Annals of Mathematics and Artificial Intelligence (pre-final version)
The final publication is available at www.springerlink.com

Combining and Automating Classical and Non-Classical Logics in Classical Higher-Order Logics

Christoph Benzmüller

Abstract Numerous classical and non-classical logics can be elegantly embedded in Church's simple type theory, also known as classical higher-order logic. Examples include propositional and quantified multimodal logics, intuitionistic logics, logics for security, and logics for spatial reasoning. Furthermore, simple type theory is sufficiently expressive to model combinations of embedded logics and it has a well understood semantics. Off-the-shelf reasoning systems for simple type theory exist that can be uniformly employed for reasoning *within* and *about* embedded logics and logics combinations.

In this article we focus on combinations of (quantified) epistemic and doxastic logics and study their application for modeling and automating the reasoning of rational agents. We present illustrating example problems and report on experiments with off-the-shelf higher-order automated theorem provers.

Keywords classical and non-classical logics, quantified multimodal logics, logic combinations, classical higher-order logic, semantic embeddings, knowledge representation, higher-order automated theorem proving

1 Introduction

Church's simple type theory ($\mathcal{STT}$) [29,4], also known as classical higher-order logic, is suited as a framework for combining and automating classical and non-classical logics. This claim is what this article investigates. The special focus in this article is on combinations of (quantified) epistemic and doxastic logics and their application for modeling and automating the reasoning of rational agents.

Building reasoning systems that support combinations of logics is a very demanding endeavor. One option is to develop a specific system for each particular logic combination in question. Doing this for all relevant and interesting combinations is hardly feasible. In fact, there is a strong discrepancy between the number of combined reasoning systems that have been sketched on paper (see e.g. [27,28,

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

Articulate Software, Angwin, CA, U.S. E-mail: c.benzmueller@googlemail.com

32, 22, 23] and the references therein), and the number of (non-trivial) combined reasoning systems that have actually been implemented. A second option is to develop flexible, plug-and-play frameworks for various logics and their combinations. Notable developments in this direction include the Logic Workbench, LoTREC, Tableaux Workbench, FaCT, ileanCoP, leanK, and the translation based MSPASS system.[1] However, these systems are mainly restricted to comparably inexpressive propositional logics and their support for flexible logic combinations is still limited. For example, none of these systems supports quantified modal logics and their combinations.

In this article we take on the logics combinations challenge from a fresh perspective, and we propose an analytical, top-down approach based on semantic embeddings in $\mathcal{STT}$. This approach is complementary to synthesis based, bottom-up approaches to combining logics. Even challenge combinations of logics can be achieved in our approach: as an example we outline a combination of spatial and epistemic reasoning. Moreover, our approach supports the analysis and verification of meta-properties of combined logics. It can thus serve as a useful tool for engineers of logic combinations.

Another advantage of our embeddings based approach is that the semantics of $\mathcal{STT}$ is well understood and that powerful proof assistants and automated theorem provers for $\mathcal{STT}$ already exist. 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 to higher-order logic, called TPTP THF [53]. Exploiting this new infrastructure we demonstrate how higher-order automated theorem provers can be employed for reasoning *within* and *about* combinations of logics.

In Sect. 2 we outline our embedding of quantified multimodal logics in $\mathcal{STT}$. Further logic embeddings in $\mathcal{STT}$ are discussed in Sect. 3; our examples comprise intuitionistic logic, access control logics and the region connection calculus. In Sect. 4 we illustrate how the reasoning *about* logics and their combinations is facilitated in our approach, and in Sect. 5 we employ simple examples to demonstrate the application of our approach for reasoning *within* combinations of (quantified) epistemic and doxastic logics. A combination of spatial and epistemic reasoning is studied in Sect. 6. The results of a small case study with off-the-shelf, TPTP THF compliant higher-order theorem provers are presented and discussed in Sect. 7.

This article significantly extends a preceding workshop paper [11].

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

$\mathcal{STT}$ [29] 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 $\to$. 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).

