Verifying the Modal Logic Cube is an Easy Task (for Higher-Order Automated Reasoners)

Christoph Benzmüller*

Articulate Software, Angwin, CA, U.S.

Dedicated to Christoph Walther

Abstract. Prominent logics, including quantified multimodal logics, can be elegantly embedded in simple type theory (classical higher-order logic). Furthermore, off-the-shelf reasoning systems for simple type type theory exist that can be uniformly employed for reasoning *within* and *about* embedded logics. In this paper we focus on reasoning *about* modal logics and exploit our framework for the automated verification of inclusion and equivalence relations between them. Related work has applied first-order automated theorem provers for the task. Our solution achieves significant improvements, most notably, with respect to elegance and simplicity of the problem encodings as well as with respect to automation performance.

1 Introduction

Church's simple type theory $\mathcal{STT}$ [15], also known as classical higher-order logic, has many prominent classical logic fragments, including propositional logic, first-order logic, and second-order logic. Interestingly, also well known non-classical logics, including propositional and quantified multimodal logics, can be elegantly embedded in $\mathcal{STT}$ [9, 6].

In this paper we exploit our embedding of quantified multimodal logic in $\mathcal{STT}$ [6] for the automated verification of inclusion relations between prominent propositional modal logics, including the logics **K**, **M** (also known as **T**), **D**, **S4**, and **S5**. Concretely, we analyze inclusion and equivalence relations for modal logics that can be defined from normal modal logic **K** by adding (combinations of) the axioms M, B, D, 4, and 5. In our problem encodings we exploit the well known correspondences of these axioms to semantic properties of accessibility relations. These correspondences can itself be elegantly formalized and effectively automated in our approach.

The automation of $\mathcal{STT}$ currently experiences a renaissance that has been fostered by the recent extension of the successful TPTP infrastructure for first-order logic [26] to higher-order logic, called TPTP THF [27, 11]. In our verification study we exploit this new infrastructure and work with different TPTP THF

* This work has been supported by the German Research Foundation (DFG) under grant BE 2501/6-1.

compliant automated higher-order reasoning systems: TPS [1], LEO-II[1] [10], Satallax [5], IsabelleP[2], Refute [28] and Nitpick [13].[3] TPS, LEO-II and IsabelleP are automated theorem provers, and Refute and Nitpick are countermodel generators. Satallax is an automated theorem prover with additional capabilities for finding countermodels.

Related work [23] has applied first-order automated theorem provers for the verification of inclusion relations between modal logics. Our solution achieves significant improvements, most notably, with respect to elegance and simplicity of the problem encodings as well as with respect to automation performance.

In Sect. 2 we outline our embedding of quantified multimodal logics in $\mathcal{STT}$ (this part is reproduced from [6]). In Sect. 3 we describe how reasoning *about* propositional modal logics and their inclusion relations is facilitated in our approach. The results of our experiments are presented in Sect. 4, and Sect. 5 concludes the paper.

2 (Normal) Quantified Multimodal Logics in $\mathcal{STT}$

$\mathcal{STT}$ [15] is based on the simply typed λ-calculus. The set $\mathcal{T}$ of simple types is usually freely generated from a set of basic types $\{o, \iota\}$ (where o is the type of Booleans and ι is the type of individuals) using the right-associative function type constructor $\rightarrow$. Instead of $\{o, \iota\}$ we here consider a set of base types $\{o, \iota, \mu\}$, providing an additional base type μ (the type of possible worlds).

The simple type theory language $\mathcal{STT}$ is defined by (where α, β, $o \in \mathcal{T}$):

$$s, t ::= p_\alpha \mid X_\alpha \mid (\lambda X_\alpha . s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta \mid (\neg_{o\rightarrow o}\, s_o)_o \mid$$
$$(s_o \vee_{o\rightarrow o\rightarrow o} t_o)_o \mid (s_\alpha =_{\alpha\rightarrow\alpha\rightarrow o} t_\alpha)_o \mid (\Pi_{(\alpha\rightarrow o)\rightarrow o}\, s_{\alpha\rightarrow o})_o$$

p_α denotes typed constants and X_α typed variables (distinct from p_α). Complex typed terms are constructed via abstraction and application. Our logical connectives of choice are $\neg_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o}$, $=_{\alpha\rightarrow\alpha\rightarrow o}$ and $\Pi_{(\alpha\rightarrow o)\rightarrow o}$ (for each type α).[4] From these connectives, other logical connectives can be defined in the usual way (e.g., $\wedge$ and $\Rightarrow$). We often use binder notation $\forall X_\alpha . s$ for $\Pi_{(\alpha\rightarrow o)\rightarrow o}(\lambda X_\alpha . s_o)$. We assume familiarity with α-conversion, β- and η-reduction, and the existence of β- and $\beta\eta$-normal forms. Moreover, we obey the usual definitions of free variable occurrences and substitutions.

The semantics of $\mathcal{STT}$ is well understood and thoroughly documented in the literature [2, 3, 7, 20]. The semantics of choice for our work is Henkin semantics.

