Reasoning about Dynamic Databases: Tableaux for Single-Step-Update Models

CHRISTIAN G. FERMÜLLER, *Institut für Computersprachen, Technische Universität, Karlsplatz 13/E185.2, A-1040 Vienna, Austria. E-mail: chrisf@logic.at*

Abstract

We study the dynamic behaviour of step-wise evolving databases using concepts from modal and many-valued logic. We focus on modelling "single-step", i.e., atomic update operations, which change the truth value of at most one basic unit of information at a time. The resulting logics form a variant of standard modal logics. These "update logics" are generalized to a many-valued context to take into account possibly incomplete and inconsistent data. In particular, Belnap's well known four-valued logic is combined with the concept of distribution modalities and underlying Kripke structures. A special family of signed and prefixed analytic tableaux is introduced as an adequate tool for proof search in this context.

1 Introduction and summary

Various approaches employing modal logics for both the representation of knowledge and mechanized reasoning about data have been investigated. See, e.g., [3, 14, 6, 10, 9] for some recent work of relevance to database theory.

Here we introduce a model of dynamic databases that is particularly simple. The states of a database are identified with assignments of truth values to basic propositions. Some states are considered as results of updating other states of the database. In other words, a binary *update relation* is defined over the set of possible states. This clearly amounts to defining usual Kripke stuctures. Standard modal logics arise if we augment classical propositional logic over the signature of basic propositions with the modalities $\Box$ and $\Diamond$ (interpreted as "in all updated states" and "in some updated state", respectively).

Literature on the so-called "update problem" usually aims at formalizing changes in databases triggered by *arbitrary complex* changes in the environment to which the database refers. Here however, we want to model only *atomic* or *"single-step" updates*. More exactly, each update operation is assumed to change the truth value of at most one basic proposition at a time. In general, single-step updates reflect this type of adaption to a changing environment only via *sequences* of such atomic update operations. However, we think that considering single-step updates leads to a *more realistic* model of the actual computational behaviour of dynamic databases at a less abstract level. Certainly the evolution of any database proceeds in basic steps, each of which corresponds to some well defined atomic action that can be performed on single data. We aim at a (conceptually as well as technically) *simple* logical model of this aspect of dynamic databases.

The subtle computational constraint on the update relation — namely: to consider only atomic update operations — seems to have dramatic effects: the set of formulae valid in all corresponding Kripke models is not closed under substitution. In response to this fact we propose to use a *two sorted* propositional language that allows to distinguish between "atoms" (basic propositions of the database) and genuine propositional variables; and consequently between "concrete" and "schematic" statements about data. We define a corresponding semantics and provide complete and sound *tableau calculi* for the resulting logics.

Moreover, we generalize the just described model of atomic dynamic databases to scenarios allowing for incomplete and inconsistent information about (atomic as well as complex) propositions. Replacing classical logic by Belnap's well-known four-valued logic [4] opens the space for new types of modal operators over corresponding update models. Some examples of such *distribution modalities* expressing properties of updates will be investigated.

A variant of the tableaux for finite-valued logics with distribution modalities introduced in [7] turns out to be adequate for formalizing reasoning in corresponding logics.

A main motivation for presenting these ideas at this particular workshop is that all four areas of research adressed by it's name — Logic, Language, Information and Computation — are in an obvious and essential way involved in the proposed approach to reasoning about dynamic databases.

We emphasize that the concepts and results presented here should be considered only as a first step in exploring the scope and limits of many-valued Kripke structures and distribution modalities in the context of reasoning about dynamic databases. Accordingly, we conclude with a list of future topics of research.

