Labelled Deduction for the Guarded Fragment*

Maarten Marx[†] Stefan Schlobach[‡] Szabolcs Mikulás[‡]

February 25, 1999

Abstract

We present the tableau calculus LC_2-TAB which is sound and complete with respect to local square modal logic. The system is a labelled deduction calculus in the spirit of those for modal **S5**. The novelty here is that the calculus works in two interacting dimensions. This 2-dimensional modal logic allows one to simulate different other modal logics, like K, KT, KTB or multi-K in quite an elegant way. The calculus is also strong enough to decide an interesting pspace complete sub-fragment of the guarded fragment, which is generally conceived of as the true modal fragment of first order logic. A PROLOG implementation of this calculus is available through the WWW.

1 Introduction

In [11], a modalised version of first order logic with n variables is presented, called MLR_n (for modal logic of n–ary relations). This modal language can be given different semantics, corresponding to different fragments of the language of first order logic. Of particular interest is the relativized semantics which gives rise to a decidable system. As we will see here, this semantics corresponds to the guarded fragment of first order logic (cf. [2, 3, 14]). This decidable fragment was introduced to account for the fact that modal logics are so robustly decidable.
Here we will focus on the case with only two variables. First –as is well known– two variables are sufficient to translate (multi–) modal logics into first order logic. Secondly, the language MLR_n for $n = 2$ has a pspace complete satisfaction problem, while for $n > 2$ this problem becomes exptime–complete [10]. But of greater importance for our purpose here is that for $n = 2$, the language has a special finite model property which gives rise to

*This research has been supported by UK EPSRC grant No.GR/K54946.

[†]University of Amsterdam, Department of W.I.N.S., Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands, marx@wins.uva.nl

[‡]King's College, DCS, Strand, London WC2R 2LS, UK, {schlobac,szabolcs}@dcs.kcl.ac.uk

a straightforward tableaux calculus based on labelled deduction ideas. For $n > 2$, the finite model property also holds [1, 8], but a simple model creating technique (like a naive tableaux system) does not work (see Section 4).
We designed and implemented a sound and complete labelled tableau calculus for LC_2, which can be used via the WWW from URL

`http://www.dcs.kcl.ac.uk/cgi-bin/LC2/lc2.cgi`.

We will show how the system works by looking at an instructive example proof. We evaluate the strength of the logic by showing how it can be used as a decision procedure for a range of other modal logics, including description logics.
The paper is organized as follows. The next section introduces two–dimensional modal logic and shows its relation with first order logic and other modal systems. Section 3 shows from a complexity point of view why our choice is correct if we want to use the system presented here to do proofs for pspace complete modal logics. Section 4 presents an example indicating the difficulties extending our system to higher dimensions. Section 5 presents the tableaux calculus. The final section deals with its implementation in PROLOG.

2 Local cubic modal logic of two dimensions

2.1 Language and semantics

Throughout this paper, we will focus on the 2-dimensional case, for which we introduce a special notation, due to Segerberg. For the logic MLR_n, we introduce n diamonds $\Diamond_i$, for every function $\sigma : n \longrightarrow n$ a unary modality $\otimes_\sigma$ and for all $i, j < n$ a modal constant id_{ij}. The reader interested in the more general ideas is referred to [11] for details about the logic MLR_n.

Definition 2.1 *Given a countable set of propositional variables $p_0, p_1, \ldots$, the set of MLR_2-formulas is defined as the smallest set such that*

1. *the propositional variables $p_0, p_1, \ldots$ and the modal constant id are LC_2-formulas (called atomic formulas),*
2. *if φ and ψ are LC_2-formulas then also $\neg\varphi, \varphi \vee \psi, \diamondvert\varphi, \diamondminus\varphi$ and $\otimes\varphi$ are LC_2-formulas.*

We abbreviate $\boxbar\varphi \equiv \neg\diamondvert\neg\varphi$, $\boxminus\varphi \equiv \neg\diamondminus\neg\varphi$ and $\varphi \wedge \psi \equiv \neg(\neg\psi \vee \neg\psi)$ as usual. Two new operators are defined by $\ominus\,\varphi \equiv \diamondminus(id \wedge \varphi)$ and $\obar\,\varphi \equiv \diamondvert(id \wedge \varphi)$.

Semantically, boxes and diamonds of MLR_2 will be interpreted as classical S5 modalities in two dimensions, with two interacting accessibility relations. We are interested in two classes of models for this language: the *squares* and the *local squares*. For U a set, we use ${}^2U$ to denote the product $U \times U$.

Definition 2.2 *Let U be a set. The set $W = U \times U$ is called a square. A set $W \subseteq {}^2U$ is called a local square if $\langle a, b\rangle \in W$ implies that ${}^2\{a, b\} \subseteq W$.*

Definition 2.3 *A model is a pair $\mathcal{M} = (W, v)$, with $W \subseteq {}^2U$ for an arbitrary set U and v a valuation, i.e. a function mapping every propositional variable to a subset of W. Such a model is called a (local) square if W is a (local) square. Truth of formulas at a pair $\langle a, b\rangle \in W$ in such models is defined as follows:*

$$
\begin{array}{lcl}
\mathcal{M}, \langle a, b\rangle \models p & \Leftrightarrow & \langle a, b\rangle \in v(p) \\
\mathcal{M}, \langle a, b\rangle \models id & \Leftrightarrow & a = b \\
\mathcal{M}, \langle a, b\rangle \models \neg\varphi & \Leftrightarrow & \mathcal{M}, \langle a, b\rangle \not\models \varphi \\
\mathcal{M}, \langle a, b\rangle \models \varphi \vee \psi & \Leftrightarrow & \mathcal{M}, \langle a, b\rangle \models \varphi \textit{ or } \mathcal{M}, \langle a, b\rangle \models \psi \\
\mathcal{M}, \langle a, b\rangle \models \otimes\varphi & \Leftrightarrow & \mathcal{M}, \langle b, a\rangle \models \varphi \\
\mathcal{M}, \langle a, b\rangle \models \Diamond\!\!\!\mid\,\varphi & \Leftrightarrow & \textit{there is a } \langle a, c\rangle \in W, \\
 & & \textit{such that } \mathcal{M}, \langle a, c\rangle \models \varphi \\
\mathcal{M}, \langle a, b\rangle \models \Diamond\!\!\!\!-\,\varphi & \Leftrightarrow & \textit{there is a } \langle c, b\rangle \in W, \\
 & & \textit{such that } \mathcal{M}, \langle c, b\rangle \models \varphi.
\end{array}
$$

A formula φ is square valid (notation $\models_{SQ} \varphi$) if φ is true in all pairs in all square models. A formula φ is locally square valid (notation $\models_{LSQ} \varphi$) if φ is true in all pairs in all locally square models.

