The Complexity of Nested Counterfactuals and Iterated Knowledge Base Revisions*

Thomas Eiter and Georg Gottlob

Christian Doppler Laboratory for Expert Systems
Information Systems Department
Technical University of Vienna
Paniglgasse 16, A-1040 Wien, Austria
(eiter|gottlob)@dbai.tuwien.ac.at

final version

Abstract

We consider the computational complexity of evaluating nested counterfactuals over a propositional knowledge base. A counterfactual $p > q$ is a conditional query with the meaning "If p would be true in the knowledge base, would it then hold that also q is true," which is different from material implication $p \Rightarrow q$. A nested counterfactual is a counterfactual statement where the premise p or the conclusion q is a counterfactual. Statements of the form $p_1 > (p_2 > \cdots (p_n > q) \cdots)$ intuitively correspond to conditional queries involving a sequence of revisions. We show that evaluating such statements is Π_2^P-complete, and that this task becomes PSPACE-complete if negation is allowed in a nesting of this form. We also consider nesting a counterfactual in the premise, i.e. $(p > q) > r$, and show that evaluating such statements is Π_4^P-complete, thus most likely much harder than evaluating $p > (q > r)$. Finally, we also address iterated nestings in the premise and the mix of nestings in the premise and the conclusion.

1 Introduction

A counterfactual is a conditional query "if p, then q," where the premise p is false in the current context [14], e.g. "This afternoon the electricity failed; if it hadn't failed,

*An extended abstract of this paper appeared in the Proceedings of 13th International Joint Conference on Artificial Intelligence (IJCAI '93), Chambery, France, 1993.

dinner would have been ready on time." This is customarily written as '$p > q$' to distinguish it from material implication '$p \Rightarrow q$', which is trivially true if p is false in the current context. The evaluation of a counterfactual in a certain context, which is described by a knowledge base, can be done using the Ramsey Test, which roughly states that $p > q$ is true if the minimal change to accept p requires accepting q. Counterfactual reasoning is nonmonotonic in the sense that by augmenting the knowledge base a previously valid counterfactual may become false. Several types of counterfactuals have been defined in the literature, based on different methods for updating a knowledge base (for overviews, cf. [37, 19, 9]). The method used in the present paper is the classical approach by Fagin, Ullman, Vardi, and Kuper [11, 10], which is one of the most important methods for updating logical databases. This method was later used in AI by Ginsberg, who demonstrated the relevance of counterfactual reasoning to a number of AI and knowledge base applications in [14]. The reader is referred to this and [10, 12, 19, 37, 23, 16] for a background.

The complexity of evaluating counterfactuals over propositional knowledge bases, i.e. finite propositional theories, was considered in [23, 16, 9]. For the method of Fagin, Ullman, and Vardi, evaluation was shown to be Π_2^P-complete [23, 9].

In the present paper we deal with nested counterfactuals over propositional knowledge bases, i.e., counterfactuals where the premise or the conclusion can be a counterfactual itself instead of a plain propositional sentence.

Counterfactuals with nestings in the conclusion, i.e., statements $p > (q > r)$, have a natural meaning. Intuitively, such statements correspond to conditional queries involving a revision p followed by a revision q. We define the semantics of such statements according to [11, 10, 14] and determine the complexity of evaluation. Counterfactuals of this form are often used in real-life contexts and are an important principle of common-sense reasoning.

Example 1: Consider the following scenario, which is taken from the realm of diagnosis. Assume that structure and functionality of the electric system of a car is stored in a knowledge base, which is maintained as a logical theory. (Note that such a logical description is essential to model-based diagnosis [26].) In this scenario, the following statement is plausible: If the headlight would be turned on and it would not shine, would it then shine if the fuse protecting the headlight would have been replaced by a new one? If $switch\_on$ represents that the headlight is turned on, $light\_shines$ that the headlight shines, and $fuse\_replaced$ that the respective fuse has been replaced by a good one, then the statement amounts to

$$(switch\_on \wedge \neg light\_shines) > (fuse\_replaced > light\_shines). \qquad (1)$$

Note that this statement is different from $(switch\_on \wedge \neg light\_shines \wedge fuse\_replaced) > light\_shines$, which clearly is always false ($\neg light\_shines$ is in the premise while

$light\_shines$ is in the conclusion). In order to properly evaluate the statement (1), the current knowledge base has to be first revised by incorporating the information $switch\_on \wedge \neg light\_shines$, and after that by incorporating $fuse\_replaced$; note that the latter step may cause that $\neg light\_shines$ is no longer true and retracted from the knowledge base. For example, if it is known that all other involved components are ok (say the battery, the wires, and the lamps), then the fact that also the fuse is ok together with the natural causal relationships between the propositions (stored as formulas in the knowledge base) implies that the headlight shines, and hence $light\_shines$ is derivable from the knowledge base.

Therefore, depending on the initial context, the statement (1) might take different values. For example, if it is known that all other components except the fuse are ok, then it naturally evaluates to true, while if it is known that e.g. the battery is broken, then it should take the value false. □

More generally, nestings into the conclusion can be iterated. A statement of the form $p_1 > (p_2 > \cdots (p_n > q) \cdots)$, which we call a *right-nested* counterfactual, is true over a knowledge base T precisely if T, after the sequence of revisions $p_1, p_2, \ldots, p_n$, implies q, i.e., $(\cdots(T \circ p_1) \cdots \circ p_n) \models q$, where $\circ$ is an update operator, i.e., an operator that allows to incorporate new information p into a theory T, yielding a consistent theory $T \circ p$, even in the case if T is inconsistent with p. Right-nested counterfactuals are for instance naturally relevant to planning and reasoning about actions (cf. [15, 36]). Thus, the complexity results derived for right-nested counterfactuals (that is, inference after iterated knowledge base revision), have practical applications in this field.

The alternative to nesting a counterfactual into the consequence is nesting into the premise, i.e. a nesting $(p > q) > r$, which we call *left-nesting*. Intuitively, $(p > q) > r$ means "Would r be true in the closest context where $p > q$ is true". Note this is different from "if $p > q$, then r", which is true if $p > q$ is false. Nestings $(p > q) > r$ are relevant to practice, as the following example shows.

Example 2: Imagine a technical system that has several components, e.g., a hardware board, whose structure and functionality is described in a logical knowledge base. The system is capable of detecting a certain error (e.g., a mismatch of two signals on the board), if, whenever this error occurs, it is reported by the system (e.g., a special error signal is set true). Suppose now that the system should be made error detecting, i.e., it should be modified so that it is capable of detecting this specific error. A natural question that rises in this context is whether a certain component (e.g., a particular resistor) will be present in the system after the modification. Let e represent the fact that the specific error occurs, r that error occurrence is reported, and c that the particular component is present in the system. Then, the question amounts to the statement $(e > r) > c$ on the current state of the system. Indeed, if d stands for capability of error detection, then the question amounts to $d > c$: If the system would be error detecting, would then c be present ? On the other hand, capability of error detection amounts to $e > r$: if the error would occur, would it then be reported ? Thus, since $d = e > r$, the question amounts to $(e > r) > c$. □

We define a semantics of left-nested counterfactuals and determine the complexity of evaluation. We also consider a possible extension to iterated nestings of this type, and to the mix of left and right-nestings. However, it seems that iterated left-nesting and mixed nextings are conceptually involved and lack simple intuitive applications.

Our study also includes allowing negation in nesting counterfactuals, i.e. statements like $p > \neg(q > r)$. This is motivated by natural relevance, shown e.g. by game applications.

Example 3: Consider a two person game in which, starting from an initial state, the players move in alternation until the game is over (e.g. chess). A player has a *forced win* at a certain state of the game, if he will win regardless of his choice for his next move.

Suppose that the game has reached a certain state after n moves $m_1, \ldots, m_n$. Player 1, who is in turn for the next move m_{n+1}, is concerned about the consequences of his choice for the move. Suppose he wants to know the following: Is it the case that, regardless of my choice for the next move, Player 2 does *not* have a forced win (i.e., not every choice of Player 2 for his next move will result in a win of Player 2).

Let m_i represent the fact that the i-th move has been made, and w_2 that Player 2 wins. Then, the question of Player 1 amounts to the statement $m_{n+1} > \neg(m_{n+2} > w_2)$ over the knowledge base S_n describing the current state of the game. Indeed, if the knowledge base S_i, $i \geq 0$, describes the state after the i-th move, then $S_i \circ m_{i+1}$ describes the state after the $i + 1$-st move. (Note that, according to the different choices for the move, different states may result.) Therefore, a forced win of Player 2 after the n+1-st move amounts to whether $m_{n+2} > w_2$ is true on S_{n+1}. Consequently, Player 1's question amounts to whether $m_{n+2} > w_2$ is false on $S_{n+1} = S_n \circ m_{n+1}$; this is equivalent to whether $m_{n+1} > \neg(m_{n+2} > w_2)$ is true on S_n. □

The main results of this paper are shown in Table 1. There, statements are assumed to be of the form $\alpha > \beta$, where α, β are propositional formulas or, in general, possibly negated and nested counterfactuals. In positive statements, no negation in front of any nested counterfactual is allowed. The nesting depth is the maximum level at which an elementary counterfactual $p > q$ is nested; an ordinary counterfactual $p > q$ has nesting depth 0 (see Section 2 for a precise definition). The results in Table 1 are commented as follows.

Deciding positive right-nested counterfactuals (statements of the form $p_1 > (p_2 > \cdots (p_n > q) \cdots)$) is Π_2^P-complete. This is rather surprising and can be viewed as a positive result. It has an interesting consequence for the two basic approaches to cope with iterated knowledge base revisions. The first incorporates each revision

$\alpha > \beta$	nesting depth 1	nesting depth $< k$	unbounded nesting
pos. right nesting	Π_2^P-complete	Π_2^P-complete	Π_2^P-complete
right nesting	Π_2^P-complete	Π_{k+1}^P-complete	PSPACE-complete
[pos.] left nesting	Π_4^P-complete	in Π_{2k}^P	in PSPACE
mixed nesting	Π_4^P-complete	in Π_{2k}^P	PSPACE-complete

Table 1: Main complexity results on evaluating nested counterfactuals.

into the knowledge base and needs in general exponential space and time, while the second stores the initial knowledge base and the syntactic sequence $[p_1, p_2, \ldots, p_n]$ of revisions separately and accounts for it in query answering. Our result guarantees that the second approach does not get substantially (i.e. exponentially) harder when the sequence of revisions increases, which strongly favors this approach.

Note that, as shown in Section 3.1, refining the evaluation of right-nested counterfactuals by the method of priorities [11, 23], which attaches a priority value to each formula, does not change the complexity. Rather, the complexity decreases to Δ_2^P if the priority values are pairwise distinct. Under the additional restriction that all formulas are Horn clauses, we obtain the optimistic result that evaluating $p_1 > (p_2 > \cdots (p_n > q) \cdots)$ is possible in polynomial time.

The second row in Table 1 shows that evaluating right-nested counterfactuals gets more complicated (PSPACE-complete) if negation can appear in front of nested counterfactuals. The entry states that for a constant bound k on the nesting depth, evaluation is Π_{k+1}^P-complete. In fact, we derive the stronger result that the same holds for negation depth at most k, i.e. the number of negations in front of nested counterfactuals.

The third row shows that checking validity of left-nested counterfactuals of the form $((\neg)(p > q)) > r$ is Π_4^P-complete, and hard already for positive such statements. It appears that evaluating left-nested counterfactuals is one of few genuine natural problems known to be complete for this class. This result and other results contribute to warrant the originally expected use of the classes of the polynomial hierarchy for classifying the complexity of natural problems, which was called in question [33]. Iterated left-nesting seems to increase complexity, but it is not clear whether it can give all of PSPACE. It is conjectured that the given upper bounds are sharp. Note that negation in left-nesting, as opposed to right-nesting, does not increase complexity; in particular, evaluating $(\neg(p > q)) > r$ is in Π_4^P.

The last row shows results about the use of mixed nesting and possible negation.

Note that the use of right-nesting and negation does not lead beyond Π^P_{2k}; in fact, this is true if only the left-nesting depth (i.e., the maximal number of left-nesting steps) and negation is limited, while arbitrarily deep positive right-nesting is allowed. Thus, bounded left-nesting and negation does not give us all of PSPACE.

The rest of this paper is organized as follows. Section 2 introduces concepts and reviews previous results. Section 3 investigates into the complexity of evaluating right-nested counterfactuals without negation. Since this type of counterfactual is the most important one from the practical point of view, we analyze for such statements the effects of restricting formulas to Horn clauses and priorities. Section 4 is devoted to right-nested counterfactuals with negation. Section 5 deals with left-nested counterfactuals, and Section 6 addresses mixed counterfactuals. The final Section 7 reviews related work and gives some conclusions.