2 Atomic databases and update models

Our first object of investigation is argueably the most simple logical model of a database: it refers to a fixed set of basic propositions (called *atoms*) and presumes that the only information explictly contained in the database is which of those atomic propostions hold and which do not hold. More formally, by a *(classical) state (of a database)* we mean a total function of type $\mathsf{atoms} \mapsto \{\mathbf{t}, \mathbf{f}\}$, where atoms is a non-empty, finite set of propositional atoms.[1] Obviously we can evaluate arbitrary propostional formulae over the signature atoms with respect to a state α as usual:

- $v_\alpha(p) = \alpha(p)$, for $p \in \mathsf{atoms}$
- $v_\alpha(\top) = \mathbf{t}$
- $v_\alpha(\neg A) = \mathbf{t}$ iff $v_\alpha(A) = \mathbf{f}$
- $v_\alpha(A \circ B) = \widetilde{\circ}(v_\alpha(A), v_\alpha(B))$

where $\circ \in \{\wedge, \vee, \supset\}$ and $\widetilde{\circ}$ is the classical truth function associated with the binary connective $\circ$.

State α' is called a *single-step-update* of a state α if the following condition is satisfied:

(ss) $\alpha(p) = \alpha'(p)$ for all *except at most one* $p \in \mathsf{atoms}$.

[1] Throughout this paper we will consider the set of atoms to be fixed for any given database and may therefore surpress reference to it in other definitions.

States of a database can be viewed as worlds of a Kripke model, where a world β is accessible from a world α iff β is a single-step-update of α.

DEFINITION 2.1
A *single-step-update model* is a pair $\mathcal{M} = (\Sigma, U)$ where

- Σ is a set of states of a database over (a fixed set) atoms, i.e., a set of functions of type $\mathsf{atoms} \mapsto \{\mathbf{t}, \mathbf{f}\}$, and
- U is a binary relation over S, subject to the restriction that $\forall \alpha, \alpha' \in \Sigma\colon \alpha U \alpha'$ implies that α' is a single step update of α.

We augment the propostional language by the modal operators $\Box$ and $\Diamond$, with the intended meaning "in all updates" and "in some update", respectively. More exactly—referring to states α of a single-step-update model $\mathcal{M} = (\Sigma, U)$—we extend the definition of v_α as follows:

- $v_\alpha^{\mathcal{M}}(\Box A) = \mathbf{t}$ iff $\forall \beta \in \Sigma\colon \alpha U \beta$ implies $v_\beta^{\mathcal{M}}(A) = \mathbf{t}$, and
- $v_\alpha^{\mathcal{M}}(\Diamond A) = \mathbf{t}$ iff $\exists \beta \in \Sigma\colon \alpha U \beta$ and $v_\beta^{\mathcal{M}}(A) = \mathbf{t}$.

This simple logical formalism—close to familiar model structures—allows to express statements that refer not only to the current state of a database but also to states that are the result of an update.

EXAMPLE 2.2
The formula $A \supset \Box A$ may be paraphrased as "If the statement A is currently supported by the database, then all possible updates will still support A". Similarly $\Diamond A \wedge \Diamond \neg A$ expresses that A is a statement that will be evaluated differently in different updates of the database. Likewise we can express that there is no possible update by "$\neg \Diamond \top$". The statement that for every update (of the current state) a further update is possible is expressed by "$\Box \Diamond \top$" etc.

By requiring the update relation to fulfill simple properties we can adapt our model to databases which obey certain dynamic constraints. E.g., requiring the update relation U to be symmetric corresponds to modelling databases for which every update is reversable. Similary in many applications it will be useful to require U to be reflexive or serial.

DEFINITION 2.3
A single-step-update model is called *update* B-, D-, T-, or TB-*model* if its update relation is symmetric, serial, reflexive, or symmetric and reflexive, respectively. The class of *all* single-step-update models is called *update* K-*models*.

3 Concrete versus schematic statements

It might seem as if—so far—we have only described just another interpretation of normal modal logic. However, in insisting that the truth value of *only one* atom can be changed in one update operation we ensured that, e.g., the formula

$$F_1 = (p \wedge q) \supset \Box(p \vee q)$$

is evaluated true in all states of all single-step-update models, if p and q are different atoms. In contrast, substituting in F_1 p for q results in a formula which is false at all states which update p from $\mathbf{t}$ to $\mathbf{f}$. In other words, the set of formulae that are true at all points of a single-step-update model is not closed under substitution.

