In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), R. Bajcsy ed, Morgan Kaufman, 1993, pp. 526–531.

The Complexity of Nested Counterfactuals and Iterated Knowledge Base Revisions*

Thomas Eiter and Georg Gottlob

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

Abstract

We consider the computational complexity of evaluating nested counterfactuals over a propositional knowledge base. Counterfactual implication $p > q$ models a statement "if p, then q," where p is known or expected to be false, and is different from material implication $p \Rightarrow q$. A nested counterfactual is a counterfactual statement where the conclusion q is a (possibly negated) counterfactual. Statements of the form $p_1 > (p_2 > \cdots (p_n > q) \cdots)$ intuitively correspond to hypothetical 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 the nesting. We also consider nesting a counterfactual in the premise, i.e. $(p > q) > r$ and show that evaluating such statements is most likely much harder than evaluating $p > (q > r)$.

1 Introduction

A counterfactual is a conditional statement "if p, then q," where the premise p is either known or expected to be false [Ginsberg, 1986], e.g. "If the electricity hadn't failed, 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. The relevance of counterfactual reasoning to a number of AI applications was first demonstrated in [Ginsberg, 1986], to which (and to [Gärdenfors, 1988; Nebel, 1991; Grahne, 1991]) the reader is referred for a background.

In this paper we mainly deal with nested counterfactuals, i.e., counterfactuals where the conclusion can be a counterfactual itself instead of a plain propositional sentence. Nested counterfactuals are often used in real-life contexts and are an important principle of commonsense reasoning.

Example 1: The statement *"If you would have bought a painting by Botticelli from John and you would notice it is a fake, you would still remain a client of John"* corresponds to a nesting of counterfactuals of the form

$$\mathit{buy\_botticelli\_from\_john} > (\mathit{fake} > \mathit{client\_of\_john}).$$

The value of this counterfactual depends, of course, on the given knowledge base. It is intuitively clear that the counterfactual will evaluate to *false* on a large number of reasonable knowledge bases. According to Ramsey's rule, the evaluation of this nested counterfactual over a given knowledge base T amounts to check whether

$$(T \circ \mathit{buy\_botticelli\_from\_john}) \circ \mathit{fake}) \models \mathit{client\_of\_john}$$

for a suitable revision operator "$\circ$". This example also shows that nesting a counterfactual in the conclusion is different from strengthening the premise, i.e., $(p > (q > r))$ is different from $p \wedge q > r$. Indeed, the conjunction of *buy_botticelli_from_john* and *fake*, is semantically inconsistent (because a Botticelli is not a fake), and therefore the sentence would be vacuously true for each knowledge base. □

More generally, a nested counterfactual of the form $p_1 > (p_2 > \ldots(\ldots > (p_n > q))\ldots)$ is true over a knowledge base T iff $T \circ p_1 \circ p_2 \ldots \circ p_n \models q$, i.e., iff T revised by p_1, revised by p_2, revised by p_3 etc. implies q. For this reason, the complexity results we will derive for nested counterfactuals are equally relevant to the problem of inferencing after *iterated* knowledge base revisions.

The complexity of evaluating unnested counterfactuals over propositional knowledge bases, i.e. finite propositional theories, was considered in [Nebel, 1991; Grahne, 1991; Eiter and Gottlob, 1992]. In this paper, we deal with evaluating nested counterfactuals based on Ginsberg's approach [Ginsberg, 1986] which uses the method by Fagin, Ullman and Vardi [Fagin *et al.*, 1983] for incorporating changes to a knowledge base. Such statements intuitively correspond to hypothetical queries involving a sequence of revisions, and are naturally relevant to planning and reasoning about actions (cf. [Ginsberg and Smith, 1988; Winslett, 1988]), for instance.

*This is a short version containing only proof sketches. An extended report containing full proofs and more results is in preparation.

