January 1986

Report No. STAN-CS-86-1096

A Model-Theoretic Approach to Updating Logical Databases

by

Marianne Winslett Wilkins

Department of Computer Science

Stanford University
Stanford, CA 94305

A Model-Theoretic Approach to Updating Logical Databases

Marianne Winslett Wilkins*
Stanford University, Computer Science Dept.

Abstract. We show that it is natural to extend the concept of database updates to encompass databases with incomplete information. Our approach embeds the incomplete database and the updates in the language of first-order logic, which we believe has strong advantages over relational tables and traditional data manipulation languages in the incomplete information situation. We present semantics for our update operators, and also provide an efficient algorithm to perform the operations.

1. Introduction

Much attention has been paid to the problem of answering queries in databases containing null values, or attribute values that are known to lie in a certain domain but whose value is currently unknown (see e.g. [Imielinski 84], [Reiter 84]). Progress on this front has encouraged research into the problem of updating such databases; as one group of researchers aptly points out [Abiteboul 85], the problem of query answering presupposes the ability to enter incomplete information into the database, and, with any luck, to remove uncertainties when more information becomes available.

Among recent work, this paper has ties to that of Abiteboul and Grahne [Abiteboul 85], who investigate the problem of updates on several varieties (with varying representational power) of tables, or relations containing null values and auxiliary constraints other than integrity constraints. They propose a definition for simple updates as set operations on the set of possible complete-information databases represented by two tables, and investigate the relationship between table type and ability to represent the result of an update correctly and completely. They do not consider updates with joins or disjunctions in selection clauses, comparisons between attribute values, or selection clauses referencing tuples other than the tuple being updated. Their conclusion was that only the most powerful and complex version of tables was able to fully support their update operators.

The work presented in this paper is also related to that of Fagin et al [Fagin 83, Fagin 84], differing chiefly in the definitions of the meaning of updates and in the inclusion of a constructive algorithm for update computation. We base the semantics of updates on the contents of the models of the theory being updated; Fagin et al lend more importance to the particular formulas currently in the theory, producing a more syntactically oriented approach. The effect of an update in our paradigm is independent of the choice of formulas (other than schema and integrity constraints) used to represent that set of models. Another difference concerns our identification of two levels of formulas in a theory-axioms and non-axioms—and the provision of very different algorithmic manipulations for the two types of formulas during an update.

* AT&T Bell Laboratories Scholar

Reiter [84, 84b] sets forth a logical framework for the null value and disjunctive information problems, where databases are represented as logical theories. (Disjunctive information occurs when one knows that one or more of a set of tuples holds true, without knowing which one.) Within this framework one may easily represent many, though not all, of the pieces of information typically encountered when dealing with missing information. Given a relational database, Reiter constructs a relational ***theory*** whose model corresponds to the world represented by the database. For our purposes here, the advantages of this logic framework are four-fold: it allows a clean formalization of incomplete information; it allows us to define the meanings of query and update operators without recourse to intuition or common knowledge; and it frees us from implicit or explicit consideration of implementation issues, by not forcing incomplete information into a tabular format. By framing the update question in this paradigm, we will also gain insights into the more general problem of updating general logical theories, and lay groundwork for use in applications beyond ordinary databases, such as AI applications using a knowledge base built on top of ground knowledge.

In the remainder of this paper, we will set forth a simple update capability that covers many useful types of updates in what we call extended relational theories. Extended relational theories, presented in Section 2, are an extension to Reiter's theories for disjunctive information in which predicate constants may appear in formulas in the theory for the database and in which formulas other than simple disjunctions may appear, thus allowing a much broader class of models for the theories. In Section 3.1 we set forth a simple data manipulation language, LDML, for extended relational theories, and give model-theoretic definitions of the meaning of LDML updates in Section 3.2. Sections 3.3 and 3.5 present an algorithm, GUA, that implements these semantics. The algorithm is shown to be correct in the sense that the alternative worlds produced by updates under this algorithm are the same as those produced by updating each alternative world individually. The algorithm can be extended to cover the case where null values appear in the theory as Skolem constants, in which case the theory may have an infinite set of models. In Section 3.4 we present necessary and sufficient conditions for two LDML updates to be equivalent when applied to any extended relational theory. Finally, Section 3.6 discusses the computational complexity of GUA.

A preliminary version of this paper appeared elsewhere [Wilkins 86].

2. Extended Relational Theories

We now give a formal presentation of our extension to Reiter's theories, called *extended* relational theories. Unlike most formalizations of incomplete information, our extended relational theories will be sufficiently powerful to represent any set of relational databases all having the same schema and integrity constraints. The language $\mathcal{L}$ for the theories contains the following strings of symbols:

1. An infinite set of variables (to be used in the axioms of the extended relational theory).

2. A set of constants, possibly empty and possibly infinite. These represent the elements in the domains of database attributes.

3. A finite set of predicates of arity 1 or more. These represent the attributes and relations of the database.

4. Punctuation symbols ‘(’, ‘)’, and ‘,’.

5. Logical connectives, quantifiers, truth values, and the equality predicate: A, V, ¬, ↔, V, 3, T, F, and = .

6. An infinite set of 0-ary predicates (predicate constants).

A given theory $\mathcal{T}$ over Cis an extended relational theory if $\mathcal{T}$ has exactly the following wffs:

1. ***Unique*** Name ***Axioms:*** For each pair of constants c_1, c_2 in $\mathcal{L}$, $\mathcal{T}$ contains the unique name axiom $\neg(c_1 = c_2)$.

2. Completion Axioms: To implement a version of the closed-world assumption so that we may prove certain ground atomic formulas to be false, we include axioms stating that the only ground atomic formulas that may be true in a model of $\mathcal{T}$ are those explicitly given somewhere in 7. As our extended relational theories do not include any axioms to generate ground atomic formulas via inference, this means that any ground atomic formula not appearing in 7 should be false in all models of 7. (A different formulation of these axioms can be used to allow for ground atomic formulas generated by inference rules.) More precisely, for each n-ary predicate $\boldsymbol{P}$ of 7 , either 7 contains an axiom of the form $\forall x_1 \ldots \forall x_n \neg P(x_1, \ldots, x_n)$, or else for some nonempty set of constants $c_{11}, c_{12}, \ldots, c_{mn}$, 7 contains exactly one axiom of the form

$$\begin{aligned}
&\forall x_1 \forall x_2 \ldots \forall x_n (P(x_1, x_2, \cdot .., x_n) \rightarrow \\
&\qquad ((x_1 = c_{11} \land x_2 = c_{12} \mathrm{A}.. . \mathrm{A}\, x_n = c_{1n}) \mathrm{V} \\
&\qquad (x_1 = c_{21} \mathrm{A}\, x_2 = c_{22} \land \ldots \mathrm{A}\, x_n = c_{2n}) \mathrm{V} \quad . \\
&\qquad \cdots \quad \mathrm{V} \\
&\qquad (x_1 = c_{m1} \mathrm{A}\, x_2 = c_{m2} \land \ldots \mathrm{A}\, x_n = c_{mn})))
\end{aligned}$$

Further, $(x_1 = c_{i1} \mathrm{A}\, x_2 = c_{i2} \mathrm{A} \ldots \mathrm{A}\, x_n = c_{in})$ is represented in (i.e., is a disjunct of) the axiom iff $P(c_{i1}, c_{i2}, \ldots, c_{in})$ appears elsewhere in 7.

Note that the completion axioms of 7 may be derived mechanically from the rest of 7.

3. Non-Axiomatic ***Section:*** The non-axiomatic formulas of 7 may be any finite set of wffs of $\mathcal{L}$ that do not contain variables or the equality predicate. ◇

A discussion of extended relational theories with type axioms (an encoding of the database schema) and dependency axioms is postponed to Section 3.5, because the complications introduced by those axioms are orthogonal to the other issues in updating extended relational theories.

In an implementation of extended relational theories, we would not actually store any of these axioms. Rather, the axioms formalize our intuitions about the behavior of a query and update processor operating on the non-axiomatic part of the database. For example, PROLOG is a query processor that shares our unique name axioms, but has an entirely different closed-world assumption.

Definition. An alternative ***world*** dof a theory 7 is a set S of truth valuations for all the ground atomic formulas of 7 of arity 1 or more, such that S holds for some model M of 7. If this relationship holds, then we say that M represents A. Intuitively, an alternative world is a snapshot of the tuples of a complete-information relational database. The alternative worlds of an extended relational theory look like a set of ordinary relational databases all having the same schema and axioms.

With the inclusion of predicate constants in $\mathcal{L}$ (as a convenience feature that makes updates easier to perform) we depart from Reiter's paradigm. Because predicate constants are "invisible" in alternative worlds, there may not be a one-to-one correspondence between the models of a relational theory and its alternative worlds, as two models may give the same truth valuation to all ground atomic formulas except some predicate constants, and still represent the same alternative world. Alternative worlds contain just the information that would be of interest to a database user, while models may be cluttered with predicate constants of no external interest.