[1] We give the system websites: Logic Workbench: `http://www.lwb.unibe.ch/`, LoTREC: `http://www.irit.fr/Lotrec/`, Tableaux Workbench: `http://twb.rsise.anu.edu.au/`, FaCT: `http://www.cs.man.ac.uk/~horrocks/FaCT/`, ileanCoP: `http://www.leancop.de/ileancop/`, MSPASS: `http://www.cs.man.ac.uk/~schmidt/mspass/`. A good systems overview is provided at: `http://www.cs.man.ac.uk/~schmidt/tools/`

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

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

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 $\neg_{o\to o}$, $\vee_{o\to o\to o}$, $=_{\alpha\to\alpha\to o}$ and $\Pi_{(\alpha\to o)\to o}$ (for each type α).[2] From these connectives, other logical connectives can be defined in the usual way (e.g., $\wedge$ and $\Rightarrow$). We often use binder notation $\forall X_\alpha \mathbin{\blacksquare} s$ for $\Pi_{(\alpha\to o)\to o}(\lambda X_\alpha \mathbin{\blacksquare} s_o)$.[3]

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 [1,2,13,38]. The semantics of choice for our work is Henkin semantics.

Quantified modal logics have been studied by Fitting [30] (further related work is available by Blackburn and Marx [24] and Braüner [25]). 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: every possible world has the same domain. While Fitting [30] 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 $\mathcal{QML}$ thus is

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

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 ($n \geq 0$). Further connectives, quantifiers, and modal operators can be defined as usual. We also obey the usual definitions of free variable occurrences and substitutions.

Fitting introduces three different notions of Kripke semantics for $\mathcal{QML}$: $\mathbf{QS5}\pi^-$, $\mathbf{QS5}\pi$, and $\mathbf{QS5}\pi^+$. In our work [16] we study related notions $\mathbf{QK}\pi^-$, $\mathbf{QK}\pi$, and $\mathbf{QK}\pi^+$ for a modal context $\mathbf{K}$, 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 $\mathcal{QML}^{\mathcal{STT}}$, 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 $\forall X \mathbin{\blacksquare} p$ as $\Pi(\lambda X \mathbin{\blacksquare}.p)$ 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 $\boldsymbol{\Pi}^{\mu\to o}$,

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

[3] We use the $\blacksquare$-notation to avoid brackets; the convention is as follows: $\blacksquare$ stands for a pair of brackets whose right counterpart reaches as far to the right as is consistent with the logical structure and the type structure of an expression.

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:

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

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, $\top = \lambda W_{\mu}.\top, \bot = \neg\top$, $\wedge = \lambda\phi,\psi.\neg(\neg\phi \vee \neg\psi)$, $\supset = \lambda\phi,\psi.\neg\phi \vee \psi$, $\Leftrightarrow = \lambda\phi,\psi.(\phi \supset \psi) \wedge (\psi \supset \phi)$, $\Diamond = \lambda R,\phi.\neg(\Box\, R\,(\neg\phi))$, $\boldsymbol{\Sigma}^{\iota} = \lambda\phi.\neg\boldsymbol{\Pi}^{\iota}(\lambda X.\neg\phi\, X)$, $\boldsymbol{\Sigma}^{\mu\to o} = \lambda\phi.\neg\boldsymbol{\Pi}^{\mu\to o}(\lambda P.\neg\phi\, P)$.

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[4] 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 $X^j \in \mathcal{IV}^{STT}$ $(j = 1, \ldots, n;\ n \geq 0)$ 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 $\boldsymbol{\Pi}^{\iota}(\lambda X.\phi) \in \mathcal{QML}^{STT}$
- if $P \in \mathcal{PV}^{STT}$ and $\phi \in \mathcal{QML}^{STT}$, then $\boldsymbol{\Pi}^{\mu\to o}(\lambda P.\phi) \in \mathcal{QML}^{STT}$

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

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 prover elegantly in the usual syntax. For example, $\Box_r\,\exists P_{\mu\to o}.\,P$ 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 we could define valid simply as $\Pi_{(\mu\to o)\to o}$):