There is a simple way of recovering closure under substitution:

DEFINITION 3.1
A formula A is *schematically valid* in a single-step-update model $\mathcal{M} = (\Sigma, U)$ if for all substitution instances A' of A and all $\alpha \in \Sigma$: $v_\alpha^M(A') = \mathbf{t}$.

The set of formulae that are schematically valid in all update $\mathcal{L}$-models is called *update-$\mathcal{L}$* (for $\mathcal{L} \in \{\mathsf{K}, \mathsf{B}, \mathsf{D}, \mathsf{T}, \mathsf{TB}\}$).

By definition these update-$\mathcal{L}$ are closed under substitution. It is easy to see that they are closed under modus ponens and the necessity rule, as well. Indeed we have the following result:

THEOREM 3.2
For $\mathcal{L} \in \{\mathsf{K}, \mathsf{B}, \mathsf{D}, \mathsf{T}, \mathsf{TB}\}$, update-$\mathcal{L}$ and (ordinary) $\mathcal{L}$ coincide.

REMARK 3.3
This result can be viewed as a strengthening of completeness for the standard normal logics $\mathcal{L}$. It says that for every non-valid formula F there is a counter-model of an instance of F that obeys the single-step-update restriction **(ss)**. Indeed, the proof of the theorem consists in an explicit construction of such a substitution instance and its corresponding update counter-model.

We are interested in reasoning about changing databases both, at the level of schematic statements and by evaluating statements referring to concrete atoms of a database. Theorem 3.2 tells us that we remain within standard normal modal logics as long as only schematic statements are considered.[2] In order to be able to refer to the schematic as well as the concrete level simultaneously we define the languange $\mathcal{UL}$ over a two-sorted signature:

- An *atomic formula* of $\mathcal{UL}$ is either an element $p \in \mathsf{atoms}$ or a propositional variable[3] or $\top$.
- *Complex formulae* of $\mathcal{UL}$ are built up as usual from the atomic formulae using the connectives $\neg, \wedge, \vee, \supset$ and the modalities $\Box$, $\Diamond$.

A formula of our extended language is called *concrete* if it does not contain propositional variables. Otherwise, it is called *schematic*. For concrete formulae F $v_\alpha^{\mathcal{M}}(F)$ is defined as in Section 2. An arbitrary (possibly schematic) formula F is called *valid* in $\mathcal{M}$ if for all concrete formulae F', that arise by substituting the propositional variables of F with concrete formulae we have $v_\alpha^{\mathcal{M}}(F') = \mathbf{t}$ for all states α of $\mathcal{M}$.

Notation. We use lower case letters for atoms. Different letters always denote different atoms. Propositional variables are denoted by upper case letters from the end of the alphabet.

EXAMPLE 3.4
The concrete formula $(a \wedge \neg b) \supset \Diamond(b \supset a)$ is valid in all update models. However the schematic formula $(X \wedge \neg b) \supset \Diamond(b \supset X)$ is not valid in most update models.

[2] By a schematic statement we mean one in which the propositional variables range over all possible (non-modal and modal) statements over the signature atoms.

[3] The set of propositional variables and atoms are disjoint.

The concrete formula $\phi = \Diamond(a \wedge b) \wedge \Diamond(a \wedge \neg b) \wedge \Diamond(\neg a \wedge b) \wedge \Diamond(\neg a \wedge \neg b)$ can never evaluate to $\mathbf{t}$. In other words $\neg\phi$ is valid in all update models. In contrast, it is easy to find counter models for $\Box(X \vee Y) \vee \Box(X \vee \neg Y) \vee \Box(\neg X \vee Y) \vee \Box(\neg X \vee \neg Y)$.

