3) from the start to the end of the sequence, for every $\text{ref}(R_i)$ turn M into $M \wedge R_i$ if this formula is consistent.

The back and forth algorithm.

The following figure shows how the algorithm proceeds when computing a formula equivalent to the set of the minimal models of P . Every formula encountered following the arrows is conjoined with it if that does not result in contradiction.



The back and forth algorithm works, because it builds the maxcon starting from P and proceeding in the same order as if the refinements were moved to the start of the sequence. Its correctness is therefore proved by Corollary 1. For the same reason, a similar mechanism determines an underformula instead of a maxcon.

This is important, because a sequence may contain lexicographic revisions, refinements and severe antiwithdrawals. Assuming that the underformulae for the latter have all been determined, at each severe antiwithdrawal encountered while going back, if B is consistent with the current maxcon M , then M is turned into $M \wedge B$ and the procedure “bounces” back in the forward direction. This is because if $M \wedge B$ is consistent then the previous $\text{lex}(L)$ in the original sequence would be turned into $\text{lex}(L \vee B)$ in the modified sequence. As a result, M is consistent with all of them, but their addition is irrelevant, because M is already conjoined with B .

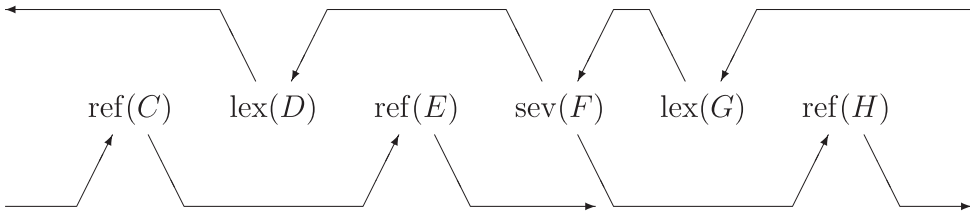
- (1) Start at the end of the sequence with $M = P$ and go back;
- (2) for every $\text{lex}(L_i)$ turn M into $M \wedge L_i$ if this formula is consistent; regardless, continue going backwards;
- (3) for every $\text{sev}(S)$, if M is consistent with its underformula B , then turn M into $M \wedge B$ and bounce forward, toward the end of the sequence; otherwise, continue going backwards;
- (4) at the start of the sequence, bounce forward, toward the end of the sequence;
- (5) when proceeding forward, for every $\text{ref}(R_i)$ turn M into $M \wedge R_i$ if this formula is consistent; regardless, continue going forward.

The back, bounce and forth algorithm

The fourth point can be omitted by placing $\text{sev}(\top)$ at the very beginning of the sequence. This marker signals the algorithm to bounce forward without the need to verify whether the sequence is at the start. At the end M has the minimal models of the final ordering.

The algorithm works, because it builds a formula that is the same that would have been produced when creating the maxcon of the modified sequence that only contains lexicographic revisions.

The following figure shows how the algorithm moves in a segment of the sequence. When it reaches $\text{sev}(F)$, if the formula that is currently being built is consistent with the underformula of this severe antiwithdrawal, then the algorithm bounces forward to $\text{ref}(H)$, otherwise it keeps going back, to $\text{lex}(D)$.



This construction produces a maxcon. The underformula of each severe antiwithdrawal is built similarly, by going back to the start of the sequence and coming back.

Example 8. The algorithm is applied to the sequence of the running example. It first determines the underformula of the severe antiwithdrawal, then the base at the end of the sequence:

$$\emptyset \text{lex}(y) \text{lex}(\neg x \wedge y) \text{ref}(x \wedge z) \text{lex}(x \wedge y \wedge z) \text{lex}(z) \text{sev}(\neg z) \text{lex}(\neg z).$$

The first step is to determine the underformula of the first severe revision in the sequence. This is done by following the back and forth procedure: first go back through the lexicographic revisions, then come forth through the refinements:

$$\begin{array}{cccccccc} 0 & \text{lex}(y) & \text{lex}(\neg x \wedge y) & \text{ref}(x \wedge z) & \text{lex}(x \wedge y \wedge z) & \text{lex}(z) & \text{sev}(\neg z) & \text{lex}(\neg z) \\ & 5 & 4 & & 3 & 2 & 1 & \\ & & & 6 & & & & \end{array}$$

Following the numbers, the formulae are in the sequence $\neg z$, z , $x \wedge y \wedge z$, $\neg x \wedge y$, y , and $x \wedge z$. As a result, the underformula of the severe antiwithdrawal is calculated on this sequence:

$$\text{under}(\neg z; z, x \wedge y \wedge z, \neg x \wedge y, y, x \wedge z).$$

This has been previously shown to be equivalent to $B = z \vee (\neg x \wedge y)$. It allows determining the final base by following the arrows as in the back, bounce and forth algorithm:

$$\begin{array}{cccccccc} 0 & \text{lex}(y) & \text{lex}(\neg x \wedge y) & \text{ref}(x \wedge z) & \text{lex}(x \wedge y \wedge z) & \text{lex}(z) & \text{sev}(\neg z) & \text{lex}(\neg z) \\ & & & & & 3? & 2 & 1 \\ & & & & & & & 3? \end{array}$$

The choice of keeping going back or bouncing forth depends on the consistency of the formula under construction with the underformula of $\text{sev}(\neg z)$. In this case, the formula is $\neg z$ and the underformula $z \vee (\neg x \wedge y)$. Since their conjunction $\neg z \wedge \neg x \wedge y$ is consistent, the algorithm bounces. Since there are no refinement after the severe antiwithdrawal, the resulting base is this formula.

The algorithm can be adapted to work with the other considered revisions without replacing them with lex , ref and sev , since each of them can be “locally” replaced with a sequence of these three. As a result, while going forward or backwards, suffices to behave in the same way as if the replacement has been done. This is done in the complexity analysis, in the next section.

5 COMPLEXITY

The reductions shown in the previous sections prove that each considered belief change operators can be turned into a lexicographic revision, possibly by first calculating a maxcon or an underformula. Both operations can be done by a polynomial number of calls to a propositional satisfiability solver. Therefore, the complexity of a sequence of arbitrary and mixed belief change operators is in the complexity class Δ_2^P , which contains all problems that can be solved by a polynomial number of calls to an NP oracle.

THEOREM 12. *The problems of establishing whether the base resulting from a sequence of lexicographic, natural, restrained, very radical and severe revisions, refinements and severe antiwithdrawals applied to the empty ordering implies a formula is in Δ_2^P .*

PROOF. The complexity class Δ_2^P contains all problems that can be solved by a polynomial number of tests of propositional satisfiability. Membership to this class is proved by showing that the considered problems can be solved by an algorithm that takes polynomial time, excluding the time needed to check the satisfiability of some propositional formulae.

The transformations of Section 3 and the algorithm of Section 4 require polynomial time, excluding the time needed to establishing the satisfiability of a polynomial number of propositional formulae.

The algorithm starts from the end of the sequence, goes back and comes forward. For each formula it meets in this journey, it performs a satisfiability check. The number of such satisfiability tests is linear in the number of formulae. The same happens when determining an underformula rather than the result of a revision.

Some transformations require no satisfiability check (e.g., $\text{msev}(P)$ is turned into $\text{sev}(P)\text{lex}(P)$); the others require the result of a revision or two (e.g., $\text{nat}(P)$ is turned into $\text{lex}(K)$, where K is the result of lexicographically revising by P). The result of a revision can be computed by a linear number of satisfiability tests, as shown in the previous paragraph. It is required at most twice for each formula in the sequence. The total number of satisfiability checks is quadratic in the length of the sequence.

The size of these results of revisions and of the underformulae is kept low by formulae $x \equiv F$, called definitions. Each formula F in the sequence is replaced by a new variable x , and $x \equiv F$ is added to a set of definitions. These definitions are used in all subsequent satisfiability or entailment tests. Every time the result of a revision or an underformula is generated, it is replaced in the sequence by a new variable, whose definition is added to the set of definitions. Since the results of revisions and the underformulae are linear combinations of the previous formulae, which are always one of these new variables each, they are linear in size. \square

The problem is also easily shown to be hard for the same class even if all operators are lexicographic revisions. This was previously published without proof [55].

THEOREM 13. *The problem of establishing whether the base resulting from a sequence of lexicographic, natural, restrained, very radical and severe revisions, refinements and severe antiwithdrawals applied to the empty ordering implies a formula is Δ_2^P -hard even if the sequence comprises lexicographic revisions only or refinements only.*

PROOF. Hardness to Δ_2^P is proved as usual by showing that a problem that is already known to be hard can be reduced in polynomial time to the problem under consideration, in this case checking entailment from a sequence of iterated revisions.

For this specific proof, the problem that is known to be Δ_2^P -hard is that of the lexicographically maximally model. An instance of this problem is a propositional formula F over an alphabet of vari-

ables $\{x_1, \dots, x_n\}$ assumed sorted by index. Given two models over this alphabet, the lexicographic order compares them according to the their value of x_1 , where true is greater than false; if they are the same, then it compares the values of x_2 in the same way. The problem is to establish whether the lexicographically maximal model of F assigns true to x_n . This check is Δ_2^P -complete [51].

A simple reduction translates this problem into the similar one where the formula is satisfiable: a possibly unsatisfiable formula F is turned into the satisfiable formula $(\neg x_0 \wedge \neg x_1 \wedge \dots \wedge \neg x_n) \vee F$, where x_0 is a new variable. A further reduction shows that the maximal model of a satisfiable formula F satisfies x_n if and only if the base of $\emptyset \text{lex}(x_n) \dots \text{lex}(x_1) \text{lex}(F)$ implies x_n . This proves that entailment for a sequence of lexicographic revisions is Δ_2^P -hard. The sequence is equivalent to $\emptyset \text{ref}(F) \text{ref}(x_1) \dots \text{ref}(x_n)$, proving the Δ_2^P -hardness for refinements only. More generally, it is hard for every alternation of these two belief change operators. \square

Since severe antiwithdrawal turns an empty total preorder into an empty total preorder, a sequence comprising only this operator has low complexity: entailment is equal to validity, coNP complete.

Sequences of mixed belief change operators are now considered. Two classes can be shown to be Δ_2^P -hard: operators that can produce a lexicographically maximal model and operators that can refine the lowest class of an ordering. In both cases, the alternation of operators does not matter, as long as they have the given behavior.

5.1 Lexicographic-finding Revisions

Entailment from a sequence of lexicographic revisions is Δ_2^P -hard by Theorem 13. Some other belief change operators can be intermixed without changing complexity. These are the ones that produce the same results when applied after a sequence of revisions whose formulae are consistent.

THEOREM 14. *If rev is any of a class of revision operators such that $\emptyset \text{rev}(S_1) \dots \text{rev}(S_n) \equiv \emptyset \text{lex}(S_1) \dots \text{lex}(S_n)$ whenever $S_1 \wedge \dots \wedge S_n$ is consistent, then entailment for rev is Δ_2^P -hard.*

PROOF. Checking whether the lexicographically maximal model of a formula F satisfies x_n is Δ_2^P -hard [51]. This model is also the only element of class zero of $\emptyset \text{lex}(x_n) \dots \text{lex}(x_1) \text{lex}(F)$. Since x_1, \dots, x_n is consistent, $\emptyset \text{lex}(x_n) \dots \text{lex}(x_1)$ is equivalent to $\emptyset \text{rev}(x_n) \dots \text{rev}(x_1)$ by assumption. Therefore, entailment from these two sequences is the same. \square

Moderate severe revision satisfies the premise of this theorem: it coincides with lexicographic revision on all consistent sequences.