[1] LEO-II integrates the first-order automated theorem prover E [24].

[2] IsabelleP applies a series of Isabelle/HOL [22] proof tactics in batch mode.

[3] Refute and Nitpick, which also belong to the Isabelle/HOL proof assistant, are sometimes called IsabelleM and IsabelleN; this is the case, for example, in the SystemOnTPTP tool `http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP`, where all reasoning systems mentioned here are available online.

[4] This choice is not minimal (from $=_{\alpha\rightarrow\alpha\rightarrow o}$ all other logical constants can already be defined [4]). It useful though in the context of resolution based theorem proving.

Quantified modal logics have been studied by Fitting [16] (further related work is available by Blackburn and Marx [12] and Braüner [14]). In contrast to Fitting we are here not interested only in **S5** structures but in the more general case of **K** from which more constrained structures (such as **S5**) can be easily obtained. First-order quantification can be constant domain or varying domain. Below we only consider the constant domain case, in which every possible world has the same domain. Like Fitting, we keep our definitions simple by not having function or constant symbols. While Fitting [16] studies quantified monomodal logic, we are interested in quantified multimodal logic. Hence, we introduce multiple $\Box_r$ operators for symbols r from an index set S. The grammar for our quantified multimodal logic $\mathcal{QML}$ is

$$s, t ::= P \mid k(X^1, \ldots, X^n) \mid \neg\, s \mid s \vee t \mid \forall X_\blacksquare s \mid \forall P_\blacksquare s \mid \Box_r\, s$$

where $P \in \mathcal{PV}$ denotes propositional variables, $X, X^i \in \mathcal{IV}$ denote first-order (individual) variables, and $k \in \mathcal{SYM}$ denotes predicate symbols of any arity. Further connectives, quantifiers, and modal operators can be defined as usual.

Fitting introduces three different notions of Kripke semantics for $\mathcal{QML}$: $\mathbf{QS5}\pi^-$, $\mathbf{QS5}\pi$, and $\mathbf{QS5}\pi^+$. In our work [8] we study related notions $\mathbf{QK}\pi^-$, $\mathbf{QK}\pi$, and $\mathbf{QK}\pi^+$ for a modal context **K**, and we support multiple modalities.

$\mathcal{STT}$ is an expressive logic and it is thus not surprising that $\mathcal{QML}$ can be elegantly modeled and even automated as a fragment of $\mathcal{STT}$. The idea of the encoding, called $\mathcal{QML}^{STT}$, is simple. Choose type ι to denote the (non-empty) set of individuals and choose the second base type μ to denote the (non-empty) set of possible worlds. As usual, the type o denotes the set of truth values. Certain formulas of type $\mu \to o$ then correspond to multimodal logic expressions. The multimodal connectives $\neg$, $\vee$, and $\Box$, become λ-terms of types $(\mu \to o) \to (\mu \to o)$, $(\mu \to o) \to (\mu \to o) \to (\mu \to o)$, and $(\mu \to \mu \to o) \to (\mu \to o) \to (\mu \to o)$, respectively.

Quantification is handled as in $\mathcal{STT}$ by modeling $\forall X_\blacksquare p$ as $\Pi(\lambda X_\blacksquare p)$ for a suitably chosen connective Π. Here we are interested in defining two particular modal $\boldsymbol{\Pi}$-connectives: $\boldsymbol{\Pi}^\iota$, for quantification over individual variables, and $\boldsymbol{\Pi}^{\mu\to o}$, for quantification over modal propositional variables that depend on worlds. They become terms of type $(\iota \to (\mu \to o)) \to (\mu \to o)$ and $((\mu \to o) \to (\mu \to o)) \to (\mu \to o)$ respectively.

The $\mathcal{QML}^{STT}$ modal operators $\neg, \vee, \Box, \boldsymbol{\Pi}^\iota$, and $\boldsymbol{\Pi}^{\mu\to o}$ are now simply defined as follows:

$$\begin{aligned}
\neg_{\,(\mu\to o)\to(\mu\to o)} &= \lambda\phi_{\mu\to o\blacksquare}\lambda W_{\mu\blacksquare}\neg\phi\, W \\
\vee_{\,(\mu\to o)\to(\mu\to o)\to(\mu\to o)} &= \lambda\phi_{\mu\to o\blacksquare}\lambda\psi_{\mu\to o\blacksquare}\lambda W_{\mu\blacksquare}\phi\, W \vee \psi\, W \\
\Box_{\,(\mu\to\mu\to o)\to(\mu\to o)\to(\mu\to o)} &= \lambda R_{\mu\to\mu\to o\blacksquare}\lambda\phi_{\mu\to o\blacksquare}\lambda W_{\mu\blacksquare}\forall V_{\mu\blacksquare}\neg R\, W\, V \vee \phi\, V \\
\boldsymbol{\Pi}^\iota_{(\iota\to(\mu\to o))\to(\mu\to o)} &= \lambda\phi_{\iota\to(\mu\to o)\blacksquare}\lambda W_{\mu\blacksquare}\forall X_{\iota\blacksquare}\phi\, X\, W \\
\boldsymbol{\Pi}^{\mu\to o}_{((\mu\to o)\to(\mu\to o))\to(\mu\to o)} &= \lambda\phi_{(\mu\to o)\to(\mu\to o)\blacksquare}\lambda W_{\mu\blacksquare}\forall P_{\mu\to o\blacksquare}\phi\, P\, W
\end{aligned}$$

Note that our encoding actually only employs the second-order fragment of $\mathcal{STT}$ enhanced with lambda-abstraction.

