Updates and Subjunctive Queries*†

Gösta Grahne‡
Department of Computer Science
University of Helsinki
Teollisuuskatu 23
SF-00510 Helsinki, Finland

Alberto O. Mendelzon
Department of Computer Science
University of Toronto
Toronto, Canada, M5S 1A4

October 27, 1993

*This work was supported by the Natural Sciences and Engineering Research Council of Canada and by the Institute for Robotics and Intelligent Systems.

†A preliminary version of this paper appeared in "Workshop on Nonstandard Queries and Answers," (R. Demolombe, L. Fariñas del Cerro, and T. Imielinski, Eds.), Toulouse 1991.

‡Work performed while visiting Department of Computer Science, University of Toronto.

Contact Author:
Alberto O. Mendelzon
Department of Computer Science
University of Toronto
Toronto, Canada, M5S 1A4
email: mendel@db.toronto.edu

Table of Contents:

Abstract

A subjunctive query of the form $\phi > \psi$ means 'if ϕ were true in the knowledgebase, would ψ also necessarily be true?' We propose the following semantics for subjunctive queries: $\phi > \psi$ will hold in the current knowledgebase `T` if ψ holds in the result of updating `T` with ϕ. This is known as the *Ramsey test* in philosophy. We adapt the *model checking approach* of Halpern and Vardi: A knowledgebase is a finite set of finite sets of positive facts interpreted in a closed world setting. We then use Winslett's possible models approach to give semantics to knowledgebase updates, and we introduce a query language which is essentially propositional logic, augmented with a subjunctive conditional that has an intensional interpretation in our model. We show that query answering and update can be performed in time polynomial in the size of the knowledgebase. However, query equivalence is shown to be complete in polynomial space, and this is also the complexity of query answering as a function of query size. We give a sound axiomatization of query equivalence and show that the update operator satisfies the postulates for updates adapted by Katsuno and Mendelzon from the Alchourrón-Gärdenfors-Makinson belief revision postulates.

1 Introduction

Imagine the following queries. 'If I place a gate at this particular position in the integrated circuit I am designing, will the design rules be violated?', or 'If my child is born in Canada, will I be eligible for Canadian citizenship?', or 'Is there any single course I can take which will let me graduate this year?', or 'If I were to make flight connections, could I fly to the Rockies?'.

The above are examples of *subjunctive queries*; they ask the database what would happen if the world were a certain way. Let us represent by formula $\phi > \psi$ the sentence 'if the world were such that ϕ, it would necessarily also be such that ψ'. As philosophers have abundantly shown, giving semantics to such statements is not without its intricacies (Goodman, 1955, Lewis, 1973). A very natural approach was proposed by F. P. Ramsey (1950). Consider all 'closest' worlds to the current one where ϕ holds, and see if they all satisfy ψ. If so, say $\phi > \psi$ holds in the current world. The difficulty now is to define what are the closest worlds where ϕ holds.

The first thing we have to settle to attack this difficulty is how our knowledge about the current world is represented. The standard approach in Knowledge Representation (and in much of Database Theory) is to assume a certain set of sentences `T` in a formal language represents the set of possible current worlds. Given a new sentence ϕ, we need to find a modified theory `T'` that describes the closest worlds wherein ϕ holds. Formulated in this way, the problem of defining the closest worlds where a certain sentence holds becomes what in Artificial Intelligence is known as the *belief revision* problem (Alchourrón *et al.*, 1985) and in Database Theory as the *update semantics* problem (Bancilhon and Spyratos, 1981, Fagin *et al.*, 1983, Cosmadakis and Papadimitriou, 1984). Hence, the semantics of a subjunctive query should be clear: update the database with sentence ϕ, and then check whether ψ holds in the result. There are two problems with this. The first problem is to define a comprehensible semantics of updates under arbitrary sentences ϕ, especially taking into account the cases where ϕ is inconsistent with `T`. The second problem is that reasonable proposals for such semantics lead to update mechanisms which are of infeasible data complexity, *i.e.*, exponential on the size of the theory `T`.

Let us examine more closely the semantic problem, that is, what should be the result of updating a theory `T` with a sentence ϕ? The departure point of recent work in belief revision is the *rationality postulates* proposed by Alchourrón, Gärdenfors and Makinson (1985) and colloquially known as the the *AGM postulates*. These are principles that every adequate belief revision operator should be expected to satisfy. For example: the new fact ϕ must be a consequence of the revised theory. And: if the new fact ϕ is consistent with `T`, then the result should be logically equivalent to $\mathtt{T} \wedge \phi$.

Katsuno and Mendelzon (1991) pointed out that these postulates are not universally desirable for all kinds of belief revision applications. In particular, they distinguished two kinds of theory change operations, *update* and *revision*. Update consists of bringing the knowledgebase up to date when the world described by it changes. For example, most database updates are of this variety, *e.g.* 'increase Joe's salary by 5%'. The second type of modification, *revision*, is used when new information is obtained about a static world. For example, we may be trying to diagnose a faulty circuit and want to incorporate into the knowledgebase the results of successive tests, where newer results may contradict old ones. Katsuno and Mendelzon came to the conclusion that the *AGM* postulates describe only revision, and they give a modified set of postulates that characterize update operators.

For example, suppose two robot vehicles V and W are orbiting Venus. We have received the message 'I have landed', but due to noise we could not determine whether it came from V or

W. Let v be the proposition 'V has landed' and w 'W has landed.' After receiving the message, the knowledgebase $\mathtt{T}$ is the sentence

$$(v \wedge \neg w) \vee (\neg v \wedge w).$$

Suppose we now send the command 'Land immediately' to V, and V replies 'I have landed'. This change can be modeled by incorporating the sentence v into $\mathtt{T}$. Since v is consistent with $\mathtt{T}$, the *AGM* postulate cited above says the result should be equivalent to $\mathtt{T} \wedge v$, which is equivalent to $v \wedge \neg w$. But upon reflection it becomes clear that this is incorrect. After V has landed, all we know is that V has landed; there is no reason to conclude that W has not. The correct answer should be v. To coin a phrase of J. Biskup (1983), we are 'opening up the world'.

Going back to our original question of how to give semantics to subjunctive queries of the form 'does $\phi > \psi$?', it seems we now have a choice of using update or revision to define the closest worlds where ϕ holds. But, as it turns out, Gärdenfors (1988) has shown that using revision to define the semantics of subjunctive statements leads to triviality, while Grahne (1991) has constructed a non-trivial, decidable propositional system in which update is used to define subjunctive implication.

In sum, we propose the following semantics for subjunctive queries: $\phi > \psi$ will hold in the current database $\mathtt{T}$ if the result of updating $\mathtt{T}$ with ϕ has ψ has a logical consequence. All update operators which satisfy the postulates of (Katsuno and Mendelzon, 1991) will be acceptable; in particular, Winslett's *possible models approach* (Winslett, 1988b) seems like a reasonable candidate.

But now we have to face the computational consequences of this definition. If $\mathtt{T}$ is a propositional theory, computing the result of updating $\mathtt{T}$ with ϕ using Winslett's operator is at least *NP-hard* on the size of $\mathtt{T}$ (Winslett, 1988a). One way of dealing with this computational complexity would be to look for worst-case exponential time algorithms that may perform well in some practical situations. This is the approach taken by Del Val (1992). He gives algorithms for updating arbitrary propositional knowledgebases in conjunctive normal form and discusses in detail their running time in certain situations, citing experimental results.

Our approach is different. We observe that the source of the complexity is conciseness. Very short theories can describe very large sets of possible worlds. On the other hand, it is not *a priori* given that any practical set of possible worlds has such a concise theory. Suppose now, that instead of storing theories, we actually store worlds. We can then replace theorem proving by *model checking*, as suggested by Halpern and Vardi (1991):

> We argue that rather than representing an agent's knowledge as a collection of formulas, and then doing theorem proving to see if a given formula follows from an agent's knowledgebase, it may be more useful to represent an agent's knowledge by a semantic model, and then do model checking to see if the given formula is true in that model.