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

Now we can formulate proof problems in $\mathcal{QML}^{STT}$, e.g., valid $\Box_r\,\exists P_{\mu\to o}.\,P$. Using rewriting or definition expanding, we can reduce such proof problems to

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

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 $\Box_r \exists P_{\mu\to o}.\, P$ unfolding and $\beta\eta$-reduction leads to $\forall W_\mu.\forall Y_\mu.\neg r\, W\, Y \vee (\neg\forall X_{\mu\to o}.\neg(X\, Y))$. It is easy to check that this formula is valid in Henkin semantics: put $X = \lambda Y_\mu.\top$.

We have proved soundness and completeness for this embedding [16], that is, for $s \in \mathcal{QML}$ and the corresponding $s_{\mu\to o} \in \mathcal{QML}^{\mathcal{STT}} \subset \mathcal{STT}$ 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 [16].

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 [40]; 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. Moreover, in some examples problems in the remainder we will employ constant symbols.

3 Embeddings of Other Logics in $\mathcal{STT}$

We have studied several other logic embeddings in $\mathcal{STT}$, some of which will be sketched next.

Intuitionistic Logics. Gödels interpretation of propositional intuitionistic logic in propositional modal logic $S4$ [35] can be combined with our results from the previous section in order to provide a sound and complete embedding of propositional intuitionistic logic into $\mathcal{STT}$ [16].

Gödel studies the propositional intuitionistic logic $\mathcal{IPL}$ defined by

$$s, t ::= p \mid \dot{\neg}\, s \mid s \,\dot{\supset}\, t \mid s \,\dot{\vee}\, t \mid p \,\dot{\wedge}\, t$$

He introduces a mapping from $\mathcal{IPL}$ into propositional modal logic $S4$ which maps $\dot{\neg}\, s$ to $\neg\, \Box_r\, s$, $s \,\dot{\supset}\, t$ to $\Box_r\, s \supset \Box_r\, t$, $s \,\dot{\vee}\, t$ to $\Box_r\, s \vee \Box_r\, t$, and $s \,\dot{\wedge}\, t$ to $s \wedge t$.[5] By simply combining Gödel's mapping with our mapping from before we obtain the following embedding of $\mathcal{IPL}$ in $\mathcal{STT}$.

Let $\mathcal{IPL}$ be a propositional intuitionistic logic with atomic primitives $p^1, \ldots,$ p^m $(m \geq 1)$. We define the set $\mathcal{IPL}^{\mathcal{STT}}$ of corresponding propositional intuitionistic logic propositions in $\mathcal{STT}$ as follows:

1. For the atomic $\mathcal{IPL}$ primitives $p^1, \ldots, p^m$ we introduce corresponding $\mathcal{IPL}^{\mathcal{STT}}$ predicate constants $p^1_{\mu\to o}, \ldots, p^m_{\mu\to o}$. Moreover, we provide the single accessibility relation constant $r_{\mu\to\mu\to o}$.

[5] Alternative mappings have been proposed and studied in the literature which we could employ here equally as well.

2. Corresponding to Gödel's mapping we introduce the logical connectives of $\mathcal{IPL}^{\mathcal{STT}}$ as abbreviations for the following λ-terms (we omit the types here):

$$\begin{aligned}
\dot{\neg} &= \lambda\phi\text{.}\lambda W\text{.}\neg\forall V\text{.}\neg r\,W\,V \vee \phi\,V \\
\dot{\supset} &= \lambda\phi\text{.}\lambda\psi\text{.}\lambda W\text{.}\neg(\forall V\text{.}\neg r\,W\,V \vee \phi\,V) \vee (\forall V\text{.}\neg r\,W\,V \vee \psi\,V) \\
\dot{\vee} &= \lambda\phi\text{.}\lambda\psi\text{.}\lambda W\text{.}(\forall V\text{.}\neg r\,W\,V \vee \phi\,V) \vee (\forall V\text{.}\neg r\,W\,V \vee \psi\,V) \\
\dot{\wedge} &= \lambda\phi\text{.}\lambda\psi\text{.}\lambda W\text{.}\neg(\neg\phi\,W \vee \neg\psi\,W)
\end{aligned}$$

