Systematic Verification of the Modal Logic Cube in Isabelle/HOL[*]

Christoph Benzmüller   Maximilian Claus
Dep. of Mathematics and Computer Science, Freie Universität Berlin, Germany
c.benzmueller|m.claus@fu-berlin.de

Nik Sultana
Computer Lab, Cambridge University, UK
nik.sultana@cl.cam.ac.uk

We present an automated verification of the well-known modal logic cube in Isabelle/HOL, in which we prove the inclusion relations between the cube's logics using automated reasoning tools. Prior work addresses this problem but without restriction to the modal logic cube, and using encodings in first-order logic in combination with first-order automated theorem provers. In contrast, our solution is more elegant, transparent and effective. It employs an embedding of quantified modal logic in classical higher-order logic. Automated reasoning tools, such as Sledgehammer with LEO-II, Satallax and CVC4, Metis and Nitpick, are employed to achieve full automation. Though successful, the experiments also motivate some technical improvements in the Isabelle/HOL tool.

1 Introduction

We present an approach to meta-reasoning about modal logics, and apply it to verify the relative strengths of logics in the well-known *modal logic cube*, which is illustrated in Figure 1. In particular, proofs are given for the equivalences of different axiomatizations and the inclusion relations shown in the cube.

Our solution makes extensive use of the fact that all modal logics found in the cube are sound and complete because they arise from base modal logic K by adding Sahlqvist axioms. This is in contrast to prior work by Rabe et al. [16], who address the more general problem of determining the relation between two arbitrary modal logics characterized by their sets of inference rules. In their article the authors apply first-order logic encodings in combination with first-order automated theorem provers to prove an inclusion relation employing a number of different decision strategies. For the subproblem of only comparing logics within the cube (and therefore taking advantage of normality as additional knowledge) our solution improves on the elegance and simplicity of the problem encodings, as well as with automation performance. One motivation of this paper is to demonstrate the advantage of a pragmatically more expressive logic environment (here classical higher-order logic) in comparison to a less expressive language such as first-order logic or decidable fragments thereof.

We exploit an embedding of quantified multimodal logic (QML) in classical higher-order logic (HOL) [7], in which we carry out the automated verification of the aforementioned inclusion relations. These include the logics **K**, **D**, **M** (also known as **T**), **S4**, and **S5**. 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 between these

[*] This work has been supported by the German Research Foundation DFG under grants BE2501/9-2 & BE2501/11-1.

C.Kaliszyk and A. Paskevich (Eds.): 4th Workshop
on Proof eXchange for Theorem Proving (PxTP'15).
EPTCS 186, 2015, pp. 27–41, doi:10.4204/EPTCS.186.5

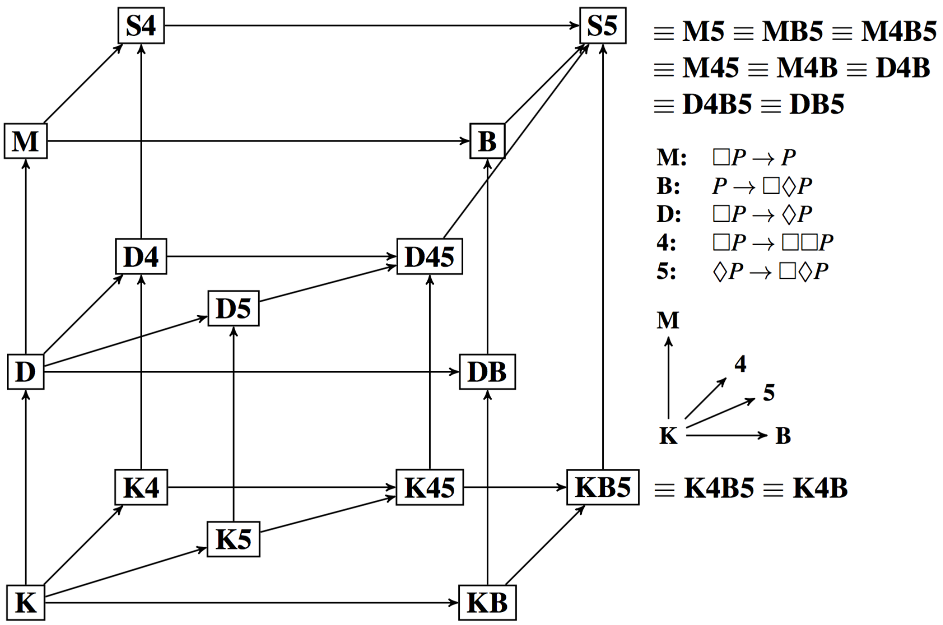


Figure 1: The modal logic cube: reasoning in modal logics is commonly done with respect to a certain set of basic axioms; different choices of basic axioms give rise to different modal logics. These modal logics can be arranged as vertices in a cube, such that the edges between them denote inclusion relations.

axioms and semantic properties of accessibility relations (i.e. Kripke models). These correspondences can themselves be elegantly formalized and effectively automated in our approach. Formalization of the modal axioms M, B, D, 4, and 5 requires quantification over propositional variables. This explains why an embedding of *quantified* modal logic in HOL is needed here, and not simply an embedding of propositional modal logic in HOL.

Our previous work (see the non-refereed, invited paper [3]) has already demonstrated the feasibility of the approach. However, instead of the development done there in pure TPTP THF [8], we here work with Isabelle/HOL [14] as the base environment, and fruitfully exploit various reasoning tools that are provided with it. This includes the Sledgehammer-based [15] interfaces from Isabelle/HOL to the external higher-order theorem provers LEO-II [9] and Satallax [1], as well as Isabelle/HOL's own reasoner Metis [11]. Moreover, the higher-order model finding capabilities of Nitpick [10] are heavily used in order to formulate and prove subsequent inclusion theorems in Isabelle/HOL. We also encountered some problems with interacting with the proof reconstruction available for LEO-II and Satallax in Isabelle/HOL.

This paper is a verified document in the sense that it has been automatically generated from Isabelle/HOL source code with the help of Isabelle's *build* tool (the entire source package is available from http://christoph-benzmueller.de/varia/pxtp2015.zip).

The paper is structured as follows: Section 2 presents an encoding of QML in HOL. This part reuses the theory provided by Benzmüller and Paulson [7], which has recently been further developed (to cover full higher-order QML) and applied for the verification of Gödel's ontological argument [5, 6]. Section 3 first establishes the well-known correspondence between properties of models and base axioms, and then investigates the equivalence of different axiomatizations. Subsequently, all inclusion relations as depicted in the modal logic cube are shown to be proper. Finally, the minimal number of possible worlds that is required to obtain proper inclusions in each case is determined and verified. Section 4 presents a short evaluation and discussion of the conducted experiments, and Section 5 concludes the paper.

2 An Embedding of Quantified Multimodal Logics in HOL

In contrast to the monomodal case, in quantified multimodal logics both modalities $\Box$ and $\Diamond$ are parametrized, such that they refer to potentially different accessibility relations. We write $\Box^R$ and $\Diamond^R$ to refer to necessity and possibility wrt. a relation R. Furthermore, in terms of quantification, we only consider the constant-domain case: this means that all possible worlds share one common domain of discourse. More details on the embedding of QML in HOL are given in earlier work [7, 6].