3. A Logical Data Manipulation Language (LDML) For Simple Updates

We now present a data manipulation language based on first-order logic, called LDML (Logical Data Manipulation Language). In this section we will consider the use of LDML for the simplest types of updates, which *we* call ***ground updates.*** The examples given will all be rather abstract; however, traditional data manipulation languages such as SQL and INGRES may be embedded in LDML.

3.1. LDML Syntax

Let $\mathcal{L}'$ be a language containing all the elements of $\mathcal{L}$ except its predicate constants, variables, and the equality predicate. Let ϕ and w be wffs over $\mathcal{L}'$, and let t be a ground atomic formula over $\mathcal{L}'$. Then LDML ground updates consist of the following four operations:

INSERT ω WHERE ϕ

DELETE t WHERE $\phi \wedge t$

MODIFY t TO BE w WHERE $\phi \wedge t$

ASSERT ϕ

Examples. Suppose the database schema contains two relations, Orders(OrderNo, PartNo, Quan) and InStock(PartNo, Quan). Then the following are ground updates:

MODIFY Orders(700,32,9) TO BE Orders(700,32,1) WHERE TA Orders(700,32,9)

DELETE Orders(700, 32, 9) WHERE TA Orders(700,32,9)

INSERT Orders(800,32,1000) V Orders(800,32,100) WHERE T

INSERT F WHERE $\neg$ InStock(32,1)

INSERT $\neg$InStock(32, 1) WHERE T

3.2. **LDML Semantics For Ground Updates**

We define the semantics of an update operating on an extended relational theory 7 by its desired effect on the models of 7. In particular, the alternative worlds (models minus predicate constants) of the updated relational theory must be the same as those obtained by applying the update separately to each original alternative world. In database terms, this may be rephrased as follows: The database with incomplete information represents a (possibly infinite) set of alternative worlds, or complete-information relational databases, each different and each one possibly the real, unknown world. The correct answers to queries and updates are those obtained by storing a separate database for each alternative world and running query processing in parallel on each separate database, pooling the query results in a final step. A necessary and sufficient guarantee of correctness for any more efficient and practical method of query and update processing is that it produce the same results for queries and updates as the parallel computation method. Equivalently, we require that the diagram below be commutative: both paths from upper-left-hand corner to lower-right-hand corner must produce the same result.

$$\begin{tikzcd}
7 \arrow[r, "\text{has alternative world}"] \arrow[d, "\text{update}"] & \mathcal{A} \arrow[d, "\text{update}"] \\
7' \arrow[r, "\text{has alternative world}"] & \mathcal{A}'
\end{tikzcd}$$

The general criteria guiding our choice of semantics are, first, that the semantics agree with traditional semantics in the case where the update request is to insert or delete a single ground atomic formula, or to modify one ground atomic formula to be another. Second, an update is to represent the most exact and most recent state of knowledge obtainable about the ground atomic formulas that the update modifies, inserts, or deletes; and the update is to override all previous information about these ground atomic formulas. These criteria have a syntactic component: one should not necessarily expect two updates with logically equivalent wffs to produce the same results. For example, the result of inserting the truth value T should be different from inserting $g \vee \neg g$; one update reports no change in the information available about g, and the other update reports that the truth valuation of g is now unknown.

We now present formal definitions of the semantics of ground updates. Let $\mathcal{B}$ be a ground update, and let $\mathcal{M}$ be a model of the extended relational theory 7. Define S to be the set of models produced by applying $\mathcal{B}$ to M as follows:

ASSERT ϕ: If ϕ is false in M, then S is the empty set; otherwise, S contains exactly M.

INSERT ω WHERE ϕ: If ϕ is false in M, then S contains one model, M. Otherwise, S contains exactly every model $\mathcal{M}^*$ such that

(1) $\mathcal{M}^*$ agrees with $\mathcal{M}$ on the truth values of all ground atomic formulas except possibly those in w; and

(2) w is true in $\boldsymbol{M}$*.

DELETE $\boldsymbol{t}$ WHERE ϕ A $\boldsymbol{t}$: If ϕ A $\boldsymbol{t}$ is false in $\boldsymbol{M}$, then S contains exactly $\boldsymbol{M}$. Otherwise, let $\boldsymbol{M}$* be the model that agrees with $\boldsymbol{M}$ on all ground atomic formulas except $\boldsymbol{t}$, which is false in $\mathcal{M}$* ; $\boldsymbol{S}$ contains exactly M* .

MODIFY $\boldsymbol{t}$ TO BE w WHERE ϕ A $\boldsymbol{t}$: If ϕ A $\boldsymbol{t}$ is false in $\boldsymbol{M}$, then S contains one model, $\boldsymbol{M}$. Otherwise, let $\mathcal{N}$ be the model created from $\boldsymbol{M}$ by assigning the truth value F to $\boldsymbol{t}$. Then S contains every model $\boldsymbol{M}$* such that

(1) $\boldsymbol{M}$* has the same truth valuations for all ground atomic formulas as $\mathcal{N}$ does, except possibly those in w ; and

(2) w is true in $\boldsymbol{M}$*.

Example. If we insert a V $\boldsymbol{b}$ into $\boldsymbol{M}$, where $\boldsymbol{a}$ and bare ground atomic formulas, then three models are created: one where a A bis true, one where a A $\neg$ ***bis*** true, and one where $\neg a \wedge b$is true-regardless of whether $\boldsymbol{a}$ or bwere true or false in $\boldsymbol{M}$ originally.

For simplicity we have defined the semantics of $\boldsymbol{B}$ in terms of its effect on the model $\boldsymbol{M}$ rather than in terms of its effect on the alternative world of $\boldsymbol{M}$. However, because the semantics are independent of the truth valuations of predicate constants in $\boldsymbol{M}$, $\boldsymbol{B}$ will have the same effect (i.e., produce the same alternative worlds) on every model representing the same alternative world as $\boldsymbol{M}$.

Note that DELETE is a special case of MODIFY and INSERT: DELETE $\boldsymbol{t}$ WHERE ϕ At is equivalent to MODIFY $\boldsymbol{t}$ TO BE $\neg\, \boldsymbol{t}$ WHERE ϕ At , and also equivalent to INSERT $\neg\, \boldsymbol{t}$ WHERE ϕ A $\boldsymbol{t}$. Similarly, ASSERT is a special case of INSERT: ASSERT $\phi$is equivalent to INSERT F WHERE $\neg\,\phi$. A little less obvious, but still immediate from the definitions, is that MODIFY is a special case of INSERT: MODIFY $\boldsymbol{t}$ TO BE w WHERE ϕ A $\boldsymbol{t}$ is equivalent to INSERT w WHERE ϕ At , if $\boldsymbol{t}$ appears in w ; and equivalent to INSERT w A $\neg\, \boldsymbol{t}$ WHERE ϕ A $\boldsymbol{t}$, otherwise.

The remarks at the beginning of this section on correctness of update algorithms may be summed up in the following definition:

Definition. The execution of a ground update $\boldsymbol{B}$ against an extended relational theory $\mathcal{T}$ to produce a new theory $\mathcal{T}'$ is correct and complete iff 7'is an extended relational theory and the alternative worlds of 7'are exactly those alternative worlds represented by the union of the models in the S sets.

Definition. A branching update occurs when some $\boldsymbol{S}$ contains more than one model. In such a case the models of 7 are said to branch, in that a model $\mathcal{M}$ before the update may map into more than one model and alternative world after the update. Intuitively, an update may cause branching when w contains the logical operation 'V', as with the ground update INSERT Orders(100, 32, 1) V Orders(100, 32, 7) WHERE T.

Branching updates are used to introduce incomplete information into the extended relational theory. ASSERT is the usual method for removing incomplete information when more exact knowledge is obtained.

3.3. An Algorithm for LDML Ground Updates: GUA

Recall that DELETE, MODIFY, and ASSERT are special cases of INSERT; it suffices to give an algorithm for performing updates of the form INSERT w WHERE ϕ.

We have semantics that describe the effect of an update on the models of a theory; the semantics gives no hints whatsoever on how to translate that effect into changes in the extended relational theory. For INSERT w WHERE ϕ, we cannot do anything so simple as to add $\phi \rightarrow$ w to 7, because wprobably contradicts the rest of 7. For example, if 7 contains $\neg$ ***a*** A $\neg$ *b*, then INSERT ***a*** V *b* WHERE T should not be interpreted as a request to add T$\rightarrow$ (a V ***b)*** to 7 ! Similarly, for MODIFY, we cannot bodily replace occurrences oft with something like $(\phi \rightarrow$ w $)$A$(\neg \phi \rightarrow t)$; consider the effect of INSERT $\neg$ ***t*** WHERE T$\wedge$ ***ton*** the non-axiomatic section *t* V $\neg$ *t* . Any update algorithm must preserve much of the structure of the old theory while changing only selected items.

Our ground update algorithm GUA may be summarized as follows: For each ground atomic formula ***f*** that appears ***in*** w , replace ***all*** occurrences off ***in the*** extended relational theory 7 by a new predicate constant p_f ***not previously*** appearing in 7. (These predicate constants are not visible externally, i.e., they may not appear in any query posed to the database.) Then add a new formula ***to 7*** that ***defines the*** correct valuation off when ϕ ***is false, and*** another formula to give ***the*** correct valuation off when ϕ ***is*** true.

