Theoretical Computer Science 105 (1992) 141–166
Elsevier

Multimodal logic programming using equational and order-sorted logic*

Françoise Debart, Patrice Enjalbert and Madeleine Lescot
Laboratoire d'Informatique, Université de Caen, 14032 Caen Cedex, France

Abstract

Debart, F., P. Enjalbert and M. Lescot, Multimodal logic programming using equational and order-sorted logic, Theoretical Computer Science 105 (1992) 141–166.

In our previous works a method for automated theorem proving in modal logic, based on algebraic and equational techniques, was proposed. In this paper we extend the method to multimodal logic and apply it to modal logic programming. Multimodal systems under consideration have a finite number of pairs of modal operators $(\Diamond_i, \Box_i)$ of any type among KD, KT, KD4, KT4, KF, and interaction axioms of the form $\Box_i A \rightarrow \Box_j A$. We define a translation from such logical systems to specially tailored equational theories of classical order-sorted logic, preserving satisfiability, and then use SLD E-resolution for theorem proving in these theories.

Introduction

In our previous works [4, 3] we proposed a method for automated theorem proving in modal logic, based on algebraic and equational techniques. The aim of this paper is twofold. Firstly, we extend the method to multimodal logic developing [9]; secondly, we investigate its application to modal logic programming.

The multimodal systems under consideration have a finite number of pairs of modal operators $\mu_i=(\Diamond_i, \Box_i)$ ("modalities" in this paper) declared with some arbitrary "modal type" among KD, KT, KD4, KT4, KF. The standard possible-worlds semantics is straightforwardly extended: with each modality μ_i, a binary "accessibility relation" R_i between worlds is associated, with the properties corresponding to the assigned modal type, respectively: seriality, reflexivity, seriality and transitivity, reflexivity and transitivity, or functionality. Moreover, we can assume inclusion relations

Correspondence to: F. Debart, Laboratoire d'Informatique, Université de Caen, 14032 Caen Cedex, France.
* This research was partially supported by the GRECO-PRC *Programmation et outils de l'IA*.

0304-3975/92/$05.00 © 1992—Elsevier Science Publishers B.V. All rights reserved

$R_j \subset R_i$, the semantical counterpart of interaction axioms of the form $\Box_i A \to \Box_j A$. Examples are given in Section 1 to illustrate the use of such logical systems in knowledge representation.

Concerning automated theorem proving (ATP in short), two different ways open. We may design specific "direct" methods, dealing with multimodal formulas themselves. Or we may first use some translation to classical logic, and then apply (or adapt) some classical ATP technique. The method presented in this paper is in the second manner. We believe that recent experience shows that it is the right way to do; this important point is discussed in [4] and in the conclusion. It works as follows. With any multimodal system S, an order-sorted signature $\Sigma(S)$ is associated, together with a set of equations $E(S)$, and a translation T is defined in such a way that a given multimodal formula B is S-satisfiable iff its translation $T(B)$ is $E(S)$-satisfiable. Then the various methods for ATP in equational order-sorted theories can be used. We have developed a method we call Σ–E-resolution, which is a combination of E-resolution defined by Plotkin [30], already used in [4, 3], and Σ-resolution (without paramodulation) as in [31, 33], which seems especially well fitted since, as in the monomodal case, all the properties of the modal operators are coded in the unification algorithm.

If one considers Horn clauses (in the usual sense) and SLD Σ–E-resolution, using standard theoretical results, we immediately get a general framework for logic programming in various multimodal systems. For instance a temporal logic programming system which subsumes Abadi and Manna's TEMPLOG [1] is obtained, whose completeness immediately follows from our general theorems.

The paper begins with a brief introduction to multimodal logic. In Section 2 we introduce the order-sorted languages and equational theories in which multimodal logic is translated, and study the translation. Section 3 addresses the main technical difficulty, unification; it presents a unification algorithm for the considered order-sorted signatures and equations, which terminates on the fragment obtained by translation from multimodal logic. Section 4 then presents SLD Σ–E-resolution and illustrates the method with two examples. Finally, a comparison with other approaches of modal logic programming is discussed in the conclusion.

1. Multimodal logic

In this section, we define the syntax and semantics of multimodal systems and mention some of their applications in knowledge representation and processing. For further details, the reader may consult e.g. [7, 18].

1.1. Multimodal systems

Let Ξ be a first-order signature consisting of a set $\boldsymbol{G}$ of function symbols (denoted as $f, g, h \ldots$) and a set $\boldsymbol{P}$ of predicate symbols (denoted as $p, q, r \ldots$) of any arity. Each

one is declared *rigid* or *flexible*. We are also given a set of *modal operators* $\Diamond_i$ and $\Box_i$ for $i=1, \ldots, r$. Each pair $\mu_i=(\Box_i, \Diamond_i)$ will be called a *modality*. In this paper, a *modal system name* is an element of {KD, KT, KD4, KT4, KF} (the terminology comes from the axiomatics of the various systems of modal logic). A *multimodal system* is $S=(\Xi, (M_i)_{i=1,\ldots,r}, <)$ consisting of

- a signature Ξ,
- for every $i=1, \ldots, r$, a modal system name M_i – the "type" of the modality $(\Box_i, \Diamond_i)$,
- a set of declarations $\mu_i<\mu_j$ for some pairs (i, j) of distinct elements of $\{1, \ldots, r\}$.

Terms and formulas are defined in a standard way using a set V of variables (denoted as $x, y, z \ldots$), the classical connectives and quantifiers $\wedge, \vee, \neg, \forall, \exists$, and the modal unary operators $(\Box_i, \Diamond_i)_{i=1,\ldots,r}$.

1.2. Semantics

Given some multimodal system S, an *S-interpretation* $\mathscr{I}$ (or *Kripke structure*) consists of

- a set W, elements of which are called *worlds*,
- a set of binary relations on W, $\{R_i/i=1, \ldots, r\}$, the *accessibility relations*,
- a set D, the *domain* of $\mathscr{I}$ (or *discourse domain*),
- for every function symbol f of arity n, and every world w, a function $f_{\mathscr{I}}^w: D^n \to D$,
- for every predicate symbol p of arity n, and every world w, a function $p_{\mathscr{I}}^w: D^n \to \{0, 1\}$.

Hence, for every world w, we have a classical interpretation $\mathscr{I}^w$ with the same domain D, where the $f_{\mathscr{I}}^w$ and $p_{\mathscr{I}}^w$ interpret the function and predicate symbols. If f (p) is a *rigid* function (predicate) symbol, then $f_{\mathscr{I}}^w$ ($p_{\mathscr{I}}^w$) does not depend on w, and does if this symbol is *flexible*. Furthermore, we suppose that

(1) for every i, R_i has the following property according to M_i: serial $(\forall x\, \exists y\, x R_i y)$ if $M_i=$KD; reflexive if $M_i=$KT; serial and transitive if $M_i=$KD4; reflexive and transitive if $M_i=$KT4; functional $(\forall x\, \exists ! y\, x R_i y)$ if $M_i=$KF; and

(2) if $\mu_j<\mu_i$ then $R_j \subset R_i$.

A *valuation* of the variables is a function $\sigma: V \to D$. If x is a variable, d an element of D, σ_x^d denotes the valuation equal to σ except that $\sigma(x)=d$. Given some interpretation $\mathscr{I}$, some valuation σ, and some world w, the interpretation of a term t relative to $\mathscr{I}, \sigma$, and w, denoted as $\langle \mathscr{I}, \sigma, w \rangle t$ is defined classically as the value of t in $\mathscr{I}^w$ for the valuation σ. Similarly, the satisfaction of a formula relatively to $\mathscr{I}, \sigma$, and w is defined by

$$\mathscr{I}, \sigma, w \models p(t_1, \ldots, t_n) \text{ iff } p_{\mathscr{I}}^w(\langle \mathscr{I}, \sigma, w \rangle t_1, \ldots, \langle \mathscr{I}, \sigma, w \rangle t_n)=1,$$

$$\mathscr{I}, \sigma, w \models \Box_i B \text{ iff for all } w' \text{ in } W \text{ such that } w R_i w',\ \mathscr{I}, \sigma, w' \models B,$$

$$\mathscr{I}, \sigma, w \models \Diamond_i B \text{ iff there is some } w' \text{ in } W \text{ such that } w R_i w' \text{ and } \mathscr{I}, \sigma, w' \models B,$$

and the classical rules for the boolean connectives and the quantifiers.

We say that a formula B is *S-satisfiable* iff there is some S-interpretation $\mathscr{I}$ and some σ, w_0 such that $\mathscr{I}, \sigma, w_0 \models B$. The notions of validity and logical consequence are then defined in a standard way.

Finally, we say that a formula is in *negation normal form* (NNF) if the scope of every negation sign is an atomic formula. Using the logical equivalence between $\neg \Box_i \neg B$ and $\Diamond_i B$, holding for every formula B, it is easy to check that for any formula there is an equivalent one in NNF.

Remarks. (1) There exist axiomatics for these multimodal systems (see [7, 18]). Especially, the relation $<$ on modalities is axiomatised by a set of interaction axiom schemas of the general form $(Ax_{ij}) \Box_i A \rightarrow \Box_j A$ if $\mu_j < \mu_i$.

(2) All modalities we consider are (at least) serial. The reason will be explained later. Also observe that all the $\mathcal{I}^w$'s have the same domain. Such interpretations are said to be "with constant domain". A smoother condition often considered is that $\mathcal{I}^w$ be included in $\mathcal{I}^{w'}$ if w' is accessible from w. This restriction of our theory could possibly be relaxed, but this should be carefully investigated. On the other hand, we deal with rigid or flexible symbols as well.

(3) If $(\Box, \Diamond)$ is of type KF, it is easy to see that for any formula A, $\Box A$ and $\Diamond A$ are equivalent.

(4) Let $<^*$ be the pre-order generated by $<$. Clearly, if $\mu_j <^* \mu_i$ and $\mu_i <^* \mu_j$, then the two modalities are equivalent, i.e. $\Box_i A$ and $\Box_j A$ are logically equivalent, for any formula A. Similarly, since all accessibility relations are serial, any modality μ_i smaller than a KF one μ_j is equivalent to μ_j. Moreover, if μ_j is of type KT, or KT4, they are degenerated: $\Box_i A$ is equivalent to A for any A. And if μ_j is of type KD4, they are quasi-degenerated: $\Box_i (A \equiv \Box_i A)$ is true in every world.

Hence, throughout the paper *we suppose that* $<^*$ *is acyclic* (*in other terms, the graph of* $<$ *is a DAG*) *and the modalities of type KF are minimal.*

1.3. Applications – Examples

The interest in multimodal logic arises from the possible mixing of the various modal operators with various interpretations.