The modal logic MLR_2 interpreted on the class of local squares will be denoted by MLR_2/LSQ. A graphical intuition for the $\otimes$ modality is given in figure 1, where $\otimes\varphi$ is true in $\langle a, b\rangle$ if and only if φ is true in $\langle b, a\rangle$.

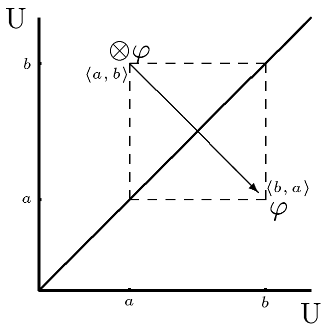

Figure 1: The semantics of a converse-formula $\otimes\varphi$

The semantics of the defined operators is an immediate result of definition 2.3.

$$
\begin{array}{lcl}
\mathcal{M}, \langle a, b\rangle \models \bigcirc\!\!\!\!\mid\;\varphi & \Leftrightarrow & \mathcal{M}, \langle a, a\rangle \models \varphi \\
\mathcal{M}, \langle a, b\rangle \models \square\!\!\!\mid\;\varphi & \Leftrightarrow & \text{for all pairs } \langle a, x\rangle \in W \\
 & & \quad\text{holds that } \mathcal{M}, \langle a, x\rangle \models \varphi
\end{array}
$$

The semantics of $\wedge, \ominus$ and $\boxminus$ is defined in a similar way.

2.2 Relation with first order logic

The language MLR_2 interpreted on squares is just as expressive as L_2 — first order logic with two variables and equality and at most binary relation symbols (no constant and function symbols). In fact, the language MLR_n was precisely designed to be the modal counterpart of first order logic. For the precise translations back and forth, we refer to [11]. Here we just give the most interesting clauses in the two–dimensional case.

$$
\begin{array}{llll}
(Rv_0v_1)^t = r & (Rv_1v_0)^t = \otimes r & (Rv_0v_0)^t = \obar\, r & (Rv_1v_1)^t = \ominus\, r \\
(v_0 = v_1)^t = id & & & \\
(\exists v_0 \varphi)^t = \diamondminus \varphi^t & (\exists v_1 \varphi)^t = \diamondbar \varphi^t. & &
\end{array}
$$

The translation in the other direction is just the converse of $(\cdot)^t$, though we need to take care of $\otimes$. One might wonder how we translate unary predicates like Pv_0. There are several standard ways of handling these, all preserving satisfiability. One way is to transform the first order formula φ into one with only binary predicates by uniformly replacing occurrences of Pv_0 by $P'v_0v_0$ and of Pv_1 by $P'v_1v_1$. For details we refer to e.g., [6].
When we interpret MLR_2 on local squares we lose the one–to–one correspondence with L_2. This is immediate since the L_2 sentence $\exists v_0 \exists v_1 Rv_0v_1 \leftrightarrow \exists v_1 \exists v_0 Rv_0v_1$ is valid, while its translation $\diamondminus \diamondbar r \leftrightarrow \diamondbar \diamondminus r$ is not valid on local squares (a counterexample is provided using our tableaux system in section 5.2).
So the question becomes whether there exists a natural fragment of first order logic which has a similar correspondence to MLR_2 interpreted on local squares as L_2 has to MLR_2 interpreted on squares. We find the answer by looking at the so called guarded fragments introduced in [2, 3, 14]. The guarded fragment is defined as follows:

Definition 2.4 *The* **atomic formulas** *are of the usual form:*

1. $v_1 = v_2$*, for variables or constants* v_1, v_2.
2. $Pv_1 \cdots v_n$*, for* n*-ary* P*, and variables or constants* $v_1, \ldots, v_n$.

Guarded formulas are defined by induction as follows.

1. *An atomic formula is a guarded formula.*
2. *If* φ, ψ *are guarded formulas, then* $\varphi \wedge \psi$ *and* $\neg\varphi$ *are guarded formulas.*
3. *If* $\bar{v}$ *is a finite non-empty sequence of variables,* ψ *is a guarded formula, and* G *is an atomic formula such that* $free(\psi) \subseteq free(G)$*, then* $\exists \bar{v}(G \wedge \psi)$ *is a guarded formula.*

The atomic formula G which appears in the above clause (3) is called the *guard*. The guarded fragment restricted to n variables and at most n-ary relation symbols is denoted by GF_n.
Obviously, we have to restrict the guarded fragment to two variables and at most binary relation symbols for our corresponding language. As we will show in section 3, the full guarded fragment with two variables is still too strong. We define the weak guarded fragment with two variables (notation WGF_2) as the subset of GF_2–formulas in which

1. no constants occur,

2. no generalized quantifier of the form $\exists v_0 v_1$ occur, and

3. every guard is of the form Rv_0v_1 or Rv_1v_0 for an arbitrary relation symbol R.

Theorem 2.5 *There exist linear translations from WGF_2 to MLR_2 and vice–versa which preserve satisfiability when MLR_2 is interpreted on local squares.*