Further operators can be introduced as usual, for example, $\top = \lambda W_\mu \centerdot \top$, $\bot = \neg\top$, $\wedge = \lambda\phi,\psi \centerdot \neg(\neg\phi \vee \neg\psi)$, $\supset = \lambda\phi,\psi \centerdot \neg\phi \vee \psi$, $\Leftrightarrow = \lambda\phi,\psi \centerdot (\phi \supset \psi) \wedge (\psi \supset \phi)$, $\Diamond = \lambda R, \phi \centerdot \neg(\Box R (\neg\phi))$, $\boldsymbol{\Sigma}^\iota = \lambda\phi \centerdot \neg \boldsymbol{\Pi}^\iota(\lambda X \centerdot \neg\phi X)$, $\boldsymbol{\Sigma}^{\mu\to o} = \lambda\phi \centerdot \neg \boldsymbol{\Pi}^{\mu\to o}(\lambda P \centerdot \neg\phi P)$.

For defining $\mathcal{QML}^{STT}$-propositions we fix a set $\mathcal{IV}^{STT}$ of individual variables of type ι, a set $\mathcal{PV}^{STT}$ of propositional variables[5] of type $\mu \to o$, and a set $\mathcal{SYM}^{STT}$ of n-ary (curried) predicate symbols of types $\underbrace{\iota \to \ldots \to \iota}_{n} \to (\mu \to o)$. Moreover, we fix a set $\mathcal{S}^{STT}$ of accessibility relation constants of type $\mu \to \mu \to o$. $\mathcal{QML}^{STT}$-propositions are now defined as the smallest set of $\mathcal{STT}$-terms for which the following hold:

- if $P \in \mathcal{PV}^{STT}$, then $P \in \mathcal{QML}^{STT}$
- if $X^j \in \mathcal{IV}^{STT}$ $(j = 1, \ldots, n)$ and $k \in \mathcal{SYM}^{STT}$, then $(k\, X^1 \ldots X^n) \in \mathcal{QML}^{STT}$
- if $\phi, \psi \in \mathcal{QML}^{STT}$, then $\neg\phi \in \mathcal{QML}^{STT}$ and $\phi \vee \psi \in \mathcal{QML}^{STT}$
- if $r \in \mathcal{S}^{STT}$ and $\phi \in \mathcal{QML}^{STT}$, then $\Box\, r\, \phi \in \mathcal{QML}^{STT}$
- if $X \in \mathcal{IV}^{STT}$ and $\phi \in \mathcal{QML}^{STT}$, then $\boldsymbol{\Pi}^\iota(\lambda X \centerdot \phi) \in \mathcal{QML}^{STT}$
- if $P \in \mathcal{PV}^{STT}$ and $\phi \in \mathcal{QML}^{STT}$, then $\boldsymbol{\Pi}^{\mu\to o}(\lambda P \centerdot \phi) \in \mathcal{QML}^{STT}$

We write $\Box_r\, \phi$ for $\Box\, r\, \phi$, $\forall X_\iota \centerdot \phi$ for $\boldsymbol{\Pi}^\iota(\lambda X_\iota \centerdot \phi)$, and $\forall P_{\mu\to o} \centerdot \phi$ for $\boldsymbol{\Pi}^{\mu\to o}(\lambda P_{\mu\to o} \centerdot \phi)$.

Note that the defining equations for our $\mathcal{QML}$ modal operators are themselves formulas in $\mathcal{STT}$. Hence, we can express $\mathcal{QML}$ formulas in a higher-order reasoner elegantly in the usual syntax. For example, $\Box_r\, \exists P_{\mu\to o} \centerdot P$ is a $\mathcal{QML}^{STT}$ proposition; it has type $\mu \to o$.