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

Example 2: Imagine a two person game and that player one wants to know whether every possible choice for his next move (m_1) does not result in a forced win for player two, i.e. player two does not win regardless of his next move ($\neg(m_2 > w_2)$). This question amounts to $m_1 > \neg(m_2 > w_2)$. □

The alternative to nesting counterfactuals into the consequence is nesting into the premise, i.e. a nesting $(p > q) > r$. 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 3: Imagine a system is error detecting if the occurrence of an error (e) is displayed (d) on some special device. The question whether a module m must occur in the system if its current state is changed to be error detecting amounts to $(e > d) > m$. □

The complexity of evaluating a single counterfactual in the propositional case was studied in [Nebel, 1991; Eiter and Gottlob, 1992], where it was shown that this problem is Π_2^P-complete. In the present paper we study the complexity of checking nested counterfactuals over propositional knowledge bases. Our main results are summarized as follows. First, we show that deciding nested counterfactuals 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 KB-revisions. The first incorporates each revision into the KB and needs in general exponential space and time, while the second stores the initial KB 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. Second, we show that things get more complicated (PSPACE-complete) if negated counterfactuals can appear in nestings (see Example 2). Third, we consider nested counterfactuals of type $(p > q) > r$, i.e. the nesting occurs in the premise, and show that checking validity of such formulas is Π_4^P-complete.

The rest of this paper is organized as follows. Section 2 introduces concepts and reviews previous results. In Sections 3 and 4 investigate into the complexity of evaluating counterfactuals nested in the conclusion without and with negation, respectively, while Section 5 deals with nesting in the premise. Section 6 gives some conclusions. Due to space limitations, we provide here for some results merely detailed proof sketches.

2 Definitions and previous results

We assume that the reader knows about the basic concepts of NP-completeness, the polynomial hierarchy, and PSPACE, rf. [Garey and Johnson, 1979]. Briefly, PSPACE is the class of problems decidable in polynomial space, and the classes Δ_k^P, Σ_k^P, and Π_k^P of the polynomial hierarchy are defined as follows: $\Delta_0^P = \Sigma_0^P = \Pi_0^P = \text{P}$ and for $k \geq 0$, $\Delta_{k+1}^P = \text{P}^{\Sigma_k^P}$, $\Sigma_{k+1}^P = \text{NP}^{\Sigma_k^P}$, $\Pi_{k+1}^P = \text{co}\Sigma_{k+1}^P$. In particular, $\Delta_1^P = \text{P}$, $\Sigma_1^P = \text{NP}$, and $\Pi_1^P = \text{coNP}$. Clearly $\Sigma_k^P \subseteq \Sigma_k^P \cup \Pi_k^P \subseteq \Delta_{k+1}^P \subseteq \Sigma_{k+1}^P$, but for $k \geq 1$ any equality is considered very unlikely similar as $\text{P} = \text{NP}$. 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 $\text{QBF}_{k,\exists}$ (resp. $\text{QBF}_{k,\forall}$) the valid QBFs Φ with j quantifier alternations and $Q_1 = \exists$ ($Q_1 = \forall$), where Φ has a "quantifier alternation" for Q_1 and every $i > 1$ with $Q_i \neq Q_{i-1}$. Deciding if $\Phi \in \text{QBF}_{k,\exists}$ (resp. $\Phi \in \text{QBF}_{k,\forall}$) is complete for Σ_k^P (Π_k^P).

Let $\mathcal{L}$ be the language of propositional logic over some set of atoms. "$\top$" and "$\perp$" are constants for truth and falsity, respectively. We assume the usual convention on the binding of the logical connectives. A *knowledge base* is a finite subset of $\mathcal{L}$. Knowledge bases are denoted by letters $S, T, \ldots$, formulas from $\mathcal{L}$ by $p, q, \ldots$, and atoms by a, b etc. A *literal* is an atom or the negation of an atom.