The first example concerns the so-called *epistemic logic*. The idea is to formalise the expression "agent i knows (or believes) that ..." by means of modal operators $\Box_i$ of a certain modal type according to the notion of knowledge or belief one has in mind: generally KT, KT4, KD4, or KT5 not to be considered in this paper (see e.g. [19] for details). A "world" w' such that $w R_i w'$ in the semantics is some "state of affairs" compatible with the knowledge of agent i in state w; we call it after Hintikka an *epistemic alternative to* w. Observe that, if $\exists ! x\ p(x)$ is true, the formula $\exists x \Box_i p(x)$ is a good formalisation of "agent i knows who has property p", while $\Box_i \exists x\ p(x)$ means only that agent i knows that *there is* such an element x. An interaction axiom (Ax_{ij}) is read "agent j knows everything agent i knows".

Another interpretation is *temporal logic*, in which we consider the set of worlds as *time instants*. A system of special interest is the *linear discrete temporal logic*, which

received much attention for parallel program verification [2]. The language possesses two modalities $\mu_1=(\Box_1, \Diamond_1)$ and $\mu_2=(\Box_2, \Diamond_2)$. The "worlds" of the Kripke structure we have in mind constitute an infinite sequence of "instants" $t_0, t_1, t_2, \ldots, t_n, \ldots$; R_1 and R_2 are respectively the "next instant" and "future" relations: $t_i R_1 t_j$ iff $j=i+1$ and $t_i R_2 t_j$ iff $i \leqslant j$. Hence, $(\Box_1, \Diamond_1)$ is of type KF and $(\Box_2, \Diamond_2)$ of type KT4, and $\mu_1 < \mu_2$. Of course, these constraints do not force the sequential structure of instants. We shall call a *standard interpretation* as the one in which this structure is indeed isomorphic to the structure of natural numbers with the successor (R_1) and the order (R_2) relations. Since R_1 is functional, $\Box_1 A$ and $\Diamond_1 A$ are equivalent; following Pnueli who introduced this system, we shall denote by "$\bigcirc$" $\Box_1$ or $\Diamond_1$ and simply write $\Diamond$ for $\Diamond_2$ and $\Box$ for $\Box_2$. An example of formula is then the following: $\Box(p \rightarrow \bigcirc q)$, which says that always (in the present and in the future) if p is true, then q will be true in the following instant.

Finally, we may consider "worlds" as different "states of affairs" obtained by performing *actions*. We obtain *dynamic logic* [20]. With each modality is associated some class of actions and $\Box_i A$ is read: after performing any action "of kind i", A will be true; conversely, $\Diamond_i A$ is read: it is possible to perform an action of kind i and then A will be true.

We give now two examples illustrating and mixing these interpretations.

Example 1.1 (The safe problem). *John must open a safe. He does not know the combination but knows that it is written on some paper which is in the desk in the room. Find a sequence of actions such that John knows it will open the safe.*

This problem was formulated in [25]: We shall first formalise the problem in some adequate multimodal logic. Later on we shall show how to solve it automatically.

The *signature* contains the following predicate symbols: comb(X, S) for "X is the combination of the safe S", written_in (Y, L) for "Y is written in location L", open(S) for "S is open"; and the constant symbols safe1, desk1 denoting the safe and the desk of the problem. All predicates are flexible since their denotation may change according to various states or alternatives; constants safe1 and desk1 are rigid: they are, so to speak, proper names.

We use the following modal operators. We write [mod] and $\langle$mod$\rangle$ for $\Box_{\text{mod}}$ and $\Diamond_{\text{mod}}$ and for each, we give its intuitive meaning and its modal type.

– [know]A	"John knows that A"	KT4 (or KT)
– [read]A	"After John performed some reading, A must be true"	KD
– [dial]A	"After John performed some dialing, A must be true"	KD
– [actions]A	"After any action, A is true"	KT4
– [s]A	"In every state A is true"	KT4

The intended meaning is that an "action" is any sequence of reading or dialing. $[s]$ is a "super modality" used to express properties true after any action and in any epistemic alternative.

Hence, we have the following interaction axioms:

$$[\text{actions}]A \rightarrow [\text{read}]A, \qquad [\text{actions}]A \rightarrow [\text{dial}]A,$$

$$[s]A \rightarrow [\text{actions}]A, \qquad [s]A \rightarrow [\text{know}]A,$$

and the corresponding order: $\mu_{\text{read}} < \mu_{\text{actions}}$, etc.

But we want more. Let R^* denote the reflexive transitive closure of the relation R. In the intended interpretation: $R_{\text{actions}} = (R_{\text{read}} \cup R_{\text{dial}})^*$ and $R_s = (R_{\text{actions}} \cup R_{\text{know}})^*$. We shall say that such an interpretation is a *standard* one.

The problem is coded in the following set of modal formulas (the variables are in capital letters).

(1) $[s]\forall S \forall L \forall X\ ((\text{comb}(X, S) \wedge \text{written\_in}(X, L))$

$\rightarrow \langle \text{read} \rangle [\text{know}] \text{comb}(X, S))$,

(2) $[s]\forall S((\exists X\ [\text{know}]\ \text{comb}(X, S)) \rightarrow \langle \text{dial} \rangle \text{open}(S))$,

(3) $[\text{know}]\exists X(\text{comb}(X, \text{safe1}) \wedge \text{written\_in}(X, \text{desk1}))$,

from which we want to infer

(4) $[\text{know}] \langle \text{actions} \rangle \text{open}(\text{safe1})$

These formulas may be read as follows:

(1) In every state, for every (safe) S, (location) L, and (number) X, if X is the combination for S and is written in L, then there is some reading action after which John knows that X is the combination of S.

(2) In every state, for every (safe) S, if John knows what is the combination of S, then there is some dialing operation John can perform and after which S will be open. (Observe the standard formalisation of "knowing what ...")

(3) John knows (in the actual present state) that the combination of safe1 is written in desk1.

(4) John knows that there is some complex action after which the safe will be open.

Note that the "goal" (4) is not quite satisfactory since we have no way to name precisely actions in the language of multimodal logic. We shall see how the translation solves this problem.

Example 1.2 (*TEMPLOG*). We now consider linear discrete temporal logic. Abadi and Manna [1] have defined a subset of the set of temporal formulas and a so-called

temporal SLD resolution for this class of formulas. The resulting system is called TEMPLOG. Here is a typical TEMPLOG program:

$$\text{fib}(0)$$

$$\bigcirc\text{fib}(1)$$

$$\Box(\bigcirc\bigcirc\,\text{fib}(X)\leftarrow\text{fib}(Y),\bigcirc\,\text{fib}(Z),X\text{ is }Y+Z)$$

which defines the flexible predicate fib in such a way that fib(X) is true at the nth next instant if X is the nth element of the Fibonacci sequence. The constants $0, 1, 2, \ldots$ and $+, =$ are rigid, since arithmetics and the identity relation does not change in time.

A query for this program is

$$\leftarrow\text{fib}(R)$$

and the answer in TEMPLOG will be the sequence of values of R at the successive instants: $0, 1, 1, 2, 3, 5, \ldots$ In other words: fib(0), $\bigcirc$fib(1), ..., $\bigcirc\bigcirc\bigcirc\bigcirc\bigcirc$fib(5), ... are consequences of the clauses of the program in all standard interpretations.

2. Path theories and the translation from multimodal logic

2.1. Introduction: frames and algebraic frames

Understanding the proposed translation from modal to classical logic needs a reconsideration of Kripke semantics we shall present now before the formal definitions. For the sake of simplicity, let us consider standard modal logic, with only one modality.

Interpretations for modal logic include a relational structure $\langle W, R\rangle$ consisting of a set of "worlds" and an "accessibility relation" in order to interpret the modal operators. Moreover, various constraints on this structure, called a *frame*, define the various modal system types. The first key idea for our method is to replace this structure by an algebraic one $\langle W, A, !\rangle$, where

- W is as usual a set of "worlds",
- A is a set, elements of which are called *operators*,
- ! is a function $W\times A\rightarrow W$.

Let us call it an *algebraic frame*. Clearly, given any algebraic frame $\langle W, A, !\rangle$ one can define a frame $\langle W, R\rangle$ by (we use infix notation for !):

(*) $\quad w\,R\,w'$ iff there exists an operator $a\in A$ such that $w!a=w'$.

Conversely, it is not difficult to see (proved in Section 2.3 in the general case) that given some frame $\langle W, R\rangle$ one can define an algebraic frame $\langle W, A, !\rangle$ such that (*) holds. Informally, we can represent things as follows. Consider $\langle W, R\rangle$ as a graph. Let A be a set of labels such that for every vertex w, and every a in A, there is one and only

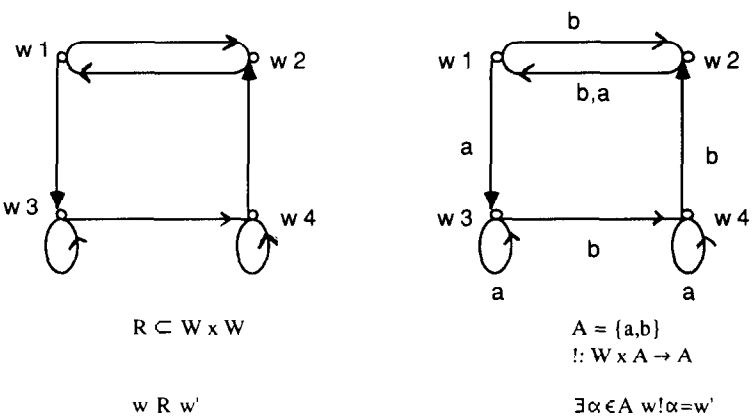


Fig. 1.

one edge with source w labelled by a. We define $w!a$ as the only w' such that (w, w') is labelled by a. The correspondence is illustrated in Fig. 1. Observe that since ! is a function (defined everywhere), R is serial. This is the reason why KD is for us the minimal modal system.

But there is one problem left: What is the counterpart in an algebraic frame of the properties of the associated "relational" frame? The nice fact and the second key idea is that the properties of reflexivity, transitivity and functionality, are mirrored in *equational constraints on the set of operators A*. Reflexivity, is ensured by assuming that there is a unit element 1, i.e. such that $w!1 = w$ for all w; we have transitivity if a composition operation $*$ is defined on A with $w!(a * a') = (w!a)!a'$ (again we use infix notation); the relation R is functional if A is reduced to a single operator (see Fig. 2).