Before a more formal presentation of the algorithm, let us examine its workings in a simple abstract example of a non-branching update. This example contains all the essential elements of the algorithm, and illustrates the principles underlying the algorithm. A similar example for branching updates is given after the presentation of GUA.

Suppose the database schema contains a single relation with at most tuples a and *b* (such as `Orders(700,34,10)` and `Orders(701,35,10)`), and that we have the following two models and alternative worlds:

Model 1: a, *b*

Mode12: a

Ignoring the axioms for this database, one non-axiomatic section of the extended relational theory for this database is the two wffs ***a*** and a $\vee$ ***b.*** Suppose a user presents the update INSERT a'A $\neg a$ WHERE $b \wedge$ CL, which is equivalent to the more familiar MODIFY a TO BE a' WHERE *b* A a. To perform this update, we replace a and ***a'***in 7 by new predicate constants p_a and $p_{a'}$, and then add formulas to give the new correct valuations for a and a'. In other words, the new models should be:

Model **1:** p_a , *b* , ***a'***

Model **2:** p_a , a .

How do we define the new formulas for a and a'? A template for the formulas will be presented later in this section; for now, looking at the MODIFY form of the update, we can intuitively say that if the selection clause CL A bis true in an alternative world, then $\neg$ ***a*** A a'should be true in that world after the update: ***(b*** A $p_a) \rightarrow (\neg a$ A a'). Similarly, if the selection clause ***a*** A *b* is false in an alternative world , then the valuations for ***a*** and ***a'***should be the same as they originally were: $\neg(b \mathrm{A} p_a) \rightarrow (a \mathrm{A} \neg a')$. We add these formulas to the non-axiomatic section of the database, resulting in the new theory

$p_a, p_a \vee b, \neg(b \wedge \mathrm{Pa}) \rightarrow (a \wedge \neg a'), (b \wedge \mathrm{Pa}) \rightarrow (\neg a \wedge a'),$

which (if we juggle the axioms appropriately) has the desired models:

Model 1: p_a , *b, a'*

Model 2: p_a, *a.*

We now present the ground update algorithm for INSERT in full detail. This first version of the algorithm shows how to perform updates on an extended relational theory without type and dependency axioms; the procedures for use with extended relational theories having those axioms will be given later.

Ground Update Algorithm (GUA)

Input. A ground INSERT update Bin LDML and an extended relational theory 7. (Express DELETE, MODIFY, and ASSERT updates as insertions.)

Output. $\mathcal{T}'$, an updated version of $\mathcal{T}$.

Procedure. A sequence of four steps:

Step 1. Add to completion axioms. For any ground atomic formula f appearing in w or ϕ but not in $\mathcal{T}$, add f to the completion axiom for its database predicate, and add $\neg f$ to the non-axiomatic section of $\mathcal{T}$, creating *7'.*

For example, if the update is INSERT Orders(700, 32, 9) V Orders(700, 32, 8) WHERE T, and neither ground atomic formula previously appeared in 7, then both must be added to the completion axiom for Orders.

Step 2. Rename. For each distinct ground atomic formula f of w , select one new predicate constant not previously appearing in $\mathcal{T}'$, which we will call p_f . For each ground atomic formula f of w , replace all occurrences of f by p_f in the non-axiomatic section of *7'.*

A convenient shorthand for this syntactic replacement is the usual substitution notation, with the semantic difference that one ground atomic formula is to be substituted for another. For example, if σ is the substitution $^{R(c)}_{p_{R(c)}}$, then the notation $(\alpha)_\sigma$ or $(\alpha)^{R(c)}_{p_{R(c)}}$ calls for the replacement of all occurrences of the ground atomic formula $R(c)$ in the wff α by the predicate constant $p_{R(c)}$. If σ_p is the substitution that replaces each ground atomic formula fin w by its predicate constant p_f , then the effect of the current step of GUA is to replace the non-axiomatic section $\mathcal{N}$ of 7'by $(\mathcal{N})_{\sigma_p}$.

Step 3. Define the update. Add the wff $(\phi)_{\sigma_p} \rightarrow$ w to 7'.

Step 4. Restrict the update. For each ground atomic formula fin w , add the wff

$$1\ (\phi)_{\sigma_p} \rightarrow (f \leftrightarrow Pf) \tag{1}$$

to 7'. Then the models of 7'represent exactly the alternative worlds that B is defined to produce from 7. ◇

Example. We present an abstract example of a branching update, again concentrating on the non-axiomatic section of an extended relational theory 7. Suppose the database schema for 7 contains a single relation with at most tuples a and ***b*,** and that we have the following two alternative worlds:

Model 1: a, b

Model 2: a

Ignoring the axioms for this database, one non-axiomatic section for the logic theory of the database is the two wffs a and $\boldsymbol{a} \vee b$.

Suppose a user presents the update INSERT $c \vee a$ WHERE $b \wedge a$ or, in its more familiar form, MODIFY $\boldsymbol{a}$ TO BE c V a WHERE b A $\boldsymbol{a}$. In Step 1, if a, b, or c does not appear in the completion axioms of 7', we add it there now. By the definition of an extended relational theory, $\boldsymbol{a}$ and bmust already appear in those axioms; we simply add c to its completion axiom and add $\neg$ c to the non-axiomatic section of 7'.

In Step 2, we replace all occurences of a and c by p_a and p_c, respectively. The non-axiomatic section of 7' now contains the three wffs p_a , $p_a \vee b$, and $\neg p_c$; the models of $\mathcal{T}'$ are:

Model 1: p_a

Model 2: p_a, $\boldsymbol{b}$.

In Step 3, we add $(\phi)_{\sigma_p} \rightarrow \omega$ (i.e., $(b \wedge p_a) \rightarrow (c \vee a)$) to 7'; and Step 4 supplies the two formulas $\neg$ (**b** A p_a)$\rightarrow$($p_a \leftrightarrow$ ***a)*** and $\neg$ (**b** A p_a)$\rightarrow$($p_c \leftrightarrow c$). The final theory is

p_a, Pa $\vee$ b, $\neg p_c$,
($b \wedge$ Pa) $\rightarrow$ ($c \vee a$),
$\neg$($b \wedge$ Pa) $\rightarrow$ (Pa $\leftrightarrow a$),
$\neg$($b \wedge$ Pa) $\rightarrow$ (PC $\leftrightarrow c$),

which has the desired models, representing four alternative worlds:

Model 1: p_a, a

Model 2: p_a, $\boldsymbol{b}$, c

Model 3: p_a, $\boldsymbol{b}$, a

Model 4: p_a, $\boldsymbol{b}$, c, $\boldsymbol{a}$.

The non-axiomatic section of 7'can be simplified to the two wffs $a \vee$ band $b \rightarrow (c \vee a)$.

Theorem 1. Given an extended relational theory 7 and a ground update B, algorithm GUA correctly and completely performs B . In particular,

(1) GUA produces a legal extended relational theory 7';

(2) The alternative worlds of 7'are the same as the alternative worlds produced by directly updating the models of 7. ◇

Readers not interested in a formal proof of correctness for Algorithm GUA should skip to section 3.4.

In the proof of Theorem 1, we will use one lemma showing that Step 1 of GUA does not change the models of 7.

Lemma 1. Let 7 be a theory containing a completion axiom α for an n-ary predicate P, such that α does not contain a disjunct of the form $(x_1 = c_1$ A $x_2 = c_2 \wedge \cdots \wedge x_n$ = c,). Then adding the new disjunct (xi = c_1 A $x_2 = c_2$ A .·· A $x_n = c_n)$ to α and adding $\neg P(c_1, \ldots, c_n)$ to the non-axiomatic section of 7 produces a new theory 7' with the same models as 7.

Proof of Lemma 1. Let f be the ground atomic formula $P(c_1, \ldots, c_n)$ and let α' be α with the disjunct added for f . Since $\alpha \rightarrow \alpha'$ and $\alpha \rightarrow \neg f$, it follows that any model of 7 is also a model of 7'. Conversely, every model of 7'satisfies all of the formulas of $\mathcal{T} - \alpha$. Every model of 7'must also satisfy α , since the only way α could be violated by 7'is if f were true in a model of $\mathcal{T}'$, which is not the case. ◇

Proof of Theorem 1. For simplicity of reference in the proof below, let 7be the original extended relational theory, $\mathcal{T}_1$ be the theory produced by Step 1 of GUA, $\mathcal{T}_2$ be the theory produced by Step 2, and so on. M will always refer to a model of the original theory, $\mathcal{M}_1$ to a model of $\mathcal{T}_1$, and so on. We first show that GUA produces a subset of the correct set of alternative worlds.

Suppose that $\mathcal{M}_4$ is a model of $\mathcal{T}_4$. Our goal is to show that B produces $\mathcal{M}_4$ from some model M of 7. It suffices to show that $\mathcal{T}_1$ has such a model M , because by Lemma 1, the models of 7 and $\mathcal{T}_1$ are the same.