QML formulas are translated as HOL terms of type $i \Rightarrow bool$, where i is the type of possible worlds. This type is abbreviated as σ.

The classical connectives $\neg, \wedge, \rightarrow$, and $\forall$ (which quantifies over individuals and over sets of individuals) and $\exists$ (over individuals) are lifted to type σ. The lifted connectives are $\neg^m$, $\wedge^m$, $\vee^m$, $\rightarrow^m$, $\equiv^m$, $\forall$, and $\exists$ (the latter two are modeled as constant symbols). Other connectives can be introduced analogously. Moreover, the modal operators $\Box$ and $\Diamond$, parametric to R, are introduced. Note that in symbols like $\neg^m$, symbol m is simply part of the name, whereas in $\Box^R$ and $\Diamond^R$, symbol R is a parameter to the modality.

abbreviation *mnot* :: $\sigma \Rightarrow \sigma$ **where** $\neg^m\ \varphi \equiv (\lambda w.\ \neg\ \varphi\ w)$
abbreviation *mand* :: $\sigma \Rightarrow \sigma \Rightarrow \sigma$ **where** $\varphi \wedge^m \psi \equiv (\lambda w.\ \varphi\ w \wedge \psi\ w)$
abbreviation *mor* :: $\sigma \Rightarrow \sigma \Rightarrow \sigma$ **where** $\varphi \vee^m \psi \equiv (\lambda w.\ \varphi\ w \vee \psi\ w)$
abbreviation *mimplies* :: $\sigma \Rightarrow \sigma \Rightarrow \sigma$ **where** $\varphi \rightarrow^m \psi \equiv (\lambda w.\ \varphi\ w \longrightarrow \psi\ w)$
abbreviation *mequiv*:: $\sigma \Rightarrow \sigma \Rightarrow \sigma$ **where** $\varphi \equiv^m \psi \equiv (\lambda w.\ \varphi\ w \longleftrightarrow \psi\ w)$
abbreviation *mforall* :: $('a \Rightarrow \sigma) \Rightarrow \sigma$ **where** $\forall\ \Phi \equiv (\lambda w.\ \forall x.\ \Phi\ x\ w)$
abbreviation *mexists* :: $('a \Rightarrow \sigma) \Rightarrow \sigma$ **where** $\exists\ \Phi \equiv (\lambda w.\ \exists x.\ \Phi\ x\ w)$
abbreviation *mbox* :: $(i \Rightarrow i \Rightarrow bool) \Rightarrow \sigma \Rightarrow \sigma$ **where** $\Box^R\ \varphi \equiv (\lambda w.\ \forall v.\ (R\ w\ v) \longrightarrow \varphi\ v)$
abbreviation *mdia* :: $(i \Rightarrow i \Rightarrow bool) \Rightarrow \sigma \Rightarrow \sigma$ **where** $\Diamond^R\ \varphi \equiv (\lambda w.\ \exists v.\ R\ w\ v \wedge \varphi\ v)$

For grounding lifted formulas, the meta-predicate $[\cdot]$, read *valid*, is introduced.

abbreviation *valid* :: $\sigma \Rightarrow bool$ **where** $[p] \equiv \forall w.\ p\ w$

3 Reasoning about Modal Logics

3.1 Correspondence Results

Axioms of the modal cube correspond to constraints on the underlying accessibility relations. These constraints are as follows:

definition *refl* $\equiv \lambda R :: (i \Rightarrow i \Rightarrow bool).\ \forall S.\ R\ S\ S$ — reflexivity
definition *sym* $\equiv \lambda R :: (i \Rightarrow i \Rightarrow bool).\ \forall S\ T.\ (R\ S\ T \longrightarrow R\ T\ S)$ — symmetry
definition *ser* $\equiv \lambda R :: (i \Rightarrow i \Rightarrow bool).\ \forall S.\ \exists T.\ R\ S\ T$ — seriality
definition *trans* $\equiv \lambda R :: (i \Rightarrow i \Rightarrow bool).\ \forall S\ T\ U.\ (R\ S\ T \wedge R\ T\ U \longrightarrow R\ S\ U)$ — transitivity
definition *eucl* $\equiv \lambda R :: (i \Rightarrow i \Rightarrow bool).\ \forall S\ T\ U.\ (R\ S\ T \wedge R\ S\ U \longrightarrow R\ T\ U)$ — Euclidean

The corresponding axioms are defined next; note that they are parametric over accessibility relation R:

definition $M \equiv \lambda R\ .\ valid\ (\forall\,(\lambda P.\ (\Box^R\ P) \rightarrow^m P))$
definition $B \equiv \lambda R\ .\ valid\ (\forall\,(\lambda P.\ P \rightarrow^m \Box^R\Diamond^R\ P))$
definition $D \equiv \lambda R\ .\ valid\ (\forall\,(\lambda P.\ (\Box^R\ P) \rightarrow^m \Diamond^R\ P))$
definition $IV \equiv \lambda R\ .\ valid\ (\forall\,(\lambda P.\ (\Box^R\ P) \rightarrow^m \Box^R\Box^R\ P))$
definition $V \equiv \lambda R\ .\ valid\ (\forall\,(\lambda P.\ (\Diamond^R\ P) \rightarrow^m \Box^R\Diamond^R\ P))$

We will see below that *correspondence theorems* (between axioms and constraints on accessibility relations) can be elegantly expressed in HOL by exploiting the embedding used above. These correspondence theorems link a constraint to every axiom—for instance, *M* is linked to *refl*. Subsequently, in order to make statements about the relationship of two logics in the cube, it is sufficient to only look at the model constraints of their respective axiomatizations. Throughout the rest of this paper, all reasoning will be done on the model-theoretic side and then interpreted on the proof-theoretic side by the means of this correspondence.

3.1.1 Axiom M corresponds to Reflexivity

theorem *A1*: $(\forall R.\ (refl\ R) \longleftrightarrow (M\ R))$ **by** (*metis M-def refl-def*)

3.1.2 Axiom B corresponds to Symmetry

lemma *A2-a*: $(\forall R.\ (sym\ R) \longrightarrow (B\ R))$ **by** (*metis B-def sym-def*)
lemma *A2-b*: $(\forall R.\ (B\ R) \longrightarrow (sym\ R))$ **by** (*simp add:B-def sym-def*, *force*)
theorem *A2*: $(\forall R.\ (sym\ R) \longleftrightarrow (B\ R))$ **by** (*metis A2-a A2-b*)

3.1.3 Axiom D corresponds to Seriality

theorem *A3*: $(\forall R.\ (ser\ R) \longleftrightarrow (D\ R))$ **by** (*metis D-def ser-def*)

3.1.4 Axiom 4 corresponds to Transitivity

theorem *A4*: $(\forall R.\ (trans\ R) \longleftrightarrow (IV\ R))$ **by** (*metis IV-def trans-def*)

3.1.5 Axiom 5 corresponds to Euclideanness

lemma *A5-a*: $(\forall R.\ (eucl\ R) \longrightarrow (V\ R))$ **by** (*metis V-def eucl-def*)
lemma *A5-b*: $(\forall R.\ (V\ R) \longrightarrow (eucl\ R))$ **by** (*simp add:V-def eucl-def*, *force*)
theorem *A5*: $(\forall R.\ (eucl\ R) \longleftrightarrow (V\ R))$ **by** (*metis A5-a A5-b*)

3.2 Alternative Axiomatisations of Modal Logics

Often the same logic within the cube can be obtained through different axiomatizations. In this section we show how to prove different axiomatizations for logic **S5** resp. **KB5** to be equivalent. Using the correspondence theorems from the previous section, the equivalences can be elegantly formulated solely using the properties of accessibility relations. In Subsections 3.2.1 and 3.2.2 we also add the corresponding statements using the modal logic axioms; this could analogously be done also for the other theorems and lemmata presented in Sections 3.2 and 3.3.