2 Preliminaries and previous results

We assume that the reader knows about the basic concepts of NP-completeness, the polynomial hierarchy, and PSPACE (rf. [13] for an excellent exposition and to [18] for quick reference). Briefly, PSPACE is the class of problems decidable in polynomial space. The classes Δ^P_k, Σ^P_k, and Π^P_k of the polynomial hierarchy are defined as follows:

$$\Delta^P_0 = \Sigma^P_0 = \Pi^P_0 = \mathrm{P}$$

and for all $k \geq 0$,

$$\Delta^P_{k+1} = \mathrm{P}^{\Sigma^P_k},\ \ \Sigma^P_{k+1} = \mathrm{NP}^{\Sigma^P_k},\ \ \Pi^P_{k+1} = \mathrm{co}\Sigma^P_{k+1}.$$

In particular, $\Delta^P_1 = \mathrm{P}$, $\Sigma^P_1 = \mathrm{NP}$, and $\Pi^P_1 = \mathrm{coNP}$. Clearly $\Sigma^P_k \subseteq \Sigma^P_k \cup \Pi^P_k \subseteq \Delta^P_{k+1} \subseteq \Sigma^P_{k+1}$, but for $k \geq 1$ any equality is considered very unlikely. The canonical PSPACE-complete problem is deciding the validity of a quantified Boolean formula (QBF) $\Phi = Q_1 a_1 Q_2 a_2 \cdots Q_n a_n E$, where each quantifier $Q_i \in \{\exists, \forall\}$ ranges over $\{\mathit{true}, \mathit{false}\}$ and E is a Boolean formula built on variables $a_1, \ldots, a_n$. Denote by $\mathrm{QBF}_{k,\exists}$ (resp. $\mathrm{QBF}_{k,\forall}$) the valid QBFs Φ with $Q_1 = \exists$ (resp. $Q_1 = \forall$) and $k-1$ quantifier alternations, i.e., occurrences of $i > 1$ such that $Q_i \neq Q_{i-1}$. Deciding if $\Phi \in \mathrm{QBF}_{k,\exists}$ (resp. $\Phi \in \mathrm{QBF}_{k,\forall}$) is complete for Σ^P_k (resp. Π^P_k).

Let $\mathcal{L}$ be the language of propositional logic over some set of atoms. The symbols "$\top$" and "$\bot$" denote constants for truth and falsity, respectively. Formulas are built over propositional atoms, which are denoted by lower case letters a, b, c from the beginning of the alphabet. We follow the common convention on the binding of the logical connectives. A *literal* is an atom or the negation of an atom. Letters $p, q, \ldots$ and $E, F, \ldots$ denote propositional (resp. Boolean) formulas. We use the following notation. If $\underline{p} = p_1, \ldots, p_n$ and $\underline{q} = q_1, \ldots, q_n$ are lists of propositional formulas, then $\underline{p} \equiv \underline{q}$ denotes $(p_1 \equiv q_1) \wedge \cdots \wedge (p_n \equiv q_n)$. As usual, juxtaposition $\underline{pq}$ denotes the concatenation of $\underline{p}$ and $\underline{q}$. Furthermore, $\neg\underline{p}$ stands for $\neg p_1, \ldots, \neg p_n$ and $r \Rightarrow \underline{p}$ denotes $r \Rightarrow p_1 \wedge \cdots \wedge p_n$. If φ is a truth value assignment to atoms $a_1, \ldots, a_n$, then E_φ denotes the formula obtained from E by replacing in E each occurrence of a_i by $\top$ if $\varphi(a_i) = \mathit{true}$ and by $\perp$ otherwise, for all $i = 1, \ldots, n$.

A *propositional knowledge base* (in the sequel, simply knowledge base) is a finite subset of $\mathcal{L}$. Knowledge bases are denoted by letters S, T, U etc.

A counterfactual "If p would be true, then q would be true" is denoted by $p > q$. The crucial link between counterfactuals and knowledge base revision is the *Ramsey Test* [12], which for our setting states that $p > q$ should be considered true on a knowledge base S if and only if the minimal change of S to incorporate p yields q true, cf. [12, p. 147]; more formally, $p > q$ is true over S if and only if $S \circ p \models q$, where $\circ$ is a suitable revision operator.

Complexity issues for counterfactuals were studied in [9]. It was shown there that for several operators $\circ$, evaluating $p > q$ is complete for some class of the second level of the polynomial hierarchy. Rather than considering ordinary counterfactuals, we here draw attention to nested counterfactuals. In the present paper we concentrate on the knowledge base revision method by Fagin, Ullman, and Vardi [11, 10], since this appears to be the most appropriate in the database context. Some of the results presented were already announced in [9].

The formal semantics of $p > q$ is defined as follows. Let

$$W(p, S) = \{T \subseteq S : T \not\models \neg p, \quad T \subset U \subseteq S \Rightarrow U \models \neg p\}$$

be the "possible worlds for p" [14],[1] and let

$$\mathcal{F}(p, S) = \{T \cup \{p\} : T \in W(p, S)\}.$$

Thus, $\mathcal{F}(p, S)$ is a collection of knowledge bases called a *flock* in [10], whose knowledge bases are considered to be independent possible outcomes of the update of S by p.

Using the Fagin et al. operator, the truth value of a counterfactual $p > q$ over a knowledge base S is defined as follows: $p > q$ has value *true* over S (in symbols, $S \models p > q$) if for every $T \in \mathcal{F}(p, S)$, $T \models q$, and value *false* otherwise ($S \not\models p > q$).

As shown in [23, 9], evaluating a counterfactual is most likely much harder than any NP-complete problem.

Proposition 2.1 [23, 9] *Deciding whether* $S \models p > q$ *is* Π_2^P*-complete.*

[1]This term may be misleading since $W(p, T)$ does *not* refer to possible worlds (in the technical sense, cf. [1]) at all but to subsets of a set of formulas.

Proof. The membership part is easy, cf. proof of Proposition 3.1. For the hardness part, it was shown in [9, Lemma 6.2] (cf. the full paper for a proof) that $\Phi \in \mathrm{QBF}_{2,\forall}$, for $\Phi = \forall \underline{a}_1 \exists \underline{a}_2 E(\underline{a}_1, \underline{a}_2)$, is equivalent to $S \models p > q$, where

$$\begin{aligned} S &= \{\underline{a}_1, \underline{a}_1', \underline{a}_2, c\}, \\ p &= [\underline{a}_1 \equiv \neg \underline{a}_1'] \wedge (c \Rightarrow E) \wedge (a_{2,1} \vee \ldots \vee a_{2,n_2} \Rightarrow c), \\ q &= c. \quad \Box \end{aligned} \tag{2}$$

We next define in a uniform way a class of formulas that contains both right and left-nested counterfactuals and is closed under nestings and negation. A semantics is given then to these formulas using the possible worlds approach, which is extended to possible worlds of counterfactuals in the spirit of Fagin et al.

The *nested counterfactuals* with possible negation are the smallest formula set $\mathcal{C}$ that satisfies the following properties:

(1) if $\alpha, \beta \in \mathcal{C} \cup \mathcal{L}$, then $\alpha > \beta \in \mathcal{C}$;

(2) if $\mathbf{c} \in \mathcal{C}$, then $\neg \mathbf{c} \in \mathcal{C}$.

We also write $\alpha \not> \beta$ for $\neg(\alpha > \beta)$, where $\alpha, \beta \in \mathcal{C} \cup \mathcal{L}$. The formulas $\mathbf{c}$ built where each α in (1) is from $\mathcal{L}$ are called *right-nested counterfactuals*, and those where each β is from $\mathcal{L}$ *left-nested counterfactuals*; all other formulas are *mixed counterfactuals*. The *nesting depth* of $\mathbf{c}$ is the maximal number of $>$-nodes on any path in the formula tree for $\mathbf{c}$ (whose leaves are the formulas from $\mathcal{L}$) minus one, and the *negation depth* the maximal number of negation signs. Here, standard counterfactuals $p > q$ are seen as nested counterfactuals with nesting depth 0. Nested counterfactuals with negation depth 0 are called *positive*.

We now give a precise semantics to nested counterfactuals. For this purpose, we need to extend the concept of a possible world from propositional formulas to nested counterfactuals (which may appear in the premise of a statement). In this course, we have to refer to the truth value of a counterfactual that is nested into the premise of a counterfactual statement. This gives rise to a mutual recursive definition of the possible worlds of nested counterfactuals and the truth values of nested counterfactuals along the formula structure.

Assume for the moment that the truth value of a nested counterfactual $\mathbf{c}$ over a knowledge base S is defined; we write $S \models \mathbf{c}$ iff $\mathbf{c}$ has value *true* over S and $S \not\models \mathbf{c}$ otherwise, i.e., iff $\mathbf{c}$ has value *false* over S.

Then, following the "possible worlds approach," the result of revising the knowledge base S with $\mathbf{c} \in \mathcal{C}$, is defined as follows:

$$\mathcal{F}(\mathbf{c}, S) = \{ T \subseteq S : T \models \mathbf{c},\ T \subset U \subseteq S \Rightarrow U \not\models \mathbf{c} \}$$

Using this definition, the truth value of a nested counterfactual $\mathbf{c} \in \mathcal{C}$ over a knowledge-base S is recursively defined as follows.

- If $\mathbf{c} = \alpha > \beta$, where $\alpha, \beta \in \mathcal{C} \cup \mathcal{L}$, then $\mathbf{c}$ has value *true* if $T \models \beta$ for every $T \in \mathcal{F}(\alpha, S)$, and value *false otherwise*;
- if $\mathbf{c} = \neg\mathbf{c}'$, then the truth value of $\mathbf{c}$ is opposite to the truth value of $\mathbf{c}'$.

It is easy to design a procedure which, given a knowledge base S and a nested counterfactual $\mathbf{c}$, decides whether $S \models \mathbf{c}$ in polynomial space. Indeed, an algorithm which for deciding $S \models \alpha > \beta$ cycles through all subsets $T \subseteq S$ and checks $T \in \mathcal{F}(\alpha, S)$ and $T \models \beta$ using recursion is straightforward. Hence, PSPACE is an upper bound for deciding $S \models \mathbf{c}$. Depending on the structure of $\mathbf{c}$, lower upper bounds will be identified in the following sections.

With each right-nested counterfactual $\mathbf{c}$ we can associate the unique sequence $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$ of nested counterfactuals such that $\mathbf{s}_1$ is of the form $p > q$, each $\mathbf{s}_i$, $2 \leq i \leq n$ results from $\mathbf{s}_{i-1}$ by applying rule (1) or (2), and $\mathbf{c} = \mathbf{c}_n$, which is referred to as the *structural sequence* of $\mathbf{c}$; each $\mathbf{s}_i$ is said to *occur* in $\mathbf{c}$.

Example 4: Let $\mathbf{c} = p_1 > (p_2 \not> q)$. The structural sequence of $\mathbf{c}$ is $p_2 > q$, $p_2 \not> q$, $p_1 > (p_2 \not> q)$. Notice that $\mathbf{c}$ has nesting depth 1 and also negation-depth 1. □

If we have a right-nested counterfactual $\mathbf{c}$, then the knowledge bases that are relevant for evaluating the subcounterfactual $\mathbf{s}_i$ from the structural sequence $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$ of $\mathbf{c}$ are determined by S and the premises of $\mathbf{s}_{i+1}, \ldots, \mathbf{s}_n$. We refer to these knowledge bases as the *context* of $\mathbf{s}_i$, which is formally defined as follows:

$$
\begin{aligned}
Cn(\mathbf{s}_n, \mathbf{c}, S) &= \{S\}, \qquad \text{and for } 2 \leq i \leq n, \\
Cn(\mathbf{s}_{i-1}, \mathbf{c}, S) &= \begin{cases} \bigcup_{T \in Cn(\mathbf{s}_i, \mathbf{c}, S)} \mathcal{F}(p, T), & \text{if } \mathbf{s}_i = p > \mathbf{s}_{i-1}; \\ Cn(\mathbf{s}_i, \mathbf{c}, S), & \text{if } \mathbf{s}_i = \neg\mathbf{s}_{i-1}. \end{cases}
\end{aligned}
$$

The following proposition is immediate from the definition, and will be used later.

Proposition 2.2 *Let $\mathbf{c} = p > \mathbf{c}'$, $\mathbf{c}' \in \mathcal{C}$, be a right-nested counterfactual. Then, $S \models \mathbf{c}$ iff $S' \models \mathbf{c}'$ for every $S' \in Cn(\mathbf{c}', \mathbf{c}, S)$.*

3 Positive right-nested counterfactuals

We start with the evaluation of nested counterfactuals where no negation occurs in the nesting. In the spirit of [25] (cf. also [34, 20]), we describe a nondeterministic algorithm for proving $S \not\models p_1 > (p_2 > \cdots (p_n > q) \cdots)$ in Figure 1. It has the following property.

ND-ALGORITHM PRNCF$(S, p_1, \ldots, p_n, q)$

input: finite $S \subseteq \mathcal{L}$, formulas $p_1, \ldots, p_n, q \in \mathcal{L}$

output: "no" iff $S \not\models p_1 > (p_2 > \cdots (p_n > q) \cdots)$

```
begin
W_1 := S;
Guess W_2, ..., W_{n+1} ⊆ S ∪ {p_1, ..., p_n} such that
  W_i ⊆ W_{i-1} ∪ {p_{i-1}} and p_{i-1} ∈ W_i, for 2 ≤ i ≤ n+1;
for i = 2 to n+1 do begin
    if (W_i ⊨ ⊥) then stop;
    for each r ∈ (W_{i-1} − W_i) do
        if (W_i ∪ {r} ⊭ ⊥) then stop;
end;
if (W_{n+1} ⊭ q) then output "no";
end.
```

Figure 1: Algorithm for evaluating positive right-nested counterfactuals

Proposition 3.1 **PRNCF**$(S, p_1, \ldots, p_n, q)$ *outputs "no" if and only if* $S \not\models p_1 > (p_2 > \cdots (p_n > q) \cdots)$.