This is the approach we take. We define a world as a finite set of atomic facts, and a knowledgebase as a finite set of worlds. The approach is somewhat similar to the use of *flocks* of theories by Fagin *et al.* (1986); however, flocks are sets of theories, while knowledgebases are sets of models. We follow the *closed world assumption*: only the facts that are explicitly stored are true in a world (Reiter, 1978, Minker, 1982). Thus defined, a knowledgebase could also be seen as a propositional formula in disjunctive normal form, equipped with the axioms for Minker's *generalized closed world assumption*. These axioms —the *circumscription axioms*— would however require quantification over infinite sets of propositions (*cf.* Cadoli and Lenzerini, 1991). By adopting the model checking approach we achieve the advantage of not having to resort to the stronger quantified language in expressing the knowledgebase (*cf.* Halpern and Vardi, 1991).

We propose a query and update formalism that allows the expression of general subjunctive queries on knowledgebases. Our approach allows us to develop query answering algorithms that run in time polynomial in the size of the knowledgebase. The price we pay is that certain updates can cause the knowledgebase to grow much faster than the corresponding propositional theory.

Our language is related to that of (Bonner, 1988a, Bonner, 1988b, Bonner *et al.*, 1989), who study intuitionistic database languages in which it is possible to have subjunctions of the form $\phi > \psi$, where ϕ is a conjunction of atomic formulas, and where ψ is atomic, embedded in the the bodies of clauses. When negated literals are used in these languages, negation is however interpreted as failure.

The proposal of Ginsberg (1986) is related to the "syntactic school" of (Fagin *et al.*, 1983, 1986). According to Ginsberg, the subjunction $\phi > \psi$ is entailed by a theory $\mathtt{T}$, if ψ is entailed by all maximal subsets of $\mathtt{T}$ that are consistent with p. Naturally, this definition is sensitive to the syntactic form of $\mathtt{T}$. Insensitivity to syntax is required both in the Alchourrón-Gärdenfors-Makinson belief revision postulates and the Katsuno-Mendelzon update postulates. It has also recently been shown by Eiter and Gottlob (1992) that the syntactic approaches are computationally more expensive than approaches satisfying the belief revision or update postulates.

Other approaches to subjunctive queries are also reported in *e.g.* (Gabbay, 1985, Gabbay and Reyle, 1984).

Section 2 of this paper contains the definition of subjunctive queries and updates in knowledgebases. In Section 3 we study the data complexity of the operators, and show that queries and updates can be computed in time polynomial in the size of the knowledgebase. The expression complexity of queries and updates is the topic of Section 4. We show that, from this point of view, query evaluation is complete in polynomial space, and that the size of the updated knowledgebase can be exponential in the size of the formula to be inserted. Section 5 contains a discussion on query optimization. We present a sound axiomatization for query transformation, and show that testing for query equivalence is complete in polynomial space. Finally, in Section 6 we consider general properties of change operators, and show that our update operator satisfies the postulates of (Katsuno and Mendelzon, 1991).

2 The Framework

Let T be a countably infinite set of *facts*, also called *terms*. We use the name $\mathcal{R}_T$ for the set $\{R : R \subset_{fin} T\}$. Since T is tacitly understood, we will write $\mathcal{R}_T$ simply as $\mathcal{R}$. Each $R \in \mathcal{R}$ is called a *world*.

For each world R we define a binary relation $\leq_R$ over $\mathcal{R}$, given by

$$S \leq_R S' \text{ iff } (S \setminus R) \cup (R \setminus S) \subseteq (S' \setminus R) \cup (R \setminus S').$$

In other words, $S \leq_R S'$ iff the symmetric difference between S and R is a subset of the symmetric difference between S' and R. It is easily verified that $\leq_R$ is a partial order on $\mathcal{R}$. We shall also use a relation $<_R$. The relation $S <_R S'$ holds, iff $S \leq_R S'$ holds, and S is not equal to S'.

Let $\mathcal{S} \subseteq \mathcal{R}$, and $\{R, S\} \subseteq \mathcal{R}$. Then we say that S is $\leq_R$*-minimal in* $\mathcal{S}$, if S is in $\mathcal{S}$, and if $S' \in \mathcal{S}$ and $S' \leq_R S$, then $S' = S$. [1]

$\mathcal{KB}$ is the set $\{KB : KB \subset_{fin} \mathcal{R}\}$. Each $KB \in \mathcal{KB}$ is called a *knowledgebase*. Note that each world in a KB stores only positive information; all terms not explicitly asserted in a world are assumed to be false in that world in our query semantics.

We say that each $t \in T$ is a *term*. From terms we can build *formulas* using the Boolean connectives $\perp, \wedge, \neg$, and the intensional connective $>$. The 0-ary connective $\perp$ denotes the constant *false*. Letters like ϕ and ψ will be used for formulas. Intuitively $\phi > \psi$ means 'if ϕ *were* true, ψ *would* be true'. The set of all formulas is denoted Φ (T is understood). The *interpretation* of a formula ϕ is denoted $[\![\phi]\!]$, and defined as follows.

$$
\begin{aligned}
[\![\perp]\!] &= \emptyset && (1)\\
[\![t]\!] &= \{R \in \mathcal{R} : t \in R\} && (2)\\
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!] && (3)\\
[\![\neg\phi]\!] &= \mathcal{R} \setminus [\![\phi]\!] && (4)\\
[\![\phi > \psi]\!] &= \{R \in \mathcal{R} : \mu(\phi, R) \subseteq [\![\psi]\!]\}, \text{where} &&\\
\mu(\phi, R) &= \{S \in \mathcal{R} : S \text{ is } \leq_R \text{-minimal in } [\![\phi]\!]\} && (5)
\end{aligned}
$$

For convenience, we shall introduce the following abbreviations: $\top$ abbreviates $\neg\perp$, $\phi \vee \psi$ abbreviates $\neg((\neg\phi) \wedge (\neg\psi))$, $\phi \rightarrow \psi$ abbreviates $(\neg\phi) \vee \psi$, $\phi \leftrightarrow \psi$ abbreviates $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, and $\phi \supset \psi$ abbreviates $\neg(\phi > \neg\psi)$. The intuitive meaning of the intensional connective $\supset$ is the following: $\phi \supset \psi$ is read 'if ϕ *were* true, ψ *might* be true'. These defined connectives have the following semantic characterizations.

Lemma 2.1

$$
\begin{aligned}
[\![\top]\!] &= \mathcal{R} && (6)\\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] \cup [\![\psi]\!] && (7)\\
[\![\phi \rightarrow \psi]\!] &= (\mathcal{R} \setminus [\![\phi]\!]) \cup [\![\psi]\!] && (8)\\
[\![\phi \leftrightarrow \psi]\!] &= ([\![\phi]\!] \cap [\![\psi]\!]) \cup ((\mathcal{R} \setminus [\![\phi]\!]) \cap (\mathcal{R} \setminus [\![\psi]\!])) && (9)\\
[\![\phi \supset \psi]\!] &= \{R \in \mathcal{R} : \mu(\phi, R) \cap [\![\psi]\!] \neq \emptyset\} && (10)
\end{aligned}
$$

Equations (5) and (10) are illustrated diagrammatically in Figure 1 below.

[1] Note that as a consequence of the fact that $\mathcal{R}$ consists of finite worlds, each nonempty subset of $\mathcal{R}$ contains at least one $\leq_R$-minimal element.

8

Figure 1. Interpretations of subjunctive sentences.

The first case shows a world R, such that all the $\leq_R$-minimal worlds in $[\![\phi]\!]$ are ψ-worlds, so R belongs to $[\![\phi > \psi]\!]$, and hence also to $[\![\phi \supset \psi]\!]$. The second case shows a world R, such that some $\leq_R$-minimal worlds are ψ-worlds, but not all; hence R belongs to $[\![\phi \supset \psi]\!]$, but not to $[\![\phi > \psi]\!]$. The third case shows a world R, such that no $\leq_R$-minimal world is a ψ-world. Hence R does not belong to $[\![\phi \supset \psi]\!]$ (and consequently not to $[\![\phi > \psi]\!]$ either).