Validity of $\mathcal{QML}^{STT}$ propositions is defined in the obvious way: a $\mathcal{QML}$-proposition $\phi_{\mu\to o}$ is valid if and only if for all possible worlds w_μ we have $w \in \phi_{\mu\to o}$, that is, if and only if $\phi_{\mu\to o}\, w_\mu$ holds. Hence, the notion of validity is modeled via the following equation (alternatively, validity could be defined simply as $\Pi_{(\mu\to o)\to o}$):

$$\text{valid} = \lambda\phi_{\mu\to o} \centerdot \forall W_\mu \centerdot \phi\, W$$

Now we can formulate proof problems in $\mathcal{QML}^{STT}$, e.g., valid $\Box_r\, \exists P_{\mu\to o} \centerdot P$. Using rewriting or definition expanding, we can reduce such proof problems to corresponding statements containing only the basic connectives $\neg$, $\vee$, $=$, Π^ι, and $\Pi^{\mu\to o}$ of $\mathcal{STT}$. In contrast to the many other approaches no external transformation mechanism is required. For our example formula valid $\Box_r\, \exists P_{\mu\to o} \centerdot P$ unfolding and $\beta\eta$-reduction leads to $\forall W_\mu \centerdot \forall Y_\mu \centerdot \neg r\, W\, Y \vee (\neg\forall X_{\mu\to o} \centerdot \neg(X\, Y))$. It is easy to check that this formula is valid in Henkin semantics: put $X = \lambda Y_\mu \centerdot \top$.

We have proved soundness and completeness for this embedding [8], that is, for $s \in \mathcal{QML}$ and the corresponding $s_{\mu\to o} \in \mathcal{QML}^{STT} \subset \mathcal{STT}$ we have:

Theorem 1. $\models^{\mathcal{STT}}$ *(valid* $s_{\mu\to o}$*) if and only if* $\models^{\mathbf{QK}\pi}$ s*.*

This result also illustrates the correspondence between $\mathbf{QK}\pi$ models and Henkin models; for more details see [8].

[5] Note that the denotation of propositional variables depends on worlds.

Obviously, the reduction of our embedding to first-order multimodal logics (which only allow quantification over individual variables), to propositional quantified multimodal logics (which only allow quantification over propositional variables) and to propositional multimodal logics (no quantifiers) is sound and complete. Extending our embedding for hybrid logics is straightforward [21]; note in particular that denomination of individual worlds using constant symbols of type μ is easily possible.

In the remainder we will often omit type information. It is sufficient to remember that worlds are of type μ, multimodal propositions of type $\mu \to o$, and accessibility relations of type $\mu \to \mu \to o$. Individuals are of type ι.

3 Reasoning about Modal Logics

3.1 Accessibility Relation Properties and Modal Logic Axioms

There are well known relationships between properties of accessibility relations and corresponding modal logic axioms (or axiom schemata) [18]. Such meta-theoretic insights can be elegantly encoded in our approach. First we encode various accessibility relation properties in $\mathcal{STT}$:

$$\begin{aligned}
\text{reflexive} &= \lambda R_\bullet \forall S_\bullet R\, S\, S \\
\text{symmetric} &= \lambda R_\bullet \forall S, T_\bullet ((R\, S\, T) \Rightarrow (R\, T\, S)) \\
\text{serial} &= \lambda R_\bullet \forall S_\bullet \exists T_\bullet (R\, S\, T) \\
\text{transitive} &= \lambda R_\bullet \forall S, T, U_\bullet ((R\, S\, T) \wedge (R\, T\, U) \Rightarrow (R\, S\, U)) \\
\text{euclidean} &= \lambda R_\bullet \forall S, T, U_\bullet ((R\, S\, T) \wedge (R\, S\, U) \Rightarrow (R\, T\, U))
\end{aligned}$$

The corresponding axioms are given next.

$$\begin{aligned}
M(\text{or } T): &\ \forall \phi_\bullet \Box_r\, \phi \supset \phi \\
B: &\ \forall \phi_\bullet \phi \supset \Box_r \Diamond_r\, \phi \\
D: &\ \forall \phi_\bullet \Box_r\, \phi \supset \Diamond_r\, \phi \\
4: &\ \forall \phi_\bullet \Box_r\, \phi \supset \Box_r \Box_r\, \phi \\
5: &\ \forall \phi_\bullet \Diamond_r\, \phi \supset \Box_r \Diamond_r\, \phi
\end{aligned}$$

Exploiting our embedding $\mathcal{QML}^{STT}$ we can now elegantly formalize well known correspondence theorems in $\mathcal{STT}$:

$$\forall R_\bullet (\text{reflexive } R) \Leftrightarrow (\text{valid } \forall \phi_\bullet \Box_R\, \phi \supset \phi) \tag{1}$$

$$\forall R_\bullet (\text{symmetric } R) \Leftrightarrow (\text{valid } \forall \phi_\bullet \phi \supset \Box_R \Diamond_R\, \phi) \tag{2}$$

$$\forall R_\bullet (\text{serial } R) \Leftrightarrow (\text{valid } \forall \phi_\bullet \Box_R\, \phi \supset \Diamond_R\, \phi) \tag{3}$$

$$\forall R_\bullet (\text{transitive } R) \Leftrightarrow (\text{valid } \forall \phi_\bullet \Box_R\, \phi \supset \Box_R \Box_R\, \phi) \tag{4}$$

$$\forall R_\bullet (\text{euclidean } R) \Leftrightarrow (\text{valid } \forall \phi_\bullet \Diamond_R\, \phi \supset \Box_R \Diamond_R\, \phi) \tag{5}$$