The theorems below can be solved directly by Metis when it is provided the minimal set of necessary definitions. Sledgehammer (with the ATPs LEO-II and Satallax or with first-order provers) can also quickly solve these problems, in which case the manual selection of the required definitions is not necessary.

3.2.1 M5 $\Longleftrightarrow$ MB5

theorem *B1*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{refl}\ R) \wedge (\mathit{sym}\ R) \wedge (\mathit{eucl}\ R))$
by (*metis eucl-def refl-def sym-def*)
theorem *B1-alt*: $\forall R.((M\ R) \wedge (V\ R)) \longleftrightarrow ((M\ R) \wedge (B\ R) \wedge (V\ R))$
by (*metis A1 A2 A5 B1*)

3.2.2 M5 $\Longleftrightarrow$ M4B5

theorem *B2*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{refl}\ R) \wedge (\mathit{trans}\ R) \wedge (\mathit{sym}\ R) \wedge (\mathit{eucl}\ R))$
by (*metis eucl-def refl-def trans-def sym-def*)
theorem *B2-alt*: $\forall R.((M\ R) \wedge (V\ R)) \longleftrightarrow ((M\ R) \wedge (IV\ R) \wedge (B\ R) \wedge (V\ R))$
by (*metis A1 A4 A5 B1-alt B2*)

3.2.3 M5 $\Longleftrightarrow$ M45

theorem *B3*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{refl}\ R) \wedge (\mathit{trans}\ R) \wedge (\mathit{eucl}\ R))$
by (*metis eucl-def refl-def trans-def*)

3.2.4 M5 $\Longleftrightarrow$ M4B

theorem *B4*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{refl}\ R) \wedge (\mathit{trans}\ R) \wedge (\mathit{sym}\ R))$
by (*metis eucl-def refl-def sym-def trans-def*)

3.2.5 M5 $\Longleftrightarrow$ D4B

theorem *B5*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{ser}\ R) \wedge (\mathit{trans}\ R) \wedge (\mathit{sym}\ R))$
by (*metis eucl-def refl-def ser-def sym-def trans-def*)

3.2.6 M5 $\Longleftrightarrow$ D4B5

theorem *B6*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{ser}\ R) \wedge (\mathit{trans}\ R) \wedge (\mathit{sym}\ R) \wedge (\mathit{eucl}\ R))$
by (*metis eucl-def refl-def ser-def sym-def trans-def*)

3.2.7 M5 $\Longleftrightarrow$ DB5

theorem *B7*: $\forall R.((\mathit{refl}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{ser}\ R) \wedge (\mathit{sym}\ R) \wedge (\mathit{eucl}\ R))$
by (*metis eucl-def refl-def ser-def sym-def*)

3.2.8 KB5 $\Longleftrightarrow$ K4B5

theorem *B8*: $\forall R.((\mathit{sym}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{trans}\ R) \wedge (\mathit{sym}\ R) \wedge (\mathit{eucl}\ R))$
by (*metis eucl-def sym-def trans-def*)

3.2.9 KB5 $\Longleftrightarrow$ K4B

theorem *B9*: $\forall R.((\mathit{sym}\ R) \wedge (\mathit{eucl}\ R)) \longleftrightarrow ((\mathit{trans}\ R) \wedge (\mathit{sym}\ R))$
by (*metis eucl-def sym-def trans-def*)

3.3 Proper Inclusion Relations between Different Modal Logics

An edge within the cube denotes an inclusion between the connected logics. In the forward direction, these can be trivially shown valid through monotonicity of entailment and equivalence of the different axiomatizations. For example, for the forward link from logic **K** to logic **B**, we need to show that every theorem of **K** is also a theorem of **B**; this simply means to disregard the additional axiom B. Below, the crucial backward directions are proved. Informally, it is shown that through moving further up in the cube (adding further axioms), theorems can be proved which were not provable before; this means that the inclusions are proper. We write $A > B$ to indicate that logic A can prove strictly more theorems than logic B.

It has to be noted that some logics are actually equivalent if the only models considered have few enough worlds; examples are given below. We introduce some useful abbreviations to formulate constraints on the number of worlds in a model.

abbreviation *one-world-model* :: $i \Rightarrow bool$ **where** $\#^1$ $w1 \equiv \forall x.\ x = w1$
abbreviation *two-world-model* :: $i \Rightarrow i \Rightarrow bool$ **where** $\#^2$ $w1\ w2 \equiv (\forall x.\ x = w1 \vee x = w2) \wedge w1 \neq w2$
abbreviation *three-world-model* :: $i \Rightarrow i \Rightarrow i \Rightarrow bool$ **where** $\#^3$ $w1\ w2\ w3 \equiv (\forall x.\ x = w1 \vee x = w2 \vee x = w3) \wedge w1 \neq w2 \wedge w1 \neq w3 \wedge w2 \neq w3$

In what follows, we reserve the symbols *i1*, *i2* and *i3* for worlds, and *r* for an accessibility relation.

We applied the following methodology in the experiments reported in this section:

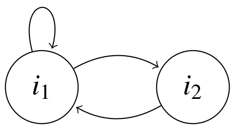
(Step A) First we deliberately made invalid conjectures about inclusion relations—e.g. for proving K4 > K we first wrongly conjectured that K4 ⊆ K, meaning that K4 entails K. We did this by conjecturing

$$lemma\ C1\text{-}A\colon \forall R.\ (trans\ R)$$

These wrongly-conjectured lemmata in Step A are uniformly named $C{*}\text{-}A$. Note that for the formulation of the $C{*}\text{-}A$-lemmata we again exploit the correspondence results given earlier, and we work with conditions on the accessibility relations instead of using the corresponding modal logic axioms. For each $C{*}\text{-}A$-lemma Nitpick quickly generates a countermodel, which it communicates in a specific syntax. For example, the countermodel it presents for *C1-A* is

$$R = (\lambda x.\ \text{-})(i_1 := (\lambda x.\ \text{-})(i_1 := True, i_2 := True), i_2 := (\lambda x.\ \text{-})(i_1 := True, i_2 := False))\ .$$

Diagrammatically this 2-world countermodel can be represented as follows



(Step B) Next, we systematically employed the arity information obtained from the countermodels for the $C{*}\text{-}A$-lemmata, reported by Nitpick, to formulate a corresponding lemma to be passed via Sledgehammer to the HOL-ATPs LEO-II, Satallax and/or CVC4 [2] (whenever it was not trivially provable by the automation tools *simp*, *force* and/or *blast* available within Isabelle/HOL). In our running example this lemma is

$$C1\text{-}B\colon \#^2\ i1\ i2 \longrightarrow \forall R.\ \neg\ trans\ R$$

All but four of these lemmata can actually be proved by either LEO-II or Satallax. Some of the easier problems can already be automated with *simp*, *force* and *blast*, which are preferred here. The four cases in which no automation attempts succeeded (we also tried all other integrated

ATPs in Isabelle) are named *C∗-ATP-challenge* below. Moreover, there are ten problems named *C∗-Isabelle-challenge*. For these problems LEO-II or Satallax found proofs, but their Metis-based integration into Isabelle failed. Hence, no verification was obtained for these problems. However, we found that five of these *C∗-Isabelle-challenge* problems can also be proved by CVC4, for which proof integration worked. Unfortunately, no other automation means (including the integrated first-order ATPs or SMT solvers) succeeded for the *C∗-Isabelle-challenge* problems.