4 Prefixed tableaux adapted to update models

We have so far defined adequate syntax and semantics of a language that allows to express various statements with respect to changing databases (of a particularly simple type). To substantiate the claim that this formalism provides a basis for *reasoning* we have to define sound and complete calculi. Often one is content with Hilbert style axiomatizations, however analytic (Gentzen style) calculi are certainly much more adequate if we aim at *automatic* deduction. Fortunately, Fitting's analytic *prefixed tableaux* [8] for standard normal logics turn out to be easily adaptable to our scenario. (See also [12] for an overview and history of related methods.)

We assume familiarity with tableaux but review some relevant terminology.

A *prefix* is a finite sequence of natural numbers intended to denote a world of a Kripke structure. A *prefixed formula* is a pair consisting of a prefix σ and formula F written as $\sigma :: F$. *Prefixed tableaux* are downward rooted trees of prefixed formulae, generated by appending new prefixed formula to a branch of a tree according to three types of rules.

Non-modal rules:

The rules for negation and disjunction are as follows:

$$(\neg\neg)\ \frac{\sigma :: \neg\neg F}{\sigma :: F} \qquad (\vee)\ \frac{\sigma :: F \vee G}{\sigma :: F \mid \sigma :: G} \qquad (\neg\vee)\ \frac{\sigma :: \neg(F \vee G)}{\begin{array}{c}\sigma :: \neg F\\ \sigma :: \neg G\end{array}}$$

The rules for conjunction and implication are similar.

Modal rules:

The rules for analyzing the modality $\Box$ in the basic modal logic K are

$$(\mathrm{K})\ \frac{\sigma :: \Box F}{\sigma.x :: F} \qquad (\pi)\ \frac{\sigma :: \neg\Box F}{\sigma.n :: \neg F}$$

where n is such that the prefix $\sigma.n$ is new to the current branch. The K-rules for $\Diamond$ are dual.

For serial, reflexive, and symmetric models we have to add the following rules, respectively:

$$(\mathrm{D})\ \frac{\sigma :: \Box F}{\sigma :: \Diamond F} \qquad (\mathrm{T})\ \frac{\sigma :: \Box F}{\sigma :: F} \qquad (\mathrm{B})\ \frac{\sigma.x :: \Box F}{\sigma :: F}$$

Closure rules:

The closure rules for standard modal logics are

$$\frac{\begin{array}{c}\sigma :: \neg F\\ \sigma :: F\end{array}}{\texttt{closed}} \qquad \frac{\sigma :: \neg\top}{\texttt{closed}}$$

A tableau is *closed* if all its branches are closed. It is well known (see, e.g., [8, 12]) that the calculus sketched above is sound and complete for the respective modal logics. I.e., there exists a closed tableau constructed according to the above rules starting with root $0 :: \neg F$ if and only if F is valid in the corresponding modal logic K, B, D, T, or TB.

To accomodate for the difference between atoms and propositional variables in the language $\mathcal{UL}$ as well as for the single-step-update condition in update models it suffices to extend the standard tableau calculi by additional closure rules.

Single-Step closure rules:

$$\frac{\begin{array}{c}\sigma :: a\\ \sigma.n :: \neg a\\ \sigma :: b\\ \sigma.n :: \neg b\end{array}}{\texttt{closed}} \qquad \frac{\begin{array}{c}\sigma :: \neg a\\ \sigma.n :: a\\ \sigma :: b\\ \sigma.n :: \neg b\end{array}}{\texttt{closed}} \qquad \frac{\begin{array}{c}\sigma :: a\\ \sigma.n :: \neg a\\ \sigma :: \neg b\\ \sigma.n :: b\end{array}}{\texttt{closed}} \qquad \frac{\begin{array}{c}\sigma :: \neg a\\ \sigma.n :: a\\ \sigma :: \neg b\\ \sigma.n :: b\end{array}}{\texttt{closed}}$$

where a and b are different atoms.