$p > q$ denotes the counterfactual "If p, then q". The formal semantics of $p > q$ is as follows. Let

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

be the "possible worlds for p" [Ginsberg, 1986], and let

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

Then, $p > q$ has value *true* over knowledge base S (in symbols, $S \models p > q$) if for every $T \in \mathcal{F}(p, S)$, $T \models q$, and value *false* otherwise ($S \not\models p > q$).

$\mathcal{F}(p, S)$ had been earlier proposed as an operator for updating logical databases in [Fagin *et al.*, 1983], where the databases in $\mathcal{F}(p, S)$ are considered to be independent possible outcomes of an update by p.

As shown in [Nebel, 1991; Eiter and Gottlob, 1992], evaluating a counterfactual is most likely much harder than any NP-complete problem.

Proposition 2.1 [Nebel, 1991; Eiter and Gottlob, 1992] *Deciding whether $S \models p > q$ is Π_2^P-complete.*

The *nested counterfactuals* are the smallest formula set $\mathcal{C}$ that contains all counterfactuals $p > q$ and satisfies the following properties:

(1) if $\mathbf{c} \in \mathcal{C}$ and $p \in \mathcal{L}$, then $p > \mathbf{c} \in \mathcal{C}$.

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

We also write $p \not> \mathbf{c}$ for $\neg(p > \mathbf{c})$. The unique sequence $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n = \mathbf{c}$ such that $\mathbf{s}_1 = p > q$ and $\mathbf{s}_i$, $2 \leq i \leq n$, results from $\mathbf{s}_{i-1}$ by (1) or (2) is referred to as the *structural sequence* of $\mathbf{c}$. Each $\mathbf{s}_i$ is said to *occur* in $\mathbf{c}$.

Example 4: The structural sequence of $p_1 > (p_2 \not> q)$ is $p_2 > q$, $p_2 \not> q$, $p_1 > (p_2 \not> q)$. □

We now give a precise semantics to nested counterfactuals. The truth value of a (possibly nested) $\mathbf{c} \in \mathcal{C}$ over a knowledge-base S is recursively defined as follows. $\mathbf{c}$ has value *true* over S (in symbols, $S \models \mathbf{c}$)

- if $\mathbf{c} = p > \mathbf{c}'$, $\mathbf{c}' \in \mathcal{C}$, and $T \models \mathbf{c}'$, for every $T \in \mathcal{F}(p, S)$.
- if $\mathbf{c} = \neg\mathbf{c}'$, $\mathbf{c}' \in \mathcal{C}$, and $\mathbf{c}'$ has value *false* over S.

and has value *false* otherwise ($S \not\models \mathbf{c}$).

The knowledge bases that are relevant for $\mathbf{s}_i$ 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:

$$Cn(\mathbf{s}_n, \mathbf{c}, S) = \{S\}, \quad \text{and for } 2 \le i \le n,$$

$$Cn(\mathbf{s}_{i-1}, \mathbf{c}, S) = \begin{cases} \bigcup_{T \in Cn(\mathbf{s}_i, \mathbf{c}, S)} \mathcal{F}(p, T) & \text{if } \mathbf{s}_i = p > \mathbf{s}_{i-1} \\ Cn(\mathbf{s}_i, \mathbf{c}, S) & \text{if } \mathbf{s}_i = \neg\mathbf{s}_{i-1} \end{cases}$$

The following proposition is immediate from the definition.

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

3 Right-nested counterfactuals

We start with the evaluation of nested counterfactuals where no negation occurs in the nesting. In the spirit of [Rabin and Scott, 1959] (cf. also [Vardi, 1989; Kautz and Selman, 1991]), we describe a nondeterministic algorithm for proving $S \not\models p_1 > (p_2 > \cdots (p_n > q) \cdots)$:

ND-ALGORITHM RNCF$(S, p_1, \ldots, p_n, q)$
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 $W_2, \ldots, W_{n+1} \subseteq S \cup \{p_1, \ldots, p_n\}$ such that
$W_i \subseteq W_{i-1} \cup \{p_{i-1}\}$ and $p_{i-1} \in W_i$, for $2 \le i \le n+1$;
for $i = 2$ **to** $n+1$ **do begin**
if $(W_i \models \perp)$ **then stop;**
for each $r \in (W_{i-1} - W_i)$ **do**
if $(W_i \cup \{r\} \not\models \perp)$ **then stop;**
end;
if $(W_{n+1} \not\models q)$ **then output** "no";
end.