(Step C) For the verification of the modal logic cube, the non-proved or non-integrated *C∗-challenge* problems of Step B are clearly unsatisfactory, since no proper verification in Isabelle is obtained. However, an easy solution for these (and all other) cases is possible by exploiting not only Nitpick's arity information on the countermodels, but by using all the information about the countermodels it presents, that is, the precise information on the accessibility relation. For example, Nitpick's countermodel for *C1-A* from above can be converted into the following theorem (where *r* denotes a fixed accessibility relation)

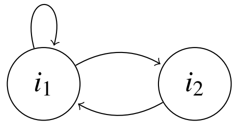
$$\textit{theorem C1-C}\colon \#^2\ i1\ i2 \wedge r\ i1\ i1 \wedge r\ i1\ i2 \wedge r\ i2\ i1 \wedge \neg r\ i2\ i2 \longrightarrow \neg\ \textit{trans}\ r.$$

The resulting theorems we generate are uniformly named *C∗-C*. It turns out that all *C∗-C*-theorems can be quickly verified in Isabelle by Metis. Thus, for each link in the modal logic we provide either a verified *C∗-B* theorem or, if this was not successful, a verified *C∗-C* theorem. Taken together, this confirms that the inclusion relation in the cube are indeed proper. Hence, these *C∗-B* resp. *C∗-C* theorems complete the verification of the modal logic cube. Below the *C∗-C* proof attempts are omitted if the corresponding *C∗-B* attempts were already successful.

(Step D) We additionally prove that the countermodels found by Nitpick in Step A are minimal (regarding the number of possible worlds). In other words, we prove here that the world model constraints as exploited in Step B are in fact minimal constraints under which the inclusion relations can be shown to be proper. Of course, if such a countermodel consists of one possible world only, nothing needs to be shown.

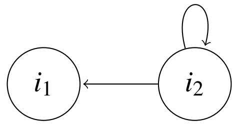
Note that the entire process sketched above, that is the schematic Steps A-D, could be fully automated, meaning that the formulation of the lemmata and theorems in each step could be obtained automatically by analyzing and converting Nitpick's output. In our experiments we still wrote and invoked the verification of each link in the modal cube manually however. Clearly, automation facilities could be very useful for the exploration of the meta-theory of other logics, for example, conditional logics [4], since the overall methodology is obviously transferable to other logics of interest.

3.3.1 K4 > K



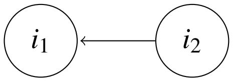
lemma *C1-A*: $\forall R.\ \textit{trans}\ R$ **nitpick oops**
theorem *C1-B*: $\#^2\ i1\ i2 \longrightarrow \neg\ (\forall R.\ \textit{trans}\ R)$ **by** (*simp add*:*trans-def*, *force*)
lemma *C1-D*: $\#^1\ i1 \longrightarrow (\forall R.\ \textit{trans}\ R)$ **by** (*metis* (*lifting*, *full-types*) *trans-def*)

3.3.2 K5 > K



lemma *C2-A*: $\forall R.\ \textit{eucl}\ R$ **nitpick oops**
theorem *C2-B*: $\#^2\ i1\ i2 \longrightarrow \neg\ (\forall R.\ \textit{eucl}\ R)$ **by** (*simp add*:*eucl-def*, *force*)
lemma *C2-D*: $\#^1\ i1 \longrightarrow (\forall R.\ \textit{eucl}\ R)$ **by** (*metis* (*lifting*, *full-types*) *eucl-def*)

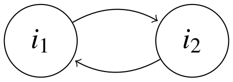
3.3.3 KB > K



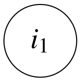
lemma *C3-A*: $\forall R.\ sym\ R$ **nitpick oops**
theorem *C3-B*: $\#^2\ i1\ i2 \longrightarrow \neg\ (\forall R.\ sym\ R)$ **by** $(simp\ add{:}sym\text{-}def, force)$
lemma *C3-D*: $\#^1\ i1 \longrightarrow (\forall R.\ sym\ R)$ **by** $(metis\ (full\text{-}types)\ sym\text{-}def)$

3.3.4 K45 > K4

lemma *C4-A*: $\forall R.\ ser\ R \longrightarrow (ser\ R \wedge eucl\ R)$ **nitpick oops**
lemma *C4-B-Isabelle-challenge*: $\#^2\ i1\ i2 \longrightarrow \neg\ (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge eucl\ R))$
— sledgehammer [remote_leo2](ser_def eucl_def)
— CPU time: 13.74 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] – timed out **oops**
theorem *C4-C*: $\#^2\ i1\ i2\ \wedge \neg r\ i1\ i1 \wedge r\ i1\ i2 \wedge r\ i2\ i1 \wedge \neg r\ i2\ i2 \longrightarrow \neg\ (ser\ r \longrightarrow (ser\ r \wedge eucl\ r))$
by $(metis\ ser\text{-}def\ eucl\text{-}def)$
lemma *C4-D*: $\#^1\ i1\ \longrightarrow (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge eucl\ R))$ **by** $(metis\ (full\text{-}types)\ eucl\text{-}def)$

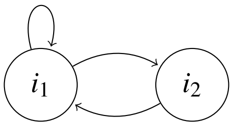


3.3.5 K45 > K5



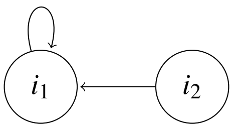
lemma *C5-A*: $\forall R.\ eucl\ R \longrightarrow (ser\ R \wedge eucl\ R)$
nitpick oops
lemma *C5-B-Isabelle-challenge*: $\#^1\ i1 \longrightarrow \neg\ (\forall R.\ (eucl\ R) \longrightarrow (ser\ R) \wedge (eucl\ R))$
— sledgehammer [remote_leo2](eucl_def ser_def) – CPU time: 14.61 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] – timed out **oops**
theorem *C5-C*: $\#^1\ i1 \wedge \neg r\ i1\ i1 \longrightarrow \neg\ (eucl\ r \longrightarrow (ser\ r \wedge eucl\ r))$ **by** $(metis\ (full\text{-}types)\ eucl\text{-}def\ ser\text{-}def)$