First consider the case where $(\phi)_{\sigma_p}$ is true in $\mathcal{M}_4$. (Recall that σ_p , defined in Step 2, is the substitution of a set of new predicate constants (i.e., predicate constants not appearing in 7) p_{f_1} through p_{f_n} for the ground atomic formulas f_1 through f_n of w , respectively.) Let $\mathcal{M}$ be a model that agrees with $\mathcal{M}_4$ on the truth valuations of all ground atomic formulas except possibly those appearing in w . For ground atomic formulas fin w , set the truth valuation of fin $\mathcal{M}$ to be the same as that of p_f in $\mathcal{M}_4$. We will show that M is a model of $\mathcal{T}_1$.

The wffs of $\mathcal{T}_4$ differ from those of $\mathcal{T}_1$ only in the addition of a number of wffs in $\mathcal{T}_4$ and in the replacement of the ground atomic formulas of w by new predicate constants in 74. Since the axioms of $\mathcal{T}_1$ are the same as those of $\mathcal{T}_4$, and do not contain ground atomic formulas of ω , $\mathcal{M}$ satisfies those axioms. M also satisfies all other wffs of $\mathcal{T}_1$ that do not contain ground atomic formulas of w , as such wffs also appear in $\mathcal{T}_4$.Without loss of generality we may assume that there is one remaining wff, α of $\mathcal{T}_1$, that may possibly not be satisfied by M . The descendant of α in $\mathcal{T}_4$ is $(\alpha)_{\sigma_p}$. Since $\mathcal{M}$ and $\mathcal{M}_4$ agree on the truth assignments to all ground atomic formulas of $(\alpha)_{\sigma_p}$, which contains no elements of ω , $(\alpha)_{\sigma_p}$ must be true in ***M*** . This implies that α will be true in ***M*** if p_{f_1} through p_{f_n} have the same set of truth assignments in ***M*** and $\mathcal{M}_4$ as do f_1 through f_n , respectively, in $\mathcal{M}$. But by the definition of ***M*** , this condition is true. We conclude that ***M*** is a model of $\mathcal{T}_1$.

It remains to show that B applied to ***M*** produces the alternative world of $\mathcal{M}_4$. Since ϕ is satisfied by ***M*** , it follows that $(\phi)_{\sigma_p}$ must be satisfied by $\mathcal{M}_4$, since $\mathcal{M}_4$ agrees with ***M*** on the truth valuations for all ground atomic formulas of $(\phi)_{\sigma_p}$ except its new predicate constants, and any new predicate constant p_f has the same valuation in $\mathcal{M}_4$ as does f , which appears in place of p_f in ϕ . By the formula $(\phi)_{\sigma_p} \rightarrow$ w of Step 3, it follows that w is true in $\mathcal{M}_4$, so rule ***2*** of the definition of INSERT is satisfied by $\mathcal{M}_4$. For rule 1, if his a ground atomic formula of ***M*** that is not in w , then by definition of $\mathcal{M}$, h has the same truth valuation in ***M*** and $\mathcal{M}_4$. Therefore B produces the alternative world of $\mathcal{M}_4$ from ***M*** .

Now consider the case where ($\phi)_{\sigma_p}$ is false in $\mathcal{M}_4$. Let $\mathcal{M}$ be the same model as $\mathcal{M}_4$. Then ***M*** satisfies all wffs of $\mathcal{T}_1$ that do not contain ground atomic formulas in ω , including the axioms of $\mathcal{T}_1$. Without loss of generality we may assume that there is one remaining wff, α of $\mathcal{T}_1$, that contains ground atomic formulas of ω and may possibly not be satisfied by $\mathcal{M}$. The descendant of α in $\mathcal{T}_4$ is (a),, . By formula (I), if f is in ω then f has the same truth value in $\mathcal{M}_4$ as p_f does; we can replace all occurrences of p_f in $(\alpha)_{\sigma_p}$ by f and create a new wff that is also true in $\mathcal{M}_4$. But this new wff is identical to α . We conclude that ***M*** is a model of $\mathcal{T}_1$ and of 7. By the same replacement technique, it follows that ϕ is false in ***M*** , and that ***B*** produces the alternative world of $\mathcal{M}_4$ from M .

We have shown that GUA is correct; we now turn to the question of completeness: does GUA produce every alternative world that should be derived under ***B*** ?

Let ***M*** be a model of 7and therefore also of $\mathcal{T}_1$. Assume first that ϕ is true in M . Select one particular set vof truth valuations for the ground atomic formulas of w such that w is true under v . If no such v exists, then ***B*** produces no alternative worlds from M , and the theorem follows.

Otherwise, let $\mathcal{M}_4$ be the model that agrees with v on all ground atomic formula valuations of v ; where p_f is assigned the same valuation as f had in ***M*** , for all new predicate constants p_f ; and that agrees with ***M*** on all other valuations. Then $\mathcal{M}_4$ is a model of an arbitrary alternative world that should be produced by ***B*** from ***M*** , and we claim that $\mathcal{M}_4$ is a model of $\mathcal{T}_4$.

First, $\mathcal{M}_4$ satisfies all wffs of $\mathcal{T}_4$ that also appear in $\mathcal{T}_1$. This includes the completion and unique name axioms of $\mathcal{T}_4$. Since p_f has the same truth valuation in $\mathcal{M}_4$ ***as*** does fin $\mathcal{M}$, it follows that $\mathcal{M}_4$ satisfies $(\mathcal{N})_{\sigma_p}$, that is, all the formulas of the non-axiomatic section of $\mathcal{T}_1$, to which σ_p was applied in Step ***2.*** Since w is true in $\mathcal{M}_4$, the wff $(\phi)_{\sigma_p} \rightarrow$ w added to $\mathcal{T}_4$ in Step 3 is satisfied in $\mathcal{T}_4$. Since ϕ is true in ***M*** , $(\phi)_{\sigma_p}$ is true in $\mathcal{M}_4$, and formula (1) from Step ***4*** is also satisfied. Therefore $\mathcal{M}_4$ is a model of $\mathcal{T}_4$, and the alternative world of $\mathcal{M}_4$ is produced by GUA.

Suppose now that ϕ is false in ***M*** . Let $\mathcal{M}_4$ be a model differing from ***M*** only in assignments to new predicate constants: p_f is assigned the same truth valuation in $\mathcal{M}_4$ as

f has in M. Then $\mathcal{M}_4$ represents the only alternative world that B will produce from M. As before, $\mathcal{M}_4$ satisfies $(\mathcal{N})_{\sigma_p}$. Since ϕ is false in M, $(\phi)_{\sigma_p}$ must be false in $\mathcal{M}_4$, since the pattern of truth valuations is identical, and therefore the wff $(\phi)_{\sigma_p} \rightarrow$ w added to $\mathcal{T}_4$ in Step 3 is satisfied by $\mathcal{M}_4$. Formula (1) is also satisfied by $\mathcal{M}_4$, so $\mathcal{M}_4$ must be produced by GUA. We conclude that GUA produces the correct set of alternative worlds from M.

It remains to verify that $\mathcal{T}_4$ meets the criteria in the definition of an extended relational theory. $\mathcal{T}_4$ still contains the set of unique name axioms for $\mathcal{L}$, and its completion axioms still contain disjuncts for exactly those ground atomic formulas appearing in the non-axiomatic section. As the non-axiomatic section still only contains wffs without variables, this concludes the proof of correctness for algorithm GUA. ◊

3.4. Equivalence of Updates

In a future publication, we will examine other possible choices for update semantics, and present in more detail the reasons why we find the semantics presented in this paper to be the best choice. (Interestingly, algorithm GUA is sufficiently general to serve under other choices of semantics simply by altering formula (1) of Step 4.) Though a qualitative discussion of the merits of different choices for semantics is indispensable, we have found that theorems on equivalence of updates go even farther toward exposing the peculiarities of a particular choice of semantics. Such theorems tell us exactly when two updates look similar but really aren't, and when two different-looking updates really are the same; they provide an impassionate demonstration of the properties of different semantics. We can use these theorems to evaluate how well a given semantics meets our intuitions: if a pair of updates should be the same according to our intuition, but an equivalence theorem tells us that they are different (or vice versa), then we can register the discrepancy as a mark against that semantics. As we analyze update equivalence under the current semantics, in the process we will exactly measure the role that syntax plays in these semantics.

Definition. If B_1 and B_2 are two updates over a language $\mathcal{L}$, then B_1 and B_2 are ***equivalent*** if for every extended relational theory 7 over $\mathcal{L}$ or any extension of $\mathcal{L}$, B_1 applied to 7 produces the same set of alternative worlds as B_2 applied to 7. B_1 and B_2 are equivalent when applied to a particular model M of 7if B_1 produces the same set of alternative worlds from M as does B_2.

The reasons for requiring equivalence over all extensions of the language $\mathcal{L}$ will become evident when we consider extended relational theories with type and dependency axioms.