The important point of this theorem for our purposes here is that any decision algorithm we design for MLR_2/LSQ will be a decision procedure for WGF_2 as well. As will be clear from the given translation, the computational costs involved are minimal.
Proof: We first provide the translation from WGF_2 into MLR_2. We assume that all relation symbols occurring in WGF_2 formulas are binary (as noted above, we can replace the unary predicates by binary ones without loss of satisfiability). The translation is just the one given above. We claim that for every WGF_2 formula φ, φ is satisfiable iff φ^t is satisfiable on a local square. From left to right is immediate: If (D, I) satisfies φ, then the (square) model $(D \times D, v)$ where $I(R) = v((Rv_0v_1)^t)$ satisfies φ^t. For the other direction, let $\mathcal{M} = (W, v)$ satisfy φ at (a_0, a_1). Let $W \subseteq U \times U$. We claim that the first order model $\mathcal{M}'$ with domain U and the interpretation I defined by $I(R) = v((Rv_0v_1)^t)$ satisfies φ at the assignment which sends v_i to a_i. The only interesting case in the inductive proof is the one for the quantifier. Our induction hypothesis is that for all $(c_0, c_1) \in W$, for all φ, $\mathcal{M}, (c_0, c_1) \models \varphi^t$ iff $\mathcal{M}' \models \varphi[v_i/c_i]$.
So let $\mathcal{M}' \models \exists v_0(Rv_0v_1 \wedge \varphi)[v_i/c_i]$. Then there exists a $c_2 \in U$ such that $\mathcal{M}' \models (Rv_0v_1 \wedge \varphi)[v_0/c_2, v_1/c_1]$. Thus $(c_2, c_1) \in I(R)$. This can only be the case if $(c_2, c_1) \in v(R^t)$, so $(c_2, c_1) \in W$. But then the induction hypothesis applies and we obtain our desired result. The other direction is proved similarly.

The translation in the other direction is given as follows. For ease of exposition we assume that $\otimes$ is only applied to propositional variables (this restriction can easily be lifted).

$$
\begin{aligned}
&(p)^t = Pv_0v_1\\
&(\otimes p)^t = Pv_1v_0\\
&(id)^t = v_0 = v_1\\
&(\Phi\varphi)^t = \exists v_1(Wv_0v_1 \wedge Wv_1v_0 \wedge Wv_1v_1 \wedge Wv_0v_0 \wedge \varphi^t)\\
&(\ominus\varphi)^t = \exists v_0(Wv_0v_1 \wedge Wv_1v_0 \wedge Wv_1v_1 \wedge Wv_0v_0 \wedge \varphi^t),
\end{aligned}
$$

and $(\cdot)^t$ commutes with the booleans. Here W is a fixed new predicate symbol.
Clearly the translation goes to WGF_2. We claim that for every MLR_2 formula φ, φ is locally square satisfiable iff $\varphi^t \wedge Wv_0v_1 \wedge Wv_1v_0 \wedge Wv_1v_1 \wedge Wv_0v_0$ is satisfiable. The direction from left to right is obvious: interpret the new predicate symbol W as the domain of the locally square model. The other direction is easy too: take the local square model whose universe is the smallest local square containing the interpretation of the predicate W. ∎

2.3 Simulating other modal logics in MLR_2

One can use MLR_2 to simulate other logics, among them the modal logics (multi-) K, KT, KB and KTB, and their tensed versions. Special properties of accessibility$\mathcal{R}$ can be encoded into the translation of the diamond operator. The advantage is that we have a single proof–system which works for a wide range of logics. Using the proof–system for MLR_2 is without increase in (theoretical) complexity since all other logics are pspace hard themselves.

Modal logic S5. Let $(\cdot)^*$ be the following translation from modal formulas to MLR_2 formulas:

$$p^* = p,\ (\cdot)^* \text{ commutes with the booleans, } (\Diamond)^* = \Phi\varphi^*.$$

One easily shows that for all φ, φ is **S5** satisfiable iff φ^* is satisfiable on a local square.

Modal logics K, KT, KB and KTB and their tensed versions. Now we give the translations for other modal logics. Note how we just mimic the standard translation for modal logic **K**, but without the cumbersome details of variables.

Definition 2.6 *Let $(\cdot)^{t_K}$ be the following translation from K to MLR_2:*

$$
\begin{aligned}
(p)^{t_K} &= p && \textit{for propositional variables } p\\
(\neg\varphi)^{t_K} &= \neg(\varphi)^{t_K}\\
(\varphi \vee \psi)^{t_K} &= (\varphi)^{t_K} \vee (\psi)^{t_K}\\
(\Diamond\varphi)^{t_K} &= \Phi(r \ \wedge \ \ominus\varphi^{t_K})
\end{aligned}
$$

where r is a new propositional variable, not occurring in the set of the propositional variables of K.

In order to simulate normal modal logics KT, KB or KTB where the accessibility relations have properties like reflexivity, symmetry or both reflexivity and symmetry, one only slightly has to change the translation of the diamond operator from definition 2.6. We use $\Diamond^{\smile}$ to denote the backwards looking (tense) modality having the same accessibility relation as $\Diamond$.

Definition 2.7 *Let $(\cdot)^{t_{K_t}}, (\cdot)^{t_{KB}}, (\cdot)^{t_{KT}}$ and $(\cdot)^{t_{KTB}}$ be modifications of the translation from K to MLR_2, corresponding exactly to $(\cdot)^K$ for all but the diamond operator and with the following translations for K_t-, KT-, KB- and KTB-diamonds $\Diamond$:*

$$\begin{array}{lcl}
(\Diamond^{\smile}\varphi)^{t_{K_t}} & = & \Phi(\otimes r \wedge \ominus\, \varphi^{t_{K_t}}) \\
(\Diamond\varphi)^{t_{KT}} & = & \Phi((r \vee id) \wedge \ominus\, \varphi^{t_{KT}}) \\
(\Diamond\varphi)^{t_{KB}} & = & \Phi(r \wedge \otimes r \wedge \ominus\, \varphi^{t_{KB}}) \\
(\Diamond\varphi)^{t_{KTB}} & = & \Phi(\ominus\, \varphi^{t_{KTB}}).
\end{array}$$

The brackets $(\cdot)$ will be omitted whenever no confusion is likely to ensue. The translations preserve satisfiability as will be shown in the following theorem.

Theorem 2.8 *For all K-formulas φ holds: $\models_{K^*} \varphi \Leftrightarrow \models_{LSQ} id \rightarrow \varphi^{t_{K^*}}$, where K^* is one of the logics K, K_t, KT, KB or KTB.*

The idea of the proof is to construct a locally square model $\mathcal{M} = (W, v)$ satisfying $id \wedge \varphi$ from a given K^*-model $\mathcal{M}^{K^*} = (W_{K^*}, \mathcal{R}, v_{K^*})$ for φ and vice versa. The most interesting part of the proof of theorem 2.8 is the proof for the translation of KTB-formulas into LC_2, because of the explicit use of the local square property. The construction of the models in the other cases is similar.