A tableau constructed according to the above rules is called *update $\mathcal{L}$-tableau*, where $\mathcal{L}$ refers on of the logics K, B, D, T, or TB, and the corresponding modal rules.

THEOREM 4.1
For $\mathcal{L} \in \{\mathsf{K}, \mathsf{B}, \mathsf{D}, \mathsf{T}, \mathsf{TB}\}$: an $\mathcal{UL}$-formula F is valid in all update $\mathcal{L}$-models iff there exists a closed *update* $\mathcal{L}$-tableau with root $0 :: \neg F$.

5 Incomplete and inconsistent data

Our model of the dynamic behaviour of a database is yet to simple to capture phenomena like possibly incomplete and inconsistent data. However, we claim that the basic formalism of single-step update models and corresponding tableaux is easily adapted to such scenarios.

Belnap's four-valued logic [4] has repeatedly been suggested as a tool for reasoning about (possibly) inconsistent and incomplete information. The main intuition in this context is that a database may not only contain information implying that a statement is *false* or *true*, but such information may also be absent or inconsistent. (I.e., the database might assign none or both classical truth values $\mathbf{t}$, $\mathbf{f}$ to an atom.) The four possible states of knowledge are represented by the four truth values $\mathbf{f}$ (*false*), $\mathbf{u}$ (*undetermined*), $\bot$(*inconsistent*), and $\mathbf{t}$ (*true*), respectively. This intended interpretation induces the following truth functions for the connectives $\neg$, $\wedge$, and $\vee$:

	$\neg$
f	**t**
u	**u**
$\bot$	$\bot$
t	**f**

$\wedge$	**f**	**u**	$\bot$	**t**
f	**f**	**f**	**f**	**f**
u	**f**	**u**	**f**	**u**
$\bot$	**f**	**f**	$\bot$	$\bot$
t	**f**	**u**	$\bot$	**t**

$\vee$	**f**	**u**	$\bot$	**t**
f	**f**	**u**	$\bot$	**t**
u	**u**	**u**	**t**	**t**
$\bot$	$\bot$	**t**	$\bot$	**t**
t	**t**	**t**	**t**	**t**

For the definition of other connectives (in particular forms of implication) and the choice of designated truth values we refer to the extensive investigations of Avron and Arieli (see, e.g., [2, 1]). Here we follow a remark in [7] and argue that Kripke structures based on four-valued "Belnap-worlds" (instead of two-valued worlds) are adequate for modelling changing databases.

The many-valued context allows to extend the classical universal and existential modalities to the more general concept of *distribution modalities*, introduced in [7]. Let $\mathcal{V}$ be the set of truth values and, correspondingly, a world of a (many-valued)

Kripke-model an assignment α : $\mathsf{atoms} \mapsto \mathcal{V}$. Then any function $\widetilde{\mu}$ of type $2^{\mathcal{V}} \mapsto \mathcal{V}$ induces a truth function of a distribution modality μ by:

$$v_\alpha^{\mathcal{M}}(\mu F) = \widetilde{\mu}(\{v_\beta^{\mathcal{M}}(F) \mid \beta \in W \colon \alpha R \beta\}).$$

Here W is the set of worlds of the model $\mathcal{M}$ and R its accessibility relation. $\{v_\beta^{\mathcal{M}}(F) \mid \beta \in W \colon \alpha R \beta\}$ is called *distribution* of F in $\mathcal{M}$ at α. This definition directly transfers to update models based on many-valued assignments. In particular, we call the update models in which the states of the database consist in assignments of type $\mathsf{atoms} \mapsto \{\mathbf{t}, \mathbf{u}, \mathbf{f}, \bot\}$ *Belnap update structures.* This context allows us to define modalities like

- $\mathit{det}(F)$ with the intended meaning: "No update renders information on F incomplete or inconsistent", and
- $\mathit{unif}(F)$ with the intended meaning: "F is evaluated uniformly in all updates".