Proposition 3.1 **RNCF**$(S, p_1, \ldots, p_n, q)$ *outputs "no" iff $S \not\models p_1 > (p_2 > \cdots (p_n > q) \cdots)$.*

Proof. (Sketch) 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 $\mathbf{c}_i = p_{n-i+1} > \mathbf{c}_{i-1}$, $2 \le i \le n$.

One can show by induction on n that $S \models \mathbf{c}$ 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 **RNCF**. It can be easily shown 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 **RNCF** correctly outputs "no". Conversely, **RNCF** outputs "no" in some computation if $S \not\models \mathbf{c}$. $\Box$

Corollary 3.2 *Given a knowledge base S and $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$, 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 $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$, 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 every $q \in S$ are Horn clauses.

Theorem 3.4 *Let S be a knowledge base and $\mathbf{c} = p_1 > (p_2 > \cdots (p_n > q) \cdots)$, where q and all p_i as well as all $p \in S$ are Horn clauses. Deciding if $S \models \mathbf{c}$ 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 **RNCF** can be done in polynomial time, and hence **RNCF** can be reformulated as an NP-algorithm. Consequently, deciding if $S \models \mathbf{c}$ is in coNP. Hardness for coNP follows for $\mathbf{c} = p_1 > q$ from [Eiter and Gottlob, 1992, Theorem 8.5], even if p_1 and q are literals. $\Box$

We remark that [Fagin *et al.*, 1983] and [Nebel, 1991] consider a refinement of the operator $\mathcal{F}(p, S)$ by the introduction of priorities. This does not only not increase the complexity (deciding $S \models p > q$ is still in Π_2^P), but makes complexity decrease when S is totally ordered (the problem is easily shown to be in Δ_2^P). Theorems 3.3 and 3.4 remain valid for the obvious generalization of priorities to nested counterfactuals. Moreover, if S is totally ordered and all formulas are Horn, we obtain the optimistic result that evaluating a nested counterfactual over S is polynomial.

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 transformation of QBFs into nested counterfactuals. Let

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

where $Q_1 = q_1$, $Q_i \neq Q_{i-1}$, for $i > 1$ and where $\underline{a}_i = a_{i,1}, \ldots, a_{i,n_i}$ is a group of $n_i \ge 1$ variables and $(Q_i \underline{a}_i)$ stands for $Q_i a_{i,1} \cdots Q_i a_{i,n_i}$, $1 \le i \le k+1$.

Let c be a new variable, and let $\underline{b}_1, \ldots, \underline{b}_k$ be groups of new variables, where $\underline{b}_i = b_{i_1}, \ldots, b_{i_{n_i}}$. Then,

- $p_i = [\underline{a}_i \not\equiv \underline{b}_i]$, for $1 \le i < k$, where $\underline{a}_i \not\equiv \underline{b}_i$ is short for $\bigwedge_{j=1}^{n_i} a_{i,j} \not\equiv b_{i,j}$.
- $p_k = p(\Phi)$, and $p(\Phi)$ is, depending on Q_{k+1}, the following formula:

  $$p(\Phi) = \bigwedge_{i=1}^{k} [\underline{a}_i \not\equiv \underline{b}_i] \wedge (E(\Phi) \vee c) \wedge (c \Rightarrow \underline{a}_{k+1}),$$

  where $E(\Phi) = \neg E$ if $Q_{k+1} = \forall$ and $E(\Phi) = E$ if $Q_{k+1} = \exists$, and $c \Rightarrow \underline{a}_{k+1}$ stands for $c \Rightarrow \bigwedge_j a_{k+1,j}$.