The reader can easily imagine how to extend these ideas to the multimodal case: one set of operators A_i will be associated with every modality μ_i, instead of the accessibility relation R_i. And each of them will have to satisfy the set of equations corresponding to the modal type of μ_i. Moreover, if $\mu_j < \mu_i$, $R_j \subset R_i$ and, therefore, $A_j \subset A_i$ also.

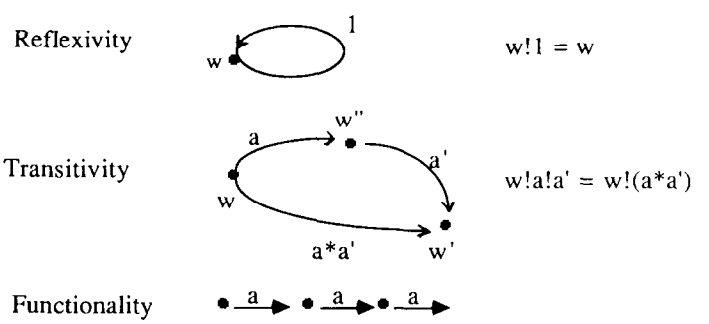


Fig. 2.

In the next section we introduce the systems of first-order logic with ordered sorts, and the equational theories (called "path theories" for reasons presented below) adequate to these semantical structures. The translation from modal to path theories is defined and studied in Section 2.3.

2.2. Path theories

2.2.1. Order-sorted logic

Syntax. We adopt notations from [23]. A signature for logic with ordered sorts is: $\Sigma = ((\mathbb{S}, \leqslant), \boldsymbol{G}, \boldsymbol{P}, \mathrm{Dec})$, where $(\mathbb{S}, \leqslant)$ is a partially ordered set of *sort symbols*, $\boldsymbol{G}$ and $\boldsymbol{P}$ are, respectively, sets of function and predicate symbols, and Dec a set of *declarations*:

$$f: \mathbb{s}_1 \times \cdots \times \mathbb{s}_n \to \mathbb{s}_{n+1} \quad \text{if } f \text{ is in } \boldsymbol{G}, \qquad p: \mathbb{s}_1 \times \cdots \times \mathbb{s}_n \to \mathbb{Boole} \quad \text{if } p \text{ is in } \boldsymbol{P}$$

where the $\mathbb{s}_i$'s are sort symbols and $\mathbb{Boole}$ a distinct symbol. $\mathbb{s}_{n+1}$ is the *range sort* of f. Note that a function or predicate symbol may have several declarations. We only impose that the *arity* n is the same in all declarations. Given some set $\{v_i: \mathbb{s}_i\}_{i=1,2,\ldots}$ of sorted variables one can build well-formed terms and formulas in an obvious way, and define a relation "the term t has sort $\mathbb{s}$" – $t: \mathbb{s}$ in symbolic notation – in such a way that if $t: \mathbb{s}$ and $\mathbb{s} \leqslant \mathbb{s}'$, then $t: \mathbb{s}'$. We assume that $\leqslant$ is the partial order generated by some relation $<$ given by some set of *order declarations*: $\mathbb{s}_i < \mathbb{s}_j$. Terms, formulas, clauses, etc., are defined as usual.

Semantics. Different interpretations may be defined for such languages [16, 17, 23, 32]. We shall use the following one, adequate for our purpose. An *interpretation* for $\Sigma = ((\mathbb{S}, \leqslant), \boldsymbol{G}, \boldsymbol{P}, \mathrm{Dec})$ as above is some triple:

$$J = \langle (D_{\mathbb{s}})_{\mathbb{s} \in \mathbb{S}}, (f_J)_{f \in G}, (q_J)_{q \in P} \rangle,$$

where

- each $D_{\mathbb{s}}$ is a nonempty set, the carriers for sorts $\mathbb{s}$. Moreover, $D_{\mathbb{s}} \subset D_{\mathbb{t}}$ if $\mathbb{s} < \mathbb{t}$;
- each f_J is a function such that for every declaration $f: \mathbb{s}_1 \times \cdots \times \mathbb{s}_n \to \mathbb{s}_{n+1}$, for every $(a_1, \ldots, a_n) \in D_{\mathbb{s}_1} \times \cdots \times D_{\mathbb{s}_n}$, $f_J(a_1, \ldots, a_n)$ is defined and belongs to $D_{\mathbb{s}_{n+1}}$; f_J is undefined otherwise;
- each q_J is a predicate such that for every declaration $q: \mathbb{s}_1 \times \cdots \times \mathbb{s}_n \to \mathbb{Boole}$, for every $(a_1, \ldots, a_n) \in D_{\mathbb{s}_1} \times \cdots \times D_{\mathbb{s}_n}$, $p_J(a_1, \ldots, a_n)$ is defined, and is undefined otherwise.

If J is some interpretation, u a term and A some formula, the value of u in J for σ, denoted as $\langle J, \sigma \rangle u$, and the relation "$A$ is true in J for σ", denoted as $J, \sigma \models A$, are defined in the usual way. The usual definitions of validity, satisfiability, etc., follow.

2.2.2. Path theories

Language. Consider some multimodal system $S = ((G, P), (M_i)_{i=1,\ldots,r}, <)$ as in Section 1. We define a signature $\Sigma(S) = ((\mathbb{S}, \leqslant), \boldsymbol{G}', \boldsymbol{P}', \mathbf{Dec})$ as follows:

- $\mathbb{S}=\{\mathbb{W}, \mathbb{A}_1, \ldots, \mathbb{A}_r, \mathbb{D}\}$, where $\mathbb{D}$ is the sort for elements of the discourse domain, $\mathbb{W}$ for worlds, and the $\mathbb{A}_i$'s for operators on worlds.
- $\mathbb{A}_j < \mathbb{A}_i$ iff $\mu_j < \mu_i$.
- $\boldsymbol{P}' = \boldsymbol{P}$.
- $\boldsymbol{G}' = \boldsymbol{G} \cup \boldsymbol{\theta}$, where $\boldsymbol{\theta} = \{\varepsilon, !, *, 1\} \cup \{a_i / M_i = \mathrm{KF}\}$, where the a_i's are fresh symbols.
- if f has arity n and p arity m in S,

$$f: \mathbb{D}^n \to \mathbb{D} \quad \text{and} \quad p: \mathbb{D}^m \to \mathbb{B}\mathbb{o}\mathbb{o}\mathbb{l} \qquad \text{(if they are rigid)}$$

$$f: \mathbb{W} \times \mathbb{D}^n \to \mathbb{D} \quad \text{and} \quad p: \mathbb{W} \times \mathbb{D}^m \to \mathbb{B}\mathbb{o}\mathbb{o}\mathbb{l} \qquad \text{(if they are flexible)}$$

are in **Dec** and also

$$\varepsilon: \mathbb{W}, \qquad !: \mathbb{W} \times \mathbb{A}_i \to \mathbb{W} \quad \text{for every } i = 1, \ldots, r,$$

$$1: \mathbb{A}_i \quad \text{for every } i \text{ such that } M_i \text{ is KT or KT4},$$

$$*: \mathbb{A}_i \times \mathbb{A}_i \to \mathbb{A}_i \text{ for every } i \text{ such that } M_i \text{ is KD4 or KT4},$$

$$a_i: \mathbb{A}_i \text{ for every } i \text{ such that } M_i \text{ is KF}.$$

We use infix notation for ! and $*$. We decide that $*$ and ! associate to the left, so that $a!b!c = (a!b)!c$ and $a*b*c = (a*b)*c$. The set of variables is split into $V' = \{x: \mathbb{D}, y: \mathbb{D}, z: \mathbb{D}, \ldots\}$ and $\Omega = \{\alpha: \mathbb{A}_i, \beta: \mathbb{A}_j, \ldots\}$. The language built on $\Sigma(S)$ is the language of the path theory associated with S. Formulas in this language will be called *path formulas*.

Note that path formulas do not contain any variable of sort $\mathbb{W}$, so that the only terms with this sort have the general form $\varepsilon!a^1!\cdots!a^k$, for some (possibly empty) sequence of terms $a^j: \mathbb{A}_{i_j}$. The reader can fruitfully interpret such an expression as denoting some world which can be reached from an "initial" world ε through some "path" whose "transitions" from one world to another are labelled by the a^j's.

Hence, an interpretation for the language of a path theory can be written as

$$I = \langle W, A_1, \ldots, A_r, D, G'_I, P'_I \rangle,$$

where W, A_i, D are the carriers for sorts $\mathbb{W}, \mathbb{A}_i, \mathbb{D}$, respectively, and G'_I, P'_I the interpretation of symbols in G' and P'. Moreover. if $\mathbb{A}_i < \mathbb{A}_j$ is in Σ, then $A_i \subset A_j$.

Equational theories. Finally, with every modal system name M_i we associate a set of equations $E(M_i)$:

$$E(\mathrm{KD}) = \emptyset,$$

$$E(\mathrm{KT}) = \{w!1 = w\},$$

$$E(\mathrm{KD4}) = \{w!(\alpha * \alpha') = (w!\alpha)!\alpha',\ (\alpha * \alpha') * \alpha'' = \alpha * (\alpha' * \alpha'')\},$$

$$E(\mathrm{KT4}) = E(\mathrm{KD4}) \cup E(\mathrm{KT}) \cup \{\alpha * 1 = \alpha,\ 1 * \alpha = \alpha\},$$

$$E(\mathrm{KF}) = \{\alpha = a_i\},$$

where the α's in $E(M_i)$ have sort $\mathbb{A}_i$ and w is a variable of sort $\mathbb{W}$ used only in these equations. With any multimodal system S we associate the set of equations $E(S)=\bigcup_{i=1,\ldots,r}E(M_i)$ the *path theory* for S.

A $\Sigma(S)$–$E(S)$-*interpretation* is an interpretation for $\Sigma(S)$ which satisfies $E(S)$. A set of closed formulas $\mathscr{F}$ on $\Sigma(S)$ is $E(S)$-*satisfiable* if it is satisfied in some $\Sigma(S)$–$E(S)$-interpretation. A formula A is an $E(S)$-*consequence* of $\mathscr{F}$ ($\mathscr{F}\models_{E(S)}A$) if the universal closure of A is true in every $\Sigma(S)$–$E(S)$-model of $\mathscr{F}$. (We shall generally write simply $E(S)$-interpretation, $E(S)$-satisfiable, etc.)

If we orient the equations of $E(S)$ from left to right, we obtain a rewriting system $R(S)$. By careful examination of the possible critical pairs one can check the following proposition.