3. We define the set of $\mathcal{IPL}^{\mathcal{STT}}$-propositions as the smallest set of simply typed λ-terms for which the following hold:
   – $p^1_{\mu\to o}, \ldots, p^m_{\mu\to o}$ define the atomic $\mathcal{IPL}^{\mathcal{STT}}$-propositions.
   – If ϕ and ψ are $\mathcal{IPL}^{\mathcal{STT}}$-propositions, then so are $\dot{\neg}\,\phi$, $\phi\,\dot{\supset}\,\psi$, $\phi\,\dot{\vee}\,\psi$, and $\phi\,\dot{\wedge}\,\psi$.

The notion of validity we adopt is the same as for $\mathcal{QML}^{\mathcal{STT}}$. However, since Gödel connects $\mathcal{IPL}$ with modal logic $S4$, we transform each proof problem $t \in \mathcal{IPL}$ into a corresponding proof problem t' in $\mathcal{STT}$ of the following form

$$t' := ((\text{valid}\,\forall\phi_{\mu\to o}\text{.}\,\Box_r\,\phi \supset \phi) \wedge (\text{valid}\,\forall\phi_{\mu\to o}\text{.}\,\Box_r\,\phi \supset \Box_r\,\Box_r\,\phi)) \Rightarrow (\text{valid}\,t_{\mu\to o})$$

where $t_{\mu\to o}$ is the $\mathcal{IPL}^{\mathcal{STT}}$ term for t according to our definition above. Alternatively we may translate t into $t'' := ((\text{reflexive}\,r) \wedge (\text{transitive}\,r)) \Rightarrow (\text{valid}\,t_{\mu\to o})$ and provide appropriate definitions for reflexivity and transitivity (cf. Section).

Combining soundness [35] and completeness [41] of Gödel's embedding with Theorem 1 we obtain the following soundness and completeness result: Let $t \in \mathcal{IPL}$ and let $t' \in \mathcal{STT}$ as constructed above. t is valid in propositional intuitionistic logic if and only if t' is valid in $\mathcal{STT}$.

Example problems in intuitionistic logic have been encoded in THF syntax [20] and added to the TPTP THF library[6] and are accessible under identifiers SYO058^4 – SYO074^4.

Access Control Logics. Garg and Abadi recently translated several prominent access control logics into modal logic S4 and proved these translations sound and complete [33]. We have combined this work with our above results in order to obtain a sound and complete embedding of these access control logics in $\mathcal{STT}$ and we have carried out experiments with the prover LEO-II [10]. Example problems have been added to the TPTP THF library and are accessible under identifiers SWV425^x – SWV436^x (for $x \in \{1, \ldots, 4\}$).

Logics for Spatial Reasoning. Evidently, the region connection calculus [48] is a fragment of $\mathcal{STT}$: choose a base type r ('region') and a reflexive and symmetric

[6] TPTP THF problems for various problem categories are available at `http://www.tptp.org/cgi-bin/SeeTPTP?Category=Problems`; all problem identifiers with an '^' in their name refer to higher-order THF problems. The TPTP library meanwhile contains about 3000 example problems in THF syntax.

relation c ('connected') of type $r \to r \to o$ and define (where X, Y, and Z are variables of type r):