Proof:($\Rightarrow$) **(KTB)** Given a locally square model $\mathcal{M}$, we construct a new KTB-model $\mathcal{M}^{KTB} = (W_{KTB}, \mathcal{R}, v_{KTB})$ where the universe $W_{KTB} = \{u \mid \langle u, u\rangle \in W\}$ is the set of all points on the diagonal in W, and with the following valuation $v_{KTB}(p) = \{u \mid \langle u, u\rangle \in v(p)\}$. The new accessibility relation $\mathcal{R}$ is defined by $\mathcal{R}(a, b) \;\Leftrightarrow\; \langle a, b\rangle \in W$ for all a and b in W_{KTB}, which forces $\mathcal{R}$ to be reflexive and symmetric. It remains to show, that for this model and for all K-formulas φ and pairs $\langle a, a\rangle \in W_{LC_2}$: $\mathcal{M}, \langle a, a\rangle \models_{LC_2} \varphi^{t_{KTB}} \Leftrightarrow \mathcal{M}^{KTB}, a \models_{KTB} \varphi$, which is done by induction over the structure of the formula φ.
($\Leftarrow$) Given a KTB-model $\mathcal{M}^{KTB}$, we construct $\mathcal{M}' = (W', v')$, where the universe $W' = \mathcal{R}$ is the set of all elements related by $\mathcal{R}$, which is by the reflexivity and symmetry of $\mathcal{R}$ local cubic. The valuation $v'(p) = \{\langle a, a\rangle \mid$

$a \in v_{KTB}(p)\}$ maps all propositional variables p to the set of diagonal points of W' corresponding to the elements mapped by $v_{KTB}(p)$. For this model, we prove for all $a \in W_{KTB}$ and all K-formulas φ, that $\mathcal{M}^{KTB}, a \models_{KTB} \varphi \Leftrightarrow \mathcal{M}', \langle a, a\rangle \models \varphi^{t_{KTB}}$ by induction on the structure of the formula φ. ■

Description logics. If one uses different new propositional variables r_i it is possible to simulate multi-modal logic K^{multi} within the same translational framework. For each different diamond $\Diamond_i$ the translation function $(\cdot)^{t_{multiK}}$ is defined by:

$$(\Diamond_i \varphi)^{t_{multiK}} = \Phi(r_i \wedge \ominus\, \varphi^{t_{multiK}})$$

the intuition of which is shown in Figure 2 for two different Diamond formulas $\Diamond_1\varphi$ and $\Diamond_2\psi$, that are K-satisfiable at a world $a \in W_K$ and the translated MLR_2-formulas $(\Diamond_1\varphi)^{t_{multiK}}$ and $(\Diamond_2\psi)^{t_{multiK}}$ are MLR_2-satisfiable in $\langle a, a\rangle \in W$. This shows that MLR_2 is also well suited to perform proofs in description logics.

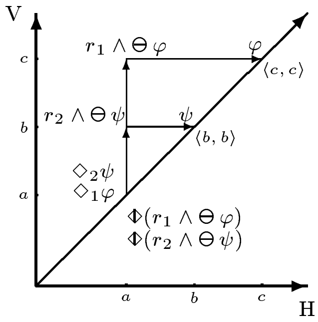

Figure 2: Simulation of multi-K with two diamonds $\Diamond_1$ and $\Diamond_2$ by MLR_2.

3 Complexity

[10] shows that the satisfiability problem of MLR_2 interpreted on local squares is pspace complete. As a corollary of the above translation we obtain that the WGF_2 fragment is in pspace as well. In fact it is pspace complete since the range of the standard translation of modal logic **K** is still inside that fragment. We restricted GF_2 in three different ways. We now show that each of them is necessary for this complexity bound. In other words, relaxing any one of these restrictions leads to an exptime hard satisfaction problem. We shall reduce the EXPTIME-complete *global* **K**-satisfaction problem for unimodal languages. The global **K**-satisfaction problem for unimodal languages is this: given a formula φ in the unimodal language, does there exist a Kripke model $\mathcal{M}$ such that $\mathcal{M} \models \varphi$ (in other words, where φ is true in *all* worlds)? The EXPTIME-completeness of this problem is an easy

consequence of (the proof of) the EXPTIME-completeness of modal logic **K** expanded with the universal modality in [13].

Adding constants. We just need to add one single constant c to obtain an exptime hard satisfaction problem. For this we modify the standard translation as follows:

$$ST_x(\Diamond\varphi) = \exists y(Rxy \land Rcy \land ST_y(\varphi)).$$

It is not hard to show that φ is globally **K** satisfiable iff $\exists y Rcy \land \forall y(Rcy \to ST_y(\varphi))$ is satisfiable. The idea of the proof is to add a new world to the Kripke model, name it c and connect all old worlds to c by R. This use of a world named by a constant as a "spy-point" originates with [5]. For a complete proof, see [4].

Allowing arbitrary guards. Let φ be a formula and W a fresh unary predicate (that is not corresponding to any proposition letter in φ). Modify the $\Diamond$–clause in the standard translation as follows:

$$ST_x(\Diamond\varphi) = \exists y(Rxy \land Wy \land ST_y(\varphi)).$$

Then φ is globally **K** satisfiable iff $\exists x Wx \land \forall x(Wx \to ST_x(\varphi))$ is satisfiable, as is easy to show.

Allowing generalized quantifiers. In this case use the following translation:

$$ST_x(\Diamond\varphi) = \exists y(Rxy \land ST_y(\varphi) \land \exists x(Wxy)).$$

Then φ is globally **K** satisfiable iff $\exists xy Wxy \land \forall xy(Wxy \to x = y) \land \forall xy(Wxy \to ST_x(\varphi))$ is satisfiable. The proof is an obvious modification of the previous one.