Proposition 2.1. *For any multimodal system S, R(S) is canonical.*

Hence, every term t has a unique normal form, denoted as $t\downarrow$, and we have an easy test for equality modulo the set of equations $E(S)$. This remark will be useful later.

2.3. Translation from modal logic to path theories

Let T be the function from the set of multimodal formulas to the corresponding set of path formulas defined by

$$T(F)=t(\varepsilon,F),$$

where t is an intermediate function which, given a $\mathbb{W}$-sorted term π and a modal formula or term, specifies a path formula or term. t is recursively defined as follows:

$t(\pi,x)=x$ if x is a $\mathbb{D}$-sorted variable,

$t(\pi,f(\tau_1,\ldots,\tau_n))=f(t(\pi,\tau_1),\ldots,t(\pi,\tau_n))$ if f is rigid,

$t(\pi,f(\tau_1,\ldots,\tau_n))=f(\pi,t(\pi,\tau_1),\ldots,t(\pi,\tau_n))$ if f is flexible,

$t(\pi,p(\tau_1,\ldots,\tau_n))=p(t(\pi,\tau_1),\ldots,t(\pi,\tau_n))$ if p is rigid,

$t(\pi,p(\tau_1,\ldots,\tau_n))=p(\pi,t(\pi,\tau_1),\ldots,t(\pi,\tau_n))$ if p is flexible,

$t(\pi,\neg F)=\neg t(\pi,F)$,

$t(\pi,F_1\vee F_2)=t(\pi,F_1)\vee t(\pi,F_2)$,

$t(\pi,F_1\wedge F_2)=t(\pi,F_1)\wedge t(\pi,F_2)$,

$t(\pi,\forall xF)=\forall x:\mathbb{D}\,t(\pi,F)$,

$t(\pi,\exists xF)=\exists x:\mathbb{D}\,t(\pi,F)$,

$t(\pi,\Box_i F)=\forall\alpha:\mathbb{A}_i\,t(\pi!\alpha,F)$ where α is not in $\mathrm{Var}(\pi)$,

$t(\pi,\Diamond_i F)=\exists\alpha:\mathbb{A}_i\,t(\pi!\alpha,F)$, where α is not in $\mathrm{Var}(\pi)$.

Example. Let $G = \Box_1 \Box_2 \exists x \Diamond_3 p(f(x))$, with p flexible and f rigid. Then

$$T(G) = \forall \alpha : \mathbb{A}_1 \forall \beta : \mathbb{A}_2 \exists x : \mathbb{D} \exists \gamma : \mathbb{A}_3 p(\varepsilon ! \alpha ! \beta ! \gamma, f(x)).$$

Proposition 2.2. *Let S be a multimodal system and B a modal formula*

B is S-satisfiable iff $T(B)$ is $E(S)$-satisfiable.

Proof

Lemma 2.3. *Let S be some multimodal system and $\mathcal{I}$ some interpretation for S, W denoting the set of worlds and R_i for $i = 1, \ldots, r$ the accessibility relations. There exist r families A_i of mappings from W to W, such that*
- *$R_i = \bigcup_{a \in A_i} a$, and*
- *if $R_i \subset R_j$, then $A_i \subset A_j$.*

Moreover, if $M_i = \mathrm{KT}$, then $\mathrm{Id} \in A_i$; if $M_i = \mathrm{KD4}$, then A_i is a sub semigroup of the semigroup of mappings from W to W; if $M_i = \mathrm{KT4}$, then A_i is a submonoid of the monoid of mappings from W to W; if $M_i = \mathrm{KF}$, *then A_i is reduced to one singleton.*

Proof of Lemma 2.3. We have assumed that the graph of the relation $<$ on modalities is a DAG. We build the A_i's step by step, beginning with the minimal μ_i's. For every $i = 1, \ldots, r$ let $B_i = \{ j / \mu_j < \mu_i \}$ and $\mathrm{BR}_i = \bigcup_{j \in B_i} R_j$. Suppose that k is such that A_j has been built for each j in B_k; it is easy to check that at each step of the construction there exists at least one such k. Let $\Delta = R_k \setminus \mathrm{BR}_k$ and for every w in W, $\Delta_w = \{ w' / w \Delta w' \}$. Let H be the least upper bound of the cardinals of all the Δ_w's, and, for every w, f_w some surjective mapping from H to Δ_w. For every $h \leqslant H$ we define a mapping $a_h : W \to W$ by $a_h(w) = f_w(h)$. Clearly, $\Delta_w = \bigcup_{h \in H} a_h$. Let $\Omega_k = (\bigcup_{j \in B_k} A_j) \cup \{ a_h / h \in H \}$. We set
- $A_k = \Omega_k$ if $M_k = \mathrm{KD}$ or KF,
- $A_k = \Omega_k \cup \{\mathrm{Id}\}$ if $M_k = \mathrm{KT}$,
- A_k is the semigroup generated by Ω_k in the monoid of mappings from W to W if $M_k = \mathrm{KD4}$, and the monoid generated by Ω_k if $M_k = \mathrm{KT4}$.

One can check easily that the requirements of the lemma are fulfilled. Observe in particular that if $M_k = \mathrm{KF}$, then B_k is empty and $H = 1$, so that A_k is reduced to one singleton. □

Proof of Proposition 2.2 (*Continued*). Let S be some multimodal system and $I = \langle W, A_1, \ldots, A_r, D, G'_I, P'_I \rangle$ some $E(S)$-interpretation. We build an S-interpretation $[I]$ as follows. The set of worlds and the discourse domain are, respectively, W and D. The accessibility relations are defined by

$$w R_i w' \text{ iff there exist some } a \text{ in } A_i \text{ such that } w' = w !_I a$$

For every w in W and every n-ary predicate symbol p of $\boldsymbol{P}$,
- $p^w_{[I]}(x_1, \ldots, x_n) = p_I(x_1, \ldots, x_n)$ if p is rigid,
- $p^w_{[I]}(x_1, \ldots, x_n) = p_I(w, x_1, \ldots, x_n)$ if p is flexible,

and we give a similar definition for $G_{[I]}$. One can check easily that $[I]$ is an S-interpretation. If σ is some valuation of the variables in $V \cup \Omega$, let σ_V be its restriction to V.

Lemma 2.4. *For any term u and every formula B of a multimodal system S, every $E(S)$-interpretation I, every term π of sort $\mathbb{W}$, and every valuation σ,*

(i) $\langle I, \sigma \rangle t(\pi, u) = \langle [I], \sigma_V, \langle I, \sigma \rangle \pi \rangle u$,

(ii) $I, \sigma \models t(\pi, B)$ iff $[I], \sigma_V, \langle I, \sigma \rangle \pi \models B$.

Proof of Lemma 2.4. The proof is easy, by induction on the structure of terms and formulas.

Lemma 2.5. *Let S be some multimodal system, $\mathcal{I}$ an S-interpretation, w_0 some world in $\mathcal{I}$. There exists some $E(S)$-interpretation I such that $[I] = \mathcal{I}$ and $\varepsilon_I = w_0$.*

Proof of Lemma 2.5. $I = \langle W, A_1 \ldots, A_r, D, G'_I, P'_I \rangle$ is defined as follows. W and D are the sets of worlds and discourse domains of $\mathcal{I}$, respectively. We build the A_i's as in Lemma 2.3. We set

- $\varepsilon_I = w_0$,
- $w!_I a = a(w)$ for all a in $\bigcup_i A_i$,
- $a *_I a' = a'$ or a for all a, a' in some A_i such that $M_i = \mathrm{KD4}$ or KT4,
- $1_I = \mathrm{Id}$, so that $1_I \in A_i$ if $M_i = \mathrm{KT}$ or KT4,
- $a_{iI} =$ the unique element of A_i if $M_i = \mathrm{KF}$.

The rest of G'_I and P'_I are defined in the obvious way. □

Proof of Proposition 2.2 (*Conclusion*). Suppose B is satisfiable, and let $\mathcal{I}, \sigma, w_0 \models B$. Let I be as in Lemma 2.5. By Lemma 2.4 $I, \sigma' \models T(B)$ for any σ' such that $\sigma'_V = \sigma$. Conversely, if $I, \sigma \models T(B)$, then $[I], \sigma_V, \langle I, \sigma \rangle \varepsilon \models B$ and B is satisfiable. □

2.4. The closure property

If R is some relation, let $R^=$, R^+, R^* denote, respectively, the reflexive, transitive and reflexive–transitive closures of R. Let $\mathcal{I}$ be some S-interpretation for some multimodal system S as in Section 1, and for every $k = 1, \ldots, r$, B_k and BR_k defined as previously.

Definition. We say that $\mathcal{I}$ has the *closure property* w.r.t. some modality μ_k if

- $R_k = \mathrm{BR}_k$ if $M_k = \mathrm{KD}$,
- $R_k = (\mathrm{BR}_k)^=$ if $M_k = \mathrm{KT}$,
- $R_k = (\mathrm{BR}_k)^+$ if $M_k = \mathrm{KD4}$,
- $R_k = (\mathrm{BR}_k)^*$ if $M_k = \mathrm{KT4}$.

The "standard models" considered in Examples 1.1 and 1.2 are precisely those having the closure property. No complete calculus w.r.t. models with the closure property can be produced for the whole language of first-order multimodal logic in the presence of "transitive" modalities [2]. But there are useful fragments for which our method is complete. In particular, the fragment corresponding to TEMPLOG program (see Section 4). Note also that the propositional version of our multimodal system, even with closure properties, is decidable since it can be embedded in propositional dynamic logic [20].

Definition. Let I be some $E(S)$-interpretation and $\mathrm{BA}_k = \bigcup_{j \in B_k} A_j$. We say that I has the *closure property* w.r.t. some sort $\mathbb{A}_k$ if
- $A_k = \mathrm{BA}_k$ if $M_k = \mathrm{KD}$,
- $A_k = \{1_I\} \cup \mathrm{BA}_k$ if $M_k = \mathrm{KT}$,
- A_k is the semigroup generated by BA_k for $*_I$ if $M_k = \mathrm{KD4}$,
- A_k is the monoid generated by BA_k for $*_I$ with 1_I as neutral element if $M_k = \mathrm{KD4}$.

A careful examination of the proof of Proposition 2.2 shows the following.

Proposition 2.6. *Let S be a multimodal system and B a modal formula. B admits a model with the closure property w.r.t. μ_k iff $T(B)$ is $E(S)$-satisfiable in some interpretation with the closure property w.r.t. $\mathbb{A}_k$.*

2.5. *"Strong" Skolemisation and the unique prefix property (UPP)*

Skolem form of formulas can be defined as usual. But there is another, nonstandard, notion which provides simpler formulas and is quite natural for the class of formulas obtained by translation from modal logic. Moreover, as we shall see in the next section, this form is needed in order to ensure the termination of our unification algorithm. We call it the "strong" Skolem form. For brevity, we present here the combination of the translation itself and the procedure of strong skolemisation.