Proof. Let $\mathbf{c}_1, \ldots, \mathbf{c}_n$ be the structural sequence of $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$. Notice that $\mathbf{c}_1 = p_n > q$ and $\mathbf{c}_i = p_{n-i+1} > \mathbf{c}_{i-1}$, for each $i = 2, \ldots, n$.

Simple induction on n shows that $S \not\models \mathbf{c}$ iff there exist $W \in Cn(\mathbf{c}_1, \mathbf{c}, S)$ and $W' \in \mathcal{F}(p_n, W)$ such that $W' \not\models q$. It follows from the definition of Cn that W is in the context of $\mathbf{c}_1$ iff there exist $W_1, W_2, \ldots, W_n = W$ such that $W_i \in Cn(\mathbf{c}_{n-i+1}, \mathbf{c}, S)$, i.e., $W_{i+1} \in \mathcal{F}(p_i, W_i)$ for $i < n$. Now consider **PRNCF**. It can be easily seen that the computation does not stop in the **for**-loop iff $W_{n+1} \in \mathcal{F}(p_n, W_n)$, where $W_n \in Cn(\mathbf{c}_1, \mathbf{c}, S)$. Thus **PRNCF** correctly outputs "no". Conversely, **PRNCF** outputs "no" in some computation if $S \not\models \mathbf{c}$. □

Corollary 3.2 *Given a knowledge base S and a positive right-nested counterfactual* $\mathbf{c}$*, deciding if $S \models \mathbf{c}$ is in* Π_2^P.

From Proposition 2.1, we thus obtain the following.

Theorem 3.3 *Given a knowledge base S and a positive right-nested counterfactual* $\mathbf{c}$*, deciding if $S \models \mathbf{c}$ is* Π_2^P*-complete.*

This result applies to knowledge bases that may contain arbitrary propositional formulas. In practice, knowledge bases are often sets of Horn clauses, i.e. disjunctions of literals of which at most one is an atom. It is well-known that deciding $S \models p$ is polynomial if p and all $q \in S$ are Horn clauses.

Theorem 3.4 *Let S be a knowledge base and $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$ such that every $p \in S$, $p_1, \ldots, p_n$ and q are Horn clauses. Then, deciding whether $S \models \mathbf{c}$ is* coNP*-complete.*

Proof. If p is a Horn clause, then every $T \in \mathcal{F}(p, S)$ is a set of Horn clauses. It follows that each classical inference test in **PRNCF** can be done in polynomial time, and hence **PRNCF** can be reformulated as an NP-algorithm. Consequently, deciding if $S \models \mathbf{c}$ is in coNP. Hardness for coNP follows from the coNP-hardness result for the case $\mathbf{c} = p_1 > q$ in [9]; from the proof of the result in the full paper (Theorem 8.5), this holds even if p_1 and q are literals. $\Box$

3.1 Priorities

A refinement of the operator $\mathcal{F}(p, S)$ by the introduction of priorities is considered in [11, 23]. Informally, in [11] a natural number $i \geq 0$ is attached to each formula with the intention that the lower the number, the higher the priority. Denote for any knowledge base S by S^i the formulas in S of priority $j \leq i$. The possible worlds for p from S are then the subsets $T \subset S$ consistent with p such that for each $i \geq 0$, T^i is maximal with respect to inclusion among these sets, i.e.,

$$W(p, S) = \{T \subseteq S : T \not\models \neg p \text{ and for all } i = 0, \ldots, n, \quad T^i \subset U \subseteq S^i \Rightarrow U \models \neg p\},$$

where n is the biggest number attached to a formula in S.[2] The generalization to iterated knowledge base updates is obvious. We write $\models^+$ instead of $\models$ if priorities are used for the knowledge base update.

It is easy to see that deciding $S \models^+ p > q$ is like deciding $S \models p > q$ in Π_2^P, as checking $T \in \mathcal{F}(p, S)$ is still in Δ_2^P. Furthermore, the upper complexity bound is preserved for evaluating nested counterfactuals, even in the Horn case. Hence,

Theorem 3.5 *Given a knowledge base S and a counterfactual $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$, deciding if $S \models^+ \mathbf{c}$ is Π_2^P-complete. If all formulas are Horn clauses, then deciding whether $S \models^+ \mathbf{c}$ is* coNP*-complete.*

[2] Fagin et al. allow that the same formula may have attached different numbers, which enables to define access priorities for different users; this has no effect on the complexity. Furthermore, they consider the formulas of S^0 as integrity constraints which should be preserved in each possible world W, i.e, $S^0 \subseteq W^0$; this does not change complexity in general.

On the contrary, priorities may even lead to a complexity decrease, as the following result tells.

Theorem 3.6 *Assume that the priority numbers attached to formulas are pairwise distinct. Then, given S and a counterfactual $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$, deciding whether $S \models^+ \mathbf{c}$ is Δ_2^P-complete. Hardness for Δ_2^P holds even for $n = 1$.*

Proof. Note that in this case, S is totally ordered according to priorities and that $\mathcal{F}(p_1, S)$ contains a single knowledge base, which can be constructed in polynomial time with an NP oracle. Thus from Proposition 2.2 it follows that deciding $S \models^+ \mathbf{c}$ is in Δ_2^P.

Δ_2^P-hardness for deciding $S \models p_1 > q$ is shown by a straightforward reduction from the Δ_2^P-complete problem MAXSAT, which is as follows [35, 21]: Given a satisfiable Boolean formula E on atoms $\underline{a} = a_1, \ldots, a_n$, decide whether $\varphi(a_n) = true$ for the lexicographically maximum truth value assignment φ to $\underline{a}$ that satisfies E. Here, the assignment φ is lexicographically greater than the assignment ψ iff $\varphi(a_i) = true$, $\psi(a_i) = false$ for the least i such that $\varphi(a_i) \neq \psi(a_i)$. Now choose priorities such that i is attached to a_i, for all $i = 1, \ldots, n$ and simply let $S = \{\, \underline{a} \,\}$, $p_1 = E$, and $q = a_n$. $\square$

A further complexity decrease can be accomplished by restricting knowledge bases to Horn clauses. In fact, the following important and optimistic result is easily proved (cf. proof of Theorem 3.6).

Theorem 3.7 *Assume that the priority numbers attached to formulas are pairwise distinct, and that all formulas are Horn. Given S and $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$, deciding whether $S \models^+ \mathbf{c}$ is possible in polynomial time.*

4 Right-nested counterfactuals with negation

We consider now evaluation of counterfactuals with negation in the nesting. It appears that negation has a drastic effect on the complexity of evaluating nested counterfactuals. Negating each counterfactual occurring in $p_1 > (p_2 > \cdots (p_k > q) \cdots)$ leads to Σ_{k+1}^P-hardness if k is a constant and to PSPACE-hardness if k is not bounded. We show this in the sequel by a polynomial time transformation of deciding the validity of QBFs into deciding the validity of nested counterfactuals.

Let Φ be a QBF with $k \geq 1$ quantifier alternations of the form

$$(Q_1 \underline{a}_1)(Q_2 \underline{a}_2) \cdots (Q_{k+1} \underline{a}_{k+1}) E(\underline{a}_1, \ldots, \underline{a}_{k+1}),$$

,i.e. $Q_i \neq Q_{i-1}$, for $i > 1$, where $\underline{a}_i = a_{i,1}, \ldots, a_{i,n_i}$ is a list of $n_i \geq 1$ variables and $(Q_i \underline{a}_i)$ stands for $Q_i a_{i,1} \cdots Q_i a_{i,n_i}$, $1 \leq i \leq k+1$.

We construct a nested counterfactual $\mathbf{c}(\Phi)$ and a theory $S(\Phi)$ as follows. Let c be a new variable, and let $\underline{b}_1, \ldots, \underline{b}_k$ be groups of new variables, where $\underline{b}_i = b_{i,1}, \ldots, b_{i,n_i}$. First define formulas $p_1, \ldots, p_k$ and q. For $i = 1, \ldots, k$ let

$$p_i = \begin{cases} \underline{a}_i \equiv \neg \underline{b}_i, & i < k; \\ p(\Phi), & i = k, \end{cases}$$

where $p(\Phi) = [\underline{a}_1 \cdots \underline{a}_k \equiv \neg \underline{b}_1 \cdots \underline{b}_k] \wedge (E(\Phi) \vee c) \wedge (c \Rightarrow \underline{a}_{k+1})$ and $E(\Phi)$ is, depending on Q_{k+1}, as follows:

$$E(\Phi) = \begin{cases} \neg E, & \text{if } Q_{k+1} = \forall; \\ E, & \text{if } Q_{k+1} = \exists. \end{cases}$$

Furthermore, let $q = \neg c$. Now $S(\Phi)$ and $\mathbf{c}(\Phi) = \mathbf{c}_k(\Phi)$ are defined as follows:

$$\begin{aligned} S(\Phi) &= \{\underline{a}_1, \underline{b}_1, \ldots, \underline{a}_k, \underline{b}_k, \neg \underline{a}_{k+1}, \neg c\}; \\ \mathbf{c}_1(\Phi) &= \begin{cases} p_k \not> q, & \text{if } Q_{k+1} = \forall; \\ p_k > q, & \text{if } Q_{k+1} = \exists, \end{cases} \qquad \text{and for } k \geq i > 1, \\ \mathbf{c}_i(\Phi) &= \begin{cases} p_{k-i} > \mathbf{c}_{i-1}(\Phi), & \text{if } Q_{k-i} = \forall; \\ p_{k-i} \not> \neg \mathbf{c}_{i-1}(\Phi), & \text{if } Q_{k-i} = \exists. \end{cases} \end{aligned}$$

Note that for every Φ, $S(\Phi)$ and $\mathbf{c}(\Phi)$ can be computed in polynomial time.

Let us describe the intuition behind $S(\Phi)$ and $\mathbf{c}(\Phi)$. The $\mathbf{c}_i(\Phi)$'s represent the (possibly negated) subformulas of Φ on the quantified variable groups $\underline{a}_{k-i+1}, \ldots, \underline{a}_{k+1}$ where all remaining variables (i.e, those in $\underline{a}_1, \ldots, \underline{a}_{k-i}$) are replaced in E with $\top$ or $\perp$ according to a truth value assignment. Every such truth value assignment is encoded by a context knowledge base C of $\mathbf{c}_i(\Phi)$. The value of variable $a_{i,j}$ is *true* if it occurs literally in C and otherwise *false* (in this case, $b_{i,j}$ is in C). Testing if $C \models \mathbf{c}_i(\Phi)$ implements a test if Φ is valid for the encoded truth value assignment and quantification of $\underline{a}_{k-i+1}, \ldots, \underline{a}_{k+1}$ with $Q_{k-i+1}, \ldots, Q_{k+1}$. Thus checking $S \models \mathbf{c}_k(\Phi)$, i.e. $S \models \mathbf{c}(\Phi)$, implements a test if the formula Φ is valid.

Example 5: Consider $\Phi = \forall a_1 \exists a_{2,1} \exists a_{2,2} \forall a_3 \, (a_1 \wedge \neg a_{2,1} \Rightarrow a_{2,2} \vee a_3)$. The formula Φ is rewritten as $(\forall \underline{a}_1)(\exists \underline{a}_2)(\forall \underline{a}_3)E$, where $\underline{a}_1 = a_1$, $\underline{a}_2 = a_{2,1}, a_{2,2}$, $\underline{a}_3 = a_3$, and $E = (a_1 \wedge \neg a_{2,1} \Rightarrow a_{2,2} \vee a_3)$. Applying the transformation ($k+2$, $Q_1 = \forall$), we get

$$\begin{aligned} S(\Phi) &= \{a_1, b_1, a_{2,1}, a_{2,2}, b_{2,1}, b_{2,2}, \neg a_3, \neg c\}, \\ p_1 &= a_1 \equiv \neg b_1, \\ p_2 &= [a_1 \equiv \neg b_1] \wedge [a_{2,1} \equiv \neg b_{2,1}] \wedge [a_{2,2} \equiv \neg b_{2,2}] \wedge (\neg E \vee c) \wedge (c \Rightarrow a_3), \\ q &= \neg c, \\ \mathbf{c}_1(\Phi) &= p_2 \not> q, \\ \mathbf{c}_2(\Phi) &= p_1 > (p_2 \not> q), \\ \mathbf{c}(\Phi) &= \mathbf{c}_2(\Phi). \end{aligned}$$

The reader may verify that Φ is valid and that $S(\Phi) \models \mathbf{c}(\Phi)$. $\Box$

We show now that checking the validity of $\mathbf{c}(\Phi)$ over $S(\Phi)$ truly resembles checking the validity of Φ.

Theorem 4.1 *$S(\Phi) \models \mathbf{c}(\Phi)$ if and only if Φ is valid.*

Before going into the proof of this theorem, we note some useful lemmata.