THEOREM 15. *If $S_1 \wedge \dots \wedge S_n$ is consistent, then*

$$\emptyset \text{msev}(S_1) \dots \text{msev}(S_n) = \emptyset \text{lex}(S_1) \dots \text{lex}(S_n).$$

PROOF. It is inductively proved that $\emptyset \text{msev}(S_1) \dots \text{msev}(S_n) = \emptyset \text{lex}(S_1) \dots \text{lex}(S_n)$ and that $\emptyset \text{msev}(S_1) \dots \text{msev}(S_n)(0) = \text{Mod}(S_1 \wedge \dots \wedge S_n)$.

The base case is with $n = 0$, where the claim holds, because $\emptyset = \emptyset$ and the conjunction of an empty sequence is \top .

Assuming that $\emptyset \text{msev}(S_1) \dots \text{msev}(S_{n-1}) = \emptyset \text{lex}(S_1) \dots \text{lex}(S_{n-1})$, and that $\emptyset \text{msev}(S_1) \dots \text{msev}(S_{n-1})(0) = \text{Mod}(S_1 \wedge \dots \wedge S_{n-1})$, the same are proved with the addition of S_n .

Let $C = [C(0), \dots, C(m)]$ be $\emptyset \text{msev}(S_1) \dots \text{msev}(S_{n-1})$, and $M = S_1 \wedge \dots \wedge S_{n-1}$. By the induction assumptions, $C = \emptyset \text{lex}(S_1) \dots \text{lex}(S_{n-1})$, and $C(0) = \text{Mod}(M)$.

The definition of $C \text{msev}(S_n)$ depends on $\min(C, S_n) = C(i) \cap \text{Mod}(S_n)$. Since $C(0) = \text{Mod}(M)$ and $M \wedge S_n$ is by assumption consistent, $i = 0$. The definition of moderate severe revision specializes

to $i = 0$ as

$$\begin{aligned}
& [C(0), \dots, C(m)]\text{msev}(S_n) \\
&= [C(0) \cap \text{Mod}(S_n), \dots, C(m) \cap \text{Mod}(S_n), \\
&\quad (C(0) \cup \dots \cup C(0)) \setminus \text{Mod}(S_n), C(1) \setminus \text{Mod}(S_n), \dots, C(m) \setminus \text{Mod}(S_n)] \\
&= [C(0) \cap \text{Mod}(S_n), \dots, C(m) \cap \text{Mod}(S_n), \\
&\quad C(0) \setminus \text{Mod}(S_n), C(1) \setminus \text{Mod}(S_n), \dots, C(m) \setminus \text{Mod}(S_n)] \\
&= [C(0), \dots, C(m)]\text{lex}(S_n).
\end{aligned}$$

The total preorder $[C(0), \dots, C(m)]\text{lex}(S_n)$ is $\emptyset\text{lex}(S_1) \dots \text{lex}(S_{n-1})\text{lex}(S_n)$ because of the induction assumption $C = \emptyset\text{lex}(S_1) \dots \text{lex}(S_{n-1})$. Since $C(0) = \text{Mod}(S_1 \wedge \dots \wedge S_{n-1})$, it follows that the class zero of this ordering is $C\text{msev}(S_n)(0) = C(0) \cap \text{Mod}(S_n) = \text{Mod}(S_1 \wedge \dots \wedge S_{n-1} \wedge S_n)$, which concludes the proof of the induction claim. \square

A consequence of the two theorems above is that entailment from a sequence of moderate severe revision is Δ_2^P -hard. The same holds even if lexicographic and moderate severe revisions are mixed.

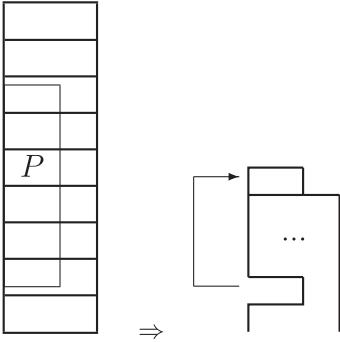
COROLLARY 2. *Entailment from a sequence of moderate severe revision is Δ_2^P -hard.*

5.2 Bottom-refining Revisions

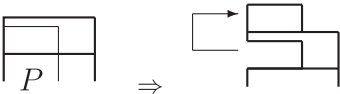
A revision operator is bottom-refining if it “refines” the lowest-index class of the ordering that has models of the revising formula.

Definition 11. An operator rev is a revision if $\min(\text{Crev}(P), \top) = \min(C, P)$ and is a bottom-revising revision if $\text{Crev}(P)(1) = C(0) \setminus \text{Mod}(P)$ also holds whenever $C(0) \cap \text{Mod}(P) \neq \emptyset$.

Removing empty classes, the minimal models of \top according to $\text{Crev}(P)$ are class zero of $\text{Crev}(P)$. The minimal models of P according to C are the non-empty intersection $C(i) \cap \text{Mod}(P)$ of minimal i . Equality between these two classes means that a revision operator makes this intersection the new class zero. No constraint is imposed on the other classes.



The new class zero is $C(0) \cap \text{Mod}(P)$ if this intersection is not empty. Bottom-refining revisions add an additional constraint in this case: revising C by P splits $C(0)$ based on P .



The name bottom-refining derives from the way the “bottom” class $C(0)$ is partitioned (refined) into the part satisfying P and the part not satisfy P . How the other classes are changed is not constrained.

THEOREM 16. *Lexicographic revision, natural revision, restrained revision, plain severe revision, severe revision, moderate severe revision, very radical revision and full meet revision are revision operators.*

PROOF. All these operators make the new class zero out of the first non-empty intersection $C(i) \cap \text{Mod}(P)$. \square

THEOREM 17. *Natural revision, restrained revision and severe revision are bottom-refining revisions.*

PROOF. All three operators are revisions, as proved by the previous theorem.

The condition of bottom-refining only concerns the case where $C(0) \cap \text{Mod}(P) \neq \emptyset$, where this index i is 0. The definition of natural revisions specializes as follows:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{nat}}(P) \\ &= [C(i) \cap \text{Mod}(M), C(0), \dots, C(i-1), C(i) \setminus \text{Mod}(P), C(i+1), \dots, C(m)] \\ &= [C(0) \cap \text{Mod}(M), C(0) \setminus \text{Mod}(P), C(1), \dots, C(m)]. \end{aligned}$$

The definition of restrained revisions specializes as follows:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{res}}(P) \\ &= [C(i) \cap \text{Mod}(P), \\ & \quad C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), \dots, C(i-1) \cap \text{Mod}(P), C(i-1) \setminus \text{Mod}(P), \\ & \quad C(i) \setminus \text{Mod}(P), \\ & \quad C(i+1) \cap \text{Mod}(P), C(i+1) \setminus \text{Mod}(P), \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)] \\ &= [C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), C(1) \cap \text{Mod}(P), C(1) \setminus \text{Mod}(P), \\ & \quad \dots, C(m) \cap \text{Mod}(P), C(m) \setminus \text{Mod}(P)]. \end{aligned}$$

The definition of severe revisions specializes as follows:

$$\begin{aligned} & [C(0), \dots, C(m)]_{\text{sevr}}(P) \\ &= [C(i) \cap \text{Mod}(P), (C(0) \cup \dots \cup C(i)) \setminus \text{Mod}(P), C(i+1), \dots, C(m)] \\ &= [C(0) \cap \text{Mod}(P), C(0) \setminus \text{Mod}(P), C(1), \dots, C(m)]. \end{aligned}$$

All three revisions makes $C(0) \cap \text{Mod}(P)$ the new class zero and $C(0) \setminus \text{Mod}(P)$ the new class one. \square

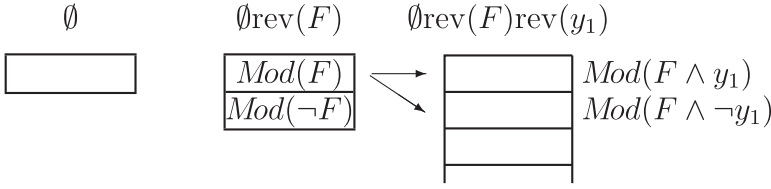
The complexity of arbitrary sequences of bottom-refining revisions is established by the following theorem.

THEOREM 18. *Inference from a sequence of bottom-refining revisions is Δ_2^P -hard.*

PROOF. The claim is proved by reduction from the problem of establishing whether the lexicographically maximal model of a consistent formula F over alphabet $\{x_1, \dots, x_n\}$ satisfies x_n . The corresponding sequence of bottom refining revisions rev is the following, where y_1, \dots, y_n are fresh variables in bijective correspondence with x_1, \dots, x_n :

$$\emptyset \text{rev}(F) \text{rev}(y_1) \text{rev}(y_1 \rightarrow x_1) \dots \text{rev}(y_n) \text{rev}(y_n \rightarrow x_n).$$

The reduction first introduces the models of F as the class zero, then refines it by x_1 if consistent, then by x_2 if consistent, and so on. A single bottom-refining revision for each variable cannot do that: if x_1 is inconsistent with F the effect of $F \text{rev}(x_1)$ is to move the models of x_1 in a class lower than that of F . This is why the new variable y_1 is introduced first.



The empty ordering has all models in class zero. A revision operator cuts $Mod(F)$ out from it to make the new class zero when revising by a consistent formula F . The class $Mod(\neg F)$ is created only if F is not tautological, but this is irrelevant.

The new class zero $Mod(F)$ is refined into $Mod(F \wedge y_1)$ and $Mod(F \wedge \neg y_1)$ because of the bottom-refining condition: Since F is consistent and does not contain y_1 , the conjunctions $F \wedge y_1$ and $F \wedge \neg y_1$ are consistent.

Revising this ordering by $y_1 \rightarrow x_1$ depends on the consistency of $F \wedge x_1$. If F is consistent with x_1 , then $F \wedge y_1 \wedge (y_1 \rightarrow x_1)$ is consistent; therefore, class zero $Mod(F \wedge y_1)$ contains some models of $y_1 \rightarrow x_1$. The resulting class zero comprises them: $Mod(F \wedge y_1 \wedge (y_1 \rightarrow x_1)) = Mod(F \wedge x_1 \wedge y_1)$. If F is inconsistent with x_1 , then $F \wedge y_1 \wedge (y_1 \rightarrow x_1)$ is inconsistent; therefore, $Mod(F \wedge y_1)$ does not contain any model of $y_1 \rightarrow x_1$. Instead, $Mod(F \wedge \neg y_1)$ does, since $Mod(F \wedge \neg y_1 \wedge (y_1 \rightarrow x_1))$ is the same as $Mod(F \wedge \neg y_1)$, which is consistent, because F is consistent and does not mention y_1 . The class zero resulting from revising by $y_1 \rightarrow x_1$ is $Mod(F \wedge \neg y_1 \wedge (y_1 \rightarrow x_1)) = Mod(F \wedge \neg y_1)$, since rev is a revision.

This proves that the result of revising first by y_1 and then by $y_1 \rightarrow x_1$ is an ordering that has $Mod(F \wedge x_1 \wedge y_1)$ as its class zero if $F \wedge x_1$ is consistent and $Mod(F \wedge \neg y_1)$ otherwise. Apart from y_1 , which is unlinked to the rest of the formula and the other variables y_i , these are the models of $F \wedge x_1$ if consistent and the models of F otherwise.