4 Dimensions higher than two

If we consider MLR_n for n larger than 2, we obtain a modal analogue of first order logic restricted to n variables and n-ary relation symbols. The obvious extensions of the translations and modal classes apply, viz [11]. The semantic analogue of the local squares is the class of local cubes, satisfying

$$(s_0, s_1, \ldots, s_{n-1}) \in W \Rightarrow {}^n\{s_0, s_1, \ldots, s_{n-1}\} \subseteq W,$$

i.e., the domains of the models are closed under all permutations and substitutions. MLR_n for n at least 3 has an undecidable satisfaction problem when interpreted on the class of cubic models with universes of the form ${}^nU$. This follows simply from similar results in first order logic (see [9] for

a strong result). When we interpret MLR_n on local cubes, the satisfaction problem becomes exptime complete [10]. In fact, an expansion of MLR_n can be used to decide the guarded fragment, as is shown in [10].
The jump in complexity coincides with a much harder to prove finite model property. In two dimensions this is an immediate result of the termination of the tableaux system presented below. The obtained model has several nice "tree–like" properties comparable to the models one obtains by a standard tableaux system for modal **K**. For higher dimensions this tree–like property is lost. Stated simply, the situation is comparable with **S4** where one also has to re–use worlds to obtain finite models. In fact the proof for fmp requires quite heavy machinery from finite model theory [1]. This fact has its repercussions when one wants to design a terminating tableaux system for MLR_n or the guarded fragment. The following example indicates that a naive tableaux system which builds a model using fresh constants all the time (thereby making as little connections as possible) cannot be guaranteed to terminate.

Example. We give an example for 3 dimensions; straightforward modifications in the argument below yield the same result for any higher dimension. We define the formula φ as

$$id \wedge p\,;(p\,;\otimes q) \wedge \neg(q\,;\neg(p\,;\otimes q))$$

where $x\,;y$ is the abbreviation for $\Diamond_2(\Diamond_1(x \wedge id_{12}) \wedge \Diamond_0(y \wedge id_{02}))$.
Now we will construct a model $\mathfrak{M} = (W, v)$ for φ by a step-by-step procedure that uses fresh points at every step when we need witnesses for $\Diamond_i\psi$ formulas. Our goal is to define $\mathfrak{M} = (W, v)$ such that $s \Vdash \varphi$ (with some $s \in W$). Clearly, we can assume that $W \subseteq {}^3\omega$, and we will denote the elements of W by natural numbers. We start with a partial model $s \in {}^3\omega$, and require that φ must hold at s. To achieve this, we have to extend our partial model.
First note that $s_0 = s_1 = s_2$ by $s \Vdash id$. We denote s_0 by 0. By $(0,0,0) \Vdash p\,;(p\,;\otimes q)$, we need a fresh point 1 such that $(0,1,1) \Vdash p$ and $(1,0,1) \Vdash p\,;\otimes q$. Now $(1,0,1) \Vdash p\,;\otimes q$ implies that there is another fresh point 2 such that $(1,2,2) \Vdash p$ and $(2,0,2) \Vdash \otimes q$. Thus $(0,2,2) \Vdash q$. Recall that $(0,0,0) \Vdash \neg(q\,;\neg(p\,;\otimes q))$. Hence $(2,0,2) \Vdash p\,;\otimes q$. So there exists a new point 3 such that $(2,3,3) \Vdash p$ and $(0,3,3) \Vdash q$.
Now let us assume that we are in the kth step of the step-by-step construction and that we already defined a partial model $\mathfrak{M}_k = (W_k, v_k)$ such that W_k is the local cube generated by the set $\{(0,i,i), (i,i+1,i+1), (i,0,i+1) : i \leq k\}$ and $v_k(p) = \{(i,i+1,i+1) : i \leq k\}$ and $v_k(q) = \{(0,i+2,i+2) : i < k\}$.
Observe that $(0,k+1,k+1) \Vdash q$. Since we want $(0,0,0) \Vdash \neg(q\,;\neg(p\,;\otimes q))$, we see that $(k+1,0,k+1) \Vdash p\,;\otimes q$ must hold. Using the definition of ; and our strategy that we take a fresh point whenever we need a witness

for a $\diamond_i$-formula, we have to extend W_k by $k+2$ such that $(k+1, k+2, k+2) \Vdash p$ and $(k+2, 0, k+2) \Vdash \otimes q$. Thus we arrive at a partial model $\mathfrak{M}_{k+1} = (W_{k+1}, v_{k+1})$ such that W_{k+1} is the local cube generated by the set $\{(0,i,i), (i,i+1,i+1), (i,0,i+1) : i \leq k+1\}$ and $v_k(p) = \{(i,i+1,i+1) : i \leq k+1\}$ and $v_k(q) = \{(0,i+2,i+2) : i < k+1\}$.
Continuing the same way, we arrive at the structure $\mathfrak{M} = (W, v)$, where W is the local cube generated by $\{(0,i,i), (i,i+1,i+1), (i,0,i+1) : i < \omega\}$ and $v(p) = \{(i,i+1,i+1) : i < \omega\}$ and $v(q) = \{(0,i+2,i+2) : i < \omega\}$. It should be clear from the above argument that s satisfies φ in $\mathfrak{M}$.
We note that it is rather easy to find a finite model satisfying φ. Indeed, the model with the singleton universe $\{(0,0,0)\}$ and the valuation $v(p) = v(q) = \{(0,0,0)\}$ satisfies φ.
Our point with the above example was to show that the straightforward step-by-step (or tableau) construction does not yield finite models, and that it is not clear how to modify the model-construction so that it yields a finite model for *every* formula.

In the case of two dimensions, matters are considerably simpler. In the next section we present a simple tableaux system which does terminate.

5 LC$_2$-TAB, a decision procedure for MLR$_2$/LSQ

The tableau calculus LC_2-TAB follows roughly the semantics given in definition 2.3. In order to show for example that a branch of a tableau containing the formula $\otimes\varphi$ labelled with $\langle a, b\rangle$ closes, it is enough to show, that a branch containing φ, labelled with $\langle b, a\rangle$ closes. The tableau rule for converse formulas thus only adds $\varphi, \langle b, a\rangle, V$ to the actual branch and ticks $\otimes\varphi, \langle a, b\rangle, V$ (meaning that no more rule is to be applied on this formula). This closely corresponds to the `converse rule` in our PROLOG implementation:

```
prove(TickedFormulas,Domain,[lf(⊗φ,X,Y)|Formulas],Result) :-
     prove([(⊗φ,X,Y)|TickedFormulas],Domain,[(φ,Y,X)|Formulas],Result).
```

The predicate `prove([],[(0,0)],[lf(¬⊖φ,0,0)],Result)` succeeds with `Result = theorem` if and only if contradictions in all possible branches are detected.