- $q = \neg c$.

Define a knowledge base $S(\Phi)$ and a counterfactual $\mathbf{c}(\Phi) = \mathbf{c}_k(\Phi)$ as follows:

$$S(\Phi) = \{\underline{a}_1, \underline{b}_1, \ldots, \underline{a}_k, \underline{b}_k, \neg a_{k+1,1}, \ldots, \neg a_{k+1,n_{k+1}}, \neg c\}$$

For $k \geq i > 1$,

$$\mathbf{c}_i(\Phi) = \begin{cases} p_{k-i} > \mathbf{c}_{i-1}(\Phi), & \text{if } Q_{k-i} = \forall \\ p_{k-i} \not> \neg\mathbf{c}_{i-1}(\Phi), & \text{if } Q_{k-i} = \exists \end{cases},$$

$$\mathbf{c}_1(\Phi) = \begin{cases} p_k \not> q, & \text{if } Q_{k+1} = \forall \\ p_k > q, & \text{if } Q_{k+1} = \exists \end{cases}$$

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

Intuitively, 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 $\bot$ according to a truth value assignment. Every such 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 (or not) for the encoded truth 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 $\Phi = \forall a_1 \exists a_{2,1} \exists a_{2,2} \forall a_3\, E$,

$$E = (a_1 \wedge \neg a_{2,1} \Rightarrow a_{2,2} \vee a_3).$$

Φ is rewritten as $(\forall \underline{a}_1)(\exists \underline{a}_2)(\forall \underline{a}_3)E$, where $\underline{a}_1 = a_1$, $\underline{a}_2 = a_{2,1}, a_{2,2}$, and $\underline{a}_3 = a_3$. Applying the transformation ($k+1 = 3$, $Q_1 = \forall$) we get

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

Verify that Φ is valid and that $S \models \mathbf{c}(\Phi)$. □

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

Before sketching a proof of this theorem, we note some useful lemmata.

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

Proof. (Sketch) Immediate from the proof of [Eiter and Gottlob, 1992, Lemma 6.2]. □

Lemma 4.3 *Let S and S' be knowledge bases 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, $S \models \mathbf{c}$ iff $S \cup S' \models \mathbf{c}$, and for each $\mathbf{s}_i$ from the structural sequence $\mathbf{s}_1, \ldots, \mathbf{s}_k$ of $\mathbf{c}$,*

$$T \in Cn(\mathbf{s}_i, \mathbf{c}, S) \Leftrightarrow T \cup S' \in Cn(\mathbf{s}_i, \mathbf{c}, S \cup S').$$

Proof. (Sketch) Can be shown by induction on k. □

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. (Sketch) By induction on k (use Proposition 2.2). □

Proof sketch of Theorem 4.1 Proof by induction on the number $k+1$ of quantifier alternations of Φ.

(Basis) $k+1 = 2$: By Lemma 4.2, it remains to consider $\Phi = (\exists \underline{a}_1)(\forall \underline{a}_2)E$. Since $\Phi \equiv \neg F'$ for $\Phi' = (\forall \underline{a}_1)(\exists \underline{a}_2)\neg E$, Φ is valid iff $S(\Phi') \not\models \mathbf{c}(\Phi')$, which can be shown to be equivalent to $S(\Phi) \models \mathbf{c}(\Phi)$.