Since $\mathit{det}(F)$ is intended to express a meta-linguistic (and therefore classical) property of F within the object language itself, it always evaluates to $\mathbf{t}$ or $\mathbf{f}$. More exactly, its semantics is fixed by:

$$\widetilde{\mathit{det}}(W) = \begin{cases} \mathbf{t} & \text{if } W = \emptyset, \{\mathbf{f}\}, \{\mathbf{t}\}, \text{or } \{\mathbf{t}, \mathbf{f}\} \\ \mathbf{f} & \text{otherwise} \end{cases}$$

On the other hand "uniform evaluation" admittedly is an ambigous concept. Certainly, we want $\mathit{unif}(F)$ to be *true* if set of all truth values to which F is evaluated in some update is $\{\mathbf{f}\}$ or $\{\mathbf{t}\}$. Likewise, it is clear that $\mathit{unif}(F)$ is *false* if the distribution contains $\mathbf{t}$ *and* $\mathbf{f}$ (i.e., if there is an update evaluating the formula to $\mathbf{t}$, but also another update that evaluates it to $\mathbf{f}$.) But we want $\mathit{unif}(F)$ to be *undetermined* if the distribution of F contains $\mathbf{u}$. One way to round off and formalize these intuitions is to define the truth function for unif as follows:

$W \subseteq \mathcal{V}$	$\widetilde{\mathit{unif}}(W)$
$\emptyset, \{\mathbf{f}\}, \{\mathbf{t}\}$	$\mathbf{t}$
$\{\mathbf{u}\}, \{\mathbf{u}, \mathbf{f}\}, \{\mathbf{u}, \mathbf{t}\}$	$\mathbf{u}$
$\{\mathbf{f}, \mathbf{t}\}, \{\mathbf{f}, \mathbf{u}, \mathbf{t}\}, \{\mathbf{f}, \bot, \mathbf{t}\}, \{\mathbf{f}, \mathbf{u}, \bot, \mathbf{t}\}$	$\mathbf{f}$
$\{\bot\}, \{\mathbf{f}, \bot\}, \{\bot, \mathbf{t}\}, \{\mathbf{u}, \bot\}, \{\mathbf{f}, \mathbf{u}, \bot\}, \{\mathbf{u}, \bot\, \mathbf{t}\}$	$\bot$

Of course, det and unif are just two simple examples. Observe that there are 4^{2^4} possible distribution modalities definable over Belnap update structures. All of them refer to properties of the "truth status" of statements with respect to the class of possible stepwise evolutions of the database. We claim that at least some of those properties are of interest in the context of modelling the logical structure of dynamic databases.[4] Moreover, we remind the reader that Belnap update models come in different variants according to the constraining properties of the update relation.

To define a particular *Belnap update logic* with respect to a class of Belnap update models we therefore have to fix three independent parameters (in addition to the set of atoms and propositional variables):

(1) a set of designated truth values $\mathcal{D}$; usually $\{\mathbf{t}\}$ or $\{\mathbf{t}, \bot\}$

[4] Space limits do not allow us to investigate more distribution modalities from this point of view, here. We plan to do so in a longer version of this paper.

(2) a set of $\{\mu_1, \dots, \mu_n\}$ of distribution modalities (with associtated truth functions $\widetilde{\mu_1}, \dots, \widetilde{\mu_n}$ and four-valued connectives (specified by their truth tables)

(3) properties like symmetry or reflexivity, which we want the update relation to observe.

We call a formula F *valid* in such a logic if $v_\alpha^{\mathcal{M}}(F) \in \mathcal{D}$ for all states α of all corresponding single-step update models $\mathcal{M}$.