Let T' be the function from the set of multimodal formulas in NNF to the corresponding set of path formulas defined by

$$T'(B) = t'(\varepsilon, \emptyset, B),$$

where, if π is a term of sort $\mathbb{W}$ and X is a set of $\mathbb{D}$-sorted variables, $t'(\pi, X, B)$ is recursively defined as follows (t is the function defined in Section 2.3):

$$t'(\pi, X, B) = t(\pi, B) \quad \text{if } B \text{ is a literal,}$$

$$t'(\pi, X, B_1 \Delta B_2) = t'(\pi, X, B_1) \Delta t'(\pi, X, B_2) \quad (\Delta \in \{\wedge, \vee\}),$$

$$t'(\pi, X, \forall x\, B) = \forall x : \mathbb{D}\, t'(\pi, X \cup \{x\}, B)$$

$$t'(\pi, X, \exists x\, B) = t'(\pi, X \cup \{x\}, B)[f(\pi, X)/x],$$

where f: $\mathbb{W} \times \mathbb{D}^n \to \mathbb{D}$ is a fresh function symbol, and n is the cardinal of X

$$t'(\pi, X, \Box_i B) = \forall \alpha : \mathbb{A}_i t'(\pi ! \alpha, X, B),$$

where α is a fresh variable

$$t'(\pi, X, \Diamond_i B) = t'(\pi ! \varphi(\pi, X), X, B),$$

where φ: $\mathbb{W} \times \mathbb{D}^n \to \mathbb{A}_i$ is a fresh function symbol and n is the cardinal of X.

Example. Let $G = \Box_1 \Box_2 \exists x \Diamond_3 p(f(x))$, with p flexible and f rigid. Then

$$T'(G) = \forall \alpha : \mathbb{A}_1 \forall \beta : \mathbb{A}_2 p(\varepsilon ! \alpha ! \beta ! \varphi(\varepsilon ! \alpha ! \beta, g(\varepsilon ! \alpha ! \beta)), f(g(\varepsilon ! \alpha ! \beta))).$$

Proposition 2.7. *Let S be some multimodal system. A formula B is S-satisfiable iff $T'(B)$ is $E(S)$-satisfiable. Moreover, B admits a model with the closure property w.r.t. μ_k iff $T'(B)$ is $E(S)$-satisfiable in some interpretation with the closure property w.r.t. $\mathbb{A}_k$.*

The formulas obtained by strong skolemisation of translated modal formulas, i.e. by the function T', possess the following unique prefix property (UPP in short): UPP, or rather a similar property formulated in his own formalism, is due to Ohlbach [26].

Definition. A set of terms or atoms S has the *unique prefix property* iff for every variable α in Ω having some occurrence in S, the terms $\pi ! \alpha$ in which it occurs are such that π is independent of the particular occurrences of α.

Example. The formula $T(G)$ in the previous example has the UPP. But $p(\varepsilon ! \alpha ! \beta ! \varphi(\varepsilon ! a ! \beta, g(\varepsilon ! \alpha ! \beta)), f(g(\varepsilon ! \alpha ! \beta)))$ has not since β occurs in two different terms $\varepsilon ! \alpha ! \beta$ and $\varepsilon ! a ! \beta$.

Proposition 2.8. *For any multimodal formula B, $T'(B)$ has the UPP.*

We shall not prove these propositions. The proof requires some long prerequisites (in fact, the correct definition of UPP itself is more technical), and is a straightforward extension of the corresponding proof in the monomodal case [4] (see [10] for details). In order to justify the second part of Proposition 2.7, we can just note that, as in the standard case, a path formula F has a model iff its strong Skolem form has one *with the same domain.*

3. Unification in path theories

3.1. Σ-substitutions and Σ–E-unifiers

Let Σ be some signature with ordered sorts. A Σ-substitution (or substitution, in short) is a mapping from a finite set D_σ of sorted variables to the set of terms (for which

we use postfix notation) such that for every $v_i : s_i$ in D_σ, $v_i\sigma$ has sort s_i. A substitution is denoted by its graph $\{v_i/t_i\}_{i:1,\dots,k}$. The empty graph $\emptyset$ denotes the identical substitution. Composition of substitutions is defined in the usual way; again we use postfix notation.

If E is some set of equations, we say that two terms t and t' are E-equal ($t =_E t'$) if $t = t'$ is a logical consequence of the theory E. A Σ–E-unifier of two terms t and t' is a Σ-substitution σ such that $t\sigma =_E t'\sigma$. Given some set of variables X, we say that two substitutions τ and σ are equal modulo E and X ($\tau =_{E,X} \sigma$) iff $v\tau =_E v\sigma$ for every variable in X. Finally, we can define order relations on substitutions by $\sigma \leqslant_{E,X} \tau$ iff there is some λ such that $\tau =_{E,X} \sigma\lambda$. Also recall that we have an easy test for $E(S)$-equality by comparing the normal form of the operands.

A *Σ–E-complete set of unifiers* (Σ–E-CSU in short) of two terms t and t' is a set U of Σ–E-unifier such that for every Σ–E-unifier τ there is some σ in U with $\sigma \leqslant_{E,X} \tau$, where $X = \mathrm{Var}(t) \cup \mathrm{Var}(t')$.

3.2. *A unification algorithm for path theories*

Let us now consider the problem of unification in path theories. An important and well-known fact (see, for instance, [22, 23]) is that Σ–E-CSUs are not in general reduced to one singleton, and may not even be finite. Indeed, if E is E(KD4), we have a situation similar to the so-called "associative unification" and CSUs are, in general, infinite. Try, for instance, to unify $\varepsilon!\alpha!c$ and $\varepsilon!c!\alpha$, where α is a variable and c a constant with the same sort $\mathbb{A}_i$ such that the associated modality μ_i is of type KD4. It is readily seen that $\{\alpha/c, \alpha/c * c, \dots, \alpha/c * c * \cdots * c, \dots\}$ is an infinite minimal CSU.

What can we do? We may use a general algorithm, as proposed in [22] to enumerate, possibly infinite, CSUs. But there is a better way on. By Proposition 2.8, we know that formulas obtained by the translation T' belong to the fragment of *UPP formulas* and we shall see that UPP guarantees the existence of finite CSUs. More precisely, it ensures termination of the algorithm presented below. (For instance, UPP clearly rules out the counterexample $\{\varepsilon!\alpha!c, \varepsilon!c!\alpha\}$.)

The algorithm is an extension of the one presented in [4] for only one modality, combined with ideas from Walther's algorithm for order-sorted unification [33]. We shall first introduce some notations and illustrate the main ideas by an example.

3.2.1. *Notations*

- If t and t' are two $\mathbb{W}$-sorted terms, we set

  $\#\, t \leqslant t'$ if t is a prefix of t' ($t' = t!a_1!\cdots!a_k$),

  $\#\, t$ and t' are *comparable* if $t \leqslant t'$ or $t' \leqslant t$.

- If σ is a substitution, $\mathscr{T}$ a term or set of terms, we write $\sigma\downarrow\mathscr{T}$ for $(\sigma\mathscr{T})\downarrow$.
- Consider some path signature associated with some multimodal system. For any sort $s = \mathbb{A}_i$, we say that s is *reflexive* (*transitive*) if 1 is a term of sort s,

i.e. the corresponding accessibility relation R_i is reflexive ($*$ has a declaration $\mathcal{A}_i \times \mathcal{A}_i \to \mathcal{A}_i$ and R_i is transitive). We denote this by REF($\mathbb{s}$) or TRANS($\mathbb{s}$) accordingly. Moreover, we define the following sets:

- $\mathrm{lb}(\mathbb{t}, \mathbb{t}') = \{\mathbb{s} \in \mathbb{S} / \mathbb{s} \leqslant \mathbb{t}, \mathbb{s} \leqslant \mathbb{t}', \text{ not } \exists \mathbb{s}'(\mathbb{s} < \mathbb{s}', \mathbb{s}' \leqslant \mathbb{t}, \mathbb{s}' \leqslant \mathbb{t}')\}$ – lower bound of $\mathbb{t}, \mathbb{t}'$;
- $\mathrm{mtrans}(\mathbb{t}) =$ the set of maximal elements in $\{\mathbb{s} \in \mathbb{S} / \mathbb{s} \leqslant \mathbb{t}$ and TRANS($\mathbb{s}$)$\}$ – maximal "transitive" sorts smaller than $\mathbb{t}$.

Example. Suppose first that we want to unify two terms of sort $\mathbb{W}$: $t_1 = \pi_1 ! \alpha_1$ and $t_2 = \pi_2 ! \alpha_2$, where $\alpha_1 : \mathcal{A}_1$ and $\alpha_2 : \mathcal{A}_2$ are variables. There are several possibilities:

(1) For every sort $\mathcal{A}_3$ such that $\mathcal{A}_3 \leqslant \mathcal{A}_1$ and $\mathcal{A}_3 \leqslant \mathcal{A}_2$ and every unifier σ of π_1 and π_2, $\sigma\{\alpha_1/\beta, \alpha_2/\beta\}$ (where $\beta : \mathcal{A}_3$ is a fresh variable) unifies t_1 and t_2: standard case, using "weakening" of α_1 and α_2 if necessary.

(2) If $\mathcal{A}_1$ is reflexive, for every unifier σ of π_1 and t_2, $\sigma\{\alpha_1/1\}$ unifies t_1 and t_2: symmetrical situation if $\mathcal{A}_2$ is reflexive.

(3) For every sort $\mathcal{A}_3$ and $\mathcal{A}_4$ such that $\mathcal{A}_3 \leqslant \mathcal{A}_1$, $\mathcal{A}_4 \leqslant \mathcal{A}_3$, $\mathcal{A}_4 \leqslant \mathcal{A}_2$, and $\mathcal{A}_3$ is transitive, let $\beta : \mathcal{A}_3$ and $\beta' : \mathcal{A}_4$ be fresh variables. For every unifier σ of $\pi_1 ! \beta$ and π_2, $\sigma\{\alpha_1/\beta * \beta', \alpha_2/\beta'\}$ unifies t_1 and t_2: symmetrical situation exchanging t_1 and t_2.