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

1. $Q_1 = \forall$: In this case, Φ is valid iff for every truth assignment ϕ to $\underline{a}_1$, the formula $\Phi_\phi = (\exists \underline{a}_2) \cdots (Q_{n+1} \underline{a}_{n+1})E_\phi$ is valid, where E_ϕ is obtained from Φ by replacing in E each occurrence of a_{1_i} by $\top$ if $\phi(a_{1_i}) =$ *true* and by $\bot$ otherwise, for all i. Applying the hypothesis for $k+1$, Φ_ϕ is valid iff $S \models \mathbf{c}$, where $S = S(\Phi_\phi)$ and $\mathbf{c} = \mathbf{c}(\Phi_\phi)$. Define

$$\begin{aligned} S_\phi &= \{a_{1,j} : \phi(a_{1,j}) = \mathit{true}, 1 \leq j \leq n_1\} \cup \\ &\quad\ \{b_{1,j} : \phi(a_{1,j}) = \mathit{false}, 1 \leq j \leq n_1\} \cup \{\underline{a}_1 \not\equiv \underline{b}_1\}, \\ p' &= [\underline{a}_1 \not\equiv \underline{b}_1] \wedge \cdots \wedge [\underline{a}_{k+1} \not\equiv \underline{b}_{k+1}] \wedge \\ &\quad\ (E(\Phi) \vee c) \wedge (c \Rightarrow \underline{a}_{k+2}), \end{aligned}$$

and let $\mathbf{s}_1 = p(\Phi_\phi) > q$ be the first counterfactual in the structural sequence of $\mathbf{c}$. It can be shown using Lemma 4.3 that $S \models \mathbf{c}$ iff $S \cup S_\phi \models \mathbf{c}$ and that

$$T \in Cn(\mathbf{s}_1, \mathbf{c}, S) \Leftrightarrow T \cup S_\phi \in Cn(\mathbf{s}_1, \mathbf{c}, S \cup S_\phi).$$

It can be further shown that for every $T \in Cn(\mathbf{s}_1, \mathbf{c}, S \cup S_\phi)$, $T \models \mathbf{s}_1$ iff $T \models p' > q$. It follows from Lemma 4.4 that $S \cup S_\phi \models \mathbf{c}$ (i.e., $S \models \mathbf{c}$) iff $S \cup S_\phi \models \mathbf{c}[p' > q]$. Consequently, $S \models \mathbf{c}$ iff $S(\Phi) \cup S_\phi \models \mathbf{c}[p' > q]$. Now observe that $p' = p(\Phi)$ and $\mathbf{c}[p' > q] = \mathbf{c}_k(\Phi)$. It follows that Φ is valid iff $S(\Phi) \cup S_\phi \models \mathbf{c}_k(\Phi)$ for every ϕ. Since $\mathbf{c}_{k+1}(\Phi) = \mathbf{c}(\Phi)$ and

$$Cn(\mathbf{c}_k(\Phi), \mathbf{c}(\Phi), S(\Phi)) = \{S \cup S_\phi : \phi \text{ a truth ass. to } \underline{a}_1\},$$

Proposition 2.2 implies that Φ is valid iff $S(\Phi) \models \mathbf{c}(\Phi)$.

2. $Q_1 = \exists$: Since $\Phi \equiv \neg\Phi'$ for $\Phi' = (\forall \underline{a}_1)(\exists \underline{a}_2) \cdots (Q'_{k+2} \underline{a}_{k+2})\neg E$, using case 1.) it is not hard to show that the statement holds for $k+2$. □

We obtain the main results of this section.

Theorem 4.5 *Let S be a knowledge base and $\mathbf{c} = p_1 \not> (p_2 \not> \cdots (p_k \not> q) \cdots)$, for constant $k \geq 1$. Deciding if $S \models \mathbf{c}$ is Σ^P_{k+1}-complete.*

Proof. (Sketch) Membership in Σ^P_{k+1} can be shown by induction on k. For $k = 1$, this holds by Proposition 2.1; for $k > 1$, by the hypothesis a guess for $W \in \mathcal{F}(p_1, S)$ with $W \not\models p_2 \not> (\cdots (p_k \not> q) \cdots)$ can be verified in polynomial time with a Σ^P_k oracle. Σ^P_{k+1}-hardness follows from Theorem 4.1. □

Theorem 4.6 *Given a knowledge base S and $\mathbf{c} \in \mathcal{C}$, deciding if $S \models \mathbf{c}$ is* PSPACE*-complete. Hardness for* PSPACE *holds if* $\mathbf{c}$ *has form* $p_1 \not> (p_2 \not> \cdots (p_n \not> q) \cdots)$.