Iterating the procedure on the remaining variables x_2, \dots, x_n produces an ordering whose class zero comprises the lexicographically maximally model of F only. Checking whether it entails x_n is the final step of the translation.

This intuition is made a formal proof by induction. For every i , class zero of $\emptyset rev(F) rev(y_1) rev(y_1 \rightarrow x_1) \dots rev(y_i) rev(y_i \rightarrow x_i)$ is $Mod(\bigwedge Y' \wedge \maxcon(F, x_1, \dots, x_i))$ for some consistent $Y' \subseteq \{y_1, \neg y_1, \dots, y_i, \neg y_i\}$. This is the lexicographically maximal partial model over variables x_1, \dots, x_i , apart from some of the variables y_1, \dots, y_i . Assuming that this condition is true for i , it is shown to remain true after revising by y_{i+1} and $y_{i+1} \rightarrow x_{i+1}$.

Let $C = \emptyset rev(F) rev(y_1) rev(y_1 \rightarrow x_1) \dots rev(y_i) rev(y_i \rightarrow x_i)$ and $M = \bigwedge Y' \wedge \maxcon(F, x_1, \dots, x_i)$. The inductive assumption is $C(0) = Mod(M)$. Since M does not mention y_{i+1} and is consistent, $M \wedge y_{i+1}$ is consistent. As a result, $C(0) \cap Mod(y_{i+1})$ is not empty. A bottom-refining revision splits the class zero in two:

$$\begin{aligned} Crev(y_{i+1})(0) &= Mod(M \wedge y_{i+1}), \\ Crev(y_{i+1})(1) &= Mod(M \wedge \neg y_{i+1}). \end{aligned}$$

This ordering is further revised by $y_{i+1} \rightarrow x_{i+1}$. The resulting class zero of $Crev(y_{i+1}) rev(y_{i+1} \rightarrow x_{i+1})$ is the first non-empty of the following two sets, since rev is by assumption a revision operator and at least the second is not empty, since M is consistent and does not contain y_{i+1} :

$$\begin{aligned} Crev(y_{i+1})(0) \cap Mod(y_{i+1} \rightarrow x_{i+1}) &= Mod(M \wedge y_{i+1} \wedge (y_{i+1} \rightarrow x_{i+1})) \\ &= Mod(M \wedge y_{i+1} \wedge x_{i+1}), \\ Crev(y_{i+1})(1) \cap Mod(y_{i+1} \rightarrow x_{i+1}) &= Mod(M \wedge \neg y_{i+1} \wedge (y_{i+1} \rightarrow x_{i+1})) \\ &= Mod(M \wedge \neg y_{i+1}). \end{aligned}$$

Since M is $\bigwedge Y' \wedge \text{maxcon}(F, x_1, \dots, x_i)$ and F is a formula over variables $\{x_1, \dots, x_n\}$ only, $M \wedge y_{i+1} \wedge x_{i+1}$ is consistent if and only if $\text{maxcon}(F, x_1, \dots, x_i) \wedge x_{i+1}$ is consistent. Depending on this condition:

maxcon(F, x_1, \dots, x_i) $\wedge x_{i+1}$ **is consistent:** $M \wedge y_{i+1} \wedge x_{i+1}$ is also consistent; therefore, the models of this formula are the new class zero; $\bigwedge Y' \wedge \text{maxcon}(F, x_1, \dots, x_i) \wedge y_{i+1} \wedge x_{i+1}$ is the same as $\bigwedge Y'' \wedge \text{maxcon}(F, x_1, \dots, x_{i+1})$ where $Y'' = Y' \cup \{y_{i+1}\}$, since by assumption $\text{maxcon}(F, x_1, \dots, x_i) \wedge x_{i+1}$ is consistent;

maxcon(F, x_1, \dots, x_i) $\wedge x_{i+1}$ **is inconsistent:** Since $M \wedge y_{i+1} \wedge x_{i+1}$ is inconsistent, the first non-empty of the two sets above is $\text{Mod}(M \wedge \neg y_{i+1})$; replacing M with its definition, $M \wedge y_{i+1}$ becomes $\bigwedge Y' \wedge \text{maxcon}(F, x_1, \dots, x_i) \wedge \neg y_{i+1}$ and this is equal to $\bigwedge Y'' \wedge \text{maxcon}(F, x_1, \dots, x_i, x_{i+1})$ where $Y'' = Y' \cup \{\neg y_i\}$. Indeed, $\text{maxcon}(F, x_1, \dots, x_i, x_{i+1}) = \text{maxcon}(F, x_1, \dots, x_i)$, since by assumption $\text{maxcon}(F, x_1, \dots, x_i) \wedge x_{i+1}$ is inconsistent. \square

5.3 Very Radical Revision

Very radical revision is neither lexicographic-finding nor bottom-refining. It is indeed easier, as the classes of $\emptyset\text{rad}(R_1) \dots \text{rad}(R_n)$ are relatively easy to determine.

THEOREM 19. *For every formulae R_1, \dots, R_n , the total preorder $\emptyset\text{rad}(R_1) \dots \text{rad}(R_n)$ is equivalent to the following preorder C :*

$$\begin{aligned} C(0) &= \text{Mod}(\neg \perp \wedge R_1 \wedge R_2 \wedge R_3 \wedge \dots \wedge R_n), \\ C(1) &= \text{Mod}(\neg R_1 \wedge R_2 \wedge R_3 \wedge \dots \wedge R_n), \\ C(2) &= \text{Mod}(\neg R_2 \wedge R_3 \wedge \dots \wedge R_n), \\ &\vdots \\ C(n-1) &= \text{Mod}(\neg R_{n-1} \wedge R_n), \\ C(n) &= \text{Mod}(\neg R_n). \end{aligned}$$

PROOF. The proof is by induction on the length of the sequence. For $n = 1$, the total preorder $C = \emptyset\text{rad}(R_1)$ splits the single class $\emptyset(0) = \text{Mod}(\top)$ into the two classes $C(0) = \text{Mod}(\top) \cap \text{Mod}(R_1) = \text{Mod}(\neg \perp \wedge R_1)$ and $C(1) = \text{Mod}(\top) \setminus \text{Mod}(R_1) = \text{Mod}(\neg R_1)$. The claim therefore holds.

Assuming that the claim holds for the preorder $C = [C(0), \dots, C(n-1)] = \emptyset\text{rad}(R_1) \dots \text{rad}(R_{n-1})$, it is proved for $\text{Crad}(R_n)$. From the definition of rad:

$$\begin{aligned} &[C(0), \dots, C(m)]\text{rad}(R_n) \\ &= [C(0) \cap \text{Mod}(R_n), \dots, C(n-1) \cap \text{Mod}(R_n), (C(0) \cup \dots \cup C(n-1)) \setminus \text{Mod}(R_n)] \\ &= [\text{Mod}(\neg \perp \wedge R_1 \wedge \dots \wedge R_{n-1}) \cap \text{Mod}(R_n), \\ &\quad \dots, \text{Mod}(\neg R_{n-1}) \cap \text{Mod}(R_n), \text{Mod}(\top) \setminus \text{Mod}(R_n)] \\ &= [\text{Mod}(\neg \perp \wedge R_1 \wedge \dots \wedge R_{n-1} \wedge R_n), \dots, \text{Mod}(\neg R_{n-1} \wedge R_n), \text{Mod}(\neg R_n)]. \end{aligned}$$

The second equality holds, because by definition C includes all models; therefore, $C(0) \cup \dots \cup C(n) = \text{Mod}(\top)$. The third holds, because $\text{Mod}(\top) \setminus \text{Mod}(R_n)$ is the set of all models but the ones of R_n . \square

This theorem tells how to determine $\min(\emptyset\text{rad}(R_1) \dots \text{rad}(R_n), \top)$: by conjoining R_n with R_{n-1} , then with R_{n-2} and so on until consistent.

Definition 12. The longest consistent conjunction of a sequence of formulae $\text{longest}(L_1, \dots, L_n)$ is $L_1 \wedge \dots \wedge L_i$ such that either $i = n$ or $L_1 \wedge \dots \wedge L_i \wedge L_{i+1}$ is inconsistent.

A longest sequence is a simplified form of maxcon: the maxcons conjoin formulae in order skipping every one that would create an inconsistency; the longest sequences stop altogether at the first. A sequence of very radical revisions from the empty ordering can be reformulated in terms of this definition.

THEOREM 20. *For every formulae R_1, \dots, R_n , formula $\text{Form}(\min(\emptyset\text{rad}(R_1) \dots \text{rad}(R_n), \top))$ is equivalent to $\text{longest}(R_n, \dots, R_1)$.*

PROOF. The previous theorem shows that the classes of $\emptyset\text{rad}(R_1) \dots \text{rad}(R_n)$ are the models of the following formulae:

$$\begin{aligned} & \neg \perp \wedge R_1 \wedge R_2 \wedge R_3 \wedge \dots \wedge R_n, \\ & \neg R_1 \wedge R_2 \wedge R_3 \wedge \dots \wedge R_n, \\ & \neg R_2 \wedge R_3 \wedge \dots \wedge R_n, \\ & \vdots \\ & \neg R_{n-1} \wedge R_n, \\ & \neg R_n. \end{aligned}$$

As a result, $\min(\emptyset\text{rad}(R_1) \dots \text{rad}(R_n), \top)$ is the set of models of the first consistent formula in the list. This formula may be the first or any of the others. The first is $\neg \perp \wedge R_1 \wedge R_2 \wedge R_3 \wedge \dots \wedge R_n$, which is equivalent to $R_1 \wedge \dots \wedge R_n$. If it is consistent, then $R_n \wedge \dots \wedge R_i$ is consistent for $i = 1$, and is therefore the same as $\text{longest}(R_n, \dots, R_1)$.

The other case is that the first consistent formula of the list is $\neg R_{i-1} \wedge R_i \wedge \dots \wedge R_n$ for some index i . Since it is consistent, its subformula $R_i \wedge \dots \wedge R_n$ is consistent too. This is the first part of the definition of the longest consistent conjunction, the second being the inconsistency of $R_{i-1} \wedge R_i \wedge \dots \wedge R_n$.

To the contrary, let M be a model of $R_{i-1} \wedge R_i \wedge \dots \wedge R_n$. If M satisfies all formulae R_1, \dots, R_{i-2} , then $R_1 \wedge \dots \wedge R_n \wedge \neg \perp$ is consistent, contrary to assumption. Therefore, M falsifies some formula among R_1, \dots, R_{i-2} . Let $j \leq i - 2$ be the highest index such that M falsifies R_j . Since M falsifies this formula, it satisfies its negation $\neg R_j$. Because of the highest index, M satisfies all formulae R_{j+1}, \dots, R_{i-2} if any. As a result, M satisfies $\neg R_j \wedge R_{j+1} \wedge \dots \wedge R_{i-2}$. Since it also satisfies $R_{i-1} \wedge R_i \wedge \dots \wedge R_n$, it satisfies $\neg R_j \wedge R_{j+1} \wedge \dots \wedge R_1$. The consistency of this sequence with $j \leq i - 2$ contradicts the assumption that i is the lowest index such that $\neg R_{i-1} \wedge R_i \wedge \dots \wedge R_1$ is consistent.