As we pointed out in section 3.2, ASSERT, MODIFY, and DELETE are special cases of INSERT under this semantics; therefore it suffices to prove conditions on equivalence for INSERT updates, rather than considering each type of operator separately. We begin with simple sufficient criteria for equivalence:

Theorem 2. Let B_1 and B_2 be two INSERT ground updates over a language $\mathcal{L}$:

B_1: INSERT ω_1 WHERE ϕ,

B_2: INSERT ω_2 WHERE ϕ.

If ω_1 and ω_2 are logically equivalent and the same ground atomic formulas appear in ω_1 and ω_2, then B_1 is equivalent to B_2. ◊

Proof of Theorem 2. Assume that ω_1 , and therefore ω_2, is satisfiable, as otherwise the theorem follows immediately. For any extended relational theory $\mathcal{T}$ over Cor an extension of $\mathcal{L}$, consider the effects of B_1 and B_2 on a model M of 7. B_1 must produce a model M^* from M , since ω_1 is satisfiable. We wish to show that M^* is also a model of an alternative world produced by B_2 acting on M . If ϕ is false in M , this follows immediately. Otherwise ω_2 must be true in M^*, because ω_1 and ω_2 are logically equivalent; and therefore rule 2 in the definition of INSERT is satisfied for B_2 by M^* . Rule 1 in the definition of INSERT is satisfied for B_2 by M^* since B_1 and B_2 contain the same ground atomic formulas. $\Diamond$

To see that the criteria of Theorem 3 are sufficient but not necessary, consider the two equivalent updates INSERT q WHERE p A q and INSERT p WHERE p A q . For necessary and sufficient criteria, we have Theorem 3, which can be summed up intuitively as follows: B_1 and B_2 are equivalent iff ω_1 and ω_2 are satisfied by the same sets of truth valuations, except that a ground atomic formula g may appear in ω_1 and not in ω_2 (or vice versa) as long as B_1 (resp. B_2) does not change the truth valuation of g .

Theorem 3. Let B_1 and B_2 be two INSERT ground updates over a language $\mathcal{L}$:

B_1: INSERT ω_1 WHERE ϕ,
B_2: INSERT ω_2 WHERE ϕ.

If ϕ is not satisfiable, then B_1 and B_2 are equivalent. Otherwise, let I be the set of ground atomic formulas appearing in both ω_1 and ω_2 . Let v_1' be a truth valuation for the ground atomic formulas of ω_1 that satisfies ω_1 . Define v_1 as the subset of v_1' containing the valuations for all the ground atomic formulas of I. Let V_1 be the set of all such valuations v_1 over all valuations v_1' , and define V_2 analogously for ω_2 . Then B_1 and B_2 are equivalent iff

(1) $V_1 = V_2$; and

(2) if the ground atomic formula g appears in ω_1 but not in ω_2 then $(\omega_1 \rightarrow$ g) A $(\phi \rightarrow$ g) or $(\omega_1 \rightarrow \neg g) \wedge (\phi \rightarrow \neg g)$ is valid; and

(3) if g appears in ω_2 but not in ω_1 then $(\omega_2 \rightarrow g) \wedge (\phi \rightarrow g)$ or $(\omega_2 \rightarrow \neg g) \wedge (\phi \rightarrow$ ¬ g) is valid. $\Diamond$

Examples. A valuation v for a wff a is a set of truth assignments to all the ground atomic formulas of α . If ω_1 is p and ω_2 is p V T, then ω_2 is satisfied by a valuation that assigns F to p , while ω_1 is not; the two formulas do not satisfy condition (1) for equivalence. Therefore INSERT p WHERE T is not equivalent to INSERT p V T WHERE T; they differ on producing models where pis false. For updates INSERT p WHERE p A q and INSERT q WHERE p A q , V_1 and V_2 are both the empty set, and Theorem 3 correctly predicts equivalence.

Proof of Theorem 3. Assume that ϕ is satisfiable, as otherwise the theorem follows immediately. We first show that if condition (1) does not hold, then B_1 and B_2 are not equivalent. Select a valuation v such that, say, v $\in V_1$ and v $\notin V_2$. By definition of V_1, there must exist a valuation v'of ω_1 that agrees with v on all the ground atomic formulas of v , such that ω_1 is true under v'. Create an extended relational theory $\mathcal{T}$ with the following axioms:

Unique Name Axioms. For each pair of distinct constants c_1 , c_2 in L, include the axiom $c_1 \neq c_2$.

Completion Axioms. Represent each occurrence of an n-place predicate $P(c_{i1}, \ldots, c_{in})$ in ω_1, ω_2 , and ϕ in a completion axiom of the form $\forall x_1 \ldots \forall x_n(P(x_1, \ldots, x_n) \rightarrow ((x_1 = c_{11}$ A ..· A $x_n = c_{1n})$ V ... V $(x_1 = c_{i1}$ A · .. A $x_n = c_{in})$ V . ·· V $(x_1 = c_{m1}$ A ··· A $x_n \ = c_{mn})))$. Include the axiom $\forall x_1$ . .· $\forall x_n \neg P(x_1, \ldots, x_n)$ for all predicates P of $\mathcal{L}$ that do not appear in ϕ or w.

Non-Axiomatic Section. Select a valuation u for the ground atomic formulas of ω_1 , ω_2 , and ϕ that satisfies ϕ. For every ground atomic formula gin ω_1 , ω_2 , and ϕ, include the wff gin $\mathcal{T}$ if g is true under u , and $\neg$ g otherwise.

Clearly 7 is an extended relational theory. 7 must be consistent because ϕ is satisfiable. By construction, 7 has one alternative world, with model M . Let M^* be a model which agrees with v' on all valuations of v' , and with M on all others. Since ω_1 is satisfied in M^* by construction, and M * agrees with M on all ground atomic formulas not in ω_1 , it follows that the alternative world of M^* is produced by B_1 from M^* . M^* cannot be a model of an alternative world produced by applying B_2 to $\mathcal{T}$, because ω_2 is false in M^*. We conclude that condition (1) is necessary.

We now show that when condition (1) is met but condition (2) is violated (or, symmetrically, condition (3)), then B_1 and B_2 are not equivalent.

By supposition, neither $(\omega_1 \rightarrow$ g) A $(\phi \rightarrow$ g) nor $(\omega_1 \rightarrow \neg g)$ A $(\phi \rightarrow \neg$ g) is valid. Simplifying, this means that either both ω_1 A g and ϕ A $\neg$ g are satisfiable, or else both ω_1 A $\neg$ g and ϕ A g are satisfiable. Suppose ω_1 A g and ϕ A $\neg$ g are satisfiable; the other case is symmetric. Construct an extended relational theory $\mathcal{T}$ as before, with non-axiomatic section such that 7 has a single alternative world, with model M , that satisfies ϕ A $\neg$ g . Then when B_1 is applied to M , since ω_1 A g is satisfiable, a model $\mathcal{M}_4$ is produced in which the truth valuation for g has been changed to T. Since B_2 cannot change the valuation of g , this means that B_1 and B_2 cannot be equivalent. We conclude that B_1 and B_2 cannot be equivalent if conditions (2) or (3) are violated.

We now turn to the reverse implication, namely, that if conditions (1), (2), and (3) are met, then B_1 and B_2 are equivalent. Assume that ω_1 , and therefore ω_2 , is satisfiable, as otherwise the theorem follows immediately.

For any extended relational theory $\mathcal{T}$ over $\mathcal{L}$ or an extension of $\mathcal{L}$, consider the effects of B_1 and B_2 on a model M of 7 where ϕ is true. Since ω_1 is satisfiable, B_1 must produce the alternative world of some model $\mathcal{M}^*$ from M . We wish to show that the alternative world of $\mathcal{M}^*$ is also an alternative world produced by B_2 acting on M.

Since $V_1 = \mathcal{Y}$, ifevery ground atomic formula of ω_2 is in I, then ω_2 is true in M *. If g is a ground atomic formula appearing in ω_2 but not in ω_1 , say, condition (3) says that the valuation of g is the same in M *as* in M^*; and similarly if g appears in ω_1 but not in ω_2 . We conclude that ω_2 must also be true in M^*. Therefore M^* satisfies rules 1 and 2 of the definition of INSERT, and is a model of an alternative world produced by B_2 from $\mathcal{M}$. ◇

What conditions govern equivalence when the two updates have different selection clauses? Intuitively, if B_1 and B_2 are two equivalent ground updates with selection clauses ϕ_1 and ϕ_2, then B_1 must not make any changes in any model where ϕ_2 is false, and vice versa.

Theorem *4.* Let B_1 and B_2 be two INSERT ground updates over a language ***L:***

B_1*:* INSERT ω_1 WHERE ϕ_1 ,

B_2*:* INSERT ω_2 WHERE ϕ_2 .