Let us now introduce two *interface functions*, *Ask* and *Tell*. Formally, *Ask* is a function from $\Phi \times \mathcal{KB}$ to $\{$*yes, no, possible*$\}$, and *Tell* is a function from $\Phi \times \mathcal{KB}$ to $\mathcal{KB}$, defined as follows.

$$\mathit{Ask}(\phi, \mathit{KB}) = \begin{cases} \mathit{yes} & \text{if } \mathit{KB} \subseteq [\![\phi]\!] \\ \mathit{no} & \text{if } \mathit{KB} \cap [\![\phi]\!] = \emptyset \\ \mathit{possible} & \text{otherwise} \end{cases} \tag{11}$$

$$\mathit{Tell}(\phi, \mathit{KB}) = \bigcup_{R \in \mathit{KB}} \mu(\phi, R) \tag{12}$$

In Theorem 3.2 we will show that the range of *Tell* is indeed $\mathcal{KB}$.

We can now show that our language satisfies the Ramsey property.

Ramsey's Lemma 2.2 $\mathit{Ask}(\phi > \psi, \mathit{KB}) = \mathit{Ask}(\psi, \mathit{Tell}(\phi, \mathit{KB}))$.

As to the properties of the *Tell* function, we shall show in Section 6 that *Tell* satisfies the update postulates of (Katsuno and Mendelzon, 1991).

A diagram of the *Tell*-function appears in Figure 2 below. The diagram shows how *KB*-worlds are mapped to their closest ϕ-worlds by $\mathit{Tell}(\phi, \mathit{KB})$. The figure only shows the case where none of the KB-worlds is itself a ϕ-world, that is, the *KB* and ϕ are inconsistent. If the *KB* is consistent with ϕ, then every ϕ-world in the *KB* is mapped to itself by $\mathit{Tell}(\phi, \mathit{KB})$.

Figure 2. The *Tell*-function.

Let us now return to the two vehicles orbiting Venus. The set of all facts T includes the facts v, "vehicle V has landed", and w, "vehicle W has landed." Suppose that before receiving the message the knowledgebase—we will call it KB_0—contains the four possible worlds $\{v\}$, $\{w\}$, $\{v, w\}$, and $\emptyset$. Note that, for any propositional formula ϕ, $Ask(\phi, KB_0) = yes$ if and only if ϕ is a tautology on the propositional letters v and w, and $Ask(\phi, KB_0) = no$ if and only if ϕ is unsatisfiable. On learning that exactly one of the vehicles has landed, we update this knowledgebase to

$$KB_1 = Tell((v \wedge \neg w) \vee (\neg v \wedge w), KB_0) = \{\{v\}, \{w\}\}.$$

We shall now evaluate Ask for some queries on KB_1:

$$Ask(v, KB_1) = Ask(w, KB_1) = possible,$$

$$Ask(\neg v, KB_1) = Ask(\neg w, KB_1) = possible,$$

$$Ask(v \wedge w, KB_1) = no,$$

$$Ask((v \wedge \neg w) \vee (\neg v \wedge w), KB_1) = yes.$$

Now consider the subjunctive query $Ask(v > \neg w, KB_1)$. That is, if V lands, will W necessarily be still orbiting? To answer the query, we need to test membership in $[\![v > \neg w]\!]$. Conceptually, $[\![v > \neg w]\!]$ is the set of worlds R in $\mathcal{R}$ such that, if we update R to make v hold, we get a world where $\neg w$ also holds. The world $\{v\}$ is such a world, but the world $\{w\}$ is not; we conclude that $Ask(v > \neg w, KB_1) = possible$. That is, if V lands, W may or may not have landed. Note that, as indicated by Ramsey's lemma, this is the same as evaluating the query $Ask(\neg w, Tell(v, KB_1))$. The update $Tell(v, KB_1) = \{\{v\}, \{v, w\}\}$ is the one we computed in the introduction.

3 Data Complexity

In this section we show that $Ask(\phi, KB)$ and $Tell(\phi, KB)$ can both be computed in time polynomial in the size of KB, for any formula $\phi \in \Phi$.

We proceed 'bottom up', by some technical results. The first objective is to show that membership of a world in $[\![\phi]\!]$ depends only on the terms mentioned explicitly in the sentences ϕ. We start by some technical lemmas.

We shall 'overload' the symbol $\supset$. It should be clear from the context whether $\supset$ stands for the intensional 'might' connective, or whether $\supset$ denotes the the usual proper superset relation.

Lemma 3.1 *$S' \leq_R S$ iff $(S' \setminus R) \subseteq (S \setminus R)$ and $(R \setminus S') \subseteq (R \setminus S)$.*

Lemma 3.2 (i). *$S \cap U \leq_{R \cap U} S$.*
(ii). *$S \cup V \leq_{R \cup V} S$.*

Proof. *(i).* We have $((S \cap U) \setminus (R \cap U)) \subseteq (S \setminus (R \cap U))$, and $((R \cap U) \setminus (S \cap U)) = ((R \cap U) \setminus S)$.
(ii). $((S \cup V) \setminus (R \cup V)) \cup ((R \cup V) \setminus (S \cup V)) = (S \setminus (R \cup V)) \cup (R \setminus (S \cup V)) \subseteq (S \setminus (R \cup V)) \cup ((R \cup V) \setminus S)$.**QED**

Lemma 3.3 (i). *If $S' \leq_R S$, then $S' \cap U \leq_{R \cap U} S \cap U$, and*
(ii). *If $S' \leq_R S$, then $S' \cup V \leq_{R \cup V} S \cup V$.*

Proof. *(ii).* Taking the fact that $(A \cup C) \setminus (B \cup C) = A \setminus (B \cup C)$, Lemma 3.1, and the symmetrical nature of the two cases, it is sufficient to show that $(S' \setminus R) \subseteq (S \setminus R)$ only if $(S' \setminus (R \cup V)) \subseteq (S \setminus (R \cup V))$.

For a contradiction, suppose that $t \in (S' \setminus (R \cup V)) \setminus (S \setminus (R \cup V))$. Since $(S' \setminus (R \cup V)) \subseteq (S' \setminus R) \subseteq (S \setminus R)$, we get $t \in (S \setminus R) \setminus (S \setminus (R \cup V))$, which implies $t \in V$, and thus $t \in (R \cup V)$; a contradiction to the assumption that $t \in (S' \setminus (R \cup V))$.

(i). Let $t \in ((S' \cap U) \setminus (R \cap U))$, i.e. $t \in (S' \setminus R) \cap U$, i.e. $t \in S' \setminus R$, and $t \in U$. We now have $t \in S \setminus R$, and $t \in U$, that is, $t \in ((S \cap U) \setminus (R \cap U))$. **QED**

For the next result we shall need a definition: For each $\phi \in \Phi$, let L_ϕ be the set of all terms mentioned in ϕ, i.e. $L_\perp = \emptyset$, $L_t = \{t\}$, $L_{\neg\phi} = L_\phi$, and $L_{\phi \wedge \psi} = L_{\phi > \psi} = L_\phi \cup L_\psi$.

Theorem 3.1 *If $R \in [\![\phi]\!]$, then*
(i). $(R \cap U) \in [\![\phi]\!]$, for all $U \supseteq L_\phi$, and
(ii). $(R \cup V) \in [\![\phi]\!]$, for all (finite) $V \subseteq (T \setminus L_\phi)$.

Proof. We proceed by an induction on the structure of ϕ.

(base). Let $t \in T$, and suppose $R \in [\![t]\!]$, i.e. $t \in R$. If $t \in U$, then obviously $t \in (R \cap U)$, i.e. $(R \cap U) \in [\![t]\!]$. Obviously also $t \in R$ implies that $t \in (R \cup V)$.

For the other base case, we note that the antecedent $R \in [\![\perp]\!]$ of the theorem is always false.