Lemma 4.2 *For $k = 1$ and $Q_1 = \forall$, $S(\Phi) \models \mathbf{c}(\Phi)$ if and only if $\Phi \in \mathrm{QBF}_{2,\forall}$.*

Proof. $S(\Phi)$ and $\mathbf{c}(\Phi)$ are identical to S and $p > q$ in the construction (2) upon renaming of literals and replacing p with the equivalent formula p_1. $\Box$

Lemma 4.3 *Let S and S' be knowledge bases and let $\mathbf{c}$ be a right-nested counterfactual, such that $S' \not\models \bot$ and no atom occurring in any $p \in S'$ occurs in any $q \in S$ or in $\mathbf{c}$. Then,*

- *(i)* *$S \models \mathbf{c}$ iff $S \cup S' \models \mathbf{c}$, and*
- *(ii)* *for each $\mathbf{s}_i$ from the structural sequence $\mathbf{s}_1, \ldots, \mathbf{s}_k$ of $\mathbf{c}$, it holds that $Cn(\mathbf{s}_i, \mathbf{c}, S \cup S') = \{T \cup S' : T \in Cn(\mathbf{s}_i, \mathbf{c}, S)\}$. (Note that $T \cap S' = \emptyset$ for each $T \in Cn(\mathbf{s}_i, \mathbf{c}, S)$.)*

Proof. Property (ii) is readily shown by induction on k. We make use of the following helpful observations, which are easy to see:

Fact 1 Let p and $S = S' \cup S''$ be such that $S'' \not\models \bot$ and $S' \cup \{p\}$ and S'' have no common atoms; then, $\mathcal{F}(p, S) = \{W \cup S'' : W \in \mathcal{F}(p, S')\}$.

Fact 2 : For a right-nested counterfactual $\mathbf{c}$ with structural sequence $\mathbf{s}_1, \ldots, \mathbf{s}_n$, $Cn(\mathbf{s}_n, \mathbf{c}, S) = \{S\}$ and $Cn(\mathbf{s}_n, \mathbf{c}, S \cup S') = \{S \cup S'\}$.

(Basis) $k = 1$: Property (ii) is obviously true by Fact 2.
(Induction) Assume that (ii) is true for $k \geq 1$, and consider $k + 1$. If $\mathbf{s}_{k+1} = \neg \mathbf{s}_k$, then (ii) is true by the induction hypothesis. Thus assume that $\mathbf{s}_{k+1} = p > \mathbf{s}_k$. For $i = k + 1$, property (ii) is clearly true by Fact 2. Thus, let $i \leq k$ and consider the right-nested counterfactual $\mathbf{c}'$ with structural sequence $\mathbf{s}_1, \ldots, \mathbf{s}_k$. As easily checked, for any knowledge base U, it holds that

$$Cn(\mathbf{s}_j, \mathbf{c}, U) = \bigcup_{T \in \mathcal{F}(p,U)} Cn(\mathbf{s}_j, \mathbf{c}', T) \qquad (*)$$

for $j = k, \ldots, 1$. Thus, it follows

$$\begin{aligned}
Cn(\mathbf{s}_i, \mathbf{c}, S \cup S') &= \textstyle\bigcup_{T \in \mathcal{F}(p, S \cup S')} Cn(\mathbf{s}_i, \mathbf{c}', T) && \text{(by (*))} \\
&= \textstyle\bigcup_{W \in \mathcal{F}(p,S)} Cn(\mathbf{s}_i, \mathbf{c}', W \cup S') && \text{(by Fact 1)} \\
&= \textstyle\bigcup_{W \in \mathcal{F}(p,S)} \{\, T \cup S' : T \in Cn(\mathbf{s}_i, \mathbf{c}', W) \,\} && \text{(by induction hyp.)} \\
&= \{\, T \cup S' : T \in Cn(\mathbf{s}_i, \mathbf{c}, S) \,\} && \text{(by definition)}
\end{aligned}$$

It follows that the claimed statement is true for $k+1$.

This completes the proof of (ii). Property (i) is an immediate consequence of (ii) and Proposition 2.2. $\square$

Lemma 4.4 *Let S be a knowledge base and let $\mathbf{s}_1, \ldots, \mathbf{s}_k$ be the structural sequence of $\mathbf{c}$, and denote by $\mathbf{c}[\mathbf{c}']$ the counterfactual obtained if $\mathbf{s}_1$ is replaced in $\mathbf{c}$ by the counterfactual $\mathbf{c}'$. If for all $T \in Cn(\mathbf{s}_1, \mathbf{c}, S)$ it holds that $T \models \mathbf{s}_1$ iff $T \models \mathbf{c}'$, then $S \models \mathbf{c}$ iff $S \models \mathbf{c}[\mathbf{c}']$.*

Proof. Straightforward by induction on k (use Proposition 2.2). $\square$

Proof of Theorem 4.1 The proof is by induction on the number $k \geq 1$ of quantifier alternations of Φ.

(Basis) $k = 1$: By Lemma 4.2, it remains to consider the case where $\Phi = (\exists \underline{a}_1)(\forall \underline{a}_2)E$. Since $\Phi \equiv \neg\Phi'$ for $\Phi' = (\forall \underline{a}_1)(\exists \underline{a}_2)\neg E$, Φ is valid iff $S(\Phi') \not\models \mathbf{c}(\Phi')$. Since $p(\Phi) \equiv p(\Phi')$ and $S(\Phi) = S(\Phi')$, it follows that Φ is valid iff $S(\Phi) \not\models p(\Phi) > q$. However, this is equivalent to $S(\Phi) \models \mathbf{c}(\Phi)$; the statement holds.

(Induction). Assume the statement holds for $k \geq 1$. Consider $k+1$, i.e. $\Phi = (Q_1\underline{a}_1) \cdots (Q_{k+2}\underline{a}_{k+2})E$, which has $k+1$ quantifier alternations. There are two cases for Q_1.

Case 1: $Q_1 = \forall$. In this case, Φ is valid iff for every truth value assignment φ to $\underline{a}_1$, the formula $\Phi_\varphi = (\exists \underline{a}_2) \cdots (Q_{k+2}\underline{a}_{k+2})E_\varphi$ is valid. Note that Φ_φ has k quantifier alternations. Applying the hypothesis for k, Φ_φ is valid iff $S \models \mathbf{c}$, where $S = S(\Phi_\varphi)$ and $\mathbf{c} = \mathbf{c}(\Phi_\varphi)$.

Let $\mathbf{s}_1$ be the first counterfactual in the structural sequence of $\mathbf{c}$. Clearly, $\mathbf{s}_1 = p(\Phi_\varphi) > q$, where

$$p(\Phi_\varphi) = [\underline{a}_2 \cdots \underline{a}_{k+1} \equiv \neg\, \underline{b}_2 \cdots \underline{b}_{k+1}] \wedge (E(\Phi_\varphi) \vee c) \wedge (c \Rightarrow \underline{a}_{k+2})$$

and $q = \neg c$. Now let S_φ be the following set of propositional formulas:

$$\begin{aligned}
S_\varphi \;\; = \;\; & \{a_{1,j} : \varphi(a_{1,j}) = \mathit{true}, 1 \leq j \leq n_1\} \cup \\
& \{b_{1,j} : \varphi(a_{1,j}) = \mathit{false}, 1 \leq j \leq n_1\} \cup \{\underline{a}_1 \equiv \neg\, \underline{b}_1\}.
\end{aligned}$$

Note that $S_\varphi \not\models \bot$, and that no variable occurring in S_φ occurs in S or in $\mathbf{c}$. Hence by Lemma 4.3, $S \models \mathbf{c}$ iff $S \cup S_\varphi \models \mathbf{c}$, and $T \in Cn(\mathbf{s}_1, \mathbf{c}, S)$ iff $T \cup S_\varphi \in Cn(\mathbf{c}_1, \mathbf{c}, S \cup S_\varphi)$.

Let p' denote the conjunction of $[\underline{a}_1 \equiv \neg\, \underline{b}_1]$ and the formula that results from $p(\Phi_\varphi)$ if the substitution of $\underline{a}_1$ by $\varphi(\underline{a}_1)$ is undone, i.e.,

$$p' \;=\; [\underline{a}_1 \cdots \underline{a}_{k+1} \equiv \neg\, \underline{b}_1 \cdots \underline{b}_{k+1}] \wedge (E(\Phi) \vee c) \wedge (c \Rightarrow \underline{a}_{k+2}).$$

We claim that for every $T \in Cn(\mathbf{s}_1, \mathbf{c}, S \cup S_\varphi)$, $T \models \mathbf{s}_1$ iff $T \models p' > q$. (Recall that $S = S(\Phi_\varphi)$ and $\mathbf{c} = \mathbf{c}(\Phi)$.) It follows from Lemma 4.4 that $S \cup S_\varphi \models \mathbf{c}$ (i.e., $S \models \mathbf{c}$) iff $S \cup S_\varphi \models \mathbf{c}[p' > q]$; as a consequence, $S \models \mathbf{c}$ iff $S(\Phi) \cup S_\varphi \models \mathbf{c}[p' > q]$.

To prove this claim, assume that $T \not\models \mathbf{s}_1$, where $\mathbf{s}_1 = p(\Phi_\varphi) > \neg c$. This implies that for some $W \in \mathcal{F}(p(\Phi_\varphi), T)$, it holds that $W \not\models \neg c$. Since $S_\varphi \subseteq T$, and since S_φ is consistent and has no variable in common with $p(F_\varphi)$ or any formula in $T - S_\varphi$, maximality of W with respect to set inclusion implies $S_\varphi \subseteq W$. Since

$$S_\varphi \models [\underline{a}_1 \equiv \neg\, \underline{b}_1] \wedge [\underline{a}_1 \equiv \varphi(a_{1,1}) \cdots \varphi(a_{1,n_1})], \tag{3}$$

we infer $W \models p'$. Thus it follows that there exists $W' \in \mathcal{F}(p', T)$ such that $W' \not\models \neg c$. It follows that $T \not\models p' > q$. Conversely, assume $T \not\models p' > q$. Thus there is some $W \in \mathcal{F}(p', T)$ such that $W \not\models \neg c$, i.e., W has a model in which c has value *true*. Then, also $W \cup S_\varphi$ is consistent: p' is satisfied for any truth value assignment to $\underline{a}_1, \underline{b}_1, \ldots, \underline{a}_{k+1}, \underline{b}_{k+1}$ that satisfies $[\underline{a}_1 \cdots \underline{a}_{k+1} \equiv \neg\, \underline{b}_1 \cdots \underline{b}_{k+1}]$ if *true* is assigned to c and the atoms $\underline{a}_{k+2}$. Since S_φ is consistent, $[\underline{a}_1 \equiv \neg\, \underline{b}_1] \in S_\varphi$, and since the variables $\underline{a}_1, \underline{b}_1$, which are all those occurring in S_φ, do not appear in any formula from $T - S_\varphi$, maximality of W with respect to set inclusion implies that $S_\varphi \subseteq W$. From (3) we infer that $W \models p(\Phi_\varphi)$. Thus it follows that there exists a $W' \in \mathcal{F}(p(\Phi_\varphi), T)$ such that $W' \not\models \neg c$. Consequently, $T \not\models p(\Phi_\varphi) > q$.

Thus $S(\Phi_\varphi) \models \mathbf{c}$ and, equivalently, Φ_φ is valid if and only if $S(\Phi) \cup S_\varphi \models \mathbf{c}[p' > q]$. However, $p' = p(\Phi)$, and it is easily verified that $\mathbf{c}[p' > q]$ is $\mathbf{c}_k(\Phi)$. It follows that Φ is valid iff $S(\Phi_\varphi) \cup S_\varphi \models \mathbf{c}_k(\Phi)$, for every truth value assignment φ to $\underline{a}_1$. Closer inspection shows that the context of $\mathbf{c}_k(\Phi)$ from $\mathbf{c}_{k+1}(\Phi)$ with respect to $S(\Phi)$ are just these knowledge bases, i.e.,

$$Cn(\mathbf{c}_k(\Phi), \mathbf{c}_{k+1}(\Phi), S(\Phi)) \;=\; \{S(\Phi_\varphi) \cup S_\varphi : \varphi \text{ a truth value assignment to } \underline{a}_1\}.$$

However, since $\mathbf{c}_{k+1}(\Phi) = \mathbf{c}(\Phi)$, we get from Proposition 2.2 that Φ is valid iff $S(\Phi) \models \mathbf{c}(\Phi)$; thus, the statement holds for $k+1$.

Case 2: $Q_1 = \exists$. Thus Φ is equivalent to $\neg\Phi'$ where $\Phi' = (\forall \underline{u}_1)(\exists \underline{u}_2) \cdots (Q'_{k+2}\underline{u}_{k+2})\neg E$. By Case 1, Φ' is valid iff $S(\Phi') \models \mathbf{c}(\Phi')$ is true.

We verify the following:

Fact 3 For each knowledge base U and $j = 1, \ldots, k+1$, it holds that $U \models \mathbf{c}_j(\Phi)$ iff $U \models \neg\mathbf{c}_j(\Phi')$.