Then B_1 and B_2 are equivalent iff

(1) INSERT ω_1 WHERE $\phi_1 \wedge \phi_2$ is equivalent to INSERT ω_2 WHERE $\phi_1 \wedge \phi_2$; and

(2) (ϕ_1 A $\neg \phi_2$)$\rightarrow \omega_1$ and (ϕ_2 A $\neg \phi_1$)$\rightarrow \omega_2$ are valid; and

(3) if ϕ_1 A $\neg \phi_2$ is satisfiable, then there exists exactly one valuation of ω_1 that makes ω_1 true; and if ϕ_2 A $\neg \phi_1$ is satisfiable, then there exists exactly one valuation of ω_2 that makes ω_2 true. ◇

Proof of Theorem *4.* We will use the idea of the conjunction c_v corresponding to a valuation v : c_v is a wff formed by including the conjunct 'g' iff g is a ground atomic formula true under v, and including the conjunct '$\neg$ g' iff g is a ground atomic formula false under v .

We first show that condition (3) is necessary. Suppose that, say, ϕ_1 A $\neg \phi_2$ is satisfiable with valuation u ; the proof will be symmetric if ϕ_2 A $\neg \phi_1$ is satisfiable. Let 7 be an extended relational theory constructed as in Theorem 4, with non-axiomatic section c_u . Then 7 has one alternative world, with model *M* , and B_2 applied to $\mathcal{T}$ does not change that alternative world. For B_1 to be equivalent to B_2 , then, B_1 cannot change the alternative world of 7. Since the number of alternative worlds B_1 produces from $\mathcal{M}$ will be equal to the number of valuations for ω_1 that satisfy ω_1 , if B_1 is equivalent to B_2 there must be only one valuation, v , that satisfies ω_1 . Further, since the alternative world produced by B_1 must be the same as the alternative world of 7, v must agree with u on all ground atomic formulas in v . Since u may be any valuation satisfying ϕ_1 A $\neg \phi_2$, v must be a subset of every valuation satisfying ϕ_1 A $\neg \phi_2$; in other words, ϕ_1 A $\neg \phi_2 \rightarrow c_v$; since c_v is logically equivalent to ω_1 , ϕ_1 A $\neg \phi_2 \rightarrow \omega_1$, implying that condition (2) is also necessary.

We now show that condition (1) is necessary. Suppose that B_1' is INSERT ω_1 WHERE ϕ_1 A ϕ_2 and B_2' is INSERT W2 WHERE ϕ_1 A ϕ_2 , and B_1' and B_2' ***axe*** not equivalent. Then B_1' and B_2' give different results when applied to some extended relational theory 7 ; in particular, they produce different sets of alternative worlds when applied to some model ***M*** of 7. We assume without loss of generality that 7 represents only a single alternative world, that of ***M .*** Since B_1' and B_2' must produce the same results on models where ϕ_1 A ϕ_2 is false, ϕ_1 A ϕ_2 must be true in ***M*** . Then B_1 applied to ***M*** will produce the same set of alternative worlds as B_1' applied to ***M*** , since both ϕ_1 and ϕ_1 A ϕ_2 are true in ***M .*** The same relation holds between B_2 and B_2' ; therefore B_1 and B_2 applied to 7 produce the same results as B_1' and B_2' , respectively, applied to $\mathcal{T}$, and B_1 and B_2 cannot be equivalent. We conclude that condition (1) is necessary for equivalence.

We now turn to the reverse implication, namely, that if conditions (1) through (3) are met, then B_1 and B_2 are equivalent. Let 7be an extended relational theory over $\mathcal{L}$ or an extension of $\mathcal{L}$, and ***M*** a model of 7. If $\neg \phi_1$ A $\neg \phi_2$ is true in ***M*** , then B_1 and B_2 produce the same set of models Sfrom ***M .*** If ϕ_1 A ϕ_2 is true in ***M*** , then since B_1' and B_2' are equivalent, again B_1 and B_2 must be equivalent when applied to ***M .*** If ϕ_1 A $\neg \phi_2$ is true in $\mathcal{M}$, then any model produced by B_2 represents the same alternative world as ***M .*** By condition (2), the same is true of B_1 applied to ***M*** . A similar line of reasoning holds if ϕ_2 A $\neg \phi_1$ is true in ***M*** **.** We conclude that B_1 and B_2 are equivalent when applied to 7. ◇

Note that although syntax is important in updates, it does not play a role in the non-axiomatic sections of extended relational theories: if two extended relational theories have the same axioms, then they will have identical sets of alternative worlds after a series of updates iff the non-axiomatic sections of the two theories are logically equivalent.

3.5. Extended Relational Theories with Type and Dependency Axioms

Until now, we have considered extended relational theories without type and dependency axioms, because the complications introduced by those axioms are orthogonal to the other issues in updating extended relational theories. We now expand the definition of an extended relational theory as follows: Distinguish a particular set ***A*** of unary predicates of 7 as the attributes of $\mathcal{T}$ **.** Then add two requirements to items l-3 in the definition of an extended relational theory in Section 2:

4. Type Axioms: The type axioms encode the schema of the database in logic. For each n**-ary** predicate ***P*** not in ***A,*** 7 contains exactly one axiom of the form

$$\forall x_1 \forall x_2 \cdots \forall x_n (P(x_1, x_2, \ldots x_n) \rightarrow (A_1(x_1) \wedge A_2(x_2) \ldots \mathrm{A}\ A_n(x_n))),$$

where $A_1, \ldots,$ ***A,*** are predicates in ***A.*** Further, each predicate in ***A*** must appear in one or more type axioms, and the non-axiomatic section of 7 must always be such that removing the type and dependency axioms from 7 does ***not change the*** models ***of 7.*** The reasons for this restriction will become apparent when we consider the changes needed in algorithm GUA to handle type axioms.

5. *Dependency Axioms:* 7 may optionally contain a set of **wffs** not containing predicate constants, designated as dependency axioms. We consider universally quantified dependencies of a template form, such as a functional or relation-inclusion dependency:

$$\forall x_1 \cdots \forall x_n (\alpha \rightarrow \beta),$$

where α is a conjunction of atomic formulas g_1 through g_m, β is quantifier-free, and x_1 through x_n appear in α. For example, a typical functional dependency would be $\forall x_1 \forall x_2 \forall x_3$ $((P(x_1, x_2) \wedge P(x_1, x_3)) \rightarrow x_2 = x_3)$.

The non-axiomatic section of 7 must always be such that removing the type and dependency axioms from 7 ***does not change the models of*** 7. The reasons for this restriction will become apparent when we consider the changes needed in algorithm GUA to handle dependency axioms. ◇

The semantics of updates must be augmented to enforce items 4 and 5. There are a number of reasonable enforcement policies that may be adopted. For dependency axioms in ordinary databases, a dependency axiom violation is usually taken as a signal to repair the database, e.g., by adding additional tuples. In a database with incomplete information, such as an extended relational theory, the dependency axioms also serve the function of automatically weeding out alternative worlds that could not possibly be the actual world. What should be the policy of the extended relational theory update algorithms on this delicate interplay of functions? We choose to delegate this issue to a higher authority, such as a database administrator, and ourselves provide only a mechanism that uses type and dependency axioms to ***weed out impossible alternative worlds,*** If desired, an additional layer may be incorporated between the user and algorithm GUA (or directly into GUA) to modify update requests in order to save models that would otherwise be inadvertently removed. For example, if the user request is INSERT $R(a, b, c)$ WHERE T, then the type and dependency layer might automatically convert this to INSERT $R(a,b,c)$ A ***Al(a)*** A $A_2(b)$ A $A_3(c)$ WHERE T, where the three additional predicates are the attributes of ***R.***

In any case, we modify the update semantics by adding one additional proviso to INSERT, DELETE, and MODIFY: A model ***M**** produced by a ground update B from ***M*** must also satisfy the type and dependency axioms of 7 ; otherwise, ***M**** is removed from $\mathcal{S}$. In the proofs below, we will refer to this new provision as rule 3 in the definition of INSERT, to be appended to rules 1 and 2 of the original definition:

(3) the type and dependency axioms of 7 are true in ***M* .***

In this discussion, the type, dependency, and unique name axioms will be permanently fixed for each database schema. It is a simple matter to extend this to allow updates to the axioms such as adding new dependencies, constants, or relations.

For algorithm GUA to handle type and dependency axioms correctly, we need to change Step 2 slightly and add three additional steps at the end of the algorithm:

Step 2’. Add to completion axioms. In addition to the functions of Step 2, for every constant c_1 appearing as a value for attribute ***A*** in a ground atomic formula of w , if c_1 is not listed in the completion axiom for ***A,*** then add c_1 to that completion axiom in $\mathcal{T}_1$, and add $\neg A(c_1)$ to the non-axiomatic section of $\mathcal{T}_1$.

For example, if the update is INSERT Orders(700, 32, 9) WHERE T, and there was no order 700 prior to the update, then “700” must be added to the completion axiom for the OrderNo attribute; and similarly for the quantity 9 and the part number 32.