Now if, for instance, α_2 were not a variable, we have a similar situation except that, of course, there can be no weakening of α_2. Observe also that if two terms t_1 and t_2 are comparable, unification fails unless, for instance, $t_2 = t_1 ! \alpha_1 ! \cdots ! \alpha_k$, where, for every $i = 1, \ldots, k$, $\alpha_i : \mathcal{A}_i$ is a variable with $\mathcal{A}_i$ reflexive, in which case we have the obvious mgu: $\{\alpha_1/1, \ldots, \alpha_k/1\}$.

Finally, suppose that $t_1 = \varepsilon ! \alpha_1$ and $t_2 = \varepsilon ! \alpha_2$, $\mathcal{A}_1$ and $\mathcal{A}_2$ are both transitive, and there is some *nontransitive* $\mathcal{A}_3$ such that $\mathcal{A}_3 \leqslant \mathcal{A}_1$ and $\mathcal{A}_3 \leqslant \mathcal{A}_2$. By simple weakening we compute, as in case 1, the unifier $\{\alpha_1/\beta, \alpha_2/\beta\}$, for some variable $\beta : \mathcal{A}_3$. But this is not a mgu! For instance, $\{\alpha_1/\beta' * \beta'', \alpha_2/\beta' * \beta''\}$, with $\beta, \beta' : \mathcal{A}_3$ unifies t_1 and t_2, while $\{\beta/\beta' * \beta''\}$ is not a well-formed substitution, since $\beta' * \beta''$ has not the sort $\mathcal{A}_3$. In fact, in order to have finite CSUs, we must rule out such situations by imposing a transitive $\mathcal{A}_4$ such that $\mathcal{A}_4 \leqslant \mathcal{A}_1$, $\mathcal{A}_4 \leqslant \mathcal{A}_2$ and $\mathcal{A}_3 \leqslant \mathcal{A}_4$. This is the restriction on the set of sorts mentioned in Proposition 3.1. In fact, this is not a real restriction since we may always add such a sort: its domain A_4 will be something between the semigroup generated by A_3 and the intersection of A_1 and A_2.

For the sake of simplicity, we present an algorithm unifying sets $\{t, t'\}$ of two terms; its extension to atoms is straightforward. It is written in functional form, the only specific feature being the use of nondeterministic "or" expressions, described below.

Syntax

or $bool_1 \Rightarrow fexp_1$; ...; $bool_n \Rightarrow fexp_n$ **end_or**

where the $bool_i$'s (the *guards*) are boolean expressions, and the $fexp_i$'s are nondeterministic functional expressions.

Semantics

- Choose some i such that $bool_i$ is true, and evaluate $fexp_i$.
- If there is not such i, return *failure*.

Hence, a functional expression including **or**-expressions may have several evaluations, leading to different results. A result can be either a proper value (here, a substitution), or *failure*. A computation is *successful* if it produces a proper value. We extend the composition of substitutions in such a way that $(\sigma\ failure) = failure$.

The *unification algorithm* is presented at the end of the section. It consists of three mutually recursive nondeterministic functions:

- Unify(t_1, t_2) is the main one. Every computation returns $E(S)$-unifiers of t_1 and t_2.
- If $L_1 = (u_1, \ldots, u_k)$ and $L_2 = (v_1, \ldots, v_k)$ are two lists of terms of the same length, Unify-list(L_1, L_2) produces substitutions that $E(S)$-unify all the pairs (u_i, v_i).
- Unify-$\mathbb{W}(t_1, t_2)$ is a specialisation of Unify(t_1, t_2) for terms of sort $\mathbb{W}$, and concentrate the specific aspects of our algorithm.

Finally, note that in each step terms will be rewritten in normal form, and we suppose that the variables introduced in weakening operations are fresh ones.

Remark. A rule-based algorithm – in fact, simply an iterative presentation of the same algorithm – is possible *using a stack policy* for the set of equations. We choose to give the recursive expression of control which reflects better the real structure of our algorithm. It must be stressed that, in any case, full nondeterminism does not seem possible for unification in path theories (roughly speaking, unification of $\mathbb{W}$-terms must be ordered from one end of the "path" to the other). Also note that extending UPP to sets of equations in a consistent way is not trivial, especially if one accepts equations of the form $\alpha = t$, where α is some variable in Ω. But this is needed for a direct proof of a standard rule-based algorithm.

Proposition 3.1. *Assume that for any transitive sorts* $\mathbb{A}_i$ *and* $\mathbb{A}_j$, lb$(\mathbb{A}_i, \mathbb{A}_j)$ *is empty or has only transitive elements. Then*

(i) *The unification algorithm terminates on UPP sets of terms or atoms and produces a (nonnecessary minimal)* $\Sigma(S)$–$E(S)$*-CSU.*

(ii) *Moreover, if* B *is some quantifier free UPP formula, for any substitution* σ *computed by the algorithm,* $B\sigma$ *has the UPP.*

Due to lack of place, we omit the proof, which is long and intricate and essentially the same as in the monomodal case [4]. After this paper was written, rule-based algorithms and simplified proofs have been elaborated [10].

$E(S)$-Unification Algorithm

Unify$(t_1, t_2) =$

if t_1 or t_2 is a variable x

then let t be the other term in

if $t = x$ **then** $\emptyset$ **else if** $x \in \mathrm{Var}(t)$ **then** *failure* **else** $\{x/t\}$

else
if t_1 and t_2 are of sort $\mathbb{W}$ **then** Unify-$\mathbb{W}(t_1, t_2)$
else
let $t_1 = f(u_1, \ldots, u_n)$ and $t_2 = g(v_1, \ldots, v_m)$ in
if $f \neq g$ **then** *failure* **else** Unify-list$((u_1, \ldots, u_n), (v_1, \ldots, v_n))$

Unify-list$((u_1, \ldots, u_n), (v_1, \ldots, v_n)) =$
begin
$k := 0$; $\tau := \emptyset$;
while $k < n$ and $\tau \neq$ *failure* **do**
begin $k := k + 1$; $\mu :=$ Unify$(\tau \downarrow u_k, \sigma \downarrow v_k)$; $\tau := \tau\mu$ **end**;
return(τ)
end

Unify-$\mathbb{W}(t_1, t_2) =$
if t_1 and t_2 are comparable
then let $\{s_1, s_2\} = \{t_1, t_2\}$ s.t. $s_2 = s_1!a_1!\cdots!a_k$ $(k > 0)$ in
if $k = 0$ **then** $\emptyset$
else if every $a_i \in \Omega$ with $a_i : s_i$ and REF(s_i)
then $\{a_i/1\}_{i=1,\ldots,k}$ **else** *failure*
else let $\{u_1, u_2\} = \{t_1, t_2\}$ nondeterministically in
or
(1) $u_1 = u_1'!\alpha : s_1$ and $u_2 = u_2'!a : s_2$ *with* $\alpha \in \Omega$, $a \notin \Omega$, $\alpha \notin \mathrm{Var}(a)$, $s_2 \leqslant s_1$
$\Rightarrow \{\alpha/a\}$ Unify-$\mathbb{W}(u_1', u_2')$;
(2) $u_1 = u_1'!\alpha : s_1$ and $u_2 = u_2'!a : s_2$ *with* $\alpha \in \Omega$, $a \notin \Omega$, $\alpha \notin \mathrm{Var}(a)$
$\Rightarrow \{\alpha/\beta : s_3 * a\}$ Unify-$\mathbb{W}(u_1'!\beta, u_2')$
where $s_3 \in \mathrm{mtrans}(s_1)$, $s_3 \geqslant s_2$, $\beta : s_3$ is a fresh variable;
(3) $u_1 = u_1'!a$ and $u_2 = u_2'!b$ *with* $a, b \notin \Omega$
$\Rightarrow$Unify-list$((u_1', a), (u_2', b))$;
(4) $u_1 = u_1'!\alpha : s_1$ *with* $\alpha \in \Omega$ and REF(s_1)
$\Rightarrow \{\alpha/1\}$Unify-$\mathbb{W}(u_1', u_2)$;
(5) $u_1 = u_1'!\alpha_1 : s_1$ and $u_2 = u_2'!\alpha_2 : s_2$ *with* $\alpha_1, \alpha_2 \in \Omega$
$\Rightarrow \{\alpha_1/\beta : s_3\}\{\alpha_2/\beta : s_3\}$ Unify-$\mathbb{W}(u_1', u_2')$
where $s_3 \in \mathrm{lb}(s_1, s_2)$ and $\beta : s_3$ is a fresh variable;
(6) $u_1 = u_1'!\alpha_1 : s_1$ and $u_2 = u_2'!\alpha_2 : s_2$ *with* $\alpha_1, \alpha_2 \in \Omega$
$\Rightarrow \{\alpha_2/\beta_2 : s_4\}\{\alpha_1/\beta_1 : s_3 * \beta_2 : s_4\}$ Unify-$\mathbb{W}(u_1'!\beta_1, u_2')$
where $s_3 \in \mathrm{mtrans}(s_1)$, $s_4 \in \mathrm{lb}(s_3, s_2)$, and $\beta_1 : s_3, \beta_2 : s_4$ are fresh variables;
end_or

4. Logic programming in path theories

By Propositions 2.2, 2.6 and 2.7, multimodal reasoning has been reduced to deduction in path theories. Without loss of generality, we may consider path formulas in clausal form. If the considered set of clauses is Horn, we can apply SLD

resolution and get a system for logic programming in path theories. We call it PATHLOG.

4.1. *PATHLOG and SLD resolution*

Throughout this section we consider some multimodal system S and the corresponding path theory $E(S)$ in the language generated by $\Sigma(S)$.

Definition. A *program clause* is any expression $A \leftarrow A_1 \wedge \cdots \wedge A_n$ with $n \geqslant 0$, where A and the A_i's are atoms. A *goal* is an expression $\leftarrow A_1 \wedge \cdots \wedge A_n$ with $n \geqslant 0$, where the A_i's are atoms. A *PATHLOG program* is a pair (P, G) consisting of a set of program clauses P and a goal G such that $P \cup \{G\}$ has the UPP. An *answer substitution* is a Σ-substitution σ such that $P \models_{E(S)} (A_1 \wedge \cdots \wedge A_n)\sigma$.

We suppose that for all atoms A, A', $\Gamma(A, A')$ is a finite $\Sigma(S)$–$E(S)$-CSU of A and A'. *SLD resolution* is defined for PATHLOG programs in an almost standard way. The only difference is that in each step, not only do we choose some atom B_i in the current goal $B_1 \wedge \cdots \wedge B_n$ and a clause $C = A \leftarrow A_1 \wedge \cdots \wedge A_n$, but also a $\Sigma(S)$–$E(S)$-unifier in $\Gamma(A, B_i)$. A *derivation* is then defined in the usual way, and the *computed substitution* is the composition of these unifiers. Observe that by Proposition 3.1 UPP is preserved, so that our unification algorithm is applicable in each step of the derivation.