5.1 The Tableaux system LC$_2$-TAB

The idea of LC_2-TAB is to start with the negated formula, theorem hood of which has to be proven, and to label it with an arbitrary point (which we will call $\langle 0,0\rangle$)[1] on the diagonal and the smallest universe containing

[1]We use the natural numbers as base for the universe which is constructed in the step-by-step approach of the calculus LC_2-TAB, because they reflect the geometrical nature of the construction in a very natural way. Due to the structure of our rules, we do only need one variable over $\mathbb{N}$, which we will call x.

this point. This universe $V = {}^2\{0\}$ is a local square. The proof consists in explicitly constructing a tableau with an open branch corresponding to a counter model for the original formula. This means that the formula φ is a theorem if and only if there is a closed tableau for $\diamondminus\neg\varphi$, labelled with $\langle 0, 0\rangle$. To be able to construct a counter model and to keep track of all existing pairs in a branch, we also label the formulas in the calculus with a set of pairs V, to construct the universe for a possible model for φ. New pairs are created if and only if formulas of the form $\diamondminus\varphi$ or $\diamondbar\varphi$ occur, where a witness pair $\langle c, b\rangle$ for the labelled formula $\diamondminus\varphi, \langle a, b\rangle, V$ has to be given, such that φ holds in $\langle c, b\rangle$ where $\langle c, b\rangle \notin V$.
If a contradiction occurs, one of the closure rules put $\perp$ in the branch. ($\checkmark$) means, that a formula (or a set of formulas) is marked *after* application of the rule and cannot be used any more. We follow Fitting [7] in his uniform notation for conjunctions and disjunctions. Fitting groups all propositional formulas of the form $X \circ Y$ and $\neg(X \circ Y)$ into two categories, those who act conjunctively, which are called α formulas and those who act disjunctively and are called β formulas. He then defines components $\alpha_1, \alpha_2, \beta_1$ and β_2 such that $\alpha \equiv \alpha_1 \wedge \alpha_2$ and $\beta \equiv \beta_1 \vee \beta_2$. The advantage of this approach is, that you only need two tableau rules for all propositional connectives: $\vee, \wedge, \rightarrow, \leftrightarrow$ and their negations.
All LC_2-TAB rules are so called extension rules, which are displayed as:

$$\begin{array}{c|c|c}
\multicolumn{3}{c}{\xi_1} \\
\multicolumn{3}{c}{\vdots} \\
\multicolumn{3}{c}{\xi_k} \\
\hline
\zeta_{11} & \cdots & \zeta_{n1} \\
\vdots & \vdots & \vdots \\
\zeta_{1m_1} & & \zeta_{nm_n}
\end{array}
\qquad \text{or} \qquad
\begin{array}{c|c}
\multicolumn{2}{c}{B} \\
\multicolumn{2}{c}{\xi} \\
\hline
B_1 & B_2 \\
\zeta_1 & \zeta_2
\end{array}$$

The left-hand side means that if there are labelled formulas $\xi_1, \ldots, \xi_k$ in the branch, then the branch will be split into n sub-branches and all labelled formulas $\zeta_{i1}, \ldots \zeta_{im_i}$ will be added to the i-th branch. n and the m_i may be 1. Since we identify the branch B with the set of labelled formulas occurring in B, the meaning of a rule of the right-hand type is, that a new set B_1 (B_2) of labelled formulas is added to the first (second) branch. These B_1 and B_2 are usually defined from B in one of the conditions of the rules. Most of the rules come with additional restrictions for their application.
To avoid cumbersome notation we use the following convention for application of the simple rules. A rule of the form $B, \varphi / B, \varphi', \varphi(\checkmark)$ will be abbreviated by $\varphi(\checkmark)/\varphi'$. The only rules which are not of this form are the diamond rules and the label rules.

Definition 5.1 *A formula φ is an LC_2-***theorem** *(denoted by $\vdash \varphi$), if and only if there is tableau proof for it, where a* **tableau proof** *for a formula φ*

in the calculus LC_2-TAB is a closed tableau starting with the labelled formula $\diamondminus(\neg\varphi), \langle 0,0\rangle, {}^2\{0\}$, *using the tableau rules stated below.*

- Propositional rules.

$(\neg)$ $\dfrac{\neg\neg Z, \langle a,b\rangle, V(\checkmark)}{Z, \langle a,b\rangle, V}$ where $a, b \in \mathbb{N} \cup \{x\}$	(α) $\dfrac{\alpha, \langle a,b\rangle, V(\checkmark)}{\begin{array}{c}\alpha_1, \langle a,b\rangle, V\\ \alpha_2, \langle a,b\rangle, V\end{array}}$ where $a, b \in \mathbb{N} \cup \{x\}$	(β) $\dfrac{\beta, \langle a,b\rangle, V(\checkmark)}{\beta_1, \langle a,b\rangle, V \mid \beta_2, \langle a,b\rangle, V}$ where $a, b \in \mathbb{N}$

- Closing rules

(id_a) $\dfrac{\neg id, \langle a,a\rangle, V(\checkmark)}{\bot}$ where $a \in \mathbb{N}$	(id_b) $\dfrac{id, \langle a,b\rangle, V(\checkmark)}{\bot}$ where $a, b \in \mathbb{N}$ and $a \neq b$	(con) $\dfrac{\begin{array}{c}\neg\varphi, \langle a,b\rangle, V(\checkmark)\\ \varphi, \langle a,b\rangle, V(\checkmark)\end{array}}{\bot}$ where $a, b \in \mathbb{N}$

- Transition rules

$(\otimes)$ $\dfrac{\otimes\varphi, \langle a,b\rangle, V(\checkmark)}{\varphi, \langle b,a\rangle, V}$ where $a, b \in \{x\} \cup \mathbb{N}$	$(\neg\otimes)$ $\dfrac{\neg\otimes\varphi, \langle a,b\rangle, V(\checkmark)}{\neg\varphi, \langle b,a\rangle, V}$ where $a, b \in \{x\} \cup \mathbb{N}$
$(\obar)$ $\dfrac{\obar\,\varphi, \langle a,b\rangle, V(\checkmark)}{\varphi, \langle a,a\rangle, V}$ where $b \in \{x\} \cup \mathbb{N}, a \in \mathbb{N}$	$(\neg\obar)$ $\dfrac{\neg\obar\,\varphi, \langle a,b\rangle, V(\checkmark)}{\neg\varphi, \langle a,a\rangle, V}$ where $b \in \{x\} \cup \mathbb{N}, a \in \mathbb{N}$
$(\diamondbar)$ $\dfrac{\begin{array}{c}B(\checkmark)\\ \diamondbar\varphi, \langle a,b\rangle, V(\checkmark)\end{array}}{\begin{array}{c\|c}B & B'\\ \varphi, \langle a,a\rangle, V & \varphi, \langle a,c\rangle, V'\end{array}}$	$(\boxbar)$ $\dfrac{\boxbar\varphi, \langle a,b\rangle, V(\checkmark)}{\varphi, \langle a,x\rangle, V}$ where $a \in \mathbb{N}$, $\;b \in \mathbb{N} \cup \{x\}$