3.3.6 KB5 > KB



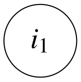
lemma *C6-A*: $\forall R.\ sym\ R \longrightarrow (sym\ R \wedge eucl\ R)$
nitpick oops
lemma *C6-B-Isabelle-challenge*: $\#^2\ i1\ i2 \longrightarrow \neg\ (\forall R.\ sym\ R \longrightarrow (sym\ R \wedge eucl\ R))$
— sledgehammer [remote_leo2,timeout=200](sym_def eucl_def) – CPU time: 29.0 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] suggested following line:
by $(metis\ (full\text{-}types)\ A4\ B8\ C1\text{-}B\ IV\text{-}def\ sym\text{-}def)$
lemma *C6-D*: $\#^1\ i1 \longrightarrow (\forall R.\ sym\ R \longrightarrow (sym\ R \wedge eucl\ R))$
by $(metis\ (full\text{-}types)\ eucl\text{-}def)$

3.3.7 KB5 > K45



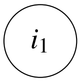
lemma *C7-A*: $\forall R.\ ser\ R \wedge eucl\ R \longrightarrow (sym\ R \wedge eucl\ R)$
nitpick oops
lemma *C7-B-Isabelle-challenge*: $\#^2\ i1\ i2 \longrightarrow \neg\ (\forall R.\ ser\ R \wedge eucl\ R \longrightarrow (sym\ R \wedge eucl\ R))$
— sledgehammer [remote_leo2] (ser_def eucl_def sym_def) – CPU time: 11.15 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] – timed out **oops**
theorem *C7-C*: $\#^2\ i1\ i2 \wedge r\ i1\ i1 \wedge \neg\ r\ i1\ i2 \wedge r\ i2\ i1 \wedge \neg\ r\ i2\ i2 \longrightarrow \neg\ (ser\ r \wedge eucl\ r \longrightarrow (sym\ r \wedge eucl\ r))$
by $(metis\ (full\text{-}types)\ ser\text{-}def\ eucl\text{-}def\ sym\text{-}def)$
lemma *C7-D*: $\#^1\ i1 \longrightarrow (\forall R.\ ser\ R \wedge eucl\ R \longrightarrow (sym\ R \wedge eucl\ R))$ **by** $(metis\ (full\text{-}types)\ sym\text{-}def)$

3.3.8 D > K



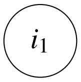
lemma *C8-A*: $\forall R.\ ser\ R$ **nitpick oops**
lemma *C8-B*: $\#^1\ i1 \longrightarrow \neg(\forall R.\ (ser\ R))$ **by** (*simp add*:*ser-def*, *force*)
theorem *C8-C*: $\#^1\ i1 \wedge \neg r\ i1\ i1 \longrightarrow \neg(ser\ r)$ **by** (*metis* (*full-types*) *ser-def*)

3.3.9 D4 > K4



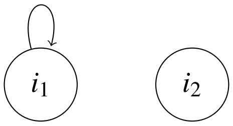
lemma *C9-A*: $\forall R.\ trans\ R \longrightarrow (ser\ R \wedge trans\ R)$
nitpick oops
theorem *C9-B*: $\#^1\ i1 \longrightarrow \neg\ (\forall R.\ trans\ R \longrightarrow (ser\ R \wedge trans\ R))$
using *C1-D C8-B* **by** *blast*

3.3.10 D5 > K5



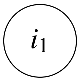
lemma *C10-A*: $\forall R.\ eucl\ R \longrightarrow (ser\ R \wedge eucl\ R)$ **nitpick oops**
theorem *C10-B*: $\#^1\ i1\ \longrightarrow \neg\ (\forall R.\ eucl\ R \longrightarrow (ser\ R \wedge eucl\ R))$ **using** *B9 C3-D C9-B* **by** *blast*

3.3.11 D45 > K45



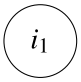
lemma *C11-A*: $\forall R.\ trans\ R \wedge eucl\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R)$
nitpick oops
theorem *C11-B*: $\#^1\ i1 \longrightarrow \neg\ (\forall R.\ trans\ R \wedge eucl\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R))$
using *B9 C3-D C9-B* **by** *blast*

3.3.12 DB > KB



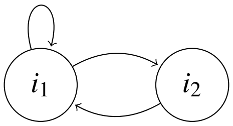
lemma *C12-A*: $\forall R.\ sym\ R \longrightarrow (ser\ R \wedge sym\ R)$
nitpick oops
theorem *C12-B*: $\#^1\ i1 \longrightarrow \neg\ (\forall R.\ sym\ R \longrightarrow (ser\ R \wedge sym\ R))$
using *C11-B C3-D* **by** *blast*

3.3.13 S5 > KB5



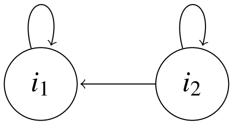
lemma *C13-A*: $\forall R.\ sym\ R \wedge eucl\ R \longrightarrow (refl\ R \wedge eucl\ R)$
nitpick oops
theorem *C13-B*: $\#^1\ i1 \longrightarrow \neg\ (\forall R.\ sym\ R \wedge eucl\ R \longrightarrow (refl\ R \wedge eucl\ R))$ **using** *B5 C12-B C6-D* **by** *blast*

3.3.14 D4 > D



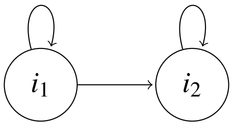
lemma *C14-A*: $\forall R.\ (ser\ R) \longrightarrow (ser\ R) \wedge (trans\ R)$
nitpick oops
theorem *C14-B-Isabelle-challenge*: $\#^2\ i1\ i2 \longrightarrow \neg(\forall R.\ ser\ R \longrightarrow (ser\ R \wedge trans\ R))$
— sledgehammer [remote_leo2] (ser_def trans_def) – CPU time: 13.08 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] suggested following line:
by (*metis* (*full-types*) *C1-B trans-def ser-def*)
lemma *C14-D*: $\#^1\ i1 \longrightarrow (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge trans\ R))$ **by** (*metis* (*full-types*) *trans-def*)

3.3.15 D5 $>$ D



lemma *C15-A*: $\forall R.\ ser\ R \longrightarrow (ser\ R \wedge eucl\ R)$
nitpick oops
theorem *C15-B-Isabelle-challenge*: #^{2} $i1\ i2 \longrightarrow \neg\ (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge eucl\ R))$
— sledgehammer [remote_leo2](ser_def eucl_def)
— CPU time: 12.9 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] suggested following line:
by (*metis* (*full-types*) *C14-B-Isabelle-challenge trans-def eucl-def*)
lemma *C15-D*: #^{1} $i1 \longrightarrow (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge eucl\ R))$ **by** (*metis* (*full-types*) *C2-D*)

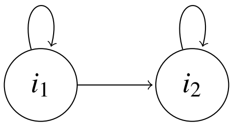
3.3.16 DB $>$ D



lemma *C16-A*: $\forall R.\ ser\ R \longrightarrow (ser\ R \wedge sym\ R)$
nitpick oops

lemma *C16-B*: #^{2} $i1\ i2 \longrightarrow \neg\ (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge sym\ R))$ **by** (*simp add*:*ser-def sym-def*, *force*)
lemma *C16-D*: #^{1} $i1 \longrightarrow (\forall R.\ ser\ R \longrightarrow (ser\ R \wedge sym\ R))$ **by** (*metis* (*full-types*) *sym-def*)