Proposition 4.1 (Soundness of SLD Σ–E-resolution). *Every computed substitution is an answer substitution.*

Proposition 4.2 (Completeness of SLD Σ–E-resolution). *Let σ be some answer substitution for a PATHLOG program (P, G). Then there exists some computed substitution τ and some γ such that $\sigma =_{E(S)} \tau\gamma$.*

The proof follows very closely the proof of the corresponding propositions in standard logic programming as presented in [24].

The main technical point concerns the definition of Herbrand interpretations. If G' is the set of function and constant symbols, let $T(G')$ be the set of ground terms on the alphabet G'. We suppose that for each minimal sort symbol $\mathbb{s}$, there is some term $t : \mathbb{s}$; otherwise, we add some constant. Let $H = T(G')/=_{E(S)}$, the quotient set of $T(G')$ by $E(S)$ equality, and for every $\mathbb{s}$, $H_{\mathbb{s}} = \{c \in H / \exists t \in c\, t : \mathbb{s}\}$. Obviously, $H_{\mathbb{s}} \subset H_{\mathbb{s}'}$ if $\mathbb{s} < \mathbb{s}'$ and each $H_{\mathbb{s}}$ is nonempty. A Herbrand $\Sigma(S)$–$E(S)$-interpretation is any $\mathcal{H} = \langle H_{\mathbb{s}_0}, H_{\mathbb{s}_1}, \ldots, H_{\mathbb{s}_p}, H_{\mathbb{s}_i}, G'_{fn}, P'_{\mathbb{s}\mathbb{s}'} \rangle$, where for any function symbol f in G', $f_{\mathcal{H}}$ is the quotient of the canonical function f from $T(G')^n$ to $T(G')$ by $=_{E(S)}$. It should be noted that this construction is made possible by the form of the sort declarations in $\Sigma(S)$ and of the equational theories $E(S)$.

Let $\simeq$ be the equivalence relation on ground atoms defined by

$$p(t_1, \ldots, t_n) \simeq p(t'_1, \ldots, t'_n) \text{ iff } t_i =_{E(S)} t'_i \text{ for all } i.$$

The Herbrand base is the set whose elements are equivalence classes for $\simeq$. Observe that since $E(S)$ is canonical, every class can be represented by some atom $p(t_1, \ldots, t_n)$, where the t_i's are in normal form. Clearly, a Herbrand interpretation $\mathcal{H}$ is completely defined by a subset $B_{\mathcal{H}}$ of the Herbrand base with $\mathcal{H} \models p(t_1, \ldots, t_n)$ iff the class of $p(t_1, \ldots, t_n)$ belongs to $B_{\mathcal{H}}$. As usual, we identify $\mathcal{H}$ and $B_{\mathcal{H}}$.

As in the standard case, the intersection of the Herbrand models of a program P is again a Herbrand model of P, the minimal model $\mathcal{H}_P$.

Proposition 4.3. *Given any PATHLOG program P, and ground atom A, the following are equivalent:*

- $P \models A$.
- A is consequence of P w.r.t. Herbrand $\Sigma(S)$–$E(S)$-interpretations.
- The $\simeq$-class of A belongs to $\mathcal{H}_P$.

The rest of the proof is a straightforward adaptation of [24, Chapter 2]. □

Note that by translation back, these results provide "for free" a notion of Herbrand interpretation for multimodal logic; other attempts are [8, 29]. Also we have a minimal Herbrand model for sets of Horn clauses, including a construction of a set of worlds "minimal" in a certain manner.

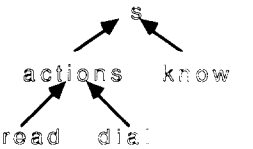
Proposition 4.4. *Suppose* (P, G) *is a PATHLOG program and that there is no function symbol with declared range sort* $\mathbb{A}_k$. *Then Herbrand models of P have the closure property w.r.t.* $\mathbb{A}_k$. *Hence, Proposition* 4.2 *holds when answer substitutions are defined relative to the class of interpretations having this property.*

Proof. Consider for instance the case where M_k is KT4. The carrier for $\mathbb{A}_k$ in the Herbrand universe is $\{1, a_1, a_1 * a_2, a_1 * a_2 * a_3, \ldots\}$, where $a_1\, a_2, a_3$, etc., are normal forms of arbitrary terms of sort $\mathbb{A}_j$ for arbitrary j's in B_k. □

We conclude with two examples of PATHLOG programs, solving the problems of Examples 1.1 and 1.2.

4.2. The safe problem (continued)

First we must define the adequate path theory and translate the given set of formulas. We have sort symbols $\mathbb{D}$, $\mathbb{W}$ and know, read, dial, actions, s associated with the considered modalities in an obvious way, with the order diagram and sort declarations shown in Fig. 3. After translation, we get the following clauses, where $\alpha, \beta, \gamma, \delta, X, Y, Z, L$ are variables and $\varphi, \psi, \eta, g, h$ skolem function symbols; we indicate the sort of $\alpha, \beta, \gamma, \delta$ and the range sort of φ, ψ, η once in each clause; $\mathbb{D}$ is the sort of X, Y, Z, L and the range sort of g and h:



ε: W ! : W x s → s ...
*: s x s → s,
actions x actions → actions
know x know → know
1: know, s, actions

Fig. 3.

(1) comb(ε!α: s!φ(ε!α, S, L, X): read!γ: know, X, S)
←comb(ε!α, X, S), written_in(ε!α, X, L).
(2) open(ε!α: s!η(ε!α, S): dial, S)
←comb(ε!α!ψ(ε!α, X, S): know, X, S).
(3) comb(ε!γ: know, h(ε!γ), safe1).
(3)′ written_in(ε!γ: know, h(ε!γ), desk1).

The goal

$$\leftarrow \text{open}(\varepsilon!\delta : \text{know}!\beta : \text{actions}, \text{safe1})$$

can be read: find the epistemic alternatives δ and the action sequences β such that in these alternatives, after performing β, the safe is open. If one finds a sequence β such that for all δ the goal formula holds, this means that John knows that β opens the safe.

We have the following successful derivation (at each step we indicate the selected clause and the unifier):

$\leftarrow \text{comb}(\varepsilon!\delta : \text{know}!\beta_1 : \text{actions}!\psi(\varepsilon!\delta!\beta_1, X, \text{safe1}) : \text{know}, X, \text{safe1})$

by (2) with $\{\beta/\beta_1 * \eta(\varepsilon!\delta!\beta_1, \text{safe1}), \alpha/\delta * \beta_1, S/\text{safe1}\}$,

$\leftarrow \text{comb}(\varepsilon!\delta : \text{know}, X, \text{safe1}), \text{written\_in}(\varepsilon!\delta, X, L)$

by (1) with X renamed in X_1 and the unifier

$\{\alpha/\delta, \beta 1/\varphi(\varepsilon!\delta, \text{safe1}, L, X), \gamma/\psi(\varepsilon!\delta!\varphi(\varepsilon!\delta, \text{safe1}, L, X, \text{safe1}), X),$

$X_1/X, S/\text{safe1}\}$,

$\leftarrow \text{written\_in}(\varepsilon!\delta : \text{know}, h(\varepsilon!\delta), L)$ by (3) with $\{X/h(\varepsilon!\delta), \gamma/\delta\}$,

$\leftarrow \square$ by (3)′ with $\{L/\text{desk1}, \gamma/\delta\}$.

Hence, we get the answer

$\beta = \varphi(\varepsilon!\delta, \text{safe1}, \text{desk1}, h(\varepsilon!\delta)) : \text{read}$

$*\eta(\varepsilon!\delta!\varphi(\varepsilon!\delta, \text{safe1}, \text{desk1}, h(\varepsilon!\delta)), \text{safe1}) : \text{dial},$

with δ unbound; the following formula is a logical consequence of the program:

$\forall\delta \text{ open}(\varepsilon!\delta : \text{know}!\varphi(\varepsilon!\delta, \text{safe1}, \text{desk 1}, h(\varepsilon!\delta)) : \text{read}$

$!\eta(\varepsilon!\delta!\varphi(\varepsilon!\delta, \text{safe1}, \text{desk1}, h(\varepsilon!\delta)), \text{safe1}) : \text{dial}, \text{safe1}).$

Similarly, the multimodal formula

$$[\text{know}]\langle\text{read}\rangle\langle\text{dial}\rangle\,\text{open}(\text{safe1})$$

is a logical consequence of the formulas (1)–(3) of Example 1.1.

Observe that, by Propositions 2.6 and 4.4, the logical consequence holds in "standard" interpretations, with the closure property w.r.t. the modalities *actions* and *s*.

4.3. TEMPLOG (continued)

Now we show how to implement TEMPLOG and program the Fibonacci example. We shall use the sort symbols $\mathbb{W}, \mathbb{D}, \mathbb{N}$ and $\mathbb{F}$. $\mathbb{N}$ corresponds to $\bigcirc$ (next) and $\mathbb{F}$ to $(\Box, \Diamond)$ (future). We have: $\mathbb{N} < \mathbb{F}$, $*: \mathbb{F} \times \mathbb{F} \to \mathbb{F}$, $1: \mathbb{F}$, and $a: \mathbb{N}$.

Suppose we write PATHLOG programs with no function symbol of range sort $\mathbb{F}$. Then, by Proposition 4.4, we have completeness w.r.t. interpretations with the closure property relative to $\mathbb{F}$. Indeed, the Herbrand models are such that

$$N = \{a\}, \qquad F = \{a, a*a, a*a*a, \ldots\}, \qquad W = \{\varepsilon,\ \varepsilon!a, \varepsilon!a!a, \varepsilon!a!a!a, \ldots\}$$

and W is isomorphic to the set of natural numbers. These interpretations correspond exactly to standard models of the temporal logic being considered. PATHLOG is complete with respect to that class of interpretations.

Now, it happens that the set of "temporal clauses" proposed by Abadi and Manna [1] for TEMPLOG (roughly, no $\Diamond$ in positive occurrence) are mapped by the translation into Horn clauses with the above restriction. Hence, we have the definition of an interpreter for TEMPLOG with completeness as a consequence of our general theory.