The $(\diamondbar)$-rule can only be applied under the conditions that:

- B is the set of all labelled formulas occurring in the branch,
- $a \in \mathbb{N}$, $\;b \in \mathbb{N} \cup \{x\}$, c is a **new** constant, i.e. $\langle c,c\rangle \notin V$,
- $V' = V \cup \{\langle c,a\rangle, \langle a,c\rangle, \langle c,c\rangle\}$,
- there is no pair $\langle a,d\rangle \in V$, such that $\varphi, \langle a,d\rangle, V \in B$,
- $B' = \{\psi, \langle x,y\rangle, V' \mid \psi, \langle x,y\rangle, V \in B\} \;\cup\; \{\diamondbar\varphi, \langle a,b\rangle, V'(\checkmark)\}$.

By definition of $\boxbar$ (and $\boxminus$), the ($\boxbar$)-rule (and ($\boxminus$)-rule) correspond to the negation of the diamond rules ($\diamondbar$)-rule (and ($\diamondminus$)-rule). There are corresponding rules for the horizontal $\diamondminus, \boxminus$ and $\ominus$ modalities.

- Label rules (lr_1) and (lr_2)

$(lr_1)\ \dfrac{\begin{array}{c}B\\ \varphi, \langle x, b\rangle, V\end{array}}{\begin{array}{c}B\\ \varphi, \langle a, b\rangle, V\end{array}}$	$(lr_2)\ \dfrac{\begin{array}{c}B\\ \varphi, \langle a, x\rangle, V\end{array}}{\begin{array}{c}B\\ \varphi, \langle a, b\rangle, V\end{array}}$

where $a, b \in \mathbb{N}$, $\langle a, b\rangle \in V$ and $\varphi, \langle a, b\rangle, V \notin B$.

The rules (lr_1) and (lr_2) are the only rules in the tableau, where the original formulas $\varphi, \langle x, b\rangle, V$ and $\varphi, \langle a, x\rangle, V$ cannot be ticked off, i.e. they can be applied as long as there are pairs in V, that have not been used yet.

It is worth mentioning, that the variable x in a branch is independent of all other occurrences in any other branch. This is by definition of the branching rules (β), ($\diamondminus$) and ($\diamondbar$). The definition of the diamond-rules blows the tableau up in a way, that all labelled formulas of a branch are copied into every new branch and no labelled formulas are deleted. But in practice (that is, in the PROLOG implementation), the formulas of the old branch are replaced by the updated labelled formulas. Even more, since the universes of all newly created labelled formulas are all the same, the domain of all these formulas is represented as only one global set of pairs.

5.2 A documented sample proof

We will construct (the open branch of) a proof in LC_2-TAB to illustrate the use of the rules. We prove, that the formula $\diamondbar\diamondminus\varphi \to \diamondminus\diamondbar\varphi$ is not a theorem of MLR_2/LSQ by showing, that there is an open branch for the tableau, starting with the negation of the formula. For reasons of readability, we only do the proof for a point on the diagonal at $\langle 0, 0\rangle$, which is sufficient to find a counter model.
We present a complete calculation until we reach a diamond formula, which forces us to branch and update the universe of the formulas in the right-hand branch. The numbers and letters in parenthesis are not part of the calculus but are needed to make it easier to talk about the example. ($\checkmark_{(i)}$) means, that a formula is ticked after application of a rule creating the new formula (i). In addition, we give a brief explanation, why we created a new formula,

on the right hand side of the tableau.

(1)	$\neg(\mathrlap{\,\mid}\Diamond\mathrlap{-}\Diamond\varphi \to \mathrlap{-}\Diamond\mathrlap{\,\mid}\Diamond\varphi), \langle 0,0\rangle, {}^2\{0\}(\checkmark_{(2)(3)})$	Negated input
(2)	$\mathrlap{\,\mid}\Diamond\mathrlap{-}\Diamond\varphi, \langle 0,0\rangle, {}^2\{0\}(\checkmark_{(5a)(5b)})$	(α) on (1)
(3)	$\neg\mathrlap{-}\Diamond\mathrlap{\,\mid}\Diamond\varphi, \langle 0,0\rangle, {}^2\{0\}(\checkmark_{(4)})$	(α) on (1)
(4)	$\neg\mathrlap{\,\mid}\Diamond\varphi, \langle x,0\rangle, {}^2\{0\}(\checkmark_{(5a)(5b)})$	$(\boxminus)$ on (3)
$(5a)$ $\mathrlap{-}\Diamond\varphi, \langle 0,0\rangle, {}^2\{0\}$	$(5b)$ $\mathrlap{-}\Diamond\varphi, \langle 0,1\rangle, {}^2\{0,1\}$	$(\mathrlap{\,\mid}\Diamond)$ on (2)
$(1a)$	$(1b)$	copies of (1)
$\vdots$	$\vdots$	

In the left branch (a), exact copies of the labelled formulas (1) to (4) are made. In the right-hand branch (b), the universes of (1b) to (4b) have been updated according to the definition of the $(\mathrlap{\,\mid}\Diamond)$-rule. The new important formulas of the branches are (1a) to (5a) and (1b) to (5b). We only proceed with the right-hand branch (b).[2] After branching on (5b), the left branch (c) closes with a contradiction on $\langle 1,1\rangle$. But the second branch can be fully expanded, until no further rule can be applied.