Proof. It is straightforward to design a procedure for deciding $S \models \mathbf{c}$ in polynomial space. PSPACE-hardness follows from Theorem 4.1. □

Since evaluating nested counterfactuals is in PSPACE, it is straightforward that evaluating a formula built using the standard propositional connectives from propositional atoms and nested counterfactuals over a knowledge base is in PSPACE, too.

5 Left-nested counterfactuals

We consider in this section evaluation of counterfactuals nested in the premise, i.e. $(p > q) > r$. A formal semantics for $(p > q) > r$ using the "possible worlds approach" can be defined as follows. Define

$$\begin{aligned}\mathcal{F}(p > q, S) \quad = \quad \{T \subseteq S : T \not\models p \not> q,\\ T \subset U \subseteq S \Rightarrow U \models p \not> q\}.\end{aligned}$$

Note that $T \models p > q$ for every $T \in \mathcal{F}(p > q, S)$. Now define that $(p > q) > r$ has over S value *true* (in symbols, $S \models (p > q) > r$) if $T \models r$ for every $T \in \mathcal{F}(p > q, S)$ and has value *false* otherwise ($S \not\models (p > q) > r$).

This definition can be extended to iterated nestings in the premise. Such statements are conceptually quite involved, however, and their relevance seems questionable; we do not know of an intuitive example. Future work will investigate into a semantics, though, which foresees besides removal also addition of formulas to reach from the relevant knowledge base a "possible world" of a repeatedly left-nested counterfactual.

Consider the following nondeterministic algorithm, where $\mathbf{CF}(p, q, S) = true$ iff $S \models p > q$:

ND-ALGORITHM LNCF(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 $S' \subseteq S$;
if $\neg\mathbf{CF}(S', p, q)$ **or** $(S' \models r)$ **then stop**
else
  for each T, $S' \subset T \subseteq S$ **do**
    if $\mathbf{CF}(T, p, q)$ **then stop;**
  output "no";
end.

Proposition 5.1 *$S \not\models (p > q) > r$ iff* **LNCF** *outputs "no".*

LNCF 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 explained by nonmonotony of counterfactuals.

It is easily seen that a proof for $S' \notin \mathcal{F}(p > q, S)$ can be given *nondeterministically* in 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. Clearly, a proof for $S \not\models (p > q) > r$ can be given nondeterministically in polynomial time with an oracle for $S' \notin \mathcal{F}(p > q)$ and classical inference. Thus,

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

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, $S' \subset T \subseteq S$, is Σ_3^P-hard.*

Proof. (Sketch) We transform deciding the validity of $\Phi = (\exists \underline{a}_1)(\forall \underline{a}_2)(\exists \underline{a}_3)E$ into this problem, where $\underline{a}_i = a_{i,1}, \ldots, a_{i,n_i}$, $i = 1, 2, 3$. 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 groups of new atoms. Define

$$\begin{aligned}
S' &= \{\underline{a}_2, \underline{a}'_2, c\},\\
S &= \{\underline{a}_1, \underline{a}'_1, \underline{a}_2, \underline{a}'_2, c\},\\
p &= [\ c \wedge \textstyle\bigwedge_{i=1}^{n_1}(\neg a_{1,i} \wedge \neg a'_{1,i} \wedge a''_{1,i})\ ] \ \vee\\
&\quad [\ \textstyle\bigwedge_{i=1}^{n_1}[(a_{1,i} \wedge a'_{1,i} \Rightarrow \neg c) \wedge (a_{1,i} \vee a'_{1,i} \Rightarrow a''_{1,i})] \wedge\\
&\quad \textstyle\bigwedge_{i=1}^{n_2}(a_{2,i} \not\equiv a'_{2,i}) \wedge (c \Rightarrow E)\ ],\\
q &= c \wedge \textstyle\bigwedge_{i=1}^{n_1} a''_{1,i}.
\end{aligned}$$