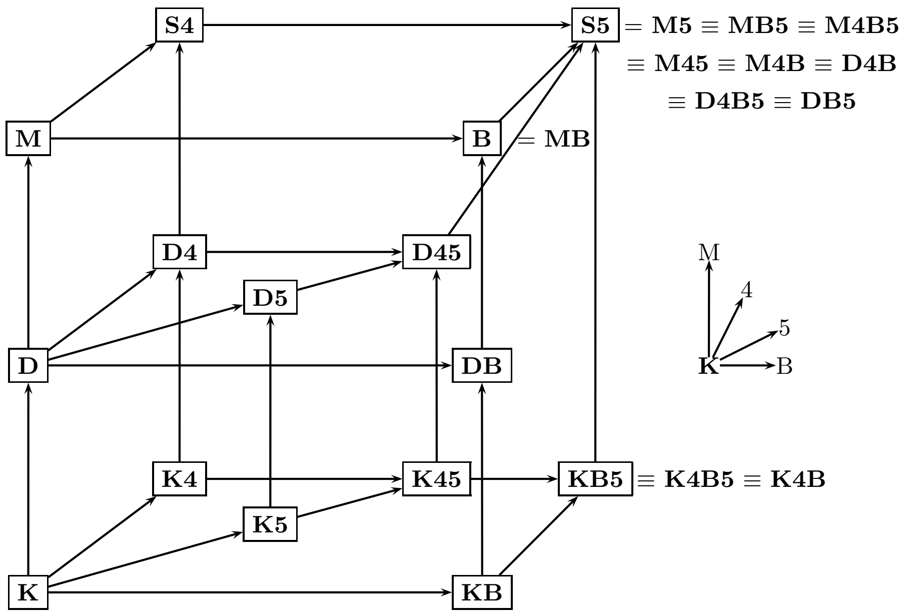

Fig. 1. The Modal Logic Cube; reproduced from [17].

3.2 Alternative Axiomatizations of Modal Logics

The cube in Figure 1 depicts the different modal logics that can be defined from normal modal logic **K** (lower left corner) by adding (combinations of) the axioms M, B, D, 4, and 5. A conjecture implicitly contained in this cube is that there are modal logics that can be axiomatized in alternative ways using these axioms.

For example, for modal logic **S5** we may choose axioms M and 5 as standard axioms. Respectively for logic **KB5** we may choose B and 5. We may then want to investigate the following conjectures about equivalent axiomatizations for **S5**, respectively, for **KB5**:

$$\begin{aligned} \mathbf{S5} = \mathbf{M5} &\Leftrightarrow \mathbf{MB5} \\ &\Leftrightarrow \mathbf{M4B5} \\ &\Leftrightarrow \mathbf{M45} \\ &\Leftrightarrow \mathbf{M4B} \\ &\Leftrightarrow \mathbf{D4B} \\ &\Leftrightarrow \mathbf{D4B5} \\ &\Leftrightarrow \mathbf{DB5} \end{aligned} \qquad \begin{aligned} \mathbf{KB5} &\Leftrightarrow \mathbf{K4B5} \\ &\Leftrightarrow \mathbf{K4B} \end{aligned}$$

Exploiting the correspondence theorems from Sect. 3.1 these problems can be formulated as follows; we give the case for **M5** $\Leftrightarrow$ **D4B**:

$$\forall R\text{.}\, (((\text{reflexive } R) \wedge (\text{euclidean } R)) \Leftrightarrow ((\text{serial } R) \wedge (\text{transitive } R) \wedge (\text{symmetric } R)))$$

3.3 Inclusion Relations between Different Modal Logics

The links in the modal logic cube in Fig. 1 describe unidirectional inclusion relations between modal logics. For example, the link between **D45** and **S5** expresses that modal logic **D45** is included in logic **S5** (we write **D45** $\Subset$ **S5**) but not vice versa. That is, all formulas that are valid in **D45** are also valid in **S5**. On the other hand, there are formulas that are valid in **S5** but not in **D45** (**S5** $\not\Subset$ **D45**).

Exploiting the equivalence (bidirectional inclusion) of **S5** and **D4B5** and monotonicity of entailment the inclusion of **D45** in **S5** is obvious: we simply add axiom B when moving in this direction. These trivial directions of the inclusion links in our modal logic cube are not further addressed in this paper.

The backward directions, however, are more interesting. It are these non-inclusion aspects of the links that we need to verify. The general task in each case is to find a countermodel to the respective inclusion statement. For example, in order to show that logic **M** is not included in logic **D** we may want to find a countermodel to the inclusion statement

$$(\text{valid}\, \forall \phi\text{.}\, \Box_r \phi \supset \Diamond_r \phi) \Rightarrow (\text{valid}\, \forall \phi\text{.}\, \Box_r \phi \supset \phi)$$

Again, by exploiting the correspondence theorems from Sect. 3.1 we may instead search for a countermodel to

$$\mathbf{M} \Subset \mathbf{D}: \qquad \forall R\text{.}\, (\text{serial } R) \Rightarrow (\text{reflexive } R)$$

The systems that are applicable to these inclusion statements are the countermodel finders Refute, Nitpick, and Satallax.