This proves that $R_{i-1} \wedge R_i \wedge \dots \wedge R_n$ is inconsistent. Since $R_i \wedge \dots \wedge R_n$ is consistent, this is $\text{longest}(R_n, \dots, R_1)$. \square

By this theorem, the complexity of $\text{longest}(L_1, \dots, L_n) \models Q$ is the same as inference from a sequence of very radical revisions from an empty sequence. This problem is investigated under the condition that each formula L_i is consistent.

THEOREM 21. *For every sequence of consistent formulae L_1, \dots, L_n and formula Q , checking whether $\text{longest}(L_1, \dots, L_n) \models Q$ is $\Delta_2^P[\log n]$ -complete, and BH_{2n-1} -complete if n is a constant.*

PROOF. The complexity classes $\Delta_2^P[\log n]$ has a definition similar to that of Δ_2^P : they both contain all problems that can be solved by an algorithm that runs in polynomial time, not counting the time spent in solving a certain number satisfiability test. This number is bounded by a polynomial in the size of the input for Δ_2^P , and by a logarithm in the size of the input for $\Delta_2^P[\log n]$.

The complexity class BH_{2n-1} is defined by identifying a decision problem (such as whether a sequence of iterated revisions entails a formula) with the set of its input instances whose answer is

“yes.” The class BH_{2n-1} is the union of a problem in BH_{2n} and a problem in NP. In turns, BH_{2n} is the intersection of a problem in BH_{2n-1} and a problem in coNP. The base of this recursive definition is that BH_0 is NP.

Entailment $\text{longest}(L_1, \dots, L_n) \models Q$ holds if $L_1 \wedge \dots \wedge L_i$ is consistent and entails Q for some $i \in \{1, \dots, n\}$. The check for the inconsistency of $L_1 \wedge \dots \wedge L_i \wedge L_{i+1}$ is not necessary: if it does not hold, then L_{i+1} is added to the conjunction $L_1 \wedge \dots \wedge L_i \models Q$, and the result still entails Q .

These consistency and entailment tests can be done in parallel; if they succeed for the same index i , then $\text{longest}(L_1, \dots, L_n)$ entails Q . The problem is therefore in $\Delta_2^P[\log n]$. If n is a constant, then an exact computation of the tests to be performed is needed; these are

- (1) $\models Q$; or
 - (2) L_1 is consistent and $L_1 \models Q$; or
 - (3) $L_1 \wedge L_2$ is consistent and $L_1 \wedge L_2 \models Q$; or
- ⋮

$n + 1$ $L_1 \wedge \dots \wedge L_n$ is consistent and entails Q .

Under the assumption that every single L_i is consistent, the first two ones can be simplified. Indeed, the second one “ L_1 is consistent and $L_1 \models Q$ ” is the same as $L_1 \models Q$. This condition is entailed by $\models Q$, which becomes unnecessary. The conditions can therefore be rewritten as

- (1) $L_1 \models Q$; or
 - (2) $L_1 \wedge L_2$ is consistent and $L_1 \wedge L_2 \models Q$; or
- ⋮

n $L_1 \wedge \dots \wedge L_n$ is consistent and entails Q .

The first test is in coNP, the other $n - 1$ ones are in D^P . By the definition of the Boolean hierarchy [79], the problem is in BH_{2n-1} .

Hardness for $\Delta_2^P[\log n]$ and BH_{2n-1} is proved for unbounded and constant n by a reduction from the following problem: given F_1, \dots, F_{2n-1} with F_i consistent implying F_{i-1} consistent, check whether the number of consistent F_i is even [24]. This problem is reduced to checking $\text{longest}(L_1, \dots, L_n) \models Q$. This implies that entailment from the longest sequence is both $\Delta_2^P[\log n]$ -hard in the general case and BH_{2n-1} -hard if n is constant.

The first step of the reduction is rewriting of each formula F_i on a private alphabet. This is assumed already done, and does not change consistency.

New variables are introduced, one for each formula: $\{y_1, \dots, y_m\}$. The query Q is $\neg y_1 \vee \neg y_3 \vee \dots$, the disjunction of all literals $\neg y_i$ with i odd.

The first formula L_1 is $y_1 \rightarrow F_1$, which is consistent, because it is satisfied by setting y_1 to false. If F_1 is inconsistent, then $\neg y_1$ is entailed. This is correct, since the number of consistent formulae is zero, which is even.

$$\begin{array}{l}
 \boxed{} \\
 y_1 \rightarrow F_1 \\
 \boxed{} \\
 \\
 \boxed{}
 \end{array}
 \begin{array}{l}
 L_1 \\
 \\
 L_2
 \end{array}$$

The second formula L_2 contains y_1 and an unrelated variable. The conjunction of the first two formulae contains $(y_1 \rightarrow F_1) \wedge y_1 \equiv F_1 \wedge y_1$. If F_1 is consistent, then this formula does not entail $\neg y_1$ and is consistent. The construction of the longest consistent sequence continues.

Overall, if F_1 is inconsistent, then $\neg y_1$ is entailed. Otherwise, $\neg y_1$ is not entailed and the construction of the longest consistent sequence continues.

The rest of the sequence works similarly: the construction stops at the first inconsistent formula F_i ; if i is odd, then $\neg y_i$ is entailed, which is correct, since the number of consistent formulae is $i - 1$, even.

The other formulae are as follows, where i is odd.

$$\begin{array}{r}
 \boxed{z_{j-1}} \quad L_{i-1} \\
 \hline
 \boxed{z_{j-1} \rightarrow F_{j-1}} \\
 \boxed{y_j \rightarrow F_j} \quad L_i \\
 \hline
 \boxed{y_j} \quad L_{i+1}
 \end{array}$$

Assuming that the construction of the longest consistent sequence includes L_{i-1} , it proceeds as follows.

Three cases are possible: F_{i-1} is inconsistent; it is consistent but F_i is not; they are both consistent.

If F_{i-1} is inconsistent, then L_i is not added to the sequence, because $L_{i-1} \wedge L_i$ contains $y_{i-1} \wedge (y_{i-1} \rightarrow F_i)$, which implies the inconsistent formula F_{i-1} . Therefore, the longest consistent sequence does not contain y_i , and it therefore does not entail $\neg y_i$. This is correct, since the number of consistent formulae is $i - 2$, odd.

If F_{i-1} is consistent, then L_i is added to the sequence, because the final part of its conjunction is $y_{i-1} \wedge (y_{i-1} \rightarrow F_{i-1}) \wedge (y_i \rightarrow F_i)$, which is equivalent to $y_{i-1} \wedge F_{i-1} \wedge (y_i \rightarrow F_i)$ and is therefore consistent. If F_i is inconsistent, then this formula entails $\neg y_i$, which is correct, because the number of consistent formulae is $i - 1$, even.

The final case is that F_i is consistent. Not only $y_{i-1} \wedge F_{i-1} \wedge (y_i \rightarrow F_i)$ is consistent but is also consistent with L_{i+1} , which is y_i plus an unrelated variable. Therefore, the construction of the longest consistent sequence continues.

Technically, the formulae L_i, L_{i+1} and Q are as follows, where i is odd:

$$\begin{aligned}
 L_1 &= y_1 \rightarrow F_1, \\
 L_2 &= y_1 \wedge y_2, \\
 &\vdots \\
 L_i &= (y_{i-1} \rightarrow F_{i-1}) \wedge (y_i \rightarrow F_i), \\
 L_{i+1} &= y_i \wedge y_{i+1}, \\
 Q &= \neg y_1 \vee \neg y_3 \vee \dots .
 \end{aligned}$$

Each formula is consistent by itself: the formulae of odd index are satisfied by the model that assigns false to all variables, the formulae of even index by that assigning true to all variables.

The claim is that $\text{longest}(L_1, \dots, L_n)$ entails Q if and only if the number of consistent formulae F_i is even.

The entailment $\text{longest}(L_1, \dots, L_m) \models Q$ simplifies, because $\text{longest}(L_1, \dots, L_m)$ is a conjunction of formulae L_i , where the variables y_i occur in separate subformulae. This means that $\text{longest}(L_1, \dots, L_m)$ entails Q if and only if it entails some variables $\neg y_i$ with i odd.

The number of consistent formulae F_i is even is the same as the consistency of $F_1 \wedge \dots \wedge F_{i-1}$ and the inconsistency of $F_1 \wedge \dots \wedge F_{i-1} \wedge F_i$ with i odd because of the separation of the variables and the consistency of all formulae preceding a consistent one.

The conjunction $L_1 \wedge \dots \wedge L_i$ with i odd is $y_1 \wedge (y_1 \rightarrow F_1) \wedge \dots \wedge y_{i-1} \wedge (y_{i-1} \rightarrow F_{i-1}) \wedge (y_i \rightarrow F_i)$. This formula is equivalent to $y_1 \wedge F_1 \wedge \dots \wedge y_{i-1} \wedge F_{i-1} \wedge (y_i \rightarrow F_i)$.

The conjunction with i even is obtained by taking $L_1 \wedge \dots \wedge L_{i+1}$ in the definition above and decreasing i by 1. It is $y_1 \wedge (y_1 \rightarrow F_1) \wedge \dots \wedge y_{i-1} \wedge (y_{i-1} \rightarrow F_{i-1}) \wedge y_i$, which is equivalent to $y_1 \wedge F_1 \wedge \dots \wedge y_{i-1} \wedge F_{i-1} \wedge y_i$.

The claim can now be proved in each direction.

If $F_1 \wedge \dots \wedge F_{i-1}$ is consistent and $F_1 \wedge \dots \wedge F_{i-1} \wedge F_i$ is not with i odd, then $L_1 \wedge \dots \wedge L_i = y_1 \wedge F_1 \wedge \dots \wedge y_{i-1} \wedge F_{i-1} \wedge (y_i \rightarrow F_i)$ is consistent and entails $\neg y_i$. The longest consistent conjunction contains it and therefore entails $\neg y_i$ as well.

To prove the other direction, the longest consistent conjunction $L_1 \wedge \dots \wedge L_i$ is assumed to entail a literal $\neg y_j$. Since this conjunction is consistent, it does not contain y_j . Since it contains all variables y_j with $j < i$ regardless of whether i is even or odd, the only possible j is i . If i is even, then $L_1 \wedge \dots \wedge L_i$ contains y_i . Therefore, i is odd. \square

Since a sequence of very radical revisions from the empty ordering is exactly the same as the longest consistent conjunction, and all formulae are consistent by assumption, the problem of entailment for very radical revision is $\Delta_2^P[\log n]$ -complete in the general case and BH_{2n-1} -complete for constant n .

5.4 Plain Severe Revision and Full Meet Revision

On total preorders comprising at most two classes, plain severe and full meet revision coincide, and always generate an ordering of at most two classes. As a result, when the initial ordering is empty, sequence of plain severe and full meet revision coincide:

$$\emptyset\text{psev}(P_1) \dots \text{psev}(P_n) \equiv \emptyset\text{full}(P_1) \dots \text{full}(P_n).$$

More generally, mixed sequences of plain severe and full meet revisions applied to an ordering comprising at most two classes are equivalent to sequence of full meet revisions only and to sequences of plain severe revisions only.