We show this by induction on j. Note that $p_j = p'_j$, for each $j \le k$, and that p_{k+1} is logically equivalent to p'_{k+1}. (Basis) For $j = 1$, the statement clearly holds, since $U \models p_{k+1} \not> q$ iff $U \models \neg(p'_{k+1} > q)$ and $U \models p_{k+1} > q$ iff $U \models \neg(p'_k \not\models q)$. (Induction) Assume the statement holds for $j \ge 1$, and consider $j + 1$. Since $p_{k-j+2} = p'_{k-j+2}$, from the induction hypothesis it follows that $U \models p_{k-j+2} > \mathbf{c}_j(\Phi)$ iff $U \models \neg(p'_{k-j+2} \not> \neg\mathbf{c}_j(\Phi'))$ and $U \models p_{k-j+2} \not> \neg\mathbf{c}_j(\Phi)$ iff $U \models \neg(p'_{k-j+2} > \mathbf{c}_j(\Phi'))$. Thus, $U \models \mathbf{c}_{j+1}(\Phi)$ iff $U \models \neg\mathbf{c}_{j+1}(\Phi')$; the statement holds for $j + 1$. This is proves Fact 3.

Fact 3 implies that $U \models \mathbf{c}(\Phi)$ iff $U \models \neg\mathbf{c}(\Phi')$. Consequently, Φ is valid iff Φ' is not valid iff $S(\Phi') \models \neg\mathbf{c}(\Phi')$ iff $S(\Phi) \models \neg\mathbf{c}(\Phi')$ iff $S(\Phi) \models \mathbf{c}(\Phi)$. Thus the statement also holds in Case 2. $\diamond$

The proof of Theorem 4.1 is complete. $\Box$

We obtain the main results of this section. We use the following lemma, whose proof is analogous to the proof of Theorem 3.3.

Lemma 4.5 *Let $\mathbf{c}$ be a right-nested counterfactual of the form $p_1 > (p_2 > (\cdots > \mathbf{c}')\cdots)$ where $\mathbf{c}' \in \mathcal{C}'$ for some $\mathcal{C}' \subseteq \mathcal{C}$ in which evaluation of a counterfactual on a knowledge base is in Δ_k^P, for some $k \ge 1$. Then, given any knowledge base S, deciding whether $S \models \mathbf{c}$ is in Π_k^P.*

Theorem 4.6 *Let S be a knowledge base and $\mathbf{c}$ be a right-nested counterfactual, whose negation-depth is bounded by a constant $k \ge 0$. Then, deciding whether $S \models \mathbf{c}$ is*

(i) Σ_{k+1}^P-complete, if $\mathbf{c}$ is of the form $p_1 \not> \alpha$, and

(ii) Π_{k+2}^P-complete otherwise.

Hardness holds even if steps (1) and (2) in the construction of $\mathbf{c}$ alternate, i.e. $\mathbf{c}$ is of the form (i) $p_1 \not> (p_2 \not> \cdots (p_k \not> q)\cdots)$, and (ii) $p_1 > (p_2 \not> (p_3 \not> \cdots))$, respectively.

Proof. Membership in Σ_{k+1}^P is shown by induction on the length n of the structural sequence $\mathbf{s}_1, \ldots, \mathbf{s}_n$ of $\mathbf{c}$ and on k.
(Basis) For $n = 1$, only $k = 0$ is possible, and membership in Π_2^P follows by Proposition 2.1.
(Induction) Assume that $n > 1$. If case (i) applies, then by the induction hypothesis, $S \models p_1 > \alpha$ lies in $\Pi_{(k-1)+2}^P$; consequently, deciding $S \models \mathbf{c}$ is in Σ_{k+1}^P. Otherwise, if case (ii) applies, $\mathbf{c} = p_1 > (p_2 > (\cdots > \mathbf{c}')\cdots)$ where $\mathbf{c}'$ is of the form $p \not> \beta$. By the induction hypothesis, deciding for any knowledge base T whether $T \models \mathbf{c}'$ is in Σ_{k+1}^P. Thus, it follows from Lemma 4.5 that deciding whether $S \models \mathbf{c}$ is in Π_{k+2}^P. This concludes the induction and the membership proof.

Σ^P_{k+1}-hardness for all $k \geq 1$ under the described restriction follows from Theorem 4.1, from which also Π^P_{k+2}-hardness follows. $\Box$

As shown in Section 2, PSPACE is an upper bound for evaluating a nested counterfactual on a knowledge base. Thus, by Theorem 4.1 we obtain the following.

Theorem 4.7 *Given a knowledge base S and a right-nested counterfactual $\mathbf{c} \in \mathcal{C}$, deciding if $S \models \mathbf{c}$ is* PSPACE*-complete. Hardness for* PSPACE *holds even if $\mathbf{c}$ is of the form $p_1 \not> (p_2 \not> \cdots (p_n \not> q) \cdots)$.*

We conclude this section with a remark on propositional formulas involving right-nested counterfactuals. Combining logical connectives and right-nested counterfactuals seems natural, and allows to connect statements on (iterated) revisions. For example, $(p > q) \vee (p > (r > \neg q))$ states that either after a revision with p, q follows from the knowledge base, or after revisions with p and r (in that order), $\neg q$ follows from the knowledge base. The semantics of formulas in the Boolean closure of the right-nested counterfactuals is clear, and by the results above, their evaluation is in PSPACE. However, for formulas that connect propositional formulas with counterfactuals, e.g., $p \vee (q > r)$, a proper formalization of the semantics is not straightforward in the framework from above. We leave this issue, which leads beyond the scope of this paper, for other investigations.

5 Left-nested counterfactuals

We consider in this section evaluation of counterfactuals nested in the premise, i.e. statements of the form $(p > q) > r$, and we address the complexity of iterated left-nestings and use of negation.

We start with positive left-nested counterfactuals of nesting depth 1. Consider the nondeterministic algorithm in Figure 2, where $\mathbf{CF}(p, q, S) = \mathit{true}$ iff $S \models p > q$. Thew following property is easily seen.

Proposition 5.1 **PLNCF1**(S, p, q, r) *outputs "no" if and only if $S \not\models (p > q) > r$.*

PLNCF1 has exponential worst case runtime even modulo the **CF** calls and $S' \models r$. An improvement to polynomial runtime seems hard to achieve. In particular, the exponential candidate space for T in the **for**-loop, which tests whether S' is closest to S such that $p > q$ holds, can most likely not be reduced efficiently to a small subset (cf. Lemma 5.3). This may be intuitively explained by nonmonotonicity of counterfactuals, i.e., $S \models p > q$ and $S \subseteq S'$ does not imply $S' \models p > q$ in general.

Example 6: Let $S = \{\neg a, a \Rightarrow b\}$, $S' = S \cup \{\neg b\}$ and $\mathbf{c} = a > (b \vee c)$. Then, $S \models \mathbf{c}$, but $S' \not\models \mathbf{c}$. Notice that $S' \cup \{c\} \models \mathbf{c}$, however. $\Box$

ND-ALGORITHM PLNCF1(S, p, q, r)

input: finite $S \subseteq \mathcal{L}$, formulas $p, q, r \in \mathcal{L}$
output: "no" iff $S \not\models (p > q) > r$

```
begin
Guess S' ⊆ S;
if ¬CF(S', p, q) or (S' ⊨ r) then stop
else
    for each T, s.t. S' ⊂ T ⊆ S, do
        if CF(T, p, q) then stop;
    output "no";
end.
```

Figure 2: Algorithm for evaluating counterfactuals $(p > q) > r$

We determine now the complexity of evaluating left-nested counterfactuals. First, we derive an upper bound for this problem.

Theorem 5.2 *Given S and $\mathbf{c} = (p{>}q){>}r$, deciding if $S \models \mathbf{c}$ is in Π_4^P.*

Proof. Let $S' \subseteq S$. It is easily seen that a proof for $S' \notin \mathcal{F}(p > q, S)$ can be given in *nondeterministic* polynomial time with an oracle for classical and counterfactual inference. As a Π_2^P oracle is suitable for that, deciding if $S' \notin \mathcal{F}(p > q, S)$ is in Σ_3^P. Since $S \not\models (p > q) > r$ iff $S' \not\models r$ for some $S' \in \mathcal{F}(p > q, S)$, it follows that deciding $S \not\models (p > q) > r$ is in Σ_4^P; hence the result. □

We will show that this upper bound is tight, as Π_4^P-hardness is a lower bound for this problem. The following lemma is useful in a proof of this result.

Lemma 5.3 *Let $S' \models p > q$ for a $S' \subseteq S$. Given S' and $p > q$, deciding whether $T \models p > q$ for any T such that $S' \subset T \subseteq S$, is Σ_3^P-hard.*

Proof. We transform deciding the validity of a QBF

$$\Phi = (\exists \underline{a}_1)(\forall \underline{a}_2)(\exists \underline{a}_3)\, E$$

into this problem, where $\underline{a}_i = a_{i,1}, \ldots, a_{i,n_i}$, for $i = 1, 2, 3$. Let c be a new atom and let $\underline{a}_1' = a_{1,1}', \ldots, a_{1,n_1}'$, $\underline{a}_1'' = a_{1,1}'', \ldots, a_{1,n_1}''$, and $\underline{a}_2' = a_{2,1}', \ldots, a_{2,n_2}'$ be lists of new atoms. Define

$$S' \;=\; \{\underline{a}_2, \underline{a}_2', c\}, \tag{4}$$

$$
\begin{aligned}
S &= \{\underline{a}_1, \underline{a}'_1, \underline{a}_2, \underline{a}'_2, c\}, \\
p &= \left[c \wedge \bigwedge_{i=1}^{n_1} (\neg a_{1,i} \wedge \neg a'_{1,i} \wedge a''_{1,i}) \right] \vee \\
&\quad \left[\bigwedge_{i=1}^{n_1} [(a_{1,i} \wedge a'_{1,i} \Rightarrow \neg c) \wedge (a_{1,i} \vee a'_{1,i} \Rightarrow a''_{1,i})] \;\wedge\; (\underline{a}_2 \equiv \neg\, \underline{a}'_2) \wedge (c \Rightarrow E) \right], \\
q &= c \wedge \bigwedge_{i=1}^{n_1} a''_{1,i}.
\end{aligned}
$$

Notice that S', S, p, and q are constructible in polynomial time. It is not hard to see that $S' \models p > q$, as $S' \cup \{p\}$ is consistent and logically implies q.

We claim that there exists T, $S' \subset T \subseteq S$, such that $T \models p > q$ iff Φ is valid.

($\Leftarrow$) Assume that Φ is valid. Let φ be any truth value assignment to $\underline{a}_1$ such that $\forall \underline{a}_2 \exists \underline{a}_3\, E_\varphi$ is valid. Let $T = S' \cup S_\varphi$, where

$$S_\varphi = \{a_{1,i} : \varphi(a_{1,i}) = \mathit{true},\ 1 \le i \le n_1\} \cup \{a'_{1,i} : \varphi(a_{1,i}) = \mathit{false},\ 1 \le i \le n_1\}.$$

We claim that $T \models p > q$. To show this, consider $\mathcal{F}(p, T)$. We have that $S' \cup \{p\} \in \mathcal{F}(p, T)$, as this is a maximal consistent subset of $T \cup \{p\}$. Note that $S' \cup \{p\} \models q$. Let $W \in \mathcal{F}(p, T)$ be different from $S' \cup \{p\}$. We note the following facts (**a**)–(**f**).

(**a**) *W must contain some atom from $\underline{a}_1$, $\underline{a}'_1$.* If not, it follows $W = S' \cup \{p\}$.

(**b**) *W logically implies the right disjunct of p.* W contains p and is consistent, and from (**a**) it follows that W is not consistent with the left disjunct of p.

(**c**) *$\{a_{2,i},\, a'_{2,i}\} \not\subseteq W$, for each $i = 1, \ldots, n_2$.* Assume that $\{a_{2,i},\, a'_{2,i}\} \in W$ for some i. Then, W does not logically imply the right disjunct of p, as there occurs the formula $a_{2,i} \equiv \neg\, a'_{2,i}$. This is a contradiction to (**b**).

(**d**) *For each $i = 1, \ldots, n_2$, $\{a_{2,i},\, a'_{2,i}\} \cap W \neq \emptyset$.* Assume that $\{a_{2,i},\, a'_{2,i}\} \cap W = \emptyset$ for some i. Then, $W \cup \{a_{2,i}\}$ or $W \cup \{a'_{2,i}\}$ is consistent. This is a contradiction to the maximality of W, i.e. would imply $W \notin \mathcal{F}(p, T)$.

(**e**) *For each $i = 1, \ldots, n_1$, if $a_{1,i} \in S_\varphi$ (resp. $a'_{1,i} \in S_\varphi$), then $a_{1,i} \in W$ (resp. $a'_{1,i} \in W$).* Assume that $a_{1,i} \in S_\varphi$ but $a_{1,i} \notin W$. Hence, neither $a_{1,i}$ nor $a'_{1,i}$ occur in W. However, W is consistent and the formula Φ is valid. Thus, regardless of the choice γ_i from $\{a_{2,j},\, a'_{2,j}\}$ taken in W for all $j = 1, \ldots, n_2$, the knowledge base $W \cup \{a_{1,i}\}$ is consistent. This is a contradiction to the maximality of W. The same argumentation applies for $a'_{1,i}$.

(**f**) *$c \in W$.* From the observations (**a**)–(**e**), we have that W resembles φ and a truth value assignment ψ to $\underline{a}_2$. Since Φ is valid, $(E_\varphi)_\psi$, i.e., the formula G such that $G = G'_\psi$ and $G' = E_\varphi$, is satisfiable. Hence, from (**a**)–(**e**) we conclude that $W \cup \{c\}$ is consistent. As $c \in T$, by maximality of W we have $c \in W$.