Step 5. Instantiate the type axioms. Although 7’now represents exactly the desired alternative worlds, 7’ may not yet be an extended relational theory. Recall that the models of an extended relational theory must not change if the type and dependency axioms are removed from the theory. To maintain this property after an update, the type axioms must be “instantiated” with the new ground atomic formulas in $\mathcal{T}'$ if (7’ − Dep − Ty) $\not\models$ 7’. If we did not do this, then “illegal” alternative worlds could suddenly become legal again after an update, if the violation was removed by the update. To meet this requirement with a minimum of effort, let $P(c_1, \ldots, c_n)$ be a ground atomic formula appearing in 7’. Suppose that the type axiom for ***P*** in 7 is $\forall x_1 \forall x_2 \cdots \forall x_n (P(x_1, \ldots, x_n) \rightarrow (A_1(x_1)$ A $A_2(x_2)$ A .. · A $A_n(x_n)))$. Then if it is the case that

(1) $P(c_1, \ldots, c_n)$ appears in w and for some i, $A_i(c_i)$ is not a formula of 7’ and $\omega \not\rightarrow A_i(c_i)$; or

(2) $A_i(c_i)$ appears in ω and $\omega \not\rightarrow A_i(c_i)$,

add the formula $P(c_1, \ldots, c_n) \rightarrow (A_1(c_1) \wedge A_2(c_2) \wedge \cdots$ A $A_n(c_n))$ to the non-axiomatic section of 7’, if it is not already present.

Step 6. Instantiate the dependency axioms. It suffices to instantiate the template dependency axiom

$$\forall x_1 \cdots \forall x_n((g_1 \wedge \ldots \wedge g_m) \rightarrow \beta),$$

where g_1 through g_m are atomic formulas, for those ground atomic formulas that unify with g_i of α . More formally, for every set of constants $c_1, \ldots, c_n$ in $\mathcal{L}$ such that the ground atomic formula

$$((g_i)^{x_1 \ldots x_n}_{c_1 \ldots c_n})$$

appears in 7'for all i, add

$$((\alpha \rightarrow \beta)^{x_1 \ldots x_n}_{c_1 \ldots c_n})$$

to the non-axiomatic section of 7'.

If we did not do this, then "illegal" alternative worlds could suddenly become legal again after an update, if the violation was removed by the update. For example, suppose that Q and P are one-place relations, and we have an inclusion dependency $\forall x(P(x) \rightarrow$ Q(x)). Then whenever a new tuple is added to ***P,*** such as ***P(a),*** the new formula ***P(a)*** $\rightarrow$ ***Q(a)*** should also be inserted unless it can be proved that ***Q(a)*** will be true in all models where ***P(a)*** is to be true. Similarly, if ***Q(a)*** is deleted from some alternative worlds while ***P(a)*** is still in the theory, then the new wff ***P(a)*** $\rightarrow$ ***Q(a)*** should be added to 7'.

Step 7. Add to completion axioms. Modify the non-axiomatic section and completion axioms of 7'as follows: For any ground atomic formula f first introduced into 7'in Steps 5 or 6, add f to the completion axiom for its predicate, and add $\neg f$ to the non-axiomatic section of 7'. Also, for every constant c appearing as a value for attribute A in a ground atomic formula of 7', if c is not listed in the completion axiom for ***A,*** then add ***c*** to that completion axiom in 7', and add $\neg A(c)$ to the non-axiomatic section of 7'. $\diamond$

Theorem 5. Given an extended relational theory 7 with type and dependency axioms and a legal ground update ***B*** , Steps l-7 of algorithm GUA correctly and completely performs ***B.*** $\diamond$

Proof of Theorem 5. First, by Lemma 1, the incorporation of Step ***2'*** does not change the result of performing Steps l-4 of algorithm GUA.

GUA will be correct if Step 4 produces exactly the correct set of alternative worlds, Steps 5-7 do not change that set of alternative worlds, and the final result 7'is an extended relational theory. First, by Lemma 1 the set of alternative worlds is not changed in Step 7. Steps 5 and 6 cannot change that set either, because the theory already contains the type and dependency axioms that are instantiated in those steps.

Now we will show that Steps l-4 produce exactly the correct set of alternative worlds. First, since every model of $\mathcal{T}_4$ satisfies the type and dependency axioms of $\mathcal{T}_4$, it follows that every alternative world that should be produced by ***B*** is represented by a model of $\mathcal{T}_4$. For the alternative world of a model $\mathcal{M}_4$ of $\mathcal{T}_4$ to be produced by ***B*** , it must be derived from a model $\mathcal{M}$ that satisfies not only the unique name axioms, completion axioms, and non-axiomatic section of $\mathcal{T}$ (as proved in Theorem 1), but also the type and dependency axioms of 7. But by the definition of an extended relational theory, the models of $\mathcal{T}$ minus all type and dependency axioms are the same as the models of 7; therefore $\mathcal{M}$ must be a model of 7.

It remains to verify that 7' meets all six criteria in the definition of an extended relational theory with type and dependency axioms. Again, we augment the proof of Theorem 1 by showing that $\mathcal{T}'$ obeys the restrictions on type and dependency axioms:

Type Axioms. We must show that (7' − Ty − D) $\models$ Ty, for Ty and D the set of type and dependency axioms, respectively. Suppose that $\mathcal{M}'$ is a model of 7' − Ty − D but not of 7', and that $\mathcal{M}'$ violates some type axiom of the form $\forall x_1 \cdot$ . . VX, $(P(x_1, \ldots, x_n) \rightarrow (A_1(x_1) \wedge A_2(x_2) \wedge \cdots$ A $A_n(x_n)))$. This type axiom was not violated by any model of 7 ; therefore for some set of constants $c_1, \ldots, c_n$ of $\mathcal{L}$, the update B must have inserted $P(c_1, \ldots, c_n)$ into an alternative world, or removed some $A_i(c_i)$ from an alternative world. More specifically, $P(c_1, \ldots, c_n)$ was in w , w $\not\rightarrow (A_1(c_1)$ A ... A $A_n(c_n))$, and $(A_1(c_1)$ A ... A $A_n(c_n))$ is not true in some model of 7'; or some $A_i(c_i)$ was in w , and w $\not\rightarrow A_i(c_i)$. But these are exactly the conditions for adding $P(c_1, \ldots, c_n) \rightarrow (A_1(c_1)$ A ... A $A_n(c_n))$ to 7' in Step 5; therefore this change in alternative worlds must take place after Step 5. But we have already shown that the alternative worlds of 7' do not change after Step 4.

Dependency ***Axioms.*** It remains to show that (7' − Ty − D) $\models$ D. Suppose that $\mathcal{M}'$ is a model of 7' − Ty − D but not of 7'. Then $\mathcal{M}'$ violates some template-style dependency axiom of the form $\forall x_1 \cdot \quad \cdot \cdot \forall x_n(\alpha \rightarrow \beta)$, where α is $g_1 \wedge g_2 \wedge \cdots g_m$, each g_i is an atomic formula, β is quantifier-free, and x_1 through x_n appear in α. It must be the case that for some set of constants $c_1, \ldots, c_n$ in $\mathcal{L}$, the ground atomic formulas $(g_i)^{x_1 \ldots x_n}_{c_1 \ldots c_n}$ are all true in $\mathcal{M}'$; and all these ground atomic formulas must appear in the non-axiomatic section of 7'. But by Step 6, the non-axiomatic section of 7' must also contain

$$(\alpha \rightarrow \beta)^{x_1 \ldots x_n}_{c_1 \ldots c_n},$$

so $\mathcal{M}'$ cannot be a model of 7' − Ty − D, contradicting our assumption. We conclude that (7' − Ty − D) $\models$ D and (7' − Ty − D) $\models$ 7'. This concludes the proof that 7' is an extended relational theory. ◇

Let us consider how type and dependency axioms interact with the results in section 3.4 on update equivalence. There is now a spurious way in which two updates B_1 and B_2 over $\mathcal{L}$ might be equivalent when applied to any extended relational theory over $\mathcal{L}$, caused by a certain relationship between $\mathcal{L}$, B_1 , and B_2 : B_1 and B_2 might be equivalent solely because certain alternative worlds produced by B_1 , say, and not by B_2 , violate the type axioms (i.e., schema) of every possible extended relational theory over $\mathcal{L}$. If this is the case, then augmenting $\mathcal{L}$ by a single one-place predicate will make B_1 and B_2 inequivalent. This is undesirable; if B_1 and B_2 are equivalent over $\mathcal{L}$, we would like them to be equivalent over all extensions of $\mathcal{L}$. For this reason, the definition of equivalence requires that B_1 and B_2 be equivalent over all extensions of $\mathcal{L}$.

Example. If $\mathcal{L}$ contains one two-place predicate, P_1, and two one-place predicates A_1 and A_2, then INSERT F WHERE T is spuriously equivalent to INSERT P_1 $(c_1, c_2) \wedge \neg A_1(c_1) \wedge \neg A_2(c_1)$WHERE T.

Theorem 6. Theorems 2 through 4 on update equivalence for extended relational theories still hold for extended relational theories with type and dependency axioms. ◇

Proof of Theorem 6. We first show that it suffices to consider extended relational theories without dependency axioms when proving results about update equivalence.

If two ground insertions B_1 and B_2 are equivalent when applied to extended relational theories with type and dependency axioms, then they must be equivalent when applied to extended relational theories without dependency axioms, as these constitute a proper subset.