These two revisions are neither lexicographic-finding nor bottom-refining. A lexicographic-finding sequence of revisions $x_1 \wedge x_3, \neg x_1, x_2$ produces $\neg x_1 \wedge x_2 \wedge x_3$, but the same sequence of full meet revisions instead produces to $\neg x_1 \wedge x_2$. A sequence of bottom-refining revisions x_1, x_2 produces an ordering with a class one equal to $x_1 \wedge \neg x_2$, but the same sequence of full meet revisions instead gives $\neg x_1 \vee \neg x_2$. They are also different from very radical revision, as seen from the sequence of revisions $x_1, x_2, \neg x_1$, where very radical revision produces $x_2 \wedge \neg x_1$ while full meet produces $\neg x_1$.

THEOREM 22. *Inference from a sequence of full meet and plain severe revisions applied to the empty ordering is $\Delta_2^P[\log n]$ -complete.*

PROOF. The $\Delta_2^P[\log n]$ class includes all problems that can be solved by a polynomial number of nonadaptive calls to an NP-oracle. Nonadaptive means that no call depends on the others. Equivalently, these calls are in parallel [13, 39].

A sequence of full meet revisions $\emptyset \text{full}(S_1) \dots \text{full}(S_n)$ requires establishing the satisfiability of the following quadratic number of formulae:

$$\begin{aligned}
& S_1, \\
& S_1 \wedge S_2, \\
& \vdots \\
& S_1 \wedge S_2 \wedge \dots \wedge S_n, \\
& S_2, \\
& S_2 \wedge S_3, \\
& \vdots \\
& S_2 \wedge S_3 \wedge \dots \wedge S_n, \\
& \vdots \\
& S_n.
\end{aligned}$$

The first group of formulae starts with S_1 and adds a formula at time until the last. The second starts with S_2 and does the same. This is repeated for all subsequent formulae S_3, \dots, S_n . All these conjunctions are checked for satisfiability regardless of the satisfiability of the others.

Given the result of these tests, the result of full meet revision is calculated in polynomial time. First, the longest continuous conjunction $F_1 \wedge \dots \wedge F_i$ is determined. If $i = n$, then it is the final result. Otherwise, since $i < n$ then $i + 1$ is less than or equal to n . Therefore, F_{i+1} is a formula of the sequence. The longest continuous conjunction $F_{i+1} \wedge \dots \wedge F_j$ is again determined. If $j = n$, then it is the final result. Otherwise, the process continues with F_{j+1} . This is repeated until F_n is in the conjunction.

Hardness was announced for full meet revision in a previous article, but without proof [55]. The proof provided here is by reduction from the problem of deciding $\text{longest}(L_1, \dots, L_n) \models Q$. Given the consistent formulae L_1, \dots, L_n , the reduction builds the following sequence:

$$\begin{aligned}
F_n &= a_n \rightarrow (L_1 \wedge \dots \wedge L_n), \\
F_{n-1} &= a_n \wedge (a_{n-1} \rightarrow (L_1 \wedge \dots \wedge L_{n-1})), \\
F_{n-2} &= a_{n-1} \wedge (a_{n-2} \rightarrow (L_1 \wedge \dots \wedge L_{n-2})), \\
&\vdots \\
F_2 &= a_3 \wedge (a_2 \rightarrow (L_1 \wedge L_2)), \\
F_1 &= a_2 \wedge L_1.
\end{aligned}$$

The sequence of revisions F_n, \dots, F_1 applied to the empty order entails Q if and only if $\text{longest}(L_1, \dots, L_n)$ does. This is the case because an inconsistent conjunction $L_1 \wedge \dots \wedge L_i$ makes F_i inconsistent with the following formulae F_{i+1} , which therefore takes its place. Otherwise, they are conjoined and the process continues.

Technically, if $L_1 \wedge \dots \wedge L_i$ is the longest consistent conjunction, then $L_1 \wedge \dots \wedge L_{i+1}$ is inconsistent. Since F_{i+1} contains $a_{i+1} \rightarrow (L_1 \wedge \dots \wedge L_{i+1})$, it is inconsistent with a_{i+1} , which is contained in F_i . As a result, full meet revision produces F_i . The subsequent formulae F_{i-1}, \dots, F_1 are consistent with F_i . Indeed, $F_i \wedge F_{i+1}$ implies $L_1 \wedge \dots \wedge L_i$, which is consistent and entails all implications $a_j \rightarrow (L_1 \wedge \dots \wedge L_j)$ in the following formulae. What remains after their removal is only a number of positive literals a_j , which are therefore consistent.

This proves that the result of the sequence of revision is $F_i \wedge \dots \wedge F_1$, which is equivalent to the longest consistent conjunction $L_1 \wedge \dots \wedge L_i$ apart from some unrelated variables a_j . Entailment of Q is therefore the same. \square

5.5 Alternating Sequences

The core of this article is that revisions may not be all of the same kind. Yet, the previous theorems show the hardness of sequences of similar revisions: all lexicographic-finding, all bottom-refining, all very radical, all full meet revisions. Does complexity change when they are intermixed? It depends on the specific alternation. Bottom-revising revisions maintain their complexity even when alternating with other revisions.

THEOREM 23. *Inference from an alternating sequence of revisions and bottom-refining revisions is Δ_2^P -hard.*

PROOF. Reduction is from the problem of inference from the lexicographically maximal model of a consistent formula F over the alphabet x_1, \dots, x_n , where y_1, \dots, y_n are new variables:

$$0\text{rev}(F)\text{botr}(y_1 \rightarrow x_1)\text{rev}(y_1)\text{botr}(y_2 \rightarrow x_2)\text{rev}(y_2) \dots$$

The (generic) revision $\text{rev}(F)$ isolates the models of F in class zero. The other classes are not of interest. Since y_1 is a new variable, $y_1 \rightarrow x_1$ is always consistent with F . By definition of bottom-refining revisions, $\text{botr}(y_1 \rightarrow x_1)$ splits $\text{Mod}(F)$ into $\text{Mod}(F \wedge (y_1 \rightarrow x_1))$ and $\text{Mod}(F \wedge \neg(y_1 \rightarrow x_1))$. The following revision $\text{rev}(y_1)$ produces $\text{Mod}(F \wedge (y_1 \rightarrow x_1) \wedge y_1)$ if consistent, otherwise $\text{Mod}(F \wedge \neg(y_1 \rightarrow x_1) \wedge y_1)$. These two sets are, respectively, the same as $\text{Mod}(F \wedge x_1 \wedge y_1)$ and $\text{Mod}(F \wedge \neg x_1 \wedge y_1)$. The new variable y_1 can be disregarded, because it does not occur anywhere else in the sequence. The result of the first pair of revision is $F \wedge x_1$ if consistent and $F \wedge \neg x_1$ otherwise. The following pairs of revisions do the same: each conjoins the result of the previous revision with x_i if consistent and with $\neg x_i$ otherwise. The final result is the lexicographically maximal model of F . \square

Some generic revisions are computationally easier than bottom-refining ones. Yet, intermixing them does not lower complexity. The same holds for lexicographic and very radical revision.

THEOREM 24. *Inference from an alternating sequence of lexicographic and very radical revisions is Δ_2^P -hard.*

PROOF. Reduction is from the problem of inference from the lexicographically maximal model of a consistent formula F over the alphabet x_1, \dots, x_n , where y_1, \dots, y_n are new variables:

$$0\text{lex}(x_n)\text{rad}(y_n) \dots \text{lex}(x_1)\text{rad}(y_1)\text{lex}(F).$$

The first revision splits $\text{Mod}(\top)$ into the classes $\text{Mod}(x_n)$ and $\text{Mod}(\neg x_n)$. The second makes three classes of out these two: $\text{Mod}(x_n \wedge y_n)$, $\text{Mod}(\neg x_n \wedge y_n)$, and $\text{Mod}(\neg y_n)$.

The second lexicographic revision duplicates the sequence: the first replica has x_{n-1} added to each of the three classes, the second has $\neg x_{n-1}$. The second very radical revision conjoins all six with y_{n-1} and adds a new class $\text{Mod}(\neg y_{n-1})$ at the end.

All models that satisfy x_{n-1} are all in lower classes than the others. Two that evaluate x_{n-1} the same are sorted according to x_n . At the end of the sequence of revisions, models are sorted according to the lexicographic order, disregarding the new variables y_1, \dots, y_n .

The final revision $\text{lex}(F)$ selects the minimal models of F according to this order. Since y_1, \dots, y_n are new variables, their value does not affect the satisfiability of F . The final result is the lexicographically minimal model of F . \square

The above two theorems show two cases where complexity does not decrease when intermixing hard and easy revisions. This is however not a general phenomenon. A counterexample is the

alternation of lexicographic and full meet revisions. It is easier than lexicographic revisions alone: $\Delta_2^P[\log n]$ -complete. This result is based on rewriting such alternations as a sequences of full meet revisions only.

LEMMA 2. *If O is a two-class ordering, then $O_{\text{lex}}(L)\text{full}(F)$ is equal to $O_{\text{full}}(L \wedge F)$ if $L \wedge F$ is consistent and to $O_{\text{full}}(F)$ otherwise.*

PROOF. Let S be a formula such that $\text{Mod}(S) = O(0)$. Since $O_{\text{lex}}(L)\text{full}(F)$ is a result of full meet revision, it always comprises two classes only, and is therefore identified by its class zero. This class comprises the models of the first consistent formula of the following list:

- $S \wedge L \wedge F$
- $L \wedge F$
- $S \wedge F$
- F

Formula F is assumed consistent. The first three formulae of the list may be all inconsistent, but the last is not.

The claim is proved by showing what happens in the four cases. The first revision applied to O is $\text{lex}(L)$. The classes of the resulting ordering comprise the models of the following formulae; each is empty if the corresponding formula is inconsistent:

- $S \wedge L$
- $\neg S \wedge L$
- $S \wedge \neg L$
- $\neg S \wedge \neg L$

The second revision is $\text{full}(F)$. It generates an ordering whose class zero comprises the models of the first consistent conjunction of F and a formula of the above list. Which one is determined by cases.

$S \wedge L \wedge F$ is consistent. The first formula conjunction $S \wedge L \wedge F$ is consistent. The resulting class zero comprises its models.

$S \wedge L \wedge F$ is inconsistent, $L \wedge F$ is consistent. The first conjunction $S \wedge L \wedge F$ is inconsistent. The second conjunction is $\neg S \wedge L \wedge F$. The inconsistency of $S \wedge L \wedge F$ implies $L \wedge F \models \neg S$. Therefore, $\neg S \wedge L \wedge F$ is equivalent to $L \wedge F$. This formula is consistent by assumption. Therefore, the first class after revising is its set of models.

$S \wedge L \wedge F$ and $L \wedge F$ are inconsistent, $S \wedge L$ consistent. The conjunction of F with the first two formulae are $S \wedge L \wedge F$ and $\neg S \wedge L \wedge F$. They are both inconsistent, since $L \wedge F$ is inconsistent.

The conjunction of F with the third formula is $S \wedge \neg L \wedge F$. The inconsistency of $L \wedge F$ implies $F \models \neg L$. Therefore, $S \wedge \neg L \wedge F$ is equivalent to $S \wedge F$, which is consistent by assumption. Its models therefore constitute the first class of the resulting order.

$S \wedge L \wedge F, L \wedge F$ and $S \wedge L$ are inconsistent. The conjunction of F with the first three classes are $S \wedge \neg L \wedge F, \neg S \wedge L \wedge F$ and $S \wedge L \wedge F$. They are all inconsistent, since they contain either $L \wedge F$ or $S \wedge F$, which are inconsistent.