From (**a**)–(**f**) we conclude that W is of the form

$$S_\varphi \cup \{a_{2,i} : \psi(a_{2,i}) = \mathit{true},\ 1 \le i \le n_2\} \cup \{a'_{2,i} : \psi(a_{2,i}) = \mathit{false},\ 1 \le i \le n_2\} \cup \{p,\ c\}$$

for some truth value assignment ψ to $\underline{a}_2$. From (**b**) and (**e**) it follows that $W \models a''_{1,i}$, for each $i = 1, \ldots, n_1$; hence, $W \models q$. It follows that $T \models p > q$.

($\Rightarrow$) Assume that for some T, $S' \subset T \subseteq S$, we have $T \models p > q$. Let T be a maximal knowledge base of this property, i.e., $T \in \mathcal{F}(p > q, S)$. We show that for each $i = 1, \ldots, n_1$, it holds that $|T \cap \{a_{1,i},\, a'_{1,i}\}| = 1$.
(≤ 1) Assume that $\{a_{1,i},\, a'_{1,i}\} \subseteq T$ for some i. We observe that the knowledge base $\{a_{1,i},\, a'_{1,i},\, p\}$ is consistent and logically implies $\neg c$. Hence, there exists $W \in \mathcal{F}(p, T)$ such that $W \models \neg c$. Consequently, $W \not\models q$, which implies $T \not\models p > q$, contradiction.
(≥ 1) Let

$$\begin{aligned} S_{\underline{a}_1} &= T \cap \{\underline{a}_1, \underline{a}'_1\}, \\ S_\psi &= \{a_{2,i} : \psi(a_{2,i}) = \mathit{true},\ 1 \leq i \leq n_2\} \cup \{a'_{2,i} : \psi(a_{2,i}) = \mathit{false},\ 1 \leq i \leq n_2\}, \\ W_\psi^{\underline{a}_1} &= S_{\underline{a}_1} \cup S_\psi \cup \{p,\, c\}, \end{aligned}$$

for any truth value assignment ψ to $\underline{a}_2$. Notice that $S_{\underline{a}_1}$ is not empty. We claim that $W_\psi^{\underline{a}_1} \in \mathcal{F}(p, T)$. It is easily checked that $W_\psi^{\underline{a}_1} - \{p\} \subseteq T$ and that $W_\psi^{\underline{a}_1} - \{p\}$ is consistent with p. It remains to show maximality of $W_\psi^{\underline{a}_1}$. Since $W_\psi^{\underline{a}_1}$ contains $S_{\underline{a}_1}$ and hence some atom from $\underline{a}_1\underline{a}'_1$, it follows that $W_\psi^{\underline{a}_1}$ logically implies the right disjunct of p. Notice that $T - W_\psi^{\underline{a}_1}$ contains only atoms from $\underline{a}_2$ and $\underline{a}'_2$; as $S_\psi \subseteq W_\psi^{\underline{a}_1}$, every W' such that $W_\psi^{\underline{a}_1} \subseteq W' \subseteq T \cup \{p\}$ contains $\{a_{2,i},\, a'_{2,i}\}$ for some $i = 1, \ldots, n_1$. Consequently, W' is not consistent with the right disjunct of p; since $W_\psi^{\underline{a}_1} \subseteq W$, W' is not consistent. It follows that $W_\psi^{\underline{a}_1} \in \mathcal{F}(p, T)$, and the claim is proved.

Next we show that for every $i = 1, \ldots, n_1$, we have $S_{\underline{a}_1} \cap \{a_{1,i},\, a'_{1,i}\} \neq \emptyset$. Assume to the contrary that $S_{\underline{a}_1} \cap \{a_{1,i},\, a'_{1,i}\} = \emptyset$ for some i. Consider any $W_\psi^{\underline{a}_1}$. We show that $W_\psi^{\underline{a}_1} \models a_{1,i}$. Since $W_\psi^{\underline{a}_1} \in \mathcal{F}(p, T)$ and $T \models p > q$, it follows that $W_\psi^{\underline{a}_1} \models a''_{1,i}$. As $W_\psi^{\underline{a}_1}$ implies the right disjunct of p, in which $a''_{1,i}$ occurs only in the subformula $a_{1,i} \vee a'_{1,i} \Rightarrow a''_{1,i}$, it follows

$$W_\psi^{\underline{a}_1} \models a_{1,i} \vee a'_{1,i}. \tag{5}$$

(Otherwise, $W_\psi^{\underline{a}_1} \cup \{\neg a_{1,i},\, \neg a'_{1,i},\, \neg a''_{1,i}\}$ would be consistent, which is a contradiction to $W_\psi^{\underline{a}_1} \models a''_{1,i}$.) Assume that $W_\psi^{\underline{a}_1} \cup \{\neg a_{1,i}\}$ is consistent. It follows, since $a'_{1,i} \notin W_\psi^{\underline{a}_1}$, that $W_\psi^{\underline{a}_1} \cup \{\neg a_{1,i},\, \neg a'_{1,i}\}$ is consistent; hence, $W_\psi^{\underline{a}_1} \not\models a_{1,i} \vee a'_{1,i}$, which is a contradiction to (5). Hence, $W_\psi^{\underline{a}_1} \models a_{1,i}$.

Now consider $T' = T \cup \{a_{1,i}\}$. We claim that $T' \models p > q$, i.e., for every $W \in \mathcal{F}(p, T')$, $W \models q$. There are two cases.
Case 1: $a_{1,i} \notin W$. It is easy to see that $W \in \mathcal{F}(p, T)$, and hence $W \models q$.
Case 2: $a_{1,i} \in W$. W logically implies the right disjunct of p; hence, for all $i = 1, \ldots, n_2$, we have $\{a_{2,i},\, a'_{2,i}\} \not\subseteq W$. On the other hand, if $\{a_{2,i},\, a'_{2,i}\} \cap W = \emptyset$, then for any $a \in \{a_{2,i},\, a'_{2,i}\}$, $W \cup \{a\}$ is consistent, which contradicts maximality

of W. Thus, $W - \{a_{1,i}\} \subseteq W_\psi^{\underline{a}_1}$ for some ψ. Since $W_\psi^{\underline{a}_1} \models a_{1,i}$, we conclude by the maximality of W that $W = W_\psi^{\underline{a}_1} \cup \{a_{1,i}\}$. Since $W_\psi^{\underline{a}_1} \models q$, it follows that $W \models q$. $\diamond$

Thus, $T' \models p > q$. However, as $T \subset T' \subseteq S$, this is a contradiction to $T \in \mathcal{F}(p, S)$. We conclude from this that $S_{\underline{a}_1} \cap \{a_{1,i},\, a'_{1,i}\} \neq \emptyset$; this concludes the proof of part (≥ 1).

Now define a truth value assignment φ to $\underline{a}_1$ by

$$\varphi(a_{1,i}) = \begin{cases} \mathit{true}, & \text{if } a_{1,i} \in S_{\underline{a}_1}; \\ \mathit{false}, & \text{if } a'_{1,i} \in S_{\underline{a}_1}, \end{cases} \qquad \text{for } i = 1, \ldots, n_1.$$

From the already shown we have that φ is well-defined. Now for any truth value assignment φ to $\underline{a}_2$, we have that $W_\psi^{\underline{a}_1} \in \mathcal{F}(p, T)$, that $c \in W_\psi^{\underline{a}_1}$, and that $W_\psi^{\underline{a}_1}$ logically implies the right disjunct of p; consequently, $(E_\varphi)_\psi$ is satisfiable. Hence, the formula

$$(\exists \underline{a}_1)(\forall \underline{a}_2)(\exists \underline{a}_3)\, E$$

i.e. Φ, is valid. This concludes the $(\Rightarrow)$ part of the proof. $\Box$

From the proof of this lemma, we obtain as a side result a lower bound of Σ_3^P-hardness for deciding a left-nested counterfactual. This is based on the observation that the condition of nonmaximality of S' with respect to satisfaction $p > q$ can be formulated as a nested counterfactual. In fact, S' is not a possible world for $p > q$ in S exactly if it holds that, for any subset T of S such that $T \models p > q$, truth of all formulas in S' (which means that T contains S' as a subset) implies that some formula from $S - S'$ is true (which means that the subset relationship is proper). Formally, this can be stated as follows (recall that $S' = \{\, \underline{a}_2, \underline{a}'_2, c\,\}$, $S - S' = \{\, \underline{a}_1, \underline{a}'_1\,\}$:

Observation 1 (proof of Lemma 5.3) *S' is not a maximal $T \subseteq S$ such that $T \models p > q$ iff $S \models \mathbf{c}$, where*

$$\mathbf{c} = (p > q) > \left[\bigwedge_{i=1}^{n_2} (a_{2,i} \wedge a'_{2,i}) \wedge c \;\Rightarrow\; \bigvee_{i=1}^{n_1} (a_{1,i} \vee a'_{1,i}) \right].$$

Proof. Check that $\mathbf{c}$ is true over S iff for every $T \in \mathcal{F}(p > q, S)$ such that $S' \subseteq T$, this containment is proper. $\Box$

On the other hand, one can show that deciding left-nested counterfactuals is Π_3^P-hard; this suggests that the problem is not in $\Sigma_3^P \cup \Pi_3^P$. We now prove the above mentioned lower bound of Π_4^P-hardness.

Theorem 5.4 *Given S and $(p{>}q){>}r$, deciding whether $S \models (p{>}q){>}r$ is Π_4^P-hard.*

Proof. We extend the construction (4) in the proof of Lemma 5.3. Let Φ be a QBF of the form

$$(\forall \underline{b}_0)(\exists \underline{a}_1)(\forall \underline{a}_2)(\exists \underline{a}_3)E,$$

where $\underline{b}_0 = b_1, \ldots, b_n$, and $\underline{a}_1, \underline{a}_2, \underline{a}_3$ are as in (4). Let $\underline{a}'_1$, $\underline{a}''_1$, $\underline{a}'_2$, and c be the same as there, and let $\underline{b} = b_1, \ldots, b_{2n}$, $\underline{b}' = b'_1, \ldots, b'_{2n}$, and $\underline{b}'' = b''_1, \ldots, b''_{2n}$. Define now

$$\begin{aligned} S &= S_1 \cup \{\, \underline{b}, \underline{b}' \,\}, && (6)\\ p &= p_1 \wedge \bigwedge_{i=1}^{2n} [(b_i \wedge b'_i \Rightarrow \neg c) \wedge (b_i \vee b'_i \Rightarrow b''_i)], \\ q &= q_1 \wedge b''_1 \wedge \cdots \wedge b''_{2n} \wedge (b_1 \cdots b_n \equiv b_{n+1} \cdots b_{2n}), \\ r &= \bigwedge_{j=1}^{n_2} (a_{2,j} \wedge a'_{2,j}) \wedge c \ \Rightarrow\ \bigvee_{i=1}^{n_1} (a_{1,i} \vee a'_{1,i}), \end{aligned}$$

where S_1, p_1, and q_1 are respectively S, p, and q in the construction (4) in the proof of Lemma 5.3, i.e.,

$$\begin{aligned} S_1 &= \{\underline{a}_1, \underline{a}'_1, \underline{a}_2, \underline{a}'_2, c\}, \\ p_1 &= \left[\, c \wedge \bigwedge_{i=1}^{n_1} (\neg a_{1,i} \wedge \neg a'_{1,i} \wedge a''_{1,i}) \,\right] \vee \\ &\quad \left[\, \bigwedge_{i=1}^{n_1} [(a_{1,i} \wedge a'_{1,i} \Rightarrow \neg c) \wedge (a_{1,i} \vee a'_{1,i} \Rightarrow a''_{1,i})] \ \wedge\ (\underline{a}_2 \equiv \neg\, \underline{a}'_2) \wedge (c \Rightarrow E) \,\right], \\ q_1 &= c \wedge a''_{1,1} \wedge \cdots \wedge a''_{1,n_1}. \end{aligned}$$

Let us explain the intuition behind this transformation. The idea is, starting from Lemma 5.3, to reduce validity checking for Φ to a collection of tests whether S'_φ, for φ a truth value assignment to $\underline{b}_0$, is a maximal $T \subseteq S$ such that $T \models p > q$; S'_φ extends S' in Lemma 5.3 conceptually by a choice γ_i from $\{b_i, b'_i\}$, for $i = 1, \ldots, n$ corresponding to φ such that the test for S'_φ resembles validity checking for $(\exists \underline{a}_1)(\forall \underline{a}_2)(\exists \underline{a}_3)\, E_\varphi$. The atoms b_{n+i} and b'_{n+i} serve to fix the choice γ_i in possible worlds of $p > q$. One can make then use of Observation 1 and describe the collection of tests as a left-nested counterfactual.