6 Prefixed signed tableaux for Belnap update models

It is well known that appropriate analytic calculi for all (truth functional) finite valued logics can be defined using *signed* versions of tableaux (see, e.g., [13]). These can be extended to finite valued modal logics by combining prefixes (denoting worlds) and signs (denoting truth values) as was shown in [7]. We describe a somewhat simplified variant of the latter calculi, adapted for update structures, for the special case of Belnap update models with serial (but otherwise general) update relation.[5]

A prefixed signed formula is a triple consisting of a finite sequence of natural numbers σ (called prefix), a truth value v, and a formula F, written as $\sigma\colon [v]\colon F$.

Remark 6.1
In classical logic the prefixed signed formulae $\sigma\colon [\mathbf{t}]\colon F$ and $\sigma\colon [\mathbf{f}]\colon F$ are just notational variants of the prefixed formulae $\sigma :: F$ and $\sigma :: \neg F$, respectively. For many-valued logics truth value signs are not only an elegant way to make semantic information explicit but are, in general, *needed* to obtain complete tableau calculi.

Again, a (prefixed signed) tableau is a downward rooted tree of prefixed signed formulae, constructed using the following rules.

Non-modal rules: can directly be read off from the truth tables of connectives. We refer to [13, 15] for general methods and results about constructing optimal rules.

Closure rules: The standard closure rule is

$$\frac{\begin{array}{c}\sigma\colon [v]\colon F\\ \sigma\colon [w]\colon F\end{array}}{\texttt{closed}}$$

where v and w are different truth values.

A modal operator μ induces an additional closure rules if a formula μF never evaluates to a particular truth value. E.g., the modality *det* triggers the following two closure rules:

$$\frac{\sigma\colon [\mathbf{u}]\colon \mathit{det}(F)}{\texttt{closed}} \qquad \frac{\sigma\colon [\bot]\colon \mathit{det}(F)}{\texttt{closed}}$$

Modal rules:

A general method for constructing modal rules from associated truth functions is described in [7]. We represent the two remaining cases of *det*-modalized formulae:

$$\frac{\sigma\colon [\mathbf{t}]\colon \mathit{det}(F)}{\begin{array}{c}\sigma.n\colon [\mathbf{t}]\colon F\\ \sigma.m\colon [\mathbf{f}]\colon F\end{array}} \qquad \frac{\sigma\colon [\mathbf{f}]\colon \mathit{det}(F)}{\sigma.n\colon [\mathbf{u}]\colon F \;\big|\; \sigma.n\colon [\bot]\colon F}$$

[5] Other properties of the update relation result in simple technical variations of some modal rules. The corresponding calculi are omitted here for space reasons.

where $\sigma.n$ and $\sigma.m$ are different (but not necessarily new on the branch).

Single-step closure rule:

$$\frac{\begin{array}{c}\sigma : [u_1] : a \\ \sigma.n : [v_1] : a \\ \sigma : [u_2] : b \\ \sigma.n : [v_2] : b\end{array}}{\texttt{closed}}$$

where a and b are different atoms and $v_i \neq u_i$ for $i \in \{1, 2\}$.

THEOREM 6.2
A formula F is valid in a Belnap update logic if and only if for all non-designated truth values v there exists a corresponding update tableau with root $0 : [v] : F$ that is closed.

It turns out that, for all mentioned variants of update logics, a systematic, exhaustive and terminating tableau construction procedure can be defined. This implies the *decidability* of these logics.

7 Open ends

We want to emphasize that the framework of update models, distribution modalities and corresponding tableaux is a flexible general tool rather than just another special non-classical formalism. To this aim we conclude with an (incomplete) list of topics for future research in this context.

Different underlying many-valued logics. Update models can be defined over all kinds of truth functional logics as mechanism for "local" evaluation. As an interesting example we mention the bi-lattice based logics suggested by M.L. Ginsberg [11] for modelling default reasoning. Also dynamic *fuzzy* databases can be modelled by building on an appropriate fuzzy logic (e.g., some finite-valued Łukasiewicz logic).