The conjunction of F with the fourth formula is $\neg S \wedge \neg L \wedge F$. The inconsistency of $L \wedge F$ implies $F \models \neg L$. The inconsistency of $S \wedge F$ implies $F \models \neg S$. As a result, $\neg S \wedge \neg L \wedge F$ is equivalent to F . Since F is consistent, its models are class zero of the resulting order.

This list proves that if O is a two-class ordering, class zero of $O_{\text{lex}}(L)\text{full}(F)$ comprises the models of: $S \wedge L \wedge F$ if consistent, otherwise $L \wedge F$ if consistent, otherwise $S \wedge F$ if consistent, otherwise F , where S is a formula whose models are class zero of O .

The claim can now be proved. Two cases are considered: either $L \wedge F$ is satisfiable, or it is not.

$L \wedge F$ is **inconsistent**. Both $S \wedge L \wedge F$ and $L \wedge F$ are inconsistent. The first two of the four cases above are not possible. The remaining two are the consistency of $S \wedge F$ and its inconsistency. The resulting class zero is, respectively, $\text{Mod}(S \wedge F)$ and $\text{Mod}(F)$. This is the same as full meet revising O with F :

$$\text{Olex}(L)\text{full}(F) = \text{Ofull}(F).$$

$L \wedge F$ is **consistent**. The consistency of $L \wedge F$ is the second of the four cases above. As a result, the third and fourth are never considered: either $S \wedge L \wedge F$ is consistent, or it is inconsistent and $L \wedge F$ is consistent. The resulting class zero is $\text{Mod}(S \wedge L \wedge F)$ if not empty and $\text{Mod}(L \wedge F)$ otherwise. This is the same as full meet revising O by $L \wedge F$:

$$\text{Olex}(L)\text{full}(F) = \text{Ofull}(L \wedge F). \quad \square$$

Every alternation of lexicographic and full meet revisions reduces to a sequence of full meet revisions:

$$\text{olex}(L_1)\text{full}(F_1) \dots \text{lex}(L_n)\text{full}(F_n).$$

Each pair $\text{lex}(L_i)\text{full}(F_i)$ is the same as either $\text{full}(L_i \wedge F_i)$ or $\text{full}(F_i)$, depending on the satisfiability of $L_i \wedge F_i$. The satisfiability of these conjunctions can be checked independently on each other. They are equivalent to a logarithmic sequence of satisfiability tests. Checking entailment from a sequence of full meet revisions only is already known to require another logarithmic number of satisfiability tests.

THEOREM 25. *Establishing entailment from the base resulting from an alternating sequence of lexicographic and full meet revisions is $\Delta_2^p[\log n]$ -complete.*

PROOF. Entailment can be established by turning the alternating sequence into one comprising only full meet revisions and then checking entailment. The first step can be done employing a linear number of independent satisfiability checks. The second also, as proved by Theorem 22. Buss and Hay [13, Theorem 9] proved that two rounds of linear number of parallel satisfiability checks are equivalent to a logarithmic number of independent ones. This proves that the problem is in $\Delta_2^p[\log n]$.

The same transformation proves that every pair $\text{lex}(L)\text{full}(F)$ turns into $\text{full}(F)$ whenever $F \models L$ and F is consistent. The hardness of full meet revision alone turns into the hardness of the considered alternation by preceding each full meet revision $\text{full}(F)$ with $\text{lex}(F)$. \square

Same hard revision, lexicographic. Different easy revisions, very radical and full meet. Different complexity: Δ_2^p -complete and $\Delta_2^p[\log n]$ -complete. Complexity differs when alternating the same hard revision with different easy ones. In one case, it is as hard as the hard revision. In the other, it is as easy as the easy one. The same for alternating full meet revisions with bottom-refining or with lexicographic revisions. Hard in one case, easy in the other. No general pattern emerges, like complexity being the same as the hardest of two revisions. It depends on the specific operators.

5.6 Arbitrarily Intermixed Sequences

Sequences of similar revisions, sequences of alternating revisions. Still, specific sequences of revisions. What happens when arbitrarily intermixing sequences?

Theorem 12 gives an upper bound: no matter how iterated revisions are intermixed, entailment is in Δ_2^p . Theorems 23, 24, and 25 negate an equally general hardness result: complexity depends on which revisions the sequence contains.

Some results easily extend to linear fractions of the same revision. A sequence that contains at least a linear fraction of bottom-refining revisions among other revisions is Δ_2^p -hard. Alternating each bottom-refining revision with a number of other revisions of the same kind is the same as

a simple alternation if these other revisions are by the same formula; this is the case for the considered revision operators (not in general, for example, it is not when iterating an improvement operator until success [49, 78]). This trick turns the sequence that proves the hardness of simple alternations into a sequence that proves the hardness of the one-two alternations.

Revising multiple times by the same formula may look like cheating. It turns a quite general scheme (bottom-refining revisions and other revisions) into a specific one (where each subsequence of other revisions is by the same formula). Yet, specific schemes are the nature of hardness results. For example, the proof of hardness for lexicographic revision employs a sequence of revisions by single variables. Such sequences are specific. In general, a sequence may contain arbitrary complex formulae.

Hardness proofs are existential proofs. Every instance of an already known Δ_2^p -hard problem turns into a sequence of lexicographic revisions. Every instance turns into a sequence. Not the other way around: some sequences are not outcomes of the translation.

The question “Are all sequences hard?” is not meaningful. Its obvious answer is always “No.” Some sequences are easy, like the repetitions of the same revisions by the same formula. Some are hard, like the ones produced by the Δ_2^p -hard reductions.

The question that makes sense is whether a specific group of sequences is hard or not. The sequences of lexicographic revisions are Δ_2^p -hard. They are because they contain all results of reducing all instances of lexicographically maximal model inference. They contain, they do not coincide.

All hardness results are of this kind. Iterated revisions are not an exception. Hardness reductions prove that certain groups of sequences are hard. A reduction that produces only sequences of a given group prove the group hard, such as the group of all sequences alternating lexicographic and very radical revisions. All supersets of such a group are equally hard, including the group that contains all possible sequences.

6 A CASE FOR SEVERE ANTIWITHDRAWAL

Severe antiwithdrawal is one of the three belief changing operators all others reduce to. While refinements are lexicographic revisions done in the opposite order, severe antiwithdrawal requires a different computation, that of the underformula. Not only is it central, it is also in a way irreplaceable. Essential.

What makes it so important? One of the referees of this article asked for an analysis of its properties. Being just severe withdrawal with the negated formula, its basic properties for a single step are the same [27, 70]. Postulates for iteration have been proposed by Darwiche and Pearl [18], with variations from Boutilier [11] and Jin and Thielsher [43].

Postulate C1 by Darwiche and Pearl [18] states that $Osev(A)sev(B)$ is the same as $Osev(B)$ whenever B entails A . This is a postulate for revision. Revision by A ensures that A is entailed; the same for B . The principle of C1 is that if A is achieved (is entailed) whenever B is achieved (is entailed), then a first revision by A is irrelevant if followed by a revision by B . Ubi major, minor cessat.

Yet, severe antiwithdrawal is not a form of revision. Achieving A is not believing A . It is accepting the possibility it may be true. It is “giving a hearing to it,” as advocated by Glaister [33]. Achieving A is making A consistent, not entailed.

Does C1 make sense for antiwithdrawal? Its essence is: If A is achieved whenever B is achieved, then antiwithdrawing A is irrelevant if followed by antiwithdrawing B . Antiwithdrawing is making a formula acceptable. Is ensuring consistency with it. Achieving A is creating consistency with A . Achieving B is creating consistency with B .

Rewriting C1 for antiwithdrawal gives: $Osev(A)sev(B)$ is the same as $Osev(B)$ whenever the consistency with B implies the consistency with A . The premise is that every formula that is

consistent with B is also consistent with A . This is only possible if every model of B is also a model of A . In other words, B entails A . Technically, this is the same property as C1 for revision. It comes from the same principle and arrives to the same condition, but via a different route, where consistency takes the place of entailment.

Postulate C2 is: $Osev(A)sev(B)$ is the same of $Osev(B)$ whenever B entails the negation of A . The premise is that entailing B makes A impossible. Achieving B negates achieving A . Achieving is consistency for antiwithdrawal. Consistency with B negates consistency with A . Every model of B negates A . This is the same as B entailing $\neg A$. Again, same technical condition, but coming from a different route.

The premise of C3 is $Osev(B) \models A$: achieving B in the present case also achieves A . Achieving is entailing for revision. For antiwithdrawal, is ensuring consistency. The consistency of $Osev(B)$ with A . This formalizes as $Osev(B) \not\models \neg A$. This is the premise of C3 for antiwithdrawal. Its consequent is the same for $Osev(A)sev(B)$. For antiwithdrawal, $Osev(A)sev(B) \not\models \neg A$. Overall, C3 for antiwithdrawal is: if $Osev(B) \not\models \neg A$, then $Osev(A)sev(B) \not\models \neg A$. Contrary to C1 and C2, it differs from its version for revision. Yet, it coincides with C4 for revision.

Adapting C4 to antiwithdrawal has the same effect: it turns it into C3.

Severe antiwithdrawal enjoys C1, C3, and C4 and does not enjoy C2.

C1 The premise is that all models of B are also models of A . If k is the minimal class of models of A and k' that for B , then $k \leq k'$. Antiwithdrawing A merges all levels up to k . Antiwithdrawing B further merges all levels up to k' . Since k' is greater than k , the second merge takes over the first.

C2 A counterexample is $O = [C(0), C(1)]$ where $Mod(B)$ intersects $C(0)$ while $Mod(A)$ does not. Antiwithdrawing B leaves the same order. Antiwithdrawing A instead merges the two classes into one; antiwithdrawing B after that does not split classes.

C3 The premise is $Osev(B) \models A$. The minimal class $C(k)$ intersecting $Mod(B)$ contains only models satisfying A . Antiwithdrawing A leaves the ordering unaffected. Not only A is entailed by the sequence of antiwithdrawing A and B , this sequence is the same as antiwithdrawing B only.

C4 The premise is that the result of antiwithdrawing B is consistent with A . If $C(k')$ is the first class that intersects $Mod(B)$, then some models of $C(0) \cup \dots \cup C(k')$ satisfy A . Therefore, the minimal k such that $C(k)$ intersects $Mod(A)$ is less than k' . Antiwithdrawing A merges the classes from 0 to k ; further antiwithdrawing B merges all classes up to k' with $k' \geq k$, incorporating all changes done by the first antiwithdrawal.

While C2 is not met by severe antiwithdrawing, its two weakenings CB and Ind may [11, 43]. The premise of CB is $Osev(A) \models \neg B$. How to recast this property in terms of achievements? Achieving A makes B impossible. Making B impossible is different than negating the achievement of B , which would be $Osev(A) \not\models B$. It is a different property, with consistency in place of entailment. The other weakening Ind has a similar premise: $Osev(B) \not\models \neg A$, still involving consistency in addition to achievements.

Severe antiwithdrawal satisfies C1, C3, and C4. It does not satisfy C2. The weakened versions CB and Ind do not seem relevant to antiwithdrawal. All of this tells something about the individual properties of severe antiwithdrawal, but does not answer the root question: What is severe antiwithdrawal, exactly?

Antiwithdrawal is a shift in mind to accommodate the belief that something is possible. Formally, it minimally and rationally changes the ordering among possibilities to ensure mutual consistency with a formula. This is what Glaister [33] calls “giving a hearing to” and separates from contraction, “not believing.” The present article further supports this operation from the technical point of view, as a central belief changing mechanism.