We prove the correctness of the transformation. In what follows, let

$$\begin{aligned} S' &= \{\underline{a}_2, \underline{a}'_2, c\}, \\ S'_\varphi &= S' \cup \{b_i,\, b_{n+i} : \varphi(b_i) = \mathit{true},\ 1 \leq i \leq n\} \cup \\ &\qquad\qquad \{b'_i,\, b'_{n+i} : \varphi(b_i) = \mathit{false},\ 1 \leq i \leq n\} \end{aligned}$$

for every truth value assignment φ to $\underline{b}$. We observe that $S'_\varphi \models p > q$ (check that $S'_\varphi \cup \{p\}$ is consistent and logically implies q), and that $S'_\varphi \not\models r$. We show that any

$W \in \mathcal{F}(p > q, S)$ such that $W \not\models r$ must be of the form S'_φ for some φ. To conclude this, we note some facts (**a**)–(**e**) about any $W \in \mathcal{F}(p > q, S)$.

(**a**) *$c \in W$.* Since $c \in S$ and for each $U \in \mathcal{F}(p, W)$, it holds $U \models c$, this follows from the maximality of W.

(**b**) *$\{b_i, b'_i\} \not\subseteq W$, for each $i = 1, \ldots, 2n$.* Assume that $\{b_i, b'_i\} \subseteq W$ for some i. Since $\{b_i, b'_i, p\}$ is consistent, there exists a $U \in \mathcal{F}(p, W)$ such that $\{b_i, b'_i\} \subseteq U$. As $p \in U$, we have $U \models \neg c$; hence, $U \not\models q$. This implies $W \not\models p > q$, which is a contradiction to $W \in \mathcal{F}(p > q, S)$.

(**c**) *$\{b_i, b'_i\} \cap W \neq \emptyset$, for each $i = n+1, \ldots, 2n$.* Assume that $\{b_i, b'_i\} \cap W = \emptyset$ for some i. Consider $U \in \mathcal{F}(p, W)$. Then, $U \cup \{\neg b''_i\}$ is consistent (note that b_i and b'_i do not occur in p_1); hence, $U \not\models q$, which implies $W \not\models p > q$. Thus, $W \notin \mathcal{F}(p > q, S)$, which is a contradiction.

(**d**) *If $S' \subseteq W$, then $\{b_i, b'_i\} \cap W \neq \emptyset$ for each $i = 1, \ldots, n$.* Assume that $\{b_i, b'_i\} \cap W = \emptyset$ for some i. From (**a**) and (**b**) we infer that $T = S' \cup (W - \{\underline{a}_1, \underline{a}'_1\})$ is consistent with p. Furthermore, T is a maximal subset of this property; hence, $T \cup \{p\} \in \mathcal{F}(p, W)$. It holds, as readily checked, that $T \cup \{p, \neg b''_i\}$ is consistent; hence, $T, p \not\models q$. This implies $W \notin \mathcal{F}(p > q, S)$, which is a contradiction.

(**e**) *If $S' \subseteq W$, then $b_i \in W$ iff $b_{n+i} \in W$, and $b'_i \in W$ iff $b'_{n+i} \in W$, for each $i = 1 \ldots, n$.* Note that from (**b**)–(**d**), exactly one of b_j and b'_j belongs to W, for each $j = 1, \ldots, 2n$. Assume the claim is false and consider the case $b_i \in W$, $b_{n+i} \notin W$ (the other cases are similar). From (**c**) it follows that $b'_{n+i} \in W$. Now observe that $\{b_i, b'_{n+i}, c\}$ is consistent with p; hence, by (**a**) there exists U, $\{b_i, b'_{n+i}, c\} \subseteq U \subseteq W$ such that $U \cup \{p\} \in \mathcal{F}(p, W)$. Notice that $U, p \models b_i \wedge \neg b_{n+i}$; hence, $U, p \not\models b_i \equiv b_{n+i}$, and thus $U, p \not\models q$. This implies $W \notin \mathcal{F}(p > q, S)$, which is a contradiction.

Now let $W \in \mathcal{F}(p > q, S)$ such that $W \not\models r$. This implies that $\{\underline{a}_1, \underline{a}'_1\} \cap W = \emptyset$. Furthermore, W contains by (**b**) at most one of b_i and b'_i, for each $i = 1, \ldots, 2n$. Consequently, $W \subseteq S'_\varphi$ for some φ; by maximality of W, it follows that $W = S'_\varphi$.

At this point we have shown that $S \models (p > q) > r$ iff S'_φ is not a maximal $T \subseteq S$ such that $T \models p > q$, for each φ. We show next that for each φ, this problem essentially reduces to deciding whether S' in construction (4) is not a maximal $T \subseteq S$ such that $S' \models p > q$. We note the following fact on construction (6).

(**f**) *Let $W \in \mathcal{F}(p > q, S)$ such that $S' \subseteq W$. Then, for each $U \in \mathcal{F}(p, W)$, we have (i) $b_i \in U$ iff $b_i \in W$, for each $i = 1, \ldots, n$, and (ii) $b'_i \in U$ iff $b'_i \in W$, for each $i = 1, \ldots, n$.* To show (i), it clearly remains to show that $b_i \in W$ implies $b_i \in U$. Assume hence that $b_i \in W$. Then, from (**e**) follows $b_{n+i} \in W$. Check that $U \cup \{p, b_{n+i}\}$ is consistent (from (**b**) follows $b'_{n+i} \notin U$); hence, by maximality of U, we have $b_{n+i} \in U$. Furthermore, we have that $U, p \models q$ (as $W \models p > q$), and $q \models b_i \equiv b_{n+i}$; hence, it follows that $U, p \models b_i$. By the maximality of U, it follows that $b_i \in U$. The proof of (ii) is analogous.

Informally, (**f**) together with the facts from above means that in testing whether

$S'_\varphi \in \mathcal{F}(p > q, S)$, we can substitute for each atom of $\underline{b}, \underline{b}'$, and $\underline{b}''$ either $\top$ or $\perp$ in S, p, and q depending on φ, such that the problem is equivalent to the issue if $S' \in \mathcal{F}(p_1 > q_1, S_1)$, given that $S' \models p_1 > q_1$, where the formula E in p_1 is replaced by E_φ.

We show now that $S \models (p > q) > r$ iff Φ is valid.

($\Leftarrow$) Let φ be any truth value assignment to $\underline{b}_0$. Then, there exists a truth value assignment ψ to $\underline{a}_1$ such that $\forall \underline{a}_2 \exists \underline{a}_3 \, (E_\varphi)_\psi$ is valid. Let

$$W^\psi_\varphi = S'_\varphi \cup \{a_{1,i} : \psi(a_{1,i}) = \mathit{true}\} \cup \{a'_{1,i} : \psi(a_{1,i}) = \mathit{false}\}.$$

We claim that $W^\psi_\varphi \models (p > q)$. To show this, consider any $U \in \mathcal{F}(p, W^\psi_\varphi)$. There are two cases.
Case 1: $\{a_{2,i},\, a'_{2,i}\} \subseteq U$, for some $i = 1, \ldots, n_2$. Then, U logically implies the left disjunct of p_1; hence, $U \models q_1$. From the maximality of U, it follows $U \cap \{\underline{b},\, \underline{b}'\} = W \cap \{\underline{b},\, \underline{b}'\}$, and hence $U \models \underline{b}''$. Thus, $U \models q$.
Case 2: $\{a_{2,i},\, a'_{2,i}\} \not\subseteq U$, for every $i = 1, \ldots, n_2$. Check that for any choice γ_i from $\{a_{2,i},\, a'_{2,i}\}$, for $i = 1, \ldots, n_2$, the set $U_\gamma = W^\psi_\varphi - \{\gamma_i : i = 1, \ldots, n_2\}$ is consistent with p_1. Consequently, U is of the form $U_\gamma \cup \{p\}$ for some γ. It follows that $U \models q$. $\diamond$

Thus $W^\psi_\varphi \models p > q$. Notice that $S'_\varphi \subset W^\psi_\varphi$ and that every $T \in \mathcal{F}(p > q, S)$ such that $W^\psi_\varphi \subseteq T$ fulfills $T \models r$. Since φ was arbitrary, it follows that $S'_\varphi \notin \mathcal{F}(p > q, S)$ for every φ. This implies that $S \models (p > q) > r$.

($\Rightarrow$) Assume that $S \models (p > q) > r$. Let φ be an arbitrary truth value assignment φ to $\underline{b}_0$. Then, we have that $S'_\varphi \notin \mathcal{F}(p > q, S)$, i.e., there exists a $T \in \mathcal{F}(p > q, S)$ such that $S'_\varphi \subset T$. From fact (**e**), it follows that $b_i, b_{n+i} \in W$ iff $\varphi(b_i) = \mathit{true}$ and $b'_i, b'_{n+i} \in W$ iff $\varphi(b_i) = \mathit{false}$, for each $i = 1, \ldots, n$. Using (**f**), one shows along the line of the ($\Rightarrow$) part of the proof of Lemma 5.3 that $|T \cap \{a_{1,i},\, a'_{1,i}\}| = 1$ holds for each $i = 1, \ldots, n_1$ and that the formula $(\exists \underline{a}_1)(\forall \underline{a}_2)(\exists \underline{a}_3)\, E_\varphi$ is valid. Since φ was arbitrary, it follows that Φ is valid. This concludes the ($\Rightarrow$) part and the proof of the theorem. $\Box$

The main result of this section follows from Theorems 5.2 and 5.4; it indicates that nesting in the premise is computationally much harder than nesting in the conclusion.

Corollary 5.5 *Given S and $\mathbf{c} = (p > q) > r$, deciding whether $S \models \mathbf{c}$ is Π^P_4-complete.*

Iteration in left-nestings appears to be conceptually quite involved, and the relevance of such statements in practice seems to be questionable; we do not know of a simple, intuitive example. Concerning evaluation, the following upper bound can be derived by straightforward induction on k (cf. Proposition 2.1 and proof of Theorem 5.2):

Proposition 5.6 *Let S be a knowledge base and* $\mathbf{c}$ *be a positive iterated left-nested counterfactual with nesting depth less than k, for a fixed $k \geq 1$. Then, deciding whether $S \models \mathbf{c}$ is in Π_{2k}^P.*

It is not clear whether this bound can be improved. We conjecture that it extends to completeness for Π_{2k}^P and that the full language of iterated left-nested counterfactuals is PSPACE-complete.

As opposed to right-nested counterfactuals, negation does not lead to an increase in the complexity of left-nested counterfactuals. While evaluating statements $p > (q \not> r)$ over a knowledge base is Π_3^P-complete, evaluating $(p \not> q) > r$ is in Π_4^P (for an algorithm, just add in the algorithm **PLNCF1** a negation in front of calls to **CF**.) Similarly negation also does not lead in iterated left-nestings to an increase in complexity, which follows by results in the next section.

6 Mixed counterfactuals

In this section, we finally consider the mix of nestings in the premise and the conclusion. The simplest statements of this type are $(p > q) > (r > s)$ and $p > ((q > r) > s)$. Notice that such statements are quite contrived. As with iterated left-nested counterfactuals, mixed counterfactuals seem to be more of theoretical interest than relevant to practice.

Intuitively, one might expect that the mix of left and right-nestings yields PSPACE-completeness. However, it appears that right-nesting does not increase complexity. In particular, both statements from above have no higher complexity than a left-nested statement $(p > q) > r$, and can be evaluated in the Π_4^P. This will be derived as a consequence of the following property.

Proposition 6.1 *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ such that for any given S and $\mathbf{c}_1 \in \mathcal{C}_1$, $\mathbf{c}_2 \in \mathcal{C}_2$ deciding $S \models \mathbf{c}_1$ is in Δ_k^P and $S \models \mathbf{c}_2$ is in Π_{k+1}^P, for some fixed $k \geq 1$. Then, deciding $S \models \mathbf{c}_1 > \mathbf{c}_2$ is in Π_{k+1}^P.*

Proof. By definition, $S \not\models \mathbf{c}_1 > \mathbf{c}_2$ iff $W \not\models \mathbf{c}_2$ for some $W \in \mathcal{F}(\mathbf{c}_1, S)$. From the hypothesis, it follows that a guess for $W \in \mathcal{F}(\mathbf{c}_1, S)$ can be verified with one call to a Π_k^P oracle (ask whether $W \models \mathbf{c}_1$ and for every W' s.t. $W \subset W' \subseteq S$, $W' \not\models \mathbf{c}_1$); moreover, since $W \not\models \mathbf{c}_2$ is in Σ_{k+1}^P, a polynomial size guess for a proof of this fact can be verified in polynomial time with a Π_k^P oracle. Consequently, deciding $S \not\models \mathbf{c}_1 > \mathbf{c}_2$ is in Σ_{k+1}^P, and therefore deciding $S \models \mathbf{c}_1 > \mathbf{c}_2$ is in Π_{k+1}^P. $\square$

The next theorem is a straightforward application of this proposition. Let for any class $\mathcal{C}' \subseteq \mathcal{C}$ denote by $cl_{prn}(\mathcal{C}')$ the closure of $\mathcal{C}'$ under positive right-nesting, which is the smallest superset of $\mathcal{C}'$ such that $\alpha > \beta \in cl_{prn}(\mathcal{C}')$ for every $\alpha \in \mathcal{C}' \cup \mathcal{L}$ and $\beta \in cl_{prn}(\mathcal{C}') \cup \mathcal{C}' \cup \mathcal{L}$.