(inductive step). Suppose $R \in [\![\psi \wedge \chi]\!]$. Thus $R \in [\![\psi]\!]$, and $R \in [\![\chi]\!]$. By the inductive hypothesis, we have $(R \cap U) \in [\![\psi]\!]$, for all $U \supseteq L_\psi$, and thus in particular for all $U \supseteq (L_\psi \cup L_\chi)$. Using a symmetric argument for $[\![\chi]\!]$, and the fact that $(L_\psi \cup L_\chi) = L_{\psi \wedge \chi}$, we get $R \cap U \in [\![\psi \wedge \chi]\!]$, for all $U \supseteq L_{\psi \wedge \chi}$.

By the inductive hypothesis we also get that $R \cup V \in [\![\psi]\!]$, for all $V \subseteq (T \setminus L_\psi)$, and thus in particular for all $V \subseteq (T \setminus L_{\psi \wedge \chi})$. The claim follows.

For the next case, suppose $R \in [\![\neg\psi]\!]$. First note that $L_{\neg\psi} = L_\psi$. Then, towards a contradiction, suppose that there is a $U \supseteq L_\psi$, such that $R \cap U \notin [\![\neg\psi]\!]$. Thus $R \cap U \in [\![\psi]\!]$. Since $(R \setminus U) \subseteq (T \setminus L_\psi)$, the inductive hypothesis yields $((R \cap U) \cup (R \setminus U)) \in [\![\psi]\!]$, i.e. $R \in [\![\psi]\!]$; a contradiction.

Towards the next contradiction, suppose that there is a $V \subseteq (T \setminus L_\psi)$, such that $R \cup V \notin [\![\neg\psi]\!]$. Then $R \cup V \in [\![\psi]\!]$. Now $(R \cup L_\psi) \supseteq L_\psi$, so by the inductive hypothesis $((R \cup V) \cap (R \cup L_\psi)) \in [\![\psi]\!]$. Since $V \cap L_\psi = \emptyset$, we get $((R \cup V) \cap (R \cup L_\psi)) = R$, which gives the desired contradiction.

For the final case, suppose that $R \in [\![\psi > \chi]\!]$. Note here that $L_{\psi > \chi} = L_\psi \cup L_\chi$. We claim that if $U \supseteq L_{\psi > \chi}$, then $(R \cap U) \in [\![\psi > \chi]\!]$. This amounts to showing that if $\mu(\psi, R) \subseteq [\![\chi]\!]$, then $\mu(\psi, R \cap U) \subseteq [\![\chi]\!]$. Let therefore $S \in \mu(\psi, R \cap U)$. From Lemma 3.2 (i), and the fact that $\leq_{R \cap U}$ is a partial order, it follows that $S = (S \cap U)$.

Now S is in $[\![\psi]\!]$, so if S is $\leq_R$-minimal in $[\![\psi]\!]$, we are done.

Suppose therefore that there is a $S' <_R S$, where $S' \in \mu(\psi, R)$. Now Lemma 3.3 (i) gives $S' \cap U \leq_{R \cap U} S \cap U$. Since $S \in \mu(\psi, R \cap U)$, and $S = (S \cap U)$, we get $S = (S' \cap U)$. By the inductive hypothesis, we have $(S' \cap U) \in [\![\chi]\!]$, and therefore $S \in [\![\chi]\!]$.

To conclude, we show that if $R \in [\![\psi > \chi]\!]$, and $V \subseteq (T \setminus L_{\psi > \chi})$, then $(R \cup V) \in [\![\psi > \chi]\!]$. Again, we show that if $\mu(\psi, R) \subseteq [\![\chi]\!]$, then $\mu(\psi, R \cup V) \subseteq [\![\chi]\!]$.

Let therefore $S \in \mu(\psi, R \cup V)$. Then it must be that $V \subseteq S$, since Lemma 3.2 (ii) gives $S \cup V \leq_{R \cup V} S$, and S is $\leq_{R \cup V}$-minimal in $[\![\psi]\!]$. By the inductive hypothesis, $S \cap (S \setminus V) = (S \setminus V)$ is in $[\![\psi]\!]$.

If $(S \setminus V) \in \mu(\psi, R)$, then, by the inductive hypothesis, $S = ((S \setminus V) \cup (V \cap S)) \in [\![\chi]\!]$.

If $(S \setminus V) \notin \mu(\psi, R)$, there must be a S', such that $S' <_R (S \setminus V)$, and $S' \in \mu(\psi, R)$. Now Lemma 3.3 (ii) gives us the fact that $S' \cup V \leq_{R \cup V} S$. Since S is in $\mu(\psi, R \cup V)$, and $\leq_{R \cup V}$ is a partial order, it must be that $(S' \cup V) = S$. Again, since $S' \in [\![\chi]\!]$, the inductive hypothesis gives the fact that $S = (S' \cup V) \in [\![\chi]\!]$.**QED**

The following key result shows that $\mu(\phi, R)$, the set of worlds in $[\![\phi]\!]$ that are 'closest' to world R, is essentially independent of any terms not mentioned in ϕ. We shall use the following notation. Let U be a subset of T. By $\mu^U(\phi, R)$ we then mean the set

$$\{S : S \text{ is an } \leq_{R \cap U} \text{ minimal element, such that } S \subseteq U \text{ and } S \in [\![\phi]\!]\}.$$

Theorem 3.2 $\mu(\phi, R) = \{S \cup (R \setminus L_\phi) : S \in \mu^{L_\phi}(\phi, R)\}$.

Proof. Let $S \in \mu^{L_\phi}(\phi, R)$. By Theorem 3.1 (ii) we have $(S \cup (R \setminus L_\phi)) \in [\![\phi]\!]$. Suppose then to the contrary that $(S \cup (R \setminus L_\phi)) \notin \mu(\phi, R)$. Then there is a $S' \in \mu(\phi, R)$, such that $S' <_R (S \cup (R \setminus L_\phi))$. From Lemma 3.3 (i) it follows that $S' \cap L_\phi \leq_{R \cap L_\phi} (S \cup (R \setminus L_\phi)) \cap L_\phi = S$. Since $S \in \mu^{L_\phi}(\phi, R)$, we have $S' \cap L_\phi = S$.

We shall now derive a contradiction. From $S' <_R (S \cup (R \setminus L_\phi))$ it follows that either $S' \setminus R \subset S \setminus R$, or $R \setminus S' \subset R \setminus (S \cup (R \setminus L_\phi))$.

Consider the first case. We have $S' \setminus R = ((S' \cap L_\phi) \setminus R) \cup ((S' \setminus L_\phi) \setminus R)$, and since $S' \cap L_\phi = S$, we get $S' \setminus R = (S \setminus R) \cup ((S' \setminus L_\phi) \setminus R)$. Therefore the first case is not possible.

Then consider the second case. We have $R \setminus S' = ((R \cap L_\phi) \setminus (S' \cap L_\phi)) \cup ((R \setminus L_\phi) \setminus (S' \setminus L_\phi))$. On the other hand we have $R \setminus (S \cup (R \setminus L_\phi)) = (R \cap L_\phi) \setminus S = (R \cap L_\phi) \setminus (S' \cap L_\phi)$. Thus the second case is not possible either. Consequently it must be our counterassumption that is wrong.

For inclusion in the other direction, let $S \in \mu(\phi, R)$. We shall first show that $S \setminus L_\phi = R \setminus L_\phi$. Suppose to the contrary that there is a $t \in (S \setminus L_\phi)$, such that $t \notin (R \setminus L_\phi)$. But this would imply that $S \setminus \{t\} <_R S$; a contradiction. Likewise, the existence of a $t \in (R \setminus L_\phi) \setminus (S \setminus L_\phi)$ would imply $S \cup \{t\} <_R S$. Therefore it must be that $S \setminus L_\phi = R \setminus L_\phi$.

We shall now proceed to show that $S \cap L_\phi \in \mu^{L_\phi}(\phi, R)$. The desired inclusion then follows, since we now are in a position to conclude that $(S \cap L_\phi) \cup (R \setminus L_\phi) = S$.