$(1d)$	$\neg(\mathrlap{\,\mid}\Diamond\mathrlap{-}\Diamond\varphi \to \mathrlap{-}\Diamond\mathrlap{\,\mid}\Diamond\varphi), \langle 0,0\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}(\checkmark)$	copy of $(1b)$
$\vdots$	$\vdots$	$\vdots$
$(6d)$	$\varphi, \langle 2,1\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	
$(7d)$	$\neg\mathrlap{\,\mid}\Diamond\varphi, \langle 0,0\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}(\checkmark_{(8d)})$	(lr_1) on $(4d)$
$(8d)$	$\neg\varphi, \langle 0,x\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	(◫) on $(7c)$
$(9d)$	$\neg\varphi, \langle 0,0\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	(lr_2) on $(8c)$
$(10d)$	$\neg\varphi, \langle 0,1\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	(lr_2) on $(8c)$
$(11d)$	$\neg\mathrlap{\,\mid}\Diamond\varphi, \langle 1,0\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}(\checkmark_{(12d)})$	(lr_1) on $(4d)$
$(12d)$	$\neg\varphi, \langle 1,x\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	(◫) on $(11c)$
$(13d)$	$\neg\varphi, \langle 1,0\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	(lr_2) on $(12c)$
$(14d)$	$\neg\varphi, \langle 1,1\rangle, {}^2\{0,1\} \cup {}^2\{1,2\}$	(lr_2) on $(12c)$

This proves that there is a model, that satisfies $\neg(\mathrlap{\,\mid}\Diamond\mathrlap{-}\Diamond\varphi \to \mathrlap{-}\Diamond\mathrlap{\,\mid}\Diamond\varphi)$ such that $\mathrlap{\,\mid}\Diamond\mathrlap{-}\Diamond\varphi \to \mathrlap{-}\Diamond\mathrlap{\,\mid}\Diamond\varphi$ is not a theorem of MLR_2/LSQ. The model $\mathcal{M} = ({}^2\{0,1\} \cup {}^2\{1,2\}, v)$, where $v(\varphi) = \{\langle 2,1\rangle\}$, is the counter model, which has been constructed in the proof and which is shown in figure 3. The reader interested in implementations might want to compare the proof and the counterexample given here with the one generated by our PROLOG program.

5.3 Soundness, Completeness and Termination

LC_2-TAB is sound and complete with respect to the logic MLR_2/LSQ and terminates. The idea behind the soundness proof is, that a formula φ is not valid, if and only if $\neg\mathrlap{-}\Diamond\varphi$ is satisfiable at an arbitrary pair $\langle 0,0\rangle$ on the diagonal. It is easily verified that all rules preserve satisfiability. For completeness, we construct a model for a formula from an open branch of the tableau, where we use the set V of pairs, which we construct during

[2] Bot sub-branches of (a) close by contradiction on pairs of the row $\langle x,0\rangle$.

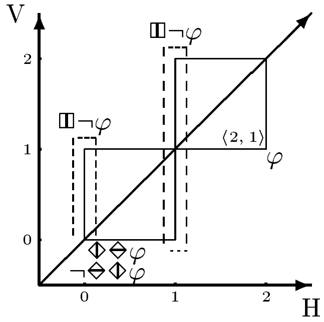

Figure 3: A counter model

application of the rules. The completeness proof is a straightforward induction on the formulas occurring in the labels. Termination guarantees that the procedure builds a model for every satisfiable formula. Termination is the only property which is a bit harder to show. The procedure terminates because the complexity of the formulas in the tableaux decreases along the diagonal. Full proofs are available in [12].

6 Implementation

The system `lc2` is a tableau theorem prover for the logic MLR_2/LSQ. The implementation method applied in `lc2` is a simple subgoaling method. This means, that we prove satisfiability of a (labelled) formula φ by proving satisfiability for one or more formulas that have been created according to the rules of LC_2-TAB. The core of `lc2` is the Prolog program `lc2.pl` using a depth-first search method to look for an open branch in the tableau, which is constructed by application of all possible rules on the set of actual formulas. If a contradiction is found in the branch, the search stops on it and is continued on a new branch. If a branch is fully evaluated, i.e. that no rule can be applied, and no contradiction has been detected, the search stops and a counter model has been found. With this method no backtracking is necessary.

The easiest way to use `lc2` is to fill out the form on the home page of LC_2-TAB. The contents of the form are sent to a central server, which activates the corresponding interface program and then `lc2` itself. The answer is immediately sent back to the user. The user can ask for satisfiability of MLR_2 formulas but also for satisfiability of modal formulas in any of the modal logics discussed in Section 2.3. In the last case they are automatically translated. In case the formula is satisfiable, the system provides a model in the respective class of frames.

The URL of the document is

`http://www.dcs.kcl.ac.uk/cgi-bin/LC2/lc2.cgi`.

More information and examples can be found following the links on that page.

Further research

The most pressing research question is whether a simple tableaux system for MLR_n interpreted on local cubes can be designed for any n. As indicated here it is not possible to simply extend the system LC_2-TAB if termination is required. It seems feasible to us to turn the mosaic deletion algorithm from [10] into a tableaux system, but we do not dare guess about its user–friendliness or even its behavior when implemented.

References

[1] H. Andréka, I. Hodkinson, and I. Németi. Finite algebras of relations are representable on finite sets. *Journal of Symbolic Logic*, 1996. Accepted for publication.

[2] H. Andréka, J. van Benthem, and I. Németi. Back and forth between modal logic and classical logic. *Bulletin of the Interest Group in Pure and Applied Logics*, 3(3):685–720, 1995.

[3] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[4] C. Areces, P. Blackburn, and M. Marx. Hybrid logics. Characterization, interpolation and complexity. Technical report, Institute for Logic, Language and Computation, University of Amsterdam, 1999.

[5] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.

[6] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Springer Verlag, 1997.

[7] M. Fitting. *First Order Logic and Automated Theorem Proving.* Springer Verlag, 1990.

[8] E. Grädel. On the restraining power of guards. manuscript, June, 1997.

[9] R. Maddux. The equational theory of CA_3 is undecidable. *Journal of Symbolic Logic*, 45(2):311–316, 1980.

[10] M. Marx. Complexity of modal logics of relations. Technical Report ML–97–02, Institute for Logic, Language and Computation, University of Amsterdam, May 1997.

[11] M. Marx and Y. Venema. *Multi-dimensional Modal Logic.* Applied Logic Series. Kluwer Academic Publishers, 1997.

[12] S. Schlobach and M. Marx. Tableaux system for modal logic of relations. Technical report, Department of Computing, Imperial College, London, UK, 1997.

[13] E. Spaan. *Complexity of modal logics.* PhD thesis, University of Amsterdam, Institute for Logic, Language and Computation, 1993.

[14] J. van Benthem. *Exploring Logical Dynamics.* Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996.