Alternatively, we may try to find a countermodel to **M** $\Subset$ **D** by tackling the negated inclusion statement with a theorem prover

$$\mathbf{M} \not\Subset \mathbf{D}: \qquad \neg\forall R\text{.}\, (\text{serial } R) \Rightarrow (\text{reflexive } R)$$

In the particular case of **M** and **D** the negated statement is clearly not a theorem though: in a model consisting of only one possible world, seriality in fact implies reflexivity. However, since we are in fact interested only in finding a countermodel to the statement **M** $\Subset$ **D** we may simply provide some help to the provers by adding an axiom expressing that there are at least two different worlds:

$$w1 \neq w2$$

When adding such an axiom the statement **M** $\not\Subset$ **D** becomes a theorem (now there is an accessibility relation which is serial but not reflexive: simply choose $\neq$). The systems that are applicable to the negated inclusion statements are the provers TPS, LEO-II, IsabelleP and Satallax.

In the experiment reported below we have actually added axioms stating there are at least three different possible worlds to all negated statements.

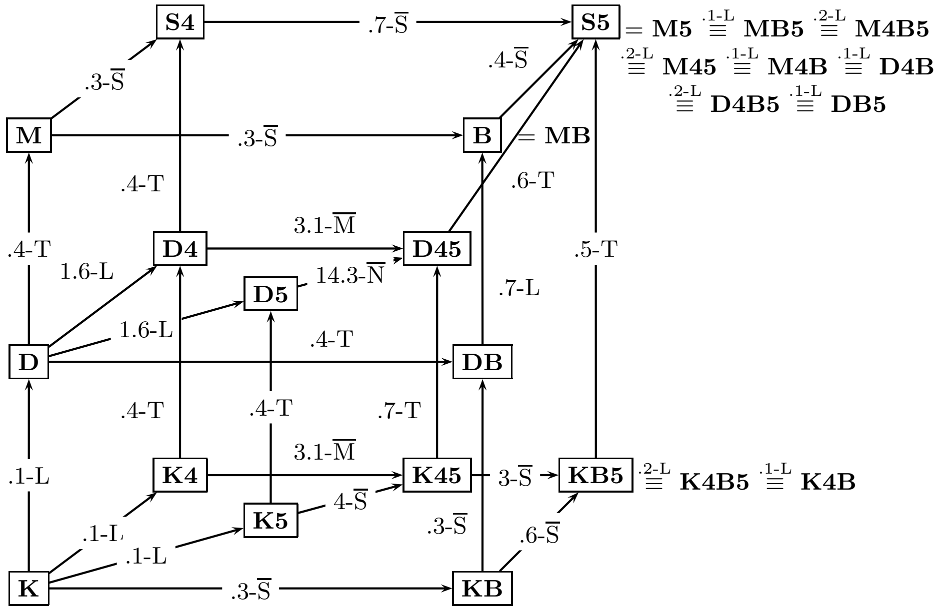

Fig. 2. Verification of the Modal Logic Cube: fastest proofs/countermodels reported.

4 Verification Results

In our experiments, we have employed the following system versions: TPS—3.080227G1d, LEO-II—v1.2, Satallax—1.4, IsabelleP—2009-2, IsabelleM—2009-2 (Refute), and IsabelleN—2009-2 (Nitpick). These systems are all available online via the SystemOnTPTP tool [25] and they support the new TPTP THF infrastructure for typed higher-order logic [11]. Exploiting the SystemOnTPTP tool all the experiment runs were done on 2.80GHz computers with 1GB memory and running the Linux operating system, with a 300s CPU limit.

The axiomatizations of $\mathcal{QML}^{\mathcal{STT}}$ and $\mathcal{IPL}^{\mathcal{STT}}$ are available as LCL013ˆ0.ax and LCL010ˆ0.ax in the TPTP library.[6] The example problems LCL698ˆ1.p and LCL695ˆ1.p ask about the satisfiability of these axiomatizations. Both questions are answered positively by the Satallax model finder in less than a second.

The correspondence theorems (1)–(5) from Sect. 3.1 are trivial: LEO-II solves problems (1),(3), and (4) in .1 seconds each, and it solves problem (2) in .3 seconds. TPS is the fastest prover to solve statement (5), for which it needs .4 seconds.

Figure 2 depicts the further results of our experiments. The timings presented for each link are given in seconds. For each link we present the fastest successful

[6] Note that the types μ and ι are unfortunately switched in the encodings available in the TPTP: the former is used for individuals and the latter for worlds. This syntactic switch is completely unproblematic.