By Theorem 3.1 (i) we have $S \cap L_\phi \in [\![\phi]\!]$. If, as a counterassumption, $S \cap L_\phi \notin \mu^{L_\phi}(\phi, R)$, there must be a $S' \in [\![\phi]\!]$, such that $S' \subseteq L_\phi$, and $S' <_{R \cap L_\phi} S \cap L_\phi$. From Lemma 3.3 (ii) it then follows that $S' \cup (R \setminus L_\phi) \leq_R S$. Since $S \in \mu(\phi, R)$, we get $S' \cup (R \setminus L_\phi) = S$, and consequently that $S' = S \cap L_\phi$; a contradiction.**QED**

Consider now the following two algorithms and the procedures they use:

```
ALGORITHM 1.  Evaluation of Ask(φ, KB).

INPUT.        A formula φ ∈ Φ, and a knowledgebase KB ∈ 𝒦ℬ.

OUTPUT.       The value of Ask(φ, KB).

METHOD.       if and({ eval(φ, R) : R ∈ KB}) = true
              then output('yes')
              else if or({ eval(φ, R) : R ∈ KB}) = true
                   then output('possible')
                   else output('no');

END OF ALGORITHM 1.
```

```
ALGORITHM 2.  Computation of Tell(φ, KB).

INPUT.        A formula φ ∈ Φ, and a knowledgebase KB ∈ 𝒦ℬ.

OUTPUT.       The value of Tell(φ, KB).

METHOD.       output( ⋃_{R∈KB} mu(φ, R));

END OF ALGORITHM 2.
```

```
function eval(φ, R) : boolean;
begin
    case the form of φ of :
        ⊥     : eval := false;
        t     : eval := t ∈ R;
        ¬ψ    : eval := not( eval(ψ, R));
        ψ ∧ χ : eval := and({ eval(ψ, R), eval(χ, R)});
        ψ > χ : eval := and({ eval(χ, S) : S ∈ mu(ψ, R)});
    end case;
end;
```

```
function mu(φ, R) : set;
begin
    closest := {S : S ⊆ L_φ};
    for each U ∈ closest do
        if not( eval(φ, U)) then closest := closest \ {U};
    for each U ∈ closest do
        for each V ∈ closest do
            if V <_{R∩L_φ} U then closest := closest \ {U};
    for each S ∈ closest do
        closest := (closest \ {S}) ∪ {S ∪ (R \ L_φ)};
    mu := closest;
end;
```

Since the function $\mathbf{mu}(\phi, R)$ clearly computes the set $\{S \cup (R \setminus L_\phi) : S \in \mu^{L_\phi}(\phi, R)\}$, an easy induction on the structure of the sentence ϕ yields the following.

Corollary 3.1 *Algorithm 1 correctly computes $Ask(\phi, KB)$, and Algorithm 2 correctly computes $Tell(\phi, KB)$.*

Let us now turn to the computational complexity of the two algorithms. As a notational convenience, for a sentence $\phi \in \Phi$, we denote by $|\phi|$ its length counted in occurrences of symbols from $T \cup \{\neg, \wedge, >\}$. For a set A, we denote by $|A|$ the number of elements in it. Note that for any $\phi \in \Phi$, we have $|L_\phi| \leq |\phi|$.

Lemma 3.4 *Let ϕ be a fixed sentence from Φ. Then the running time of function* $\mathbf{mu}(\phi, R)$ *is* $\mathcal{O}(|R|)$.

Proof. Since $|\texttt{closest}| = 2^{|L_\phi|} \leq 2^{|\phi|}$, the total number of iterations in the loops in **mu** is bounded by a constant. Also, for each $U \in \texttt{closest}$, we have $|U| \leq |L_\phi| \leq |\phi|$. Therefore the computation of $\mathbf{eval}(\phi, U)$ within **mu** does not depend on $|R|$ at all, i.e. it can be carried out in constant time.

In the nested **for each**-loop we test whether $V <_{R \cap L_\phi} U$. Clearly, this can be done in time proportional to $|R|$. Likewise, the step in the last loop takes time proportional to $|R|$. **QED**

Lemma 3.5 *Let ϕ be a fixed sentence from Φ. Then the running time of function* $\mathbf{eval}(\phi, R)$ *is* $\mathcal{O}(|R|)$.

Proof. We proceed by an induction on the structure of ϕ.

Base. If ϕ is $\perp$ the claim is obvious. If ϕ is of the form t, we have to test whether $t \in R$. This takes time proportional to $|R|$.

Inductive step. If ϕ is of the form $\neg\psi$ or $\psi \wedge \chi$, then the claim is obvious. Suppose therefore that ϕ is of the form $\psi > \chi$.

First we compute $\mathbf{mu}(\psi, R)$. Since $|\psi| < |\phi|$, Lemma 3.4 tells us that this computation requires time proportional to $|R|$. Next, the number of elements in $\mathbf{mu}(\psi, R)$ is less than

$2^{|L_\psi|} \leq 2^{|\phi|}$ and therefore bounded by a constant, say c. We thus have to compute **eval** no more than c times. The size of each S in $\mathbf{mu}(\psi, R)$ is at most $|R| + |\psi| < |R| + |\phi|$. Since $|\chi| < |\phi|$, the inductive hypothesis yields the fact that the computation of $\mathbf{eval}(\chi, S)$ can be carried out in time $\mathcal{O}(|R| + |\phi|) = \mathcal{O}(|R|)$.**QED**

By the size of a knowledgebase we mean the sum of the sizes of the worlds in it. We now have the following straightforward consequence of the two previous lemmas.

Theorem 3.3 *Algorithms 1 and 2 both run in time linear in the size of KB.*

4 Expression Complexity

In this section we shall study the complexity of computing $Ask(\phi, KB)$ and $Tell(\phi, KB)$, as a function of $|\phi|$, the length of ϕ.

It is clear that Algorithm 2 takes time that is exponential in $|\phi|$. The following theorem says that the bound cannot be improved. We note that Fagin *et al.* (1986) mention a similar result.

Theorem 4.1 *There is a knowledgebase KB and a family* $\{\phi_n\}_{n\in\omega}$ *of formulas, such that the size of* $Tell(\phi_n, KB)$ *is exponential in the length of* ϕ_n.

Proof. Let $KB = \{\emptyset\}$, and let ϕ_n be the formula $(t_1 \vee u_1) \wedge \ldots \wedge (t_{n+1} \vee u_{n+1})$. Then $\mu(\phi_n, \emptyset) = \{\{v_1, \ldots, v_{n+1}\} : v_i \in \{t_i, u_i\}, i \in \{1, \ldots, n+1\}\}$.**QED**

On the other hand, if ϕ is a propositional formula in disjunctive normal form, then the size of $Tell(\phi, KB)$ is linear in the length of ϕ. We also note that for updates of the form ϕ_n above, the *model trees* of (Fernández and Minker, 1991) might provide a compact representation of the resulting KB. This question would however require further investigations.

In order to characterize the expression complexity of Ask, we shall study the problem of deciding whether $Ask(\phi, KB) = yes$, for a fixed KB. We provide a constructive upper bound, that actually also computes the value of $Ask(\phi, KB)$. We also provide a matching lower bound for the decision problem.

Formally, by the *expression complexity* of *Ask wrt KB* we mean the complexity of the set

$$Query_{KB} = \{\phi \in \Phi : Ask(\phi, KB) = yes\}.$$

We shall first interest ourselves in the *satisfiability* problem, that is, the problem of determining if $[\![\phi]\!] \neq \emptyset$, for a given ϕ, in which case we say that ϕ is satisfiable. Formally, we are interested in the set

$$Sat_\Phi = \{\phi \in \Phi : [\![\phi]\!] \neq \emptyset\}.$$

Our first result concerns decidability.

Lemma 4.1 Sat_Φ *is effectively decidable.*

Proof. We shall show that $\phi \in Sat_\Phi$, if and only if $[\![\phi]\!] \cap \{S : S \subseteq L_\phi\} \neq \emptyset$. The latter problem is decidable as a consequence of Corollary 3.1.