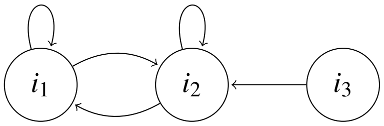
3.3.17 D45 $>$ D4



lemma *C17-A*: $\forall R.\ ser\ R \wedge trans\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R)$
nitpick oops

lemma *C17-B-ATP-challenge*: #^{2} $i1\ i2 \longrightarrow \neg(\forall R.\ ser\ R \wedge trans\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R))$
oops — All ATPs time out
theorem *C17-C*: #^{2} $i1\ i2 \wedge r\ i1\ i1 \wedge r\ i1\ i2 \wedge \neg\ r\ i2\ i1 \wedge r\ i2\ i2 \longrightarrow \neg\ (ser\ r \wedge trans\ r \longrightarrow (ser\ r \wedge trans\ r \wedge eucl\ r))$
by (*metis* (*full-types*) *ser-def trans-def eucl-def*)
lemma *C17-D*: #^{1} $i1 \longrightarrow (\forall R.\ ser\ R \wedge trans\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R))$
by (*metis* (*full-types*) *eucl-def*)

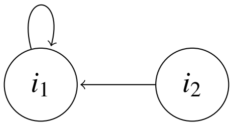
3.3.18 D45 $>$ D5



lemma *C18-A*: $\forall R.\ ser\ R \wedge eucl\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R)$
nitpick oops

lemma *C18-ATP-challenge*: #^{3} $i1\ i2\ i3 \longrightarrow \neg\ (\forall R.\ ser\ R \wedge eucl\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R))$
oops — All ATPs time out
theorem *C18-C*: #^{3} $i1\ i2\ i3 \wedge r\ i1\ i1 \wedge r\ i1\ i2 \wedge \neg\ r\ i1\ i3 \wedge r\ i2\ i1 \wedge r\ i2\ i2 \wedge \neg\ r\ i2\ i3 \wedge \neg\ r\ i3\ i1 \wedge r\ i3\ i2 \wedge \neg\ r\ i3\ i3 \longrightarrow \neg\ (ser\ r \wedge eucl\ r \longrightarrow (ser\ r \wedge trans\ r \wedge eucl\ r))$ **by** (*metis* (*full-types*) *eucl-def ser-def trans-def*)
lemma *C18-D*: #^{2} $i1\ i2 \longrightarrow (\forall R.\ ser\ R \wedge eucl\ R \longrightarrow (ser\ R \wedge trans\ R \wedge eucl\ R))$
by (*metis* (*full-types*) *eucl-def trans-def*)

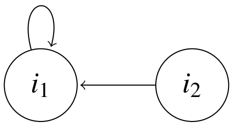
3.3.19 M $>$ D



lemma *C19-A*: $\forall R.\ ser\ R \longrightarrow refl\ R$
nitpick oops

theorem *C19-B-Isabelle-challenge*: $\#^2$ *i1 i2* $\longrightarrow \neg$ ($\forall$*R*. *ser R* $\longrightarrow$ *refl R*)
— sledgehammer [remote_leo2,timeout=200] (ser_def refl_def) – CPU time: 29.15 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] suggested following line:
by (*metis* (*full-types*) *C14-B-Isabelle-challenge trans-def refl-def*)
lemma *C19-D*: $\#^1$ *i1* $\longrightarrow$ ($\forall$*R*. *ser R* $\longrightarrow$ *refl R*) **by** (*metis* (*full-types*) *ser-def refl-def*)

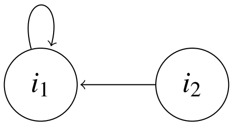
3.3.20 S4 > D4



lemma *C20-A*: $\forall$*R*. *ser R* $\wedge$ *trans R* $\longrightarrow$ (*refl R* $\wedge$ *trans R*)
nitpick oops

lemma *C20-B-Isabelle-challenge*: $\#^2$ *i1 i2* $\longrightarrow \neg$ ($\forall$*R*. *ser R* $\wedge$ *trans R* $\longrightarrow$ (*refl R* $\wedge$ *trans R*))
— sledgehammer [remote_leo2](ser_def trans_def refl_def) – CPU time: 12.5 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] – timed out
oops
theorem *C20-C*: $\#^2$ *i1 i2* $\wedge$ *r i1 i1* $\wedge \neg$ *r i1 i2* $\wedge$ *r i2 i1* $\wedge \neg$ *r i2 i2* $\longrightarrow \neg$ (*ser r* $\wedge$ *trans r* $\longrightarrow$ (*refl r* $\wedge$ *trans r*))
by (*metis* (*full-types*) *ser-def refl-def trans-def*)
lemma *C20-D*: $\#^1$ *i1* $\longrightarrow$ ($\forall$*R*. *ser R* $\wedge$ *trans R* $\longrightarrow$ (*refl R* $\wedge$ *trans R*))
by (*metis* (*full-types*) *ser-def refl-def*)

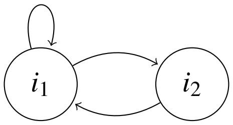
3.3.21 S5 > D45



lemma *C21-A*: $\forall$*R*. *ser R* $\wedge$ *trans R* $\wedge$ *eucl R* $\longrightarrow$ (*refl R* $\wedge$ *eucl R*)
nitpick oops

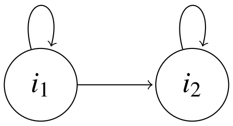
lemma *C21-B-Isabelle-challenge*: $\#^2$ *i1 i2* $\longrightarrow \neg$ ($\forall$*R*. *ser R* $\wedge$ *trans R* $\wedge$ *eucl R* $\longrightarrow$ (*refl R* $\wedge$ *eucl R*))
— sledgehammer [remote_leo2](ser_def trans_def eucl_def refl_def) – CPU time: 12.51 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] – timed out
oops
theorem *C21-C*: $\#^2$ *i1 i2* $\wedge$ *r i1 i1* $\wedge \neg$ *r i1 i2* $\wedge$ *r i2 i1* $\wedge \neg$ *r i2 i2* $\longrightarrow \neg$ (*ser r* $\wedge$ *trans r* $\wedge$ *eucl r* $\longrightarrow$ (*refl r* $\wedge$ *eucl r*))
by (*metis* (*full-types*) *ser-def trans-def eucl-def refl-def*)
lemma *C21-inclusion*: $\#^1$ *i1* $\longrightarrow$ ($\forall$*R*. *ser R* $\wedge$ *trans R* $\wedge$ *eucl R* $\longrightarrow$ (*refl R* $\wedge$ *eucl R*))
by (*metis* (*full-types*) *ser-def refl-def*)

3.3.22 B > DB



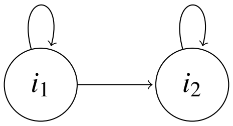
lemma *C22-A*: $\forall$*R*. *ser R* $\wedge$ *sym R* $\longrightarrow$ (*refl R* $\wedge$ *sym R*)
nitpick oops
lemma *C22-B-Isabelle-challenge*: $\#^2$ *i1 i2* $\longrightarrow \neg$ ($\forall$*R*. *ser R* $\wedge$ *sym R* $\longrightarrow$ (*refl R* $\wedge$ *sym R*))
— sledgehammer [remote_leo2,timeout=200](ser_def sym_def refl_def) – CPU time: 31.18 s. Metis reconstruction failed.
— sledgehammer [cvc4,timeout=300] suggested following line:
— by (smt C14_B sym_def trans_def refl_def) **oops**
theorem *C22-C*: $\#^2$ *i1 i2* $\wedge$ *r i1 i1* $\wedge$ *r i1 i2* $\wedge$ *r i2 i1* $\wedge \neg$ *r i2 i2* $\longrightarrow \neg$ (*ser r* $\wedge$ *sym r* $\longrightarrow$ (*refl r* $\wedge$ *sym r*))
by (*metis* (*full-types*) *ser-def sym-def refl-def*)
lemma *C22-D*: $\#^1$ *i1* $\longrightarrow$ ($\forall$*R*. *ser R* $\wedge$ *sym R* $\longrightarrow$ (*refl R* $\wedge$ *sym R*))
by (*metis* (*full-types*) *ser-def refl-def*)