However, the specific form of antiwithdrawal that is severe antiwithdrawal exhibits a serious drawback. To accept the possibility that something may be true, it also accepts everything else currently judged equally unlikely. This is too much a shift in mind. When the hunter accepts that a red bird may be in the thicket, he still maintains that no black boar is in the field next to the river.

Yet, if the red bird is indeed there, then the black boar will no longer be so unlikely. The possibility that some exotic animal wanders around the place increases the likeliness of some other doing as well. Something unlikely becoming a possibility opens the door to something else equally unlikely. At the same time, opening a door is not walking through it. Severe antiwithdrawal is an element in this process. It raises the threshold of believable unlikeliness. Some restriction is then necessary to ensure not everything is believed, only what is supported by some kind of evidence. This use of severe antiwithdrawal is supported by the technical analysis as well, which fruitfully employs it as a building block for translating other belief changing operations.

7 CONCLUSIONS

This article advocates and studies mixed sequences of belief change operators, in which revisions, refinements, and withdrawals may occur. With some exceptions [7, 19, 29, 42, 48, 53], the semantics for iterated belief revision mostly work on objects that are equivalent to total preorders, which lets using different kinds of changes at different times. Even the memoryless operators such as full meet revision [1] and the distance-based revision [17, 65] can be embedded in this framework: They produce a plausibility order that does not depend at all on the previous one except for their zero class.

The main technical result of this article is a method for computing the result of a mixed sequence of revisions. It directly works on sequences of lexicographic revisions, refinements, and severe antiwithdrawals, which may result from translating an arbitrary sequence of lexicographic revisions, refinements, severe withdrawal, natural, severe, plain severe, moderate severe, and very radical revisions, alternating in every possible way. The requirement of being able to solve propositional satisfiability problems is not too demanding, given the current efficiency of SAT algorithms [3, 5] and given that belief revision is built upon propositional logic, whose basic operations of satisfiability and validity are at the first level of the polynomial hierarchy [21, 58].

Example 9. The running example can be solved by an explicit representation of the preorders:

$$\emptyset \text{lex}(y) \text{nat}(\neg x) \text{ref}(x \wedge z) \text{rad}(\neg z).$$

The initial preorder is empty: $\emptyset = [\text{Mod}(\top)]$. The revisions change it as follows, where $\text{Mod}()$ is omitted from the classes for simplicity.

\emptyset	$\text{lex}(y)$	$\text{nat}(\neg x)$	$\text{res}(x \wedge z)$	$\text{rad}(\neg z)$
\top	y	$\neg x \wedge y$	$x \wedge y \wedge z$	$\neg x \wedge y \wedge \neg z$
	$\neg y$	$x \wedge y$	$\neg x \wedge y$	$x \wedge y \wedge \neg z$
		$\neg y$	$x \wedge y \wedge \neg z$	$\neg y \wedge \neg z$
			$\neg y \wedge x \wedge z$	$\neg z$
			$\neg y \wedge (\neg x \vee \neg z)$	

The final result is the same as obtained by the algorithm: the base of the last preorder is $\neg x \wedge y \wedge \neg z$. However, explicitly storing the preorder means representing all its classes, which in this example increased in number up to five. In general, with n variables there may be as many as 2^n models, and therefore as many as 2^n nonempty classes. In this case, the bound $2^3 = 8$ was almost reached after $\text{res}(x \wedge z)$.

Other articles explored the translations from different belief change operators into a single formalism. Rott has shown that severe withdrawal, irrevocable and irrefutable revision can be expressed in terms of revision by comparison [67], natural and lexicographic in terms of bounded revision [69]. Several single-step revisions can be recast in some forms of circumscription [57].

Several computational complexity results about belief revision are known. Eiter and Gottlob [22] proved that most distance-based and syntax approaches are Π_2^P -complete in the single-step case. In a further article [23], the same authors proved (among other results) that the same applies to positive right-nested counterfactuals, which are equivalent to a form of iterated revision. Nebel [62] proved a number of results, the most relevant to the present article being the that one-step syntactic-lexicographic revision is Δ_2^P -complete. This operator can encode lexicographic revision as defined in the iterated case by placing each formula in a separate priority class. Other iterated revisions have a similar degree of complexity [55].

A number of problems are left open. The algorithm requires a SAT solver, which is unavoidable given that the underlying language is propositional logic and SAT expresses its basic problems of satisfiability, mutual consistency and entailment. However, some restricted languages such as Horn and Krom require only polynomial time for checking satisfiability [74]. As a result, it makes sense to investigate their computational properties on iterated change. The analysis would not be obvious, because underformulae include both disjunction and conjunction, which may result in a non-Horn and non-Krom formula. The Horn restriction has been studied in single-step revisions by Eiter and Gottlob [20], and has recently been considered as a contributor to the semantics of revision [16].

Some iterated belief change operators such as radical revision (as opposed to very radical revision, considered in this article) consider some models “inaccessible” [25, 75]. In terms of total preorders, this amounts to shifting from a partition into ordered classes into a sequence of non-overlapping subsets; the models that are not in any of them are the inaccessible one. Alternatively, the highest-level class is given the special status of inaccessible model container. These operators have not been considered in this article, but the analysis could be extended to them.

Other operators not considered in this article include the ones based on numerical rankings [43, 72, 76, 80] and bidimensional ones [15, 28, 69]. They allow for specifying the strength of a revision either by a number or indirectly by referring to that of another formula. Either way, revision is by a pair of a formula and an expression of its strength. A preliminary analysis suggests that at least a form of bidimensional change, revision by comparison, can be recast in terms of lexicographic and severe antiwithdrawal, at the cost of first determining an underformula and a maxcon of the previous lexicographic revisions. Other two-place operators may be amenable to such reductions. Other recent work include iterated contraction [6, 50, 73] and operators where conditions on the result are specified, rather than mandating a mechanism for obtaining them [38].

Memoryless revision operators [17, 71] may be treated as if they had memory: this is the case of full meet revision, which is indeed oblivious to the previous history of revision. The ordering it generates is always $[Mod(K), Mod(\neg K)]$. In spite of its simplicity, it is still useful to characterize a *tabula rasa* step of reasoning, forgetting all previously acquired data to start over from a single simple information.

Operators with full memory [19, 29, 48, 53] require a different analysis, since they work from the complete history of revisions rather than from a total preorder that is modified at each step. The same applies to operators working from structure more complex than total preorders over models [7, 42].

Finally, given that a revision may be performed using different operators, a question is how to decide which. This is related to merging and non-prioritized revision. An answer may be to

use the history of previous revision to find out the credibility of a source [56], which affects the kind of incorporation. For example, trustworthy sources produce lexicographic revisions, plausible but not very reliable sources produce natural revisions, the others refinements. Still better, sources providing information that turned out to be valid after all subsequent changes are better treated by lexicographic revisions; source providing information that turned out to be specific to the current case are formalized by natural revision. As an alternative, every new information may be initially treated as a natural revision; if observations suggest its generality, then they are promoted to lexicographic.

Example 10 (Cont.). Sound of feathers. A bird, after all?

The hunter and the policeman turn their head, eager to find out. What comes out from the bushes is a drag queen in red feathers, who stopped by the thicket for the obvious reason while coming for the parade at the village fête. Not a bird ($\neg b$) but red (r), not to be hunted anyway ($\neg h$).

REFERENCES

- [1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. 1985. On the logic of theory change: Partial meet contraction and revision functions. *J. Symbol. Logic* 50 (1985), 510–530.
- [2] Carlos Areces and Verónica Becher. 2001. Iterable AGM functions. In *Frontiers in Belief Revision*, M. Williams and H. Rott (Eds.). Springer Netherlands, 261–277. https://doi.org/10.1007/978-94-015-9817-0_13
- [3] T. Balyo, M. J. H. Heule, and M. Järvisalo. 2017. SAT competition 2016: Recent developments. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. AAAI Press/The MIT Press, 5061–5063. Retrieved from <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14977>.
- [4] S. Benferhat, D. Dubois, and H. Prade. 2001. A computational model for belief change and fusing ordered belief bases. In *Frontiers in Belief Revision*, M. Williams and H. Rott (Eds.). Springer, 109–134.
- [5] A. Biere, M. J. M. Heule, H. van Maaren, and T. Walsh. 2009. *Handbook of Satisfiability*. IOS Press.
- [6] R. Booth and J. Chandler. 2019. From iterated revision to iterated contraction: Extending the Harper Identity. *Artific. Intell.* 277 (2019). <https://doi.org/10.1016/j.artint.2019.103171>
- [7] R. Booth and J. Chandler. 2020. On strengthening the logic of iterated belief revision: Proper ordinal interval operators. *Artific. Intell.* 285 (2020), 103–289. <https://doi.org/10.1016/j.artint.2020.103289>
- [8] R. Booth and T. Meyer. 2006. Admissible and restrained revision. *J. Artific. Intell. Res.* 26 (2006), 127–151.
- [9] R. Booth and A. Nittka. 2008. Reconstructing an Agent’s epistemic state from observations about its beliefs and non-beliefs. *J. Logic Comput.* 18, 5 (2008), 755–782. <https://doi.org/10.1093/logcom/exm091>
- [10] R. Booth, T. A. Meyer, and K. Wong. 2006. A bad day surfing is better than a good day working: How to revise a total preorder. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*. AAAI Press/The MIT Press, 230–238.
- [11] C. Boutilier. 1996. Abduction to plausible causes: An event-based model of belief update. *Artific. Intell.* 83 (1996), 143–166.
- [12] C. Boutilier. 1996. Iterated revision and minimal change of conditional beliefs. *J. Philos. Logic* 25, 3 (1996), 263–305.
- [13] S. R. Buss and L. Hay. 1991. On truth-table reducibility to SAT. *Info. Comput.* 91, 1 (1991), 86–102. [https://doi.org/10.1016/0890-5401\(91\)90075-D](https://doi.org/10.1016/0890-5401(91)90075-D)
- [14] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf. 2000. Space efficiency of propositional knowledge representation formalisms. *J. Artific. Intell. Res.* 13, 1 (2000), 1–31.
- [15] J. Cantwell. 1997. On the logic of small changes in hypertheories. *Theoria* 63, 1-2 (1997), 54–89.
- [16] N. Creignou, R. Ktari, and O. Papini. 2018. Belief update within propositional fragments. *J. Artific. Intell. Res.* 61 (2018), 807–834. <https://doi.org/10.1613/jair.5541>
- [17] M. Dalal. 1988. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88)*. 475–479.
- [18] A. Darwiche and J. Pearl. 1997. On the logic of iterated belief revision. *Artific. Intell.* 89, 1-2 (1997), 1–29.
- [19] J. P. Delgrande, D. Dubois, and J. Lang. 2006. Iterated revision as prioritized merging. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*. 210–220.
- [20] T. Eiter and G. Gottlob. 1992. *Complexity Results for Disjunctive Logic Programming and Application to Nonmonotonic Logics*. Technical Report CD-TR 92/41. Technische Universität Wien, Vienna Austria, Christian Doppler Labor für Expertensysteme.
- [21] T. Eiter and G. Gottlob. 1992. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artific. Intell.* 57 (1992), 227–270.