Other useful distribution modalities. As explained above, every function of type $2^{\mathcal{V}} \mapsto \mathcal{V}$ induces a distribution modality. A systematic investigation of expressibility, complexitiy of corresponding rules and functional dependency between different sets of modalities is still lacking.

Modelling global update constraints. Many kinds of general integrity constraints can easily be modelled by imposing further restrictions on the update relation between states. As a simple example consider the condition—for Belnap update models—that updates can only *increase knowledge* about data. Technically this corresponds to requiring $\alpha(a) \leq_k \beta(a)$ if $\alpha U \beta$, where $\leq_k$ is the partial "knowledge order" defined by $\mathbf{u} \leq_k \mathbf{t} \leq_k \bot$ and $\mathbf{u} \leq_k \mathbf{f} \leq_k \bot$.

Other types of update operations. Obviously single-step-updates as defined by condition **(ss)** (Section 2) are only a special case.[6] One might, e.g., study multiple update relations that are indexed by the "new information" that triggers the update. This information is often represented by a boolean combination of atoms and thus naturally induces a corresponding algebra of update relations.

[6] For a thorough computational analysis of different related concepts see [5].

First-order reasoning. Both, update models and corresponding tableau, are readily generalized to the first-order level. This move, of course, vastly improves the expressibility of the corresponding logics. Their strength and limits should also be explored.

References

[1] Ofer Arieli and Arnon Avron. The logical role of the four-valued bilattice. In *13th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 118–126, Indianapolis (USA), 1998. IEEE Computer Society.

[2] Ofer Arieli and Arnon Avron. The value of the four values. *Artificial Intelligence*, 102(1):97–141, 1998.

[3] Philippe Balbiani. A modal logic for data analysis. In Wojciech Penczek and Andrzej Szalas, editors, *Proc. Mathematical Foundations of Computer Science, 21st MFCS*, volume 1115 of *Lecture Notes in Computer Science*, pages 167–179, Cracow, Poland, 1996. Springer-Verlag.

[4] Nuel D. Belnap. A useful four-valued logic. In J.M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37. D. Reidel Publishing Co., 1977.

[5] Thomas Eiter and Georg Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.

[6] Dieter Fensel, Rix Groenboom, and G.R.Renardel de Lavalette. Modal change logic (MCL): Specifying the reasoning of knowledge-based systems. *Data Knowl. Eng.*, 25(1-2):243–269, 1998.

[7] Christian G. Fermüller and Herbert Langsteiner. Tableaux for finite-valued logics with arbitrary distribution modalities. In Harrie de Swart, editor, *Proc. Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Computer Science*, pages 156–171, Oisterwijk (Netherlands), 1998. Springer-Verlag.

[8] Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics.* D. Reidel Publishing Co., Dordrecht, 1983.

[9] Melvin C. Fitting. Modality and databases. In Roy Dyckhoff, editor, *Proc. Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*, Lecture Notes in Computer Science, St. Andrews, Scotland, 2000. Springer-Verlag. to appear.

[10] Nir Friedman and Joseph Y. Halpern. Modeling belief in dynamic systems. II: Revision and update. *J. Artif. Intell. Res. (JAIR)*, 10:17–167, 1999.

[11] Matthew L. Ginsberg. Multi-valued logics. *Computational Intelligence*, 4(3), 1988.

[12] Rajeev Goreé. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.

[13] Reiner Hähnle. *Automated Deduction in Multiple-valued Logics.* Clarendon Press, Oxford, 1993.

[14] Mark Ryan and Pierre-Yves Schobbens. Counterfactuals and updates as inverse modalities. *J. of Logic Lang. Inf.*, 6(2):123–146, 1997.

[15] Gernot Salzer. Optimal axiomatizations for multiple-valued operators and quantifiers based on semilattices. In Michael McRobbie and John Slaney, editors, *Proc. 13th Conference on Automated Deduction, New Brunswick/NJ, USA*, volume 1104 of *Lecture Notes in Computer Science*, pages 688–702. Springer-Verlag, 1996.