Example. The "fibonacci program" is translated into

$$\text{fib}(\varepsilon, 0)$$

$$\text{fib}(\varepsilon!a, 1)$$

$$\text{fib}(\varepsilon!\alpha!a!a, X) \leftarrow \text{fib}(\varepsilon!\alpha, Y), \text{fib}(\varepsilon!\alpha!a, Z), X \text{ is } Y+Z$$

with the goal

$$\leftarrow \text{fib}(\varepsilon!\beta, X)$$

The answers are ("is" is as in standard Prolog):

$$\beta = 1,\ X = 0; \quad \beta = a,\ X = 1; \quad \beta = a*a,\ X = 1; \quad \beta = a*a*a,\ X = 2; \text{ etc.}$$

In other words, the following formulas are consequences of the given set of clauses:

$$\text{fib}(\varepsilon, 0),\ \text{fib}(\varepsilon!a, 1),\ \text{fib}(\varepsilon!a!a, 1), \ldots,$$

and fib(0), fib(1), fib(1), fib(2), ... are consequences of the original TEMPLOG program, as expected.

Comments. (1) Certainly, much should be done in order to define a methodology for knowledge representation in this formalism, either directly in PATHLOG or in multimodal logic, followed by translation. These examples intend only to illustrate the evaluation mechanism of PATHLOG programs and show encouraging indications concerning their expressive power. In particular, note that we get answers not expressible in the modal language itself.

(2) One might argue that this is not *modal* logic programming, strictly speaking. Indeed, further work, should be done in order to define the class of modal formulas which produces sets of Horn clauses by translation, and to examine how answers and proofs might be "decompiled". But this sounds like a rather routine task without special difficulties.

Conclusions: relation with other works and discussion

(1) Ohlbach [26] and Farinas and Herzig [13] have proposed ATP methods for modal logic (with one modality) strongly related to ours as given in [4, 3]. The difference essentially lies in the *purely algebraic techniques* used in our approach. We believe that it gives our formalism a better mathematical tractability, since we can either directly use or easily adapt the known results and techniques of classical logic. The results presented in Section 4, concerning the declarative semantics of PATHLOG (Proposition 4.3) and the completeness of SLD resolution (Propositions 4.2 and 4.4) with the consequences for TEMPLOG, are quite significant.

(2) Modal logic programming is something like a current technical challenge and several approaches have been proposed in the past few years. All the following use "direct" methods, without translation in classical logic.

We already presented Abadi and Manna's TEMPLOG [1] and showed how PATHLOG subsumes it. In [15] Gabbay proposes another approach to temporal logic programming. He defines a notion of "Horn temporal clauses" with a specific computation rule for "temporal programs". Compared to ours, on the one hand, his language deals as well with linear or branching time with future and past operators, but, on the other hand, there is no "next" operator and the set of "Horn clauses" is more restricted. Moreover, the computation rule is an ad hoc one, quite intricate and far from the standard PROLOG.

Farinas's MOLOG [12] accepts a class of very expressive multimodal languages in which modal operators are indexed by first-order terms, and is, for that reason, very attractive and stimulating. For instance, one can write such a clause as "[believe(x)] [know(father(x))] $P(x)$", where x is a variable, which can be read "everybody believes that his father knows that he possesses property P". It is inspired by Farinas's "modal resolution" [11]. But there is no clear logical semantics (and, hence, completeness

results) for MOLOG. Attempts restricted to fragments corresponding to quantifier-free basic modal systems from K to KT5 (S5) are made in [6] and [5]. In fact, the former may be seen as a premisse to the method of Farinas and Herzig [13]. And the difficulties encountered in the latter were the determining factors for switching to a "translation" method.

Orgun and Wadge [29] propose a "general theory of intensional logic programming", which owes a part of its inspiration to the Lucid experience. Their approach can deal with modal operators with a more general semantic than the standard "possible worlds" one. But restricted to this case, there are limitations we do not have; for instance, they cannot cope with KT4 operators, the considered "Horn clauses" are less general, and the set of worlds in Kripke structures is always isomorphic to ω. Also, and this is a major problem, they provide only a declarative semantic, not a procedural one. Finally, in [28] Okada presents proof-theoretic preliminaries for modal logic programming, but the theory is not fully developed.

Hence, it seems that the path theories approach is placed quite well w.r.t. expressivity of the accepted modal languages and theoretical results. We think this pleads for the translation from modal to classical logic when *automated reasoning* is concerned, and *provided* one uses a "good translation".

(3) Now, what is a "good translation"? Two related criteria can be invoked. First, it should preserve "something" of the structure of modal formulas. Second, there should be some special device to deal with the structure of worlds. In other words, we should be able to determine a fragment of classical logic, for which specific methods can be used. In our approach, the fragment is characterised by UPP and the specialised method is $\Sigma(S)$–$E(S)$-unification. The situation is similar in Ohlbach's work. Another interesting approach filling these conditions is that of Frisch and Scherl [14], where the target language and the method are those of some constraint logic.

(4) After this paper was written, an extension to *à la MOLOG* multimodal languages has been elaborated [10]. Indeed, we believe that this is the kind of formalism needed if some application is to be considered. The semantics of such a logic and translations in classical logic are also investigated in [26] and [21], but no specific automated method is provided there.

References

[1] M. Abadi and Z. Manna, Temporal logic programming, in: *Proc. IEEE Symp. on Logic Programming*, San Francisco (1987) 4–116.

[2] E. Audureau, P. Enjalbert and L. Farinas, *Logique temporelle: sémantique et validation de programmes parallèles* (Masson, Paris, 1990).

[3] Y. Auffray and P. Enjalbert, Modal theorem proving using equational methods, in: *Proc. 11th Internat. Joint Conf. on Artificial Intelligence*, Detroit (1989) 441–445.

[4] Y. Auffray and P. Enjalbert, Modal theorem proving: an equational viewpoint, *J. Logic and Comput.* 1992, to appear.

[5] Y. Auffray, P. Enjalbert and J.J. Hebrard, Strategies for propositional modal resolution, results and problems. *J. Automat. Reason.* **6** (1990) 1–38.

[6] Ph. Balbiani, L. Farinas and A. Herzig, Declarative semantics for modal logic programs, in: *Proc. Internat. Conf. on Fifth Generation Computer Systems* (1988).
[7] L. Catach, Normal multimodal logics, in: *Proc. Nat. Conf. on Artificial Intelligence*, St. Paul, Minnesota (1988) 491–495.
[8] M. Cialdea, Une méthode de déduction automatique en logique modale, Thèse de l'Université Paul Sabatier, Toulouse, France, 1986.
[9] F. Debart, P. Enjalbert and M. Lescot, Multi-modal automated deduction using equational and order-sorted logic, in: *Proc. 2nd Conf. on Conditional and Typed Rewriting Systems*, Montreal 120–124 (1990); in: M. Okada and S. Kaplan, eds., Lecture Notes in Computer Science (Springer, Berlin, 1991), to appear.
[10] F. Debart, Théories équationelles et de constraints pour la démonstration automatique en logique multi-modale, Thèse de l'Université de Caen, 1992.
[11] P. Enjalbert and L. Farinas, Modal resolution in clausal form, *Theoret. Comput. Sci.* **65** (1989) 1–33.
[12] L. Farinas, MOLOG: a system that extends PROLOG with modal logic, *New Generation Computing* **4**(1) (1986) 35–50.
[13] L. Farinas and A. Herzig, Quantified modal logic and unification theory, Rapport LSI 293, Université P. Sabatier, Toulouse, France, 1988.
[14] A.M. Frisch and R.B. Scherl, A general framework for modal deduction, in: *Proc. 2nd Internat. Conf. on Principles of Representation and Reasoning KG '91* (Morgan Kaufmann, Los Altos, CA, 1991) 196–207.
[15] D.M. Gabbay, Modal and temporal logic programming, in: A. Galton, ed., *Temporal Logics and Their Applications* (Academic Press, New York, 1987) 197–237.
[16] J.A. Goguen and J. Meseguer, Models and equality for logical programming, in: *Proc. TAPSOT '87*, Lecture Notes in Computer Science, Vol. 250 (Springer, Berlin, 1987) 1–22.
[17] J.A. Goguen and J. Meseguer, Order-sorted algebras I: equational deduction for multiple inheritance, overloading, exceptions and partial operations, Tech. Report SRI-CSL 89-10, 1989.
[18] R. Goldblatt, *Logics of Time and Computation* (CSLI-SRI Publications, Stanford, CA, 1987).
[19] J.Y. Halpern and Y.O. Moses, A guide to the modal logics of knowledge and belief, in: *Proc. Internat. Joint Conf. on Artificial Intelligence* (1985) 480–490.
[20] D. Harel, *First Order Dynamics Logic*, Lecture Notes in Computer Science, Vol. 68 (Springer, Berlin, 1979).
[21] A. Herzig and H.J. Ohlbach, Parameter structures for parametrized modal operators, in: *Proc. IJCAI '91* (1991).
[22] J.P. Jouannaud and C. Kirchner, Solving equations in abstract algebras: a rule-based survey on unification, Tech. Report 561, Laboratoire de Recherche en informatique, Orsay, France, 1990.
[23] C. Kirchner, Order-sorted equational unification, in: *Proc. 5th Internat. Conf. on Logic Programming* (1988).
[24] J.W. Lloyd, *Foundations of Logic Programming* (Springer, Berlin, 1984)
[25] R.C. Moore, Reasoning about knowledge and action, Ph.D. Thesis, MIT, Cambridge, MA, 1980.
[26] H.J. Ohlbach, A resolution calculus for modal logics, in: *Proc. 9th Internat. Conf. on Automated Deduction*, Lecture Notes in Computer Science, Vol. 310 (Springer, Berlin, 1988) 500–516.
[27] H.J. Ohlbach, Semantics-based translation methods for modal logics, *J. Logic Comput.* **1** (1991) 691–746.
[28] M. Okada, Mathematical basis of modal logic programming, in: *Proc. JELIA '88*, Cahier du LIUC 89-5, Université de Caen, France, 1988.
[29] M.A. Orgun and W.W. Wadge, Towards a unified theory of intensional logic programming, Tech. Report, Dept. of Computer Science, Univ. of Victoria, BC, Canada, 1989.
[30] G. Plotkin, Building in equational theories, *Mach. Intell.* 7 (1972) 73–90.
[31] M. Schmidt-Schauss, Unification in many-sorted equational theories, in: *Proc. 8th Internat. Conf. on Automated Deduction*, Oxford, England, Lecture Notes in Computer Science, Vol. 230 (Springer, Berlin, 1986) 538–552.
[32] G. Smolka, Order-sorted Horn logic: semantics and deduction, SEKI Report SR-86-17, Kaiserslautern Univ., 1986.
[33] C. Walther, *A Many-sorted Calculus Based on Resolution and Paramodulation*, Research notes in Artificial Intelligence (Pitman, London 1987).