3.3.23 B > M



lemma *C23-A*: ∀*R*. *refl R* ⟶ (*refl R* ∧ *sym R*) **nitpick oops**
lemma *C23-B-ATP-challenge*: #^{2} *i1 i2* ⟶ ¬ (∀*R*. *refl R* ⟶ (*refl R* ∧ *sym R*))
oops — All ATPs time out
theorem *C23-C*: #^{2} *i1 i2* ∧ *r i1 i1* ∧ *r i1 i2* ∧ ¬ *r i2 i1* ∧ *r i2 i2* ⟶ ¬ (*refl r* ⟶ (*refl r* ∧ *sym r*))
by (*metis refl-def sym-def*)
lemma *C23-D*: #^{1} *i1* ⟶ (∀*R*. *refl R* ⟶ (*refl R* ∧ *sym R*)) **by** (*metis* (*full-types*) *sym-def*)

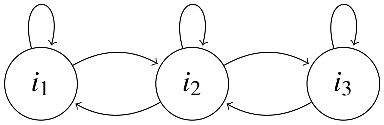
3.3.24 S5 > S4



lemma *C24-A*: ∀*R*. *refl R* ∧ *trans R* ⟶ (*refl R* ∧ *eucl R*)
nitpick oops

lemma *C24-B-ATP-challenge*: #^{2} *i1 i2* ⟶ ¬ (∀*R*. *refl R* ∧ *trans R* ⟶ (*refl R* ∧ *eucl R*))
oops — All ATPs time out
theorem *C24-C*: #^{2} *i1 i2* ∧ *r i1 i1* ∧ *r i1 i2* ∧ ¬ *r i2 i1* ∧ *r i2 i2* ⟶ ¬ (*refl r* ∧ *trans r* ⟶ (*refl r* ∧ *eucl r*))
by (*metis* (*full-types*) *trans-def refl-def eucl-def*)
lemma *C24-D*: #^{1} *i1* ⟶ (∀*R*. *refl R* ∧ *trans R* ⟶ (*refl R* ∧ *eucl R*)) **by** (*metis* (*full-types*) *eucl-def*)

3.3.25 S5 > B



lemma *C25-A*: ∀*R*. *refl R* ∧ *sym R* ⟶ (*refl R* ∧ *eucl R*)
nitpick oops

lemma *C25-B-ATP-challenge*: #^{3} *i1 i2 i3* ⟶ ¬ (∀*R*. (*refl R* ∧ *sym R*) ⟶ (*refl R* ∧ *eucl R*))
oops — All ATPs time out
theorem *C25-C*: #^{3} *i1 i2 i3* ∧ *r i1 i1* ∧ *r i1 i2* ∧ ¬ *r i1 i3* ∧ *r i2 i1* ∧ *r i2 i2* ∧ *r i2 i3* ∧ ¬ *r i3 i1* ∧ *r i3 i2* ∧ *r i3 i3* ⟶ ¬ ((*refl r* ∧ *sym r*) ⟶ (*refl r* ∧ *eucl r*))
by (*metis* (*full-types*) *eucl-def refl-def sym-def*)
lemma *C25-D*: #^{2} *i1 i2* ⟶ (∀*R*. (*refl R* ∧ *sym R*) ⟶ (*refl R* ∧ *eucl R*))
by (*metis* (*full-types*) *refl-def sym-def eucl-def*)

4 Discussion and Future Work.

The entire Isabelle document can be verified by Isabelle2014 in less than 60s on a semi-modern computer (2.4 GHz Core 2 Duo, 8 GB of memory). When including all (commented) remote calls to the external ATPs in the calculation the verification time sums up to a few minutes, which is still very reasonable.

The improvements in comparison to the first-order based verification of the modal logic cube done earlier by Rabe et al. [16], are: clarity and readability of the problem encodings, methodology, reliability (our proofs are verifiable in Isabelle/HOL) and, most importantly, automation performance. For the latter note that the experiments by Rabe et al. [16] required several days of reasoning time in first-order theorem provers. Most importantly, however, their solution relied on an enormous manual coding effort. However, we want to point again to the more general aims of their work.

Our solution instead requires a small amount of resources in comparison. In fact, as indicated before, the entire process (Steps A-D) is schematic, so that it should eventually be possible to fully automate our method. For this it would be beneficial to have a flexible and accessible conversion of the countermodels delivered by Nitpick back into Isabelle/HOL input syntax. In fact, an automated conversion of Nitpick's countermodels into the corresponding $C*$-B and $C*$-C conjectures would eventually enable a truly automated exploration and verification of of the modal logic cube with no or minimal handcoding effort. Similarly, for the interactive user a more intuitive presentation of Nitpick's countermodels would be welcome (perhaps similar to the illustrations we used in this paper).

Using the first-order provers E [17], SPASS [19], Z3 [13] and Vampire [12] proved unsuccessful for all *C∗-Isabelle-challenge* problems (unless the right lemmas were given to them). Analyzing the reason for their weakness, as compared to the better performing higher-order automated theorem provers, remains future work. In contrast, the SMT solver CVC4 (via Sledgehammer) was quite successful and contributed five *C∗-Isabelle-challenge* proofs.

Our work motivates further improvements regarding the integration of LEO-II and Satallax: While these systems are able to prove all *∗-Isabelle-challenge* problems their proofs cannot yet be easily replayed or integrated in Isabelle/HOL. There have been recent improvements regarding the transformation of proofs from LEO-II and Satallax to Isabelle/HOL [18], using which all the proofs produced by Satallax and LEO-II in our work could be checked in Isabelle/HOL,[1] but this process still requires some manual work to adapt the output from the ATPs.

Our work also motivates further improvements in higher-order automated theorem provers. For example, for these systems it should be possible to also prove the remaining two *∗-ATP-challenge* problems. Moreover, they needed more than 10 seconds of CPU time in our experiments for the *∗-Isabelle-challenge* problems; it should be possible to prove these theorems much faster.

5 Conclusion

We have fully verified the modal logic cube in Isabelle/HOL. Our solution is simple, elegant, easy to follow, effective and efficient. Proof exchange between systems played a crucial role in our experiments. In particular, we have exploited and combined Nitpick's countermodel-finding capabilities with subsequent calls to the higher-order theorem provers LEO-II and Satallax and the SMT solver CVC4 via Isabelle's Sledgehammer tool. Our experiments also point to several improvement opportunities for Isabelle and the higher-order reasoners, in particular, regarding interaction and proof exchange.