Suppose now that B_1 and B_2 are equivalent when applied to any extended relational theory without dependency axioms, but are not equivalent when applied to some theory 7 that contains dependency axioms. Then for some model M of $\mathcal{T}$, B_1 must produce a different set of models from M than B_2 does. Let $\mathcal{N}$ be a model produced by, say, B_1 but not by B_2. Using the construction procedure of Theorem 4, construct a new extended relational theory $\mathcal{T}'$ with the same type and unique name axioms as 7, no dependency axioms, and a non-axiomatic section containing the wff gfor each ground atomic formula g that is true in M. Let the completion axioms of 7'include disjuncts only for the ground atomic formulas that are true in M. Let $\mathcal{M}'$ be a model of the single alternative world of 7'. Then B_1 applied to $\mathcal{M}'$ must produce the alternative world of $\mathcal{N}$, but B_2 cannot produce the alternative world of $\mathcal{N}$, a contradiction of our assumption. We conclude that it suffices to consider extended relational theories without dependency axioms when proving results about update equivalence.

We now consider the effect of adding type axioms to extended relational theories. Suppose that B_1 and B_2 are equivalent when applied to any extended relational theory with type axioms. Suppose $\mathcal{T}$ is an extended relational theory with no type axioms over some language $\mathcal{L}$, and B_1 and B_2 are not equivalent when applied to 7. Let $\mathcal{T}'$ be the same extended relational theory as $\mathcal{T}$, but with a set of type axioms added:

$$\forall x_1 \cdots \forall x_n(P(x_1, \ldots, x_n) \rightarrow (A(x_1) \mathrm{\ A} \ldots \mathrm{A\ } A(x_n)),$$

for all predicates P of $\mathcal{T}$ of arity greater than zero, where A is a one-place predicate not in $\mathcal{L}$. We also need a completion axiom for A:

$$\forall x(A(x) \rightarrow (\mathrm{x} = c_1 \vee \cdots \mathrm{v\ x} = c_m),$$

where x = c_i is a disjunct iff c_i appears in ϕ, w , or elsewhere in 7'. To the non-axiomatic section of 7', add $A(c)$ for every constant c appearing in the completion axiom for A. Then there is a one-to-one correspondence between models $\mathcal{M}$ of 7 and models $\mathcal{M}'$ of $\mathcal{T}'$, such that M agrees with M'on the truth valuations of all ground atomic formulas except those of A. Further, this mapping is preserved under application of B_1 and B_2, as every model produced by B_1 and B_2 from $\mathcal{M}'$ is guaranteed to satisfy the type axioms of 7'. By assumption, B_1 and B_2 are equivalent when applied to 7'; we conclude that they are also equivalent when applied to $\mathcal{T}$, or to any extended relational theory without type axioms.

Now suppose that B_1 and B_2 are equivalent when applied to any extended relational theory without type axioms. Let 7'be an extended relational theory over a language $\mathcal{L}$, and let 7 be an extended relational theory derived from 7'by relabeling its type axioms as dependency axioms. Then B_1 and B_2 are equivalent when applied to 7. But the semantics for INSERT do not differentiate between type and dependency axioms; the result of applying an insertion to 7is by definition the same as applying that insertion to 7'. We conclude that B_1 and B_2 must also be equivalent when applied to $\mathcal{T}'$, and that it suffices to consider extended relational theories without type and dependency axioms when proving results about equivalence. ◇

3.6. Cost of Algorithm GUA

Let g be the number of instances of ground atomic formulas in the ground update ***B;*** and let ***R*** be the greatest number of distinct occurrences in the extended relational theory 7of any predicate. If no dependency axioms are present, an optimized form of GUA runs in time O(glog(R)) (the same asymptotic cost as for ordinary database updates) and increases the size of 7 by O(g) worst case. This is not to say that an O(g log(R)) implementation of updates is the best choice; rather, it is advisable to devote extra time to heuristics for minimizing the length of the formulas to be added to 7. Nonetheless, a worst-case time estimate for GUA is informative, as it tells us how much time must be devoted to the algorithm proper.

To obtain this estimate, all ground atomic formulas in the non-axiomatic section of 7 must appear in indices, with one index per predicate, so that lookup and insertion time is O(log(R)). I?redicate constants, however, are referenced through a single separate index. The renaming step (Step 2) is the bottleneck for GUA. To make renaming fast, we assume that the ground atomic formulas of the non-axiomatic section are stored only as pointers. In particular, we assume that all occurrences of a ground atomic formula or predicate constant in the non-axiomatic section of 7 are linked together in a list whose head is an index entry, so that renaming may be done rapidly. Additionally, the names of ground atomic formulas cannot be physically stored with the non-axiomatic wffs they appear in; however, the non-axiomatic wffs may contain pointers into a separate name space where names of ground atomic formulas are kept.

Finally, we assume that the schema is fixed, i.e., that the number of predicates is a constant.

We now show the running time of each optimized algorithmic step:

Step 1. O(g log(R)) to add new negative formulas.

Step 2. O(g log(R)), if the cost of renaming is constant, as outlined above.

Steps 3 and 4. To improve efficiency, put all instantiations of formula (1) into one large implication. Then it will cost O(g log(R)) to add the new formulas to 7'.

Step 5. $O(g \log(R))$, if the testing of logical implications is reduced to a test of whether $A_i(c_i)$ is a conjunct of w .

Step 6. (Discussed below)

Step 7. Combine with Steps 5 and 6 at no extra asymptotic cost.

The cost of Step 6 (dependency checking) depends entirely on the type of dependency axioms. We derive costs for the simplest types of dependencies here, which can be given optimized enforcement algorithms .

If the dependency axiom is a functional dependency, then the cost of Step 6 is ***O(gR)*** worst case (when every updated tuple seems to conflict with every other tuple in its relation) and ***O(g*** log(R)) best case (when no conflicts occur). If the dependency axiom is a relation-inclusion dependency, then the cost is also is ***O(gR)*** worst case (when the removal of a tuple seems to invalidate every tuple in some other relation) and O(g log(R)) best case (when no conflicts occur). The same cost functions hold for a multivalued dependency as well.

4. Summary and Conclusion

We have defined extended relational theories as extensions of Reiter's theories for disjunctive information. Formulas in the body of an extended relational theory may be any ground wffs, and may contain auxiliary predicate constants that are not part of the database schema, thereby increasing the representational power of Reiter's theories. Within this context, we set forth a simple data manipulation language, LDML, and give model-theoretic definitions of the meaning of LDML updates. We concentrate on the concept of a ground update, or an LDML update without variables; updates with variables can be reduced to the problem of performing a set of ground updates simultaneously. We present an algorithm for performing ground updates, and prove it correct in the sense that the alternative worlds produced by updates under this algorithm are the same as those produced by updating each alternative world individually. For a particular extended relational theory $\mathcal{T}$, this algorithm runs in time proportional to the product of the number of atomic formulas in the update request and the logarithm of the size of the predicate with the largest number of distinct ground atomic formulas in $\mathcal{T}$; this is the same asymptotic cost as for ordinary complete-information database updates.

We conclude that, first, one may extend the concept of a database update to databases with incomplete information in a natural way; second, that first-order logic is a fruitful paradigm for the investigation; and third, that one may construct an algorithm to perform these updates with a reasonable running time.

An important topic that we have not found room to discuss here is the simplification of extended relational theories, as they grow steadily longer under the update algorithms presented. This is a vital concern, since it is in large part the possibility of heuristic simplification that makes the LDML algorithms more attractive than simply keeping a record of past updates and recomputing the state of the theory on each new query. A heuristic algorithm for simplification will be a vital part of any implementation of these algorithms, and is at the core of the implementation coded by the author.

5. Acknowledgements

This paper would not have been possible without the encouragement of Christos Papadimitriou and the emphasis on rigor of Moshe Vardi.

6. References

[Abiteboul 85] S. Abiteboul, G. Grahne, "Update Semantics for Incomplete Databases," *Proc. VLDB Conf.*, Stockholm, August 1985.

[Fagin 84] R. Fagin, G. M. Kuper, J. D. Ullman, and M. Y. Vardi, "Updating Logical Databases," *Proc.* of the 3rd *ACM* PODS, April 1984.

[Fagin 83] R. Fagin, J. D. Ullman, and M. Y. Vardi, "On the Semantics of Updates in Databases," *Proc.* of the 2nd ACM PODS, April 1983.

[Imielinski 84] T. Imielinski and W. Lipski, "Incomplete Information in Relational Databases," *Journal* of the *ACM*, **31**:4, October 1984.

[Reiter 84] R. Reiter, "Towards a Logical Reconstruction of Relational Database Theory," in M. Brodie, J. Myopoulos, and J. Schmidt (eds.) ***On Conceptual Modelling***, Springer-Verlag, 1984.

[Reiter 84b] R. Reiter, "Logical Foundations for Database Theory," ***Conference On Theory in Data Bases***, Benodet, France, July 1984.

[Vardi 85] M. Vardi, "Querying Logical Databases," ***Proc. of the 4th ACM PODS***, March 1985.

[Wilkins 86] M. Wilkins, "A Model-Theoretic Approach to Updating Logical Databases (Extended Abstract)," *Proc.* of the *5th* ACM PODS, March 1986.