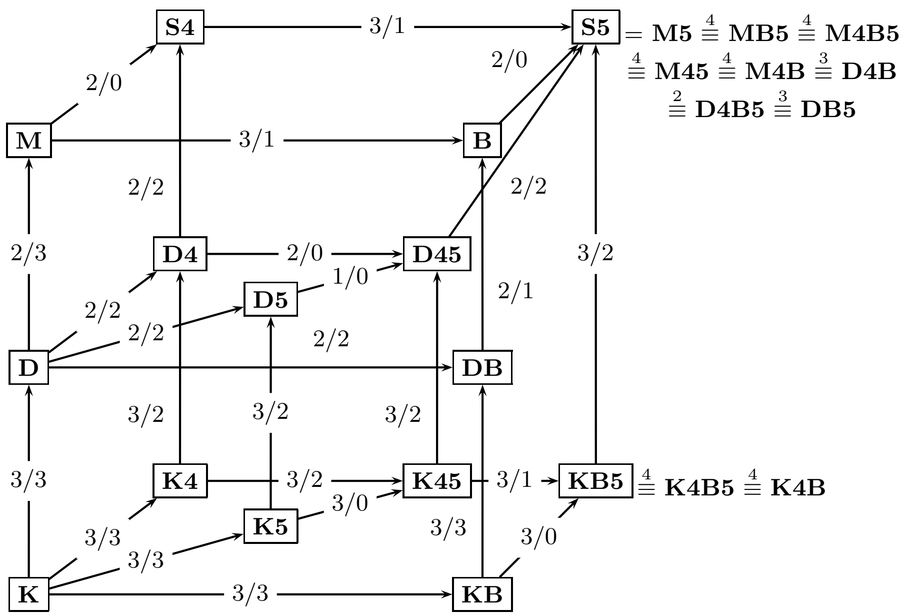

Fig. 3. Verification of the Modal Logic Cube: number of different successful results.

attempt reported by one of our reasoners. Moreover, we indicate with the suffixes -T, -L, -$\overline{\text{N}}$, -$\overline{\text{M}}$, and -$\overline{\text{S}}$ which system contributed this result. T stands for TPS, L for LEO-II, $\overline{\text{N}}$ for Nitpick, $\overline{\text{M}}$ for Refute, and $\overline{\text{S}}$ for Satallax as countermodel finder. IsabelleP and Satallax as prover never contributed a fastest result. As mentioned before, for each link the inclusion statement and the negated inclusion statement as described in Sect. 3.3 were presented to the provers.

Figure 3 presents how many different successful results were reported for each link. For example, the proper subset relation between **K4** and **D4** has been confirmed five times: the link annotation 3/2 says that we received 3 countermodels for **D4** $\Subset$ **K4** and 2 proofs for the negated statement **D4** $\not\Subset$ **K4**. In the equivalence statements the annotations express the number of successful proof attempts, for example, the statement **M5** $\equiv$ **MB5** was confirmed by each of our four provers.

In summary, all links in the modal logic cube can be verified effectively by at least one of the reasoners and most steps take only milliseconds. Furthermore, all equivalence statements in the cube can be solved in a few milliseconds. In all but one case different certificates are provided by the reasoners, raising the level of trust in the results significantly. Summing up all fastest times in our entire experiments results in a sum of less than 40 seconds.

5 Conclusion

The automated analysis and verification of bidirectional and unidirectional inclusions between propositional modal logics has originally been posed as a challenge problem for automated theorem provers by John Halleck and Geoff Sutcliffe. John Halleck was in need for a program as an aid to maintaining his logic systems overview [19].

Subsequently the challenge has been addressed with first-order automated theorem provers [23]. However, the solution presented there employs technically complex and hard to follow problem encodings in first-order logic, and even the fastest automated analysis of a subset relation in this study already requires more than 11 minutes of total reasoning time with state-of-the-art first-order automated theorem provers (if the pre-processing times required in this approach are also taken into account).

In our framework the automated analysis of bidirectional and unidirectional inclusion relations between well known modal logics becomes an easy task for higher-order automated theorem reasoning systems. Most notably, our problem encodings are elegant, simple and straightforward and the verification of the entire modal logic cube takes less than 40 seconds.

Future work includes the application of our framework for the exploration of inclusion relations between further modal logics. Note in particular, that our embedding of modal logics in simple type theory is not restricted to propositional logics: it also supports quantifiers and multiple box operators (cf. the Examples 4–6 in [6]). This calls for the development of a workbench for the automated analysis of propositional and quantified multimodal logics based on the approach presented in this paper.

Acknowledgment: I thank Jasmin Blanchette for proof reading this paper and for his valuable input (he helped in finding a minor issue in the problem encodings which initially prevented his Nitpick system in finding a counterexample to **D45** $\Subset$ **D5**). Moreover, I am indebted to Larry Paulson and Geoff Sutcliffe. Larry Paulson, together with the author, initiated the LEO-II project at Cambridge University (EPRSC grant LEO-II EP/D070511/1). Geoff Sutcliffe, in collaboration with the author and supported by several further contributors, developed the new higher-order TPTP THF infrastructure (EU FP7 grant THFTPTP PIIF-GA-2008-219982). Both projects had a significant impact on the work presented in this article.