Define sequences $\Gamma^i[\mathcal{C}']$ and $E^i[\mathcal{C}]$, $i \geq 0$, by

$$\Gamma^0[\mathcal{C}'] = \mathcal{C}',\ \Gamma^{i+1}[\mathcal{C}'] = cl_{prn}(\Gamma^i[\mathcal{C}']),\ \text{and}$$
$$E^0[\mathcal{C}'] = \mathcal{C}',\ E^{i+1}[\mathcal{C}'] = cl_{prn}(E^i[\mathcal{C}'] \cup \{\neg\mathbf{c} : \in E^i[\mathcal{C}']\}).$$

Since $cl_{prn}(\cdot)$ is monotonic, clearly $\Gamma^i[\mathcal{C}'] \subseteq E^i[\mathcal{C}']$ for every $i \geq 0$, and the sequences have the limits $\Gamma^\infty[\mathcal{C}'] = \bigcup_{i\geq 0} \Gamma^i[\mathcal{C}']$ and $E^\infty[\mathcal{C}'] = \bigcup_{i \geq 0} E^i[\mathcal{C}']$, respectively. It is easily seen that $\Gamma^\infty[\emptyset]$ is the class of all positive counterfactuals, and that $E^\infty[\emptyset] = E^\infty[\mathcal{C}'] = \mathcal{C}$. Note that $\Gamma^2[\emptyset]$, and hence also $E^2[\emptyset]$ contains the statements $(p > q) > (r > s)$ and $p > ((q > r) > s)$ from the beginning of this section.

Theorem 6.2 *Let $\mathcal{C}' \subseteq \mathcal{C}$ such that evaluation of every $\mathbf{c} \in \mathcal{C}'$ over any given knowledge base S is in Δ^P_k, for some $k \geq 1$. Then, deciding $S \models \mathbf{c}$ for any $\mathbf{c} \in E^i[\mathcal{C}']$, $i \geq 1$, is in Π^P_{k+2i-1}.*

Proof. From Proposition 6.1 and the hypothesis, it follows by structural induction on $\mathbf{c} \in cl_{prn}(\mathcal{C}')$ that deciding $S \models \mathbf{c}$ is in Π^P_{k+1}, and hence the statement holds for $i = 1$. Induction on i and the structure of formulas in $E^i[\mathcal{C}']$ gives the result (note that deciding $S \models \mathbf{c}$ for $\mathbf{c} \in E^i[\mathcal{C}'] \cup \{\neg\mathbf{c}' : \mathbf{c}' \in E^i[\mathcal{C}']\}$ is in $\Pi^P_{k+2i-1} \cup \Sigma^P_{k+2i-1} \subseteq \Delta^P_{k+2i}$). $\Box$

This result allows to derive a number of complexity bounds for different classes of mixed nested counterfactuals. It is not difficult to see that $\Gamma^i[\emptyset]$ (and thus also $E^i[\emptyset]$) contains precisely those nested counterfactuals $\mathbf{c}$ which are positive and whose left-nesting depth is less than i, where the left-nesting depth of $\mathbf{c}$ is the maximal number of $>$-nodes v on any path in the formula tree for $\mathbf{c}$ such that v is a left descendant of the preceding $>$-node. (A similar characterization of $E^i[\emptyset]$ can be found.) For example, $(p > q) > ((r > s) > t)$ has left-nesting depth 1. In particular, $\Gamma^1[\emptyset]$ is the class of positive right-nested counterfactuals. Concerning $E^i[\emptyset]$, we note that this class includes all left- and right-nested counterfactuals of the form $\alpha > \beta$ whose nesting depth is less than i. From Theorems 4.6,4.7, 6.2 and Corollary 5.5 we thus obtain the following.

Corollary 6.3 *Given a knowledge base S and $\mathbf{c} \in E^k[\emptyset]$ for constant $k \geq 1$, then deciding $S \models \mathbf{c}$ is in Π^P_{2k} and Π^P_{k+1}-hard. For $k = 2$, deciding $S \models \mathbf{c}$ is Π^P_4-complete, and hard even for positive left-nested $\mathbf{c}$. Moreover, deciding $S \models \mathbf{c}$ for $\mathbf{c} \in E^\infty[\emptyset]$ is PSPACE-complete.*

The complexity of $\Gamma^\infty[\emptyset]$, i.e. the positive mixed counterfactuals, seems to be strictly tied to the one of iterated left-nestings. This view is supported by the results on $E^i[\emptyset]$ above, which allow us to derive that bounded left-nesting, even with arbitrary positive right-nesting but bounded negative right-nesting, does not give all of PSPACE; in fact, it can be easily shown that each nested counterfactual $\mathbf{c}$ with left-nesting depth less than k and negation depth less or equal l is contained in $E^{k+l}[\emptyset]$.

7 Related work and conclusion

Complexity characterizations of evaluating counterfactuals are given in [37, 23, 16, 17]. Grahne and Mendelzon [17] considered subjunctive queries in a different framework, where the knowledge base is given by a set of models KB and updates are performed according to Winslett's method [36], which operates as follows. Let for each model M in KB, denote $Min(M,p)$ denote the set of models of the update formula p that are closest to M, which are those models whose sets of atoms on which they differ from M are minimal with respect to set inclusion. Then, the result of updating KB with p is the union $\bigcup_{M\in KB} Min(M,p)$. The relation between Winslett's update operator and the one by Fagin et al. has been investigated in [19, 9]. In particular, Corollary 4.2 in [17] implies that evaluating nested counterfactuals under this update semantics is PSPACE-complete.

A large account to hypothetical reasoning in databases has been given by Bonner [2, 3, 4], who studied extensions of Horn clause logic in which rules can hypothetically add facts to a relational database by means of embedded implications [22]. For example, a rule of the form $A \leftarrow B[add : C]$ intuitively means "infer A if inserting C allows the inference of B." Note, however, that the fact C is always consistent with the current database, and that no facts are removed from the database in the course of a hypothetical addition. This is a salient difference to the revision framework we considered. Bonner showed that function-free Horn logic extended with hypothetical additions is PSPACE-complete [3]. Moreover, he extended that logic, which has an intuitionistic semantics, by negation-as-failure and analyzed stratified rule-bases; as shown in [2], stratified rule-bases with at most k strata are data-complete for Σ_k^P. The results of the previous sections imply that there exists an efficient translation from evaluating right-nested counterfactuals to inference from stratified rule-bases and vice versa. However, a short, intuitive such translation is not at hand. Likewise, a simple efficient transformation from intuitionistic propositional logic, which is well-known PSPACE-complete [30], is not evident.

Our work contributes to the recent effort in giving a precise complexity characterization of nonmonotonic reasoning in the full propositional context, cf. [24, 37, 23, 27, 9, 32] (see [6] for an overview), extending previous results for restricted contexts, cf. [20, 31, 29, 5]. Such a characterization supports a better understanding of the computational relationships between various forms of nonmonotonic reasoning, e.g. efficient intertranslatability.

Furthermore, the precise complexity of a problem gives us a clue of its computational difficulty and may provide insight to sources of complexity. For counterfactuals, these sources are classical inference ($S \models p$) and the many knowledge bases that are possible after incorporating a change. Fortunately, a sequence of changes is not a source of complexity. Since Π_2^P-complete problems are most likely much harder than NP-complete problems, our results suggest that methods such as GSAT [28] for efficient handling of NP-complete problems are most likely not applicable to nested counterfactuals. However, GSAT can be fruitfully applied for proving $S \not\models p_1 > (p_2 > \cdots (p_n > q) \cdots)$ if all propositional formulas are Horn clauses. It remains to find alternatives for the inadequate GSAT method that allow to handle nested counterfactuals satisfactorily. One way would be to find an approximation of the semantics which can be fast computed, along the ideas of the work in [7]. Another way would be to look whether more general methods and techniques for approximating PSPACE-complete problems are applicable; however, the work on the design of approximation algorithms for such problems is currently at an early stage [8].

Acknowledgments The authors would like to thank the referees and Helmut Veith for their useful hints and constructive comments on an earlier version of this paper.

References

[1] S. Abiteboul, P. Kanellakis, and G. Grahne. On the Representation and Querying of Sets of Possible Worlds. *Theoretical Computer Science*, 78:159–187, 1991.

[2] A. Bonner. Hypothetical Datalog: Negation and Linear Recursion. In *Proceedings PODS-89*, pages 286–300, 1989.

[3] A. Bonner. Hypothetical Datalog: Complexity and Expressibility. *Theoretical Computer Science*, 76:3–51, 1990.

[4] A. Bonner. *Hypothetical Reasoning in Deductive Databases.* PhD thesis, Dept. of Computer Science, Rutgers University, 1991.

[5] M. Cadoli and M. Lenzerini. The Complexity of Closed World Reasoning and Circumscription. *Journal of Computer and Systems Sciences*, 43:165–211, 1994.

[6] M. Cadoli and M. Schaerf. A Survey on Complexity Results for Non-monotonic Logics. *Journal of Logic Programming*, 17:127–160, 1993.

[7] M. Cadoli and M. Schaerf. Tractable Reasoning via Approximation. *Artificial Intelligence*, 74(2):249–310, 1995.

[8] A. Condon. Approximate Solutions to Problems in PSPACE. *SIGACT News*, 26(2):4–13, June 1995.

[9] T. Eiter and G. Gottlob. On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals. In *Proceedings PODS-92*, pages 261–273, 1992. Extended version *Artificial Intelligence*, 57(2–3):227–270, 1992.

[10] R. Fagin, G. Kuper, J. Ullman, and M. Y. Vardi. Updating Logical Databases. In P. Kanellakis and F. Preparata, editors, *Advances in Computing Research*, volume 3, pages 1–18, Greenwich, CT, 1986. JAI Press.

[11] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the Semantics of Updates in Databases. In *Proceedings PODS-83*, pages 352–365, 1983.

[12] P. Gärdenfors. *Knowledge in Flux*. Bradford Books, MIT Press, 1988.

[13] M. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.

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

[15] M. L. Ginsberg and D. E. Smith. Reasoning About Action I: A Possible Worlds Approach. *Artificial Intelligence*, 35:165–195, 1988.

[16] G. Grahne. Updates and Counterfactuals. In *Proceedings KR-91*, pages 269–276, 1991.

[17] G. Grahne and A. O. Mendelzon. Updates and Subjunctive Queries. Technical Report KRR-TR-91-4, University of Toronto, CS Dept., July 1991. *Information and Computation*, 116(2):241–252, 1995.

[18] D. S. Johnson. A Catalog of Complexity Classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science* A, ch. 2. Elsevier Science Pub., 1990.

[19] H. Katsuno and A. O. Mendelzon. Propositional Knowledge Base Revision and Minimal Change. *Artificial Intelligence*, 52:253–294, 1991.

[20] H. Kautz and B. Selman. Hard Problems for Simple Default Logics. *Artificial Intelligence*, 49:243–279, 1991.

[21] M. Krentel. Generalizations of OptP to the Polynomial Hierarchy. *Theoretical Computer Science*, 97(2):183–198, 1992.

[22] L.T. McCarty. Clausal Intuitionistic Logic. I. Fixed-Point Semantics. *Journal of Logic Programming*, 5:1–31, 1993.

[23] B. Nebel. Belief Revision and Default Reasoning: Syntax-Based Approaches. In *Proceedings KR-91*, pages 417–428, 1991.

[24] I. Niemelä. Towards Automatic Reasoning. In *Proc. Europ. Workshop on Logics in AI, Amsterdam, September 1990*, #478 in LNCS. Springer, 1991.

[25] M. Rabin and D. Scott. Finite Automata and Their Decision Problem. *IBM Journal of Research and Development*, 3:114–125, 1959.

[26] R. Reiter. A Theory of Diagnosis From First Principles. *Artificial Intelligence*, 32:57–95, 1987.

[27] V. Rutenburg. Complexity Classification in Truth Maintenance Systems. In *Proceedings STACS-91*, LNCS 480, pages 373–383, 1991. Extended version in *Annals of Mathematics and Artificial Intelligence*, 10:207–231, 1994.

[28] B. Selman, H. Levesque, and D. Mitchell. A New Method for Solving Hard Satisfiability Problems. In *Proceedings AAAI-92*, pages 440–446, 1992.

[29] B. Selman and H. J. Levesque. Abductive and Default Reasoning: A Computational Core. In *Proceedings AAAI-90*, pages 343–348, July 1990.

[30] R. Statman. Intuitionistic Propositional Logic is Polynomial-Space Complete. *Theoretical Computer Science*, 9(2):67–72, 1979.

[31] J. Stillman. It's Not My Default: The Complexity of Membership Problems in Restricted Propositional Default Logic. In *Proceedings AAAI-90*, pages 571–579, 1990.

[32] J. Stillman. The Complexity of Propositional Default Logic. In *Proceedings AAAI-92*, pages 794–799, 1992.

[33] L. Stockmeyer. Classifying the Computational Complexity of Problems. *Journal of Symbolic Logic*, 52(1):1–43, 1987.

[34] M. Vardi. On the Complexity of Epistemic Reasoning. In *Proceedings Fourth Annual Symposium on Logic in Computer Science LICS-89*, pages 243–252. IEEE Computer Science Press, 1989.

[35] K. Wagner. Bounded query classes. *SIAM J. Comp.*, 19(5):833–846, 1990.

[36] M. Winslett. Reasoning about Action Using a Possible Models Approach. In *Proceedings AAAI-88*, pages 89–93, 1988.

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