- [22] T. Eiter and G. Gottlob. 1992. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artific. Intell.* 57 (1992), 227–270.
- [23] T. Eiter and G. Gottlob. 1996. The complexity of nested counterfactuals and iterated knowledge base revisions. *J. Comput. Syst. Sci.* 53, 3 (1996), 497–512.
- [24] T. Eiter and G. Gottlob. 1997. The complexity class Θ_2^P : Recent results and applications in AI and modal logic. In *Proceedings of the 11th International Symposium on Fundamentals of Computer Theory (FCT'97)*. Springer, 1–18. <https://doi.org/10.1007/BFb0036168>
- [25] E. L. Fermé. 2000. Irrevocable belief revision and epistemic entrenchment. *J. Interest Group Pure Appl. Logic* 8, 5 (2000), 645–652. <https://doi.org/10.1093/jigpal/8.5.645>
- [26] E. Fermé and M. D. L. Reis. 2013. Epistemic entrenchment-based multiple contractions. *Rev. Symbol. Logic* 6, 3 (2013), 460–487.
- [27] E. Fermé and R. Rodriguez. 1998. A brief note about Rott contraction. *J. Interest Group Pure Appl. Logic* 6, 6 (1998), 835–842.
- [28] E. Fermé and H. Rott. 2004. Revision by comparison. *Artific. Intelligence* 157, 1 (2004), 5–47.
- [29] D. M. Gabbay, G. Pigozzi, and J. Woods. 2003. Controlled revision—An algorithmic approach for belief revision. *J. Logic Comput.* 13, 1 (2003), 3–22. <https://doi.org/10.1093/logcom/13.1.3>
- [30] P. Gärdenfors. 1988. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. Bradford Books, MIT Press, Cambridge, MA.
- [31] P. Gärdenfors and D. Makinson. 1988. Revision of knowledge systems using epistemic entrenchment. In *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'88)*. 83–95.
- [32] M. Girlando, B. Lellmann, N. Olivetti, and G. L. Pozzato. 2017. Hypersequent calculi for Lewis' conditional logics with uniformity and reflexivity. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. 131–148.
- [33] S. M. Glaister. 1998. Symmetry and belief revision. *Erkenntnis* 49, 1 (1998), 21–56.
- [34] S. M. Glaister. 2000. Recovery recovered. *J. Philos. Logic* 29, 2 (2000), 171–206.
- [35] A. Grove. 1988. Two modellings for theory change. *J. Philos. Logic* (1988), 157–170.
- [36] S. O. Hansson. 2011. Logic of belief revision. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.
- [37] S. O. Hansson. 2006. In praise of full meet contraction. *Análisis filosófico* 26, 1 (2006), 134–146.
- [38] S. O. Hansson. 2016. Iterated descriptor revision and the logic of Ramsey test conditionals. *J. Philos. Logic* 45, 4 (2016), 429–450. <https://doi.org/10.1007/s10992-015-9381-7>
- [39] L. A. Hemachandra. 1989. The strong exponential hierarchy collapses. *J. Comput. Syst. Sci.* 39, 3 (1989), 299–322. [https://doi.org/10.1016/0022-0000\(89\)90025-1](https://doi.org/10.1016/0022-0000(89)90025-1)
- [40] A. Hunter and J. P. Delgrande. 2011. Iterated belief change due to actions and observations. *J. Artific. Intell. Res.* 40 (2011), 269–304. <https://doi.org/10.1613/jair.3132>
- [41] A. Hunter and J. P. Delgrande. 2015. Belief change with uncertain action histories. *J. Artific. Intell. Res.* 53 (2015), 779–824. <https://doi.org/10.1613/jair.4558>
- [42] T. I. I. Aravanis, P. Peppas, and M.-A. Williams. 2019. Observations on Darwiche and Pearl's approach for iterated belief revision. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. 1509–1515. <https://doi.org/10.24963/ijcai.2019/209>
- [43] Y. Jin and M. Thielscher. 2007. Iterated belief revision, revised. *Artific. Intell.* 171, 1 (2007), 1–18.
- [44] H. Katsuno and A. O. Mendelzon. 1991. Propositional knowledge base revision and minimal change. *Artific. Intell.* 52 (1991), 263–294.
- [45] G. Kern-Isberner and D. Huvermann. 2017. What kind of independence do we need for multiple iterated belief change? *J. Appl. Logic* 22 (2017), 91–119. <https://doi.org/10.1016/j.jal.2016.11.033>
- [46] G. Kern-Isberner, N. Skovgaard-Olsen, and W. Spohn. 2021. *Ranking Theory*. The MIT Press, Chapter 5.3.
- [47] S. Konieczny. 1998. *Operators with Memory for Iterated Revision*. Technical Report IT-314. Laboratoire d'Informatique Fondamentale de Lille.
- [48] S. Konieczny and R. Pino Pérez. 2000. A framework for iterated revision. *J. Appl. Non-class. Logics* 10, 3-4 (2000), 339–367.
- [49] S. Konieczny and R. Pino Pérez. 2008. Improvement operators. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*. AAAI Press/The MIT Press, 177–187.
- [50] S. Konieczny and R. Pino Pérez. 2017. On iterated contraction: Syntactic characterization, representation theorem and limitations of the Levi identity. In *Proceedings of the 11th International Conference on Scalable Uncertainty Management (SUM'17)*. Springer, 348–362. https://doi.org/10.1007/978-3-319-67582-4_25
- [51] M. W. Krentel. 1988. The complexity of optimization problems. *J. Comput. System Sci.* 36 (1988), 490–509.

- [52] M. Langlois, R. H. Sloan, B. Szörényi, and G. Turán. 2008. Horn complements: Towards Horn-to-Horn belief revision. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*. 466–471.
- [53] D. Lehmann. 1995. Belief revision, revised. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. 1534–1540.
- [54] P. Liberatore. 1997. The complexity of belief update. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*. 68–73.
- [55] P. Liberatore. 1997. The complexity of iterated belief revision. In *Proceedings of the 6th International Conference on Database Theory (ICDT'97)*. 276–290.
- [56] P. Liberatore. 2016. Belief merging by examples. *ACM Trans. Comput. Logic* 17, 2 (2016), 9:1–9:38. <https://doi.org/10.1145/2818645>
- [57] P. Liberatore and M. Schaerf. 1997. Reducing belief revision to circumscription (and viceversa). *Artific. Intell.* 93, 1–2 (1997), 261–296.
- [58] P. Liberatore and M. Schaerf. 2001. Belief revision and update: Complexity of model checking. *J. Comput. Syst. Sci.* 62, 1 (2001), 43–72. <https://doi.org/10.1006/jcss.2000.1698>
- [59] T. Meyer, A. Ghose, and S. Chopra. 2002. Syntactic representations of semantic merging operations. In *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI'02)*. 620. https://doi.org/10.1007/3-540-45683-X_88
- [60] A. Nayak. 1994. Iterated belief change based on epistemic entrenchment. *Erkenntnis* 41 (1994), 353–390.
- [61] A. Nayak, M. Pagnucco, and P. Peppas. 2003. Dynamic belief revision operators. *Artific. Intell.* 146, 2 (2003), 193–228. [https://doi.org/10.1016/S0004-3702\(03\)00017-1](https://doi.org/10.1016/S0004-3702(03)00017-1)
- [62] B. Nebel. 1998. How hard is it to revise a belief base? In *Belief Change—Handbook of Defeasible Reasoning and Uncertainty Management Systems*, Vol. 3, D. Dubois and H. Prade (Eds.). Kluwer Academic.
- [63] O. Papini. 2001. Iterated revision operations stemming from the history of an agent's observations. In *Frontiers in Belief Revision*. Applied Logic Series, Vol. 22. Springer, 279–301.
- [64] P. Peppas, A. M. Fotinopoulos, and S. Seremetaki. 2008. Conflicts between relevance-sensitive and iterated belief revision. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*. IOS Press, 85–88. <https://doi.org/10.3233/978-1-58603-891-5-85>
- [65] P. Peppas and M. A. Williams. 2016. Kinetic consistency and relevance in belief revision. In *Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA'16)*. 401–414. https://doi.org/10.1007/978-3-319-48758-8_26
- [66] H. Rott. 2003. Coherence and conservatism in the dynamics of belief II: Iterated belief change without dispositional coherence. *J. Logic Comput.* 13, 1 (2003), 111–145. <https://doi.org/10.1093/logcom/13.1.111>
- [67] H. Rott. 2006. Revision by comparison as a unifying framework: Severe withdrawal, irrevocable revision and irrefutable revision. *Theoret. Comput. Sci.* 355, 2 (2006), 228–242. <https://doi.org/10.1016/j.tcs.2006.01.011>
- [68] H. Rott. 2009. Shifting priorities: Simple representations for twenty-seven iterated theory change operators. In *Towards Mathematical Philosophy*, D. Makinson, J. Malinowski, and H. Wansing (Eds.). Trends in Logic, Vol. 28. Springer Netherlands, 269–296. https://doi.org/10.1007/978-1-4020-9084-4_14
- [69] H. Rott. 2012. Bounded revision: Two-dimensional belief change between conservative and moderate revision. *J. Philos. Logic* 41, 1 (2012), 173–200. <https://doi.org/10.1007/s10992-011-9206-2>
- [70] H. Rott and M. Pagnucco. 1999. Severe withdrawal (and recovery). *J. Philos. Logic* 28, 5 (1999), 51–547.
- [71] K. Satoh. 1988. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88)*. 455–462.
- [72] K. Sauerwald, J. Haldimann, M. von Berg, and C. Beierle. 2020. Descriptor revision for conditionals: Literal descriptors and conditional preservation. In *Proceedings of KI-2020: Advances in Artificial Intelligence—43rd German Conference on AI*. Springer, 204–218. https://doi.org/10.1007/978-3-030-58285-2_15
- [73] K. Sauerwald, G. Kern-Isberner, and C. Beierle. 2020. A conditional perspective for iterated belief contraction. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI'20)*. IOS Press, 889–896. <https://doi.org/10.3233/FAIA200180>
- [74] T. J. Schaefer. 1978. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing (STOC'78)*. 216–226.
- [75] K. Segerberg. 1998. Irrevocable belief revision in dynamic doxastic logic. *Notre Dame J. Formal Logic* 39, 3 (1998), 287–306. <https://doi.org/10.1305/ndjfl/1039182247>
- [76] W. Spohn. 1988. Ordinal conditional functions: A dynamic theory of epistemic states. In *Causation in Decision, Belief Change, and Statistics*. Kluwer Academic, 105–134.
- [77] W. Spohn. 1999. Ranking functions, AGM style. In *Internet Festschrift for Peter Gärdenfors*, B. Hansson, S. Halldén, N.-E. Sahlin, and W. Rabinowicz (Eds.).

- [78] F. R. Velázquez-Quesada. 2017. On subtler belief revision policies. In *Proceedings of the 6th International Workshop on Logic, Rationality, and Interaction (Lecture Notes in Computer Science, Vol. 10455)*. Springer, 314–329. https://doi.org/10.1007/978-3-662-55665-8_22
- [79] K. Wagner. 1987. More complicated questions about maxima and minima, and some closures of NP. *Theoret. Comput. Sci.* 51 (1987), 53–80. [https://doi.org/10.1016/0304-3975\(87\)90049-1](https://doi.org/10.1016/0304-3975(87)90049-1)
- [80] M. Williams. 1994. Transmutations of knowledge systems. In *Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning (KR'94)*. 619–629.

Received 26 February 2022; revised 30 October 2022; accepted 23 January 2023