Notice that S', S, p and q are constructible in polynomial time. It is not hard to see that $S' \models p > q$. Moreover, it can be shown that there exists T, $S' \subset T \subseteq S$, such that $T \models p > q$ iff Φ is valid. □

Theorem 5.4 *Deciding if $S \models (p>q)>r$ from S and $(p>q)>r$ is Π_4^P-hard.*

Proof. (Sketch) The proof is an extension of the transformation in the proof of Lemma 5.3. Let

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

$\underline{b}^* = b_1, \ldots, b_{n_b}$, and let $\underline{b} = b_1, \ldots, b_{2n_b}$, $\underline{b}' = b'_1, \ldots, b'_{2n_b}$, $\underline{b}'' = b''_1, \ldots, b''_{2n_b}$ using new atoms. Define

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

where S_1 (resp. p_1, q_1) is constructed as S (resp. p, q) in the proof of Lemma 5.3. It can be shown that $S \models (p > q) > r$ iff Φ is valid. □

Corollary 5.5 *Deciding if $S \models (p > q) > r$ from S and $(p > q) > r$ is Π_4^P-complete.*

6 Related work and conclusion

Complexity characterizations of evaluating counterfactuals are given in [Winslett, 1990; Nebel, 1991; Grahne, 1991; Grahne and Mendelzon, 1991]. Grahne and Mendelzon [Grahne and Mendelzon, 1991] considered subjunctive queries in a different framework, where the knowledge base is given by a set of models and updates are performed according to Winslett's method [Winslett, 1988]. In particular, [Grahne and Mendelzon, 1991, Corollary 4.2] implies that evaluating nested counterfactuals under this update semantics is PSPACE-complete.

Our work contributes to the recent effort in giving a precise complexity characterization of nonmonotonic reasoning in the full propositional context, cf. [Niemelä, 1991; Winslett, 1990; Nebel, 1991; Rutenburg, 1991; Eiter and Gottlob, 1992; Stillman, 1992] (see [Cadoli and Schaerf, 1992] for an overview), extending previous results for restricted contexts, e.g. [Kautz and Selman, 1991; Stillman, 1990; Selman and Levesque, 1990; Cadoli and Lenzerini, 1990; Provan, 1990]. 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 [Selman *et al.*, 1992] 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.

Acknowledgments

The authors would like to thank two referees for their valuable comments.

References

[Cadoli and Lenzerini, 1990] Marco Cadoli and Maurizio Lenzerini. The Complexity of Closed World Reasoning and Circumscription. In *Proceedings AAAI-90*, pages 550–555, 1990.

[Cadoli and Schaerf, 1992] M. Cadoli and M. Schaerf. A Survey on Complexity Results for Non-monotonic Logics. Technical report, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1992. Journal of Logic Programming, to appear.

[Eiter and Gottlob, 1992] Thomas Eiter and Georg Gottlob. On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals. *Artificial Intelligence*, 57(2–3):227–270, 1992.

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

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

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

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

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

[Grahne and Mendelzon, 1991] Gösta Grahne and Alberto O. Mendelzon. Updates and Subjunctive Queries. Technical Report KRR-TR-91-4, University of Toronto, CS Dept., July 1991. *Information and Computation*, to appear.

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

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

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

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

[Provan, 1990] Gregory Provan. The Computational Complexity of Multiple-Context Truth Maintenance Systems. In *Proceedings ECAI-90*, pages 522–527, 1990.

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

[Rutenburg, 1991] V. Rutenburg. Complexity Classification in Truth Maintenance Systems. In *Proceedings STACS-91*, pages 373–383, 1991.

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

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

[Stillman, 1990] 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.

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

[Vardi, 1989] 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.

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

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