The if direction of the above claim is obvious. For the only if direction, let $R \in [\![\phi]\!]$. By Theorem 3.1 (i), we have $(R \cap L_\phi) \in [\![\phi]\!]$, and obviously $(R \cap L_\phi) \in \{S : S \subseteq L_\phi\}$.**QED**

The next problem to tackle is the computational complexity of Sat_Φ. For this, consider the following variant of the function **eval**(ϕ, R):

```
function eval(φ, R) : boolean;
begin
    case the form of φ of :
        ⊥     : eval := false;
        t     : eval := t ∈ R;
        ¬ψ    : eval := not( eval(ψ, R));
        ψ ∧ χ : eval := and({ eval(ψ, R), eval(χ, R)});
        ψ > χ : begin
                    temp := true;
                    for each U ⊆ L_ψ do
                        if eval(ψ, U) then
                        begin
                            close := true;
                            for each V ⊆ L_ψ do
                                if and( {eval(ψ, V), V <_{R∩L_ψ} U}) then
                                  close := false;
                            if close then
                                temp := and( {temp, eval(χ, U ∪ (R \ L_ψ) )});
                        end;
                    eval := temp;
                end;
    end case;
end;
```

This version of **eval** is a depth-first search. The depth of the recursion is clearly bounded by $|\phi|$, the length of ϕ. In an implementation, the stack will therefore have at most as many nodes as there are subformulas of ϕ. In each node we need to store the current subformula, and when the subformula is $\psi \wedge \chi$ we need to record whether we are evaluating ψ or χ. In the case where the current subformula is $\psi > \chi$, we need, in addition to the values of variables `temp` and `close`, to record which subset of L_ψ is under consideration. We also need a marker that tells us where in the block for $\psi > \chi$ we are. None of these pieces of information requires more amount of space than a polynomial of the current subformula.

In order to test if ϕ is in Sat_Φ, as in the proof of Lemma 4.1 we cycle through all subsets S of L_ϕ, and for each S we run the modified version **eval**(ϕ, S). Since $|S| \leq |\phi|$ we are thus lead to the following result.

Theorem 4.2 *Sat_Φ is in* PSPACE.

We shall now provide a matching lower bound for Sat_Φ. For this we shall use a reduction from the Quantified Boolean Formula problem (QBF, Garey and Johnson, 1979, see also Hopcroft and Ullman, 1979). Formally, QBF is the set of all expressions of the form

$$Q_n t_n(\ldots(Q_1 t_1(\phi))),$$

where $n \geq 0$, where each $Q_i \in \{\forall, \exists\}$, where each $t_i \in T$, and where ϕ is a propositional formula in Φ, such that $L_\phi = \{t_1, \ldots, t_n\}$.

The value of a QBF $Q_n t_n(\ldots(Q_1 t_1(\phi)))$ is *true* or *false*. The value is defined recursively as follows. A QBF $\forall t_i(Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\psi))))$ is *true* if and only if both QBF's $Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\top_i(\psi))))$, and $Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\bot_i(\psi))))$ are *true*. By $\top_i(\psi)$ we mean the formula ψ with each occurrence of t_i replaced by $\top$. Likewise, $\bot_i(\psi)$ denotes ψ with each occurrence of t_i replaced by $\bot$. Also, the QBF $\exists t_i(Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\psi))))$ is *true* if and only if at least one of the QBF's $Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\top_i(\psi))))$, and $Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\bot_i(\psi))))$ is *true*. In the base case of the recursion we will have a formula, say χ, that contains only Boolean connectives, $\top$ and $\bot$. This QBF is *true* if $[\![\chi]\!] = \mathcal{R}$, and *false* if $[\![\chi]\!] = \emptyset$.

Theorem 4.3 *Sat_Φ is* PSPACE*-hard.*

Proof. Let $Q_n t_n(\ldots(Q_1 t_1(\phi))) \in$ QBF. We shall construct a formula $\rho \in \Phi$, such that $Q_n t_n(\ldots(Q_1 t_1(\phi)))$ is *true* if and only if $\rho \in Sat_\Phi$. We will have $L_\rho = \{t_1, \ldots, t_n, u_1, \ldots, u_n\}$, where

$$\rho = \sigma_n \wedge \rho_n.$$

Here

$$\sigma_n = \bigwedge_{i \in \{1,\ldots,n\}} \neg t_i \wedge \neg u_i,$$

and σ_0 is $\top$. The formula ρ_n as well as each ρ_i is defined recursively as follows:

$$\rho_i = \begin{cases} (t_i \vee u_i) > \rho_{i-1} & \text{if } Q_i = \forall \\ (t_i \vee u_i) \supset \rho_{i-1} & \text{if } Q_i = \exists \end{cases}$$

The formula ρ_0 is simply ϕ. Clearly, the length of ρ is linear in the length of $Q_n t_n(\ldots(Q_1 t_1(\phi)))$.

As an illustration of the transformation, consider the QBF

$$\forall t_3(\exists t_2(\forall t_1((t_3 \wedge t_2) \rightarrow t_1))).$$

We now have

$$\rho = \sigma_3 \wedge ((t_3 \vee u_3) > ((t_2 \vee u_2) \supset ((t_1 \vee u_1) > ((t_3 \wedge t_2) \rightarrow t_1)))).$$

We shall now show that $Q_n t_n(\ldots(Q_1 t_1(\phi)))$ is *true* if and only if ρ is satisfiable. We start by some notation. Let $R \in \mathcal{R}$, and $i \in \{0, \ldots, n\}$. By ϕ_i^R we mean the formula ϕ with each occurrence of t_j, where $i + 1 \leq j \leq n$, replaced by $\top$, if $t_j \in R$, and replaced by $\bot$, otherwise. Note that $\phi_n^R = \phi$, and that ϕ_0^R contains only Boolean connectives, $\top$ and $\bot$.

Now we claim that for all $i \in \{0, \ldots, n\}$, if $R \in [\![\sigma_i]\!]$, then

$$Q_i t_i(\ldots(Q_1 t_1(\phi_i^R))) \text{ is } \textit{true} \text{ iff } R \in [\![\rho_i]\!].$$

We verify the claim by an induction on i.

(base). Clearly $[\![\phi_0^R]\!] = \mathcal{R}$ if and only if $R \in [\![\phi]\!]$. Thus the QBF ϕ_0^R is *true* if and only if $R \in [\![\rho_0]\!]$. The value of R is irrelevant here, since $[\![\sigma_0]\!] = \mathcal{R}$.

(inductive step). First note that if $R \in [\![\sigma_i]\!]$, then $\mu(t_i \vee u_i, R) = \{R \cup \{t_i\}, R \cup \{u_i\}\} \subseteq [\![\sigma_{i-1}]\!]$. Let Q_i be $\forall$ (resp. $\exists$). Then we have

$$R \in [\![\rho_i]\!]$$

iff

$$(R \cup \{t_i\}) \in [\![\rho_{i-1}]\!], \text{ and (resp. or) } (R \cup \{u_i\}) \in [\![\rho_{i-1}]\!]$$

iff (by the inductive hypothesis)

$$Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\phi_{i-1}^{R\cup\{t_i\}}))) \text{ is } \textit{true}, \text{ and (resp. or) } Q_{i-1}t_{i-1}(\ldots(Q_1 t_1(\phi_{i-1}^{R\cup\{u_i\}}))) \text{ is } \textit{true}$$

iff

$$(Q_{i-1}t_{i-1}(\ldots Q_1 t_1(\top_i(\phi_i^R)))) \text{ is } \textit{true}, \text{ and (resp. or) } (Q_{i-1}t_{i-1}(\ldots Q_1 t_1(\bot_i(\phi_i^R)))) \text{ is } \textit{true}$$

iff

$$Q_i t_i(\ldots(Q_1 t_1(\phi_i^R))) \text{ is } \textit{true}.$$

Thus the claim is proved. Suppose then that $\sigma_n \wedge \rho_n$ is satisfiable. As in Lemma 4.1, we can restrict our attention to worlds that are subsets of $\{t_1, \ldots, t_n, u_1, \ldots, u_n\}$. Thus we have that $\sigma_n \wedge \rho_n$ is satisfiable iff $\emptyset \in [\![\rho_n]\!]$ iff $Q_n t_n(\ldots(Q_1 t_1(\phi_n^{\emptyset})))$ is *true* iff $Q_n t_n(\ldots(Q_1 t_1(\phi)))$ is *true*.**QED**

It is clear that the modified version of procedure **eval** also gives an algorithm for computing $Ask(\phi, KB)$ in space polynomial in the length of ϕ. For the decision problem, we conclude the following.