References

1. P.B. Andrews and C. Brown. TPS: A Hybrid Automatic-Interactive System for Developing Proofs. *Journal of Applied Logic*, 4(4):367–395, 2006.
2. Peter B. Andrews. General models and extensionality. *Journal of Symbolic Logic*, 37:395–397, 1972.
3. Peter B. Andrews. General models, descriptions, and choice in type theory. *Journal of Symbolic Logic*, 37:385–394, 1972.

4. Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Kluwer Academic Publishers, second edition, 2002.
5. Julian Backes and Chad E. Brown. Analytic tableaux for higher-order logic with choice. In Jürgen Giesl and Reiner Haehnle, editors, *IJCAR 2010 - 5th International Joint Conference on Automated Reasoning*, LNAI, Edinburgh, UK, July 2010. Springer.
6. Christoph Benzmüller. Combining logics in simple type theory. In *The 11th International Workshop on Computational Logic in Multi-Agent Systems*, Lecture Notes in Artifical Intelligence, Lisbon, Portugal, 2010. Springer.
7. Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Higher order semantics and extensionality. *Journal of Symbolic Logic*, 69:1027–1088, 2004.
8. Christoph Benzmüller and Lawrence C. Paulson. *Quantified Multimodal Logics in Simple Type Theory*. SEKI Report SR–2009–02 (ISSN 1437-4447). SEKI Publications, DFKI Bremen GmbH, Safe and Secure Cognitive Systems, Cartesium, Enrique Schmidt Str. 5, D–28359 Bremen, Germany, 2009. http://arxiv.org/abs/0905.2435.
9. Christoph Benzmüller and Lawrence C. Paulson. Multimodal and intuitionistic logics in simple type theory. *The Logic Journal of the IGPL*, 2010.
10. Christoph Benzmüller, Lawrence C. Paulson, Frank Theiss, and Arnaud Fietzke. LEO-II — A Cooperative Automatic Theorem Prover for Higher-Order Logic. In P. Baumgartner, A. Armando, and D. Gilles, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning*, number 5195 in Lecture Notes in Artificial Intelligence, pages 162–170, 2008.
11. Christoph Benzmüller, Florian Rabe, and Geoff Sutcliffe. THF0 — The Core TPTP Language for Classical Higher-Order Logic. In P. Baumgartner, A. Armando, and D. Gilles, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning*, number 5195 in Lecture Notes in Artificial Intelligence, pages 491–506, 2008.
12. Patrick Blackburn and Maarten Marx. Tableaux for quantified hybrid logic. In Uwe Egly and Christian G. Fermüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002, Copenhagen, Denmark, July 30 - August 1, 2002, Proceedings*, volume 2381 of *Lecture Notes in Computer Science*, pages 38–52. Springer, 2002.
13. Jasmin Christian Blanchette and Tobias Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In Matt Kaufmann and Lawrence C. Paulson, editors, *International Conference on Interactive Theorem Proving (ITP-10)*, LNCS. Springer, 2010.
14. Torben Braüner. Natural deduction for first-order hybrid logic. *Journal of Logic, Language and Information*, 14(2):173–198, 2005.
15. Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.
16. Melvin Fitting. Interpolation for first order S5. *Journal of Symbolic Logic*, 67(2):621–634, 2002.
17. James Garson. Modal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2009 edition, 2009.
18. Robert Goldblatt. *Logics of Time and Computation (2nd edition)*. Number 7 in Center for the Study of Language and Information - Lecture Notes. Stanford, CA, USA, 1992.
19. J. Halleck. John Halleck's Logic Systems. `http://www.cc.utah.edu/~nahaj/logic/structures/systems/index.html`.

20. Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.
21. Mark Kaminski and Gert Smolka. Terminating tableau systems for hybrid logic with difference and converse. *Journal of Logic, Language and Information*, 18(4):437–464, Oct 2009.
22. Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002.
23. F. Rabe, P. Pudlak, G. Sutcliffe, and W. Shen. Solving the $100 Modal Logic Challenge. *Journal of Applied Logic*, page To appear, 2008.
24. S. Schulz. E: A Brainiac Theorem Prover. *AI Communications*, 15(2-3):111–126, 2002.
25. Geoff Sutcliffe. TPTP, TSTP, CASC, etc. In V. Diekert, M. Volkov, and A. Voronkov, editors, *Proceedings of the 2nd International Computer Science Symposium in Russia*, number 4649 in Lecture Notes in Computer Science, pages 7–23. Springer-Verlag, 2007.
26. Geoff Sutcliffe. The TPTP problem library and associated infrastructure. *J. Autom. Reasoning*, 43(4):337–362, 2009.
27. Geoff Sutcliffe and Christoph Benzmüller. Automated reasoning in higher-order logic using the TPTP THF infrastructure. *Journal of Formalized Reasoning*, 3(1):1–27, 2010.
28. Tjark Weber. *SAT-Based Finite Model Generation for Higher-Order Logic*. Ph.D. thesis, Dept. of Informatics, T.U. München, 2008.