Related experiments have been carried out earlier in collaboration with Geoff Sutcliffe. Similar to and improving on the work reported in [3], these unpublished experiments used the TPTP THF infrastructure directly. However, in that work we did not achieve a 'trusted verification' in the sense of the work presented in this paper. Another improvement in this article has been the use of schematic meta-level working steps (Steps A-D) to systematically convert (counter)models found by Nitpick into conjectures to be investigated.

Future work will explore and evaluate similar logic relationships for other non-classical logics, for example, conditional logics. Any improvements in the mentioned systems, as motivated above, would be very beneficial towards this planned work. Moreover, it would be useful to fully automate the schematic, meta-level working steps (Steps A-D) as applied in our experiments. This would produce a system that would explore logic relations truly automatically (for example, in conditional logics), analogous to what has been achieved here for the modal logic cube.

[1] The proofs and the evaluation workflow can be downloaded from http://christoph-benzmueller.de/papers/pxtp2015-eval.zip

Acknowledgements: We thank Florian Rabe and the anonymous reviewers of this paper for their valuable feedback.

References

[1] J. Backes & C.E. Brown (2010): *Analytic Tableaux for Higher-Order Logic with Choice*. In J. Giesl & R. Hähnle, editors: *Automated Reasoning*, *Lecture Notes in Computer Science* 6173, Springer Berlin Heidelberg, pp. 76–90, doi:10.1007/978-3-642-14203-1_7.

[2] C. Barrett, C.L. Conway, M. Deters, L. Hadarean, D. Jovanovic, T. King, A. Reynolds & C. Tinelli (2011): *CVC4*. In Ganesh Gopalakrishnan & Shaz Qadeer, editors: *Computer Aided Verification*, *LNCS* 6806, Springer Berlin Heidelberg, pp. 171–177, doi:10.1007/978-3-642-22110-1_14.

[3] C. Benzmüller (2010): *Verifying the Modal Logic Cube is an Easy Task (for Higher-Order Automated Reasoners)*. In S. Siegler & N. Wasser, editors: *Verification, Induction, Termination Analysis - Festschrift for Christoph Walther on the Occasion of His 60th Birthday*, *LNCS* 6463, Springer, pp. 117–128, doi:10.1007/978-3-642-17172-7_7. Available at http://christoph-benzmueller.de/papers/B12.pdf.

[4] C. Benzmüller (2013): *Automating Quantified Conditional Logics in HOL*. In F. Rossi, editor: *23rd International Joint Conference on Artificial Intelligence (IJCAI-13)*, Beijing, China, pp. 746–753. Available at http://ijcai.org/papers13/Papers/IJCAI13-117.pdf.

[5] C. Benzmüller & B. Woltzenlogel Paleo (2013): *Gödel's God in Isabelle/HOL*. *Archive of Formal Proofs*. http://afp.sf.net/entries/GoedelGod.shtml, Formal proof development.

[6] C. Benzmüller & B. Woltzenlogel Paleo (2014): *Automating Gödel's Ontological Proof of God's Existence with Higher-order Automated Theorem Provers*. In T. Schaub, G. Friedrich & B. O'Sullivan, editors: *ECAI 2014*, *Frontiers in Artificial Intelligence and Applications* 263, IOS Press, pp. 93 – 98, doi:10.3233/978-1-61499-419-0-93. Available at http://christoph-benzmueller.de/papers/C40.pdf.

[7] C. Benzmüller & L.C. Paulson (2013): *Quantified Multimodal Logics in Simple Type Theory*. *Logica Universalis (Special Issue on Multimodal Logics)* 7(1), pp. 7–20, doi:10.1007/s11787-012-0052-y. Available at http://christoph-benzmueller.de/papers/J23.pdf.

[8] C. Benzmüller, F. Rabe & G. Sutcliffe (2008): *THF0 – The Core of the TPTP Language for Classical Higher-Order Logic*. In A. Armando, P. Baumgartner & G. Dowek, editors: *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, *LNCS* 5195, Springer, pp. 491–506, doi:10.1007/978-3-540-71070-7_41. Available at http://christoph-benzmueller.de/papers/C25.pdf.

[9] C. Benzmüller, F. Theiss, L.C. Paulson & A. Fietzke (2008): *LEO-II - A Cooperative Automatic Theorem Prover for Higher-Order Logic (System Description)*. In A. Armando, P. Baumgartner & G. Dowek, editors: *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, *LNCS* 5195, Springer, pp. 162–170, doi:10.1007/978-3-540-71070-7_14. Available at http://christoph-benzmueller.de/papers/C26.pdf.

[10] J.C. Blanchette & T. Nipkow (2010): *Nitpick: A counterexample generator for higher-order logic based on a relational model finder*. In M. Kaufmann & L. Paulson, editors: *Interactive Theorem Proving (ITP 2010)*, 6172, pp. 131–146, doi:10.1007/978-3-642-14052-5_11.

[11] J. Hurd (2003): *First-Order Proof Tactics in Higher-Order Logic Theorem Provers*. In M. Archer, B. Di Vito & C. Muñoz, editors: *Design and Application of Strategies/Tactics in Higher Order Logics (STRATA 2003)*, NASA Technical Reports, pp. 56–68. Available at http://www.gilith.com/research/papers.

[12] L. Kovács & A. Voronkov (2013): *First-Order Theorem Proving and Vampire*. In N. Sharygina & H. Veith, editors: *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, *LNCS* 8044, Springer, pp. 1–35, doi:10.1007/978-3-642-39799-8_1.

[13] L.M. de Moura & N. Bjørner (2008): *Z3: An Efficient SMT Solver*. In C.R. Ramakrishnan & J. Rehof, editors: *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, *LNCS* 4963, Springer, pp. 337–340, doi:10.1007/978-3-540-78800-3_24.

[14] T. Nipkow, L.C. Paulson & M. Wenzel (2002): *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. *LNCS* 2283, Springer.

[15] L.C. Paulson & J.C. Blanchette (2012): *Three years of experience with Sledgehammer, a Practical Link Between Automatic and Interactive Theorem Provers*. In G. Sutcliffe, S. Schulz & E. Ternovska, editors: *IWIL 2010*, *EPiC Series* 2, EasyChair, pp. 1–11.

[16] F. Rabe, P. Pudlak, G. Sutcliffe & W. Shen (2009): *Solving the $ 100 modal logic challenge*. *Journal of Applied Logic* 7, pp. 113–130, doi:10.1016/j.jal.2007.07.007.

[17] S. Schulz (2013): *System Description: E 1.8*. In K.L. McMillan, A. Middeldorp & A. Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings*, *LNCS* 8312, Springer, pp. 735–743, doi:10.1007/978-3-642-45221-5_49.

[18] N. Sultana (2015): *Higher-order proof translation*. Ph.D. thesis, Computer Laboratory, University of Cambridge. Available as Tech Report UCAM-CL-TR-867.

[19] C. Weidenbach, D. Dimova, A. Fietzke, R. Kumar, M. Suda & P. Wischnewski (2009): *SPASS Version 3.5*. In R.A. Schmidt, editor: *Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction, Montreal, Canada, August 2-7, 2009. Proceedings*, *LNCS* 5663, Springer, pp. 140–145, doi:10.1007/978-3-642-02959-2_10.