Corollary 4.1 *For all $KB \in \mathcal{KB}$, $Query_{KB}$ is in* PSPACE.

For a lower bound, it is easy to see that, in the notation of the proof of Theorem 4.3, $Ask(\rho_n, \{\emptyset\}) = yes$ if and only if the QBF ϕ is *true*. Thus we have the following result.

Corollary 4.2 *$Query_{\{\emptyset\}}$ is* PSPACE*-complete.*

We note that some further complexity-theoretic results for updates have recently been obtained by Eiter and Gottlob (1992). Instead of data and expression complexity, they study (among other things) the *combined complexity* of unnested subjunctive queries, which in the notation of the present paper would be the complexity of the set

$$\{\langle KB, \phi > \psi\rangle : Ask(\phi > \psi, KB) = yes\},$$

where ϕ and ψ are propositional sentences. Eiter and Gottlob show that this set is Π_2^p-complete. They also show that if the knowledgebase is *represented* by a set of propositional Horn clauses $\mathtt{T}$, and if ϕ and ψ also are propositional Horn clauses, then the problem of verifying whether $Ask(\phi > \psi, [\![\mathtt{T}]\!]) = yes$ is *co-NP* complete (in the size of the input $\langle \mathtt{T}, \phi, \psi\rangle$). If, furthermore, the size of ϕ is bounded by a constant, then this problem can be solved in polynomial time.

5 Query Transformation

In this section we provide a sound axiomatization of query equivalence, and show that the problem of testing whether two queries are equivalent is *PSPACE*-complete.

Formally, we say that ϕ and ψ are *equivalent*, denoted $\phi \equiv \psi$, and defined as

$$\phi \equiv \psi \text{ iff } Ask(\phi, KB) = Ask(\psi, KB),$$

for all $KB \in \mathcal{KB}$.

Let Θ be the smallest subset of Φ that contains all formulas of the following form

All truth-functional axioms,

$\phi \rightarrow (\psi > \phi)$, *where* $L_\phi \cap L_\psi = \emptyset$,

$\phi > \phi$,

$(\phi > \neg\phi) \rightarrow (\psi > \neg\phi)$,

$((\phi > \chi) \wedge (\psi > \chi)) \rightarrow ((\phi \vee \psi) > \chi)$,

$(\phi > \psi) \rightarrow (\phi \rightarrow \psi)$,

$(\phi \wedge \psi) \rightarrow (\phi > \psi)$,

and that is closed under Modus Ponens and the following rule:

For any $n \geq 1$,

$((\chi_1 \wedge \ldots \wedge \chi_n) \rightarrow \psi) \mapsto ((\phi > \chi_1 \wedge \ldots \wedge \phi > \chi_n) \rightarrow (\phi > \psi))$.

We can now show that the axiomatization is *sound*.

Theorem 5.1 *If* $\phi \leftrightarrow \psi \in \Theta$, *then* $\phi \equiv \psi$.

The proof of this theorem is straightforward and thus omitted. We are currently investigating the completeness of the axiomatic system.

As to the computational complexity of equivalence, we are interested in the complexity of the set

$$Equ_\Phi = \{(\phi, \psi) \in \Phi \times \Phi : \phi \equiv \psi\}.$$

The equivalence problem is a variation of the satisfiability problem, since it is easily shown that $(\phi, \psi) \in Equ_\Phi$ if and only if $\neg(\phi \leftrightarrow \psi) \notin Sat_\Phi$. Therefore, we have the following result.

Corollary 5.1 Equ_Φ *is* PSPACE-*complete.*

6 Change Algebras

Our *Tell* function is a *change* operator, and represents a particular solution to the change problem. The change problem appears in many guises, such as data and knowledgebase updates, view updates, belief revisions, truth maintenance, abduction, defeasible reasoning, and so forth. Most generally, change is a form of nonmonotonic reasoning (see e.g. Makinson and Gärdenfors, 1991).

In order to fit our *Tell* operator into a more general pattern, we shall introduce some notation. Let T' be a henceforth fixed finite subset of T, and let Φ' be the set of all propositional formulas ϕ in Φ, such that $L_\phi \subseteq T'$. Furthermore, let $\circ$ be a mapping from $\Phi' \times \Phi'$ to Φ'. Then the tuple

$$\langle \Phi', \circ \rangle$$

is called a (free) *change algebra*, and for ϕ and ψ in Φ', $\phi \circ \psi$ is meant to represent the formula obtained from changing ϕ by ψ.

Properties of change operators that have appeared in the literature are examined in (Katsuno and Mendelzon, 1991). In particular, two classes of change algebras have recently been identified: revision algebras (Alchourrón *et al.*, 1985), and update algebras (Katsuno and Mendelzon, 1991). A revision corresponds to a change in our knowledge about a static world, whereas an update corresponds to a change in the world. The conditions characterizing these algebras will be given below. We shall first need some additional notation.

Let $\{\phi, \psi\} \subseteq \Phi'$, and define $[\![\phi]\!]' = [\![\phi]\!] \cap \{S : S \subseteq T'\}$. Then we say that ϕ is *satisfiable*, if $[\![\phi]\!]' \neq \emptyset$ (as before), that ϕ *implies* ψ, if $[\![\phi]\!]' \subseteq [\![\psi]\!]'$, that ϕ is *equivalent* to ψ, if $[\![\phi]\!]' = [\![\psi]\!]'$ (as before), and that ϕ is *complete*, if $[\![\phi]\!]'$ is a singleton. We now have the following definitions.

A tuple $\langle \Phi', \circ \rangle$ is a *revision algebra* if it is a change algebra and the following conditions hold (Alchourrón *et al.*, 1985, Katsuno and Mendelzon, 1991):

(R1) $\phi \circ \psi$ implies ψ

(R2) If $\phi \wedge \psi$ is satisfiable, then $\phi \circ \psi$ is equivalent to $\phi \wedge \psi$.

(R3) If ψ is satisfiable, then $\phi \circ \psi$ is satisfiable.

(R4) If ϕ is equivalent to ξ and ψ to χ, then $\phi \circ \psi$ is equivalent to $\xi \circ \chi$.

(R5) $(\phi \circ \psi) \wedge \chi$ implies $\phi \circ (\psi \wedge \chi)$.

(R6) If $(\phi \circ \psi) \wedge \chi$ is satisfiable, then $\phi \circ (\psi \wedge \chi)$ implies $(\phi \circ \psi) \wedge \chi$.

For an intuitive account of these conditions or postulates, see (Alchourrón *et al.*, 1985, Katsuno and Mendelzon, 1991).

It is clear that changes in the knowledge about a static world, i.e. revisions, do not capture all change operators. In particular, most database changes are changes in the world. Such changes are called updates. Formally, a tuple $\langle \Phi', \circ \rangle$ is an *update algebra* if it is a change algebra and the following conditions hold (Katsuno and Mendelzon, 1991):

(U1) $\phi \circ \psi$ implies ψ

(U2) If ϕ implies ψ, then then $\phi \circ \psi$ is equivalent to ϕ.

(U3) If both ϕ and ψ are satisfiable, then $\phi \circ \psi$ is satisfiable.

(U4) If ϕ is equivalent to ξ and ψ to χ, then $\phi \circ \psi$ is equivalent to $\xi \circ \chi$.

(U5) $(\phi \circ \psi) \wedge \chi$ implies $\phi \circ (\psi \wedge \chi)$.

(U6) If $\phi \circ \psi$ implies χ, and $\phi \circ \chi$ implies ψ, then $\phi \circ \psi$ is equivalent to $\phi \circ \chi$.

(U7) If ϕ is complete, then $(\phi \circ \psi) \wedge (\phi \circ \chi)$ implies $\phi \circ (\psi \vee \chi)$.

(U8) $(\phi \vee \varphi) \circ \psi$ is equivalent to $(\phi \circ \psi) \vee (\varphi \circ \psi)$.

These conditions are given intuitive motivation in (Katsuno and Mendelzon, 1991), which paper also gives model theoretic characterizations of revision and update operators.

In order to embed our *Tell* operator into a change algebra we shall need some more notation. Let $\mathbf{S} = \{S_1, \ldots, S_n\}$ be a set of subsets of T'. Then $\varphi_{\mathbf{S}}$ denotes the formula $\varphi_{S_1} \vee \ldots \vee \varphi_{S_n}$, where

$$\varphi_{S_i} = (\bigwedge_{t \in S_i} t) \wedge (\bigwedge_{t \in \{T' \setminus S_i\}} \neg t).$$

Also, if $\phi \in \Phi'$, then we define

$$KB_\phi = [\![\phi]\!]'.$$

Now define a change operator $\circ_{Tell}$ as follows:

$$\phi \circ_{Tell} \psi = \varphi_{Tell(\psi, KB_\phi)}.$$

An examination of the space example in Section 2, reveals that $\circ_{Tell}$ does not satisfy condition (R2). We conclude the following.

Theorem 6.1 $\langle \Phi', \circ_{Tell} \rangle$ *is not a revision algebra.*

On the other hand, the next theorem shows that our *Tell* mapping indeed reflects a change in the world, i.e. it is an update.

Theorem 6.2 $\langle \Phi', \circ_{Tell} \rangle$ *is an update algebra.*

This fact is proved using a representation theorem of (Katsuno & Mendelzon, 1991) which states that any update algebra can alternatively be characterized by a partial orders on worlds, of which the partial orders $\leq_R$ of this paper are an instance.

7 Conclusion

We have given a framework for subjunctive queries, defining their semantics from the point of view of belief revision theory. By taking a model checking approach, we obtain queries and updates of polynomial data complexity. From an expression complexity point of view, it turns out that an update can increase the size of the knowledgebase exponentially, and that query evaluation is complete in polynomial space, as is also query equivalence, i.e. the problem of testing whether two queries returns the same answer for all knowledgebases. For query equivalence we have provided a sound axiomatization.

The following direction now suggests itself: The language for Φ can be extended with variables, equality and quantification. The natural question to study is then the expressive power of such extended knowledgebase languages, since it seems that our μ-operator can be made to generalize the fixpoint queries of (Chandra and Harel, 1982). This direction is pursued in (Grahne, Mendelzon and Revesz, 1992).

Acknowledgements

Many thanks to Hirofumi Katsuno and the anonymous referees for their comments.

C. E. Alchourrón, P. Gärdenfors & D. Makinson. (1985), On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic* **50**, 510–530.

F. Bancilhon & N. Spyratos. (1981), Update semantics of relational views. *ACM Transactions on Database Systems* **4**, 557–575.

J. Biskup. (1983), A foundation of Codd's relational maybe-operations. *ACM Transactions on Database Systems* **8**, 608–636.

A. J. Bonner. (1988a), Hypothetical Datalog: complexity and expressibility, *in* "Proceedings of the Second International Conference on Database Theory.", pp. 144–160.

A. J. Bonner. (1988b), A logic for hypothetical reasoning, *in* "Proceedings of the Seventh AAAI National Conference on Artificial Intelligence," pp. 480–484.

A. J. Bonner, L. T. McCarty & K. Vadaparty. (1989), Expressing database queries with intuitionistic logic, *in* "Proceedings of the North American Conference on Logic Programming," pp. 831–850.

M. Cadoli & M. Lenzerini. (1991), The Complexity of Propositional Closed World Reasoning and Circumscription. Technical report, Dipartimento di Informatica e Sistemica, Università di Roma "La Sapienza."

A. K. Chandra & D. Harel. (1982), Structure and complexity of relational queries. *Journal of Computer and System Sciences* **25**, 99–28.

S. S. Cosmadakis & C. H. Papadimitriou. (1984), Updates of relational views. *Journal of the Association for Computing Machinery* **31**, 742–760.

A. Del Val. (1992), Computing Knowledge Base Updates. To appear in "Proceedings of the Third Conference on Principles of Knowledge Representation."

T. Eiter & G. Gottlob. (1992), On the complexity of propositional knowledge base revision, updates, and counterfactuals, *in* "Proceedings of the Eleventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems," pp. 261–273.

R. Fagin, G. M. Kuper, J. D. Ullman & M. Y. Vardi. (1986), Updating logical databases, *in* "Advances in Computing Research — vol. 3: The Theory of Databases," (P. C. Kanellakis & F. P. Preparata, Eds.), pp. 1–18, JAI Press, Greenwich, Ct.

R. Fagin, J. D. Ullman & M. Y. Vardi. (1983), On the semantics of updates in databases, *in* "Proceedings of the Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems," pp. 352–365.

J. A. Fernández & J. Minker. (1991), Bottom-up evaluation of Hierarchical disjunctive databases, *in* "Proceedings of the Seventh International Conference on Logic Programming," pp. 660–675.

D. M. Gabbay. (1985), N-Prolog: an extension of Prolog with hypothetical implications. II. Logical foundations and negation as failure. *Journal of Logic Programming* **2**, 251–283.

D. M. Gabbay & U. Reyle. (1984), N-Prolog: an extension of Prolog with hypothetical implications. I. *Journal of Logic Programming* **1**, 319–355.

P. Gärdenfors. (1988), "Knowledge in Flux: Modeling the Dynamics of Epistemic States," Bradford Books, MIT Press, Cambridge, MA.

M. R. Garey & D. S. Johnson. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco.

M. L. Ginsberg. (1986), Counterfactuals. *Artificial Intelligence* **30**, 35–79.

N. Goodman. (1955), "Fact, Fiction, and Forecast," Harvard University Press, Cambridge.

G. Grahne. (1991) Updates and counterfactuals, *in* "Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning," pp. 269–276.

G. Grahne, A. O. Mendelzon & P. Z. Revesz. (1992) Knowledgebase transformations, *in* "Proceedings of the Eleventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems," pp. 246–260.

J. Halpern & M. Y. Vardi. (1991), Model checking vs. theorem proving: a manifesto, *in* "Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning," pp. 325–334.

J. E. Hopcroft & J. D. Ullman. (1979), "Introduction to Automata Theory, Languages and Computation," Addison-Wesley.

H. Katsuno & A. O. Mendelzon. (1989), A unified view of propositional knowledge base updates, *in* " Proceedings of the 11th International Joint Conference on Artificial Intelligence," pp. 1413–1419.

H. Katsuno & A. O. Mendelzon. (1991), On the difference between updating a knowledge base and revising it, *in* "Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning," pp. 387–394.

D. K. Lewis. (1973), "Counterfactuals," Blackwell, Oxford.

D. Makinson & P. Gärdenfors. (1991), Relations between the logic of theory change and nonmonotonic logic, *in* Lecture Notes in Artificial Intelligence, Vol. 465 : "The Logic of Theory Change," (A. Fuhrmann & M. Morreau, Eds.), pp. 185–205, Springer Verlag, Berlin.

J. Minker. (1982), On indefinite databases and the closed world assumption, *in* " Proceedings of the Sixth International Conference on Automated Deduction," pp. 292–398

F. P. Ramsey. (1950), General propositions and causality, *in* "Foundations of Mathematics and Other Logical Essays," (R. B. Braithwaite, Ed.), pp. 237–257, Routledge and Kegan Paul, New York.

R. Reiter. (1978), On closed world databases, *in* "Logic and Databases," (H. Gallaire & J. Minker, Eds.), pp. 55–76, Plenum Press, New York.

M. Winslett. (1988a), A framework for comparison of update semantics, *in* "Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems," pp. 315–324.

M. Winslett. (1988b), Reasoning about action using a possible models approach, *in* "Proceedings of the Seventh National Conference on Artificial Intelligence," pp. 89–93.