$$\begin{aligned}
\text{disconnected} : dc &= \lambda X, Y_\centerdot \neg(c\ X\ Y) \\
\text{part of} : p &= \lambda X, Y_\centerdot \forall Z_\centerdot (c\ Z\ X) \Rightarrow (c\ Z\ Y) \\
\text{identical with} : eq &= \lambda X, Y_\centerdot (p\ X\ Y) \wedge (p\ Y\ X) \\
\text{overlaps} : o &= \lambda X, Y_\centerdot \exists Z_\centerdot (p\ Z\ X) \wedge (p\ Z\ Y) \\
\text{partially overlaps} : po &= \lambda X, Y_\centerdot (o\ X\ Y) \wedge \neg(p\ X\ Y) \wedge \neg(p\ Y\ X) \\
\text{externally connected} : ec &= \lambda X, Y_\centerdot (c\ X\ Y) \wedge \neg(o\ X\ Y) \\
\text{proper part} : pp &= \lambda X, Y_\centerdot (p\ X\ Y) \wedge \neg(p\ Y\ X) \\
\text{tangential proper part} : tpp &= \lambda X, Y_\centerdot (pp\ X\ Y) \wedge \exists Z_\centerdot (ec\ Z\ X) \wedge (ec\ Z\ Y) \\
\text{nontang. proper part} : ntpp &= \lambda X, Y_\centerdot (pp\ X\ Y) \wedge \neg\exists Z_\centerdot (ec\ Z\ X) \wedge (ec\ Z\ Y)
\end{aligned}$$

4 Reasoning about Logics and Combinations of Logics

We illustrate how our approach supports reasoning *about* modal logics and their combinations. First, we focus on the well known relationships between properties of accessibility relations and corresponding modal axioms (respectively axiom schemata) [37]. Such meta-theoretic insights can be elegantly encoded (and, as we will later see, automatically proved) in our approach. We begin with the encoding of various accessibility relation properties in $\mathcal{STT}$:

$$\begin{aligned}
\text{reflexive} &= \lambda R_\centerdot \forall S_\centerdot (R\,S\,S) && (1) \\
\text{symmetric} &= \lambda R_\centerdot \forall S, T_\centerdot (R\,S\,T) \Rightarrow (R\,T\,S) && (2) \\
\text{serial} &= \lambda R_\centerdot \forall S_\centerdot \exists T_\centerdot (R\,S\,T) && (3) \\
\text{transitive} &= \lambda R_\centerdot \forall S, T, U_\centerdot ((R\,S\,T) \wedge (R\,T\,U)) \Rightarrow (R\,S\,U) && (4) \\
\text{euclidean} &= \lambda R_\centerdot \forall S, T, U_\centerdot ((R\,S\,T) \wedge (R\,S\,U)) \Rightarrow (R\,T\,U) && (5) \\
\text{partially_functional} &= \lambda R_\centerdot \forall S, T, U_\centerdot ((R\,S\,T) \wedge (R\,S\,U)) \Rightarrow T = U && (6) \\
\text{functional} &= \lambda R_\centerdot \forall S_\centerdot \exists T_\centerdot (R\,S\,T) \wedge \forall U_\centerdot (R\,S\,U) \Rightarrow T = U && (7) \\
\text{weakly_dense} &= \lambda R_\centerdot \forall S, T_\centerdot (R\,S\,T) \Rightarrow \exists U_\centerdot (R\,S\,U) \wedge (R\,U\,T) && (8) \\
\text{weakly_connected} &= \lambda R_\centerdot \forall S, T, U_\centerdot ((R\,S\,T) \wedge (R\,S\,U)) \Rightarrow \\
&\qquad ((R\,T\,U) \vee T = U \vee (R\,U\,T)) && (9) \\
\text{weakly_directed} &= \lambda R_\centerdot \forall S, T, U_\centerdot ((R\,S\,T) \wedge (R\,S\,U)) \Rightarrow \\
&\qquad \exists V_\centerdot (R\,T\,V) \wedge (R\,U\,V) && (10)
\end{aligned}$$

Remember, that R is of type $\mu \to \mu \to o$ and S, T, U are of type μ. The corresponding axioms are given next.

$$
\begin{array}{rlr}
M: & \forall\phi.\, \Box_r \phi \supset \phi & (11) \\
B: & \forall\phi.\, \phi \supset \Box_r \Diamond_r \phi & (12) \\
D: & \forall\phi.\, \Box_r \phi \supset \Diamond_r \phi & (13) \\
4: & \forall\phi.\, \Box_r \phi \supset \Box_r \Box_r \phi & (14) \\
5: & \forall\phi.\, \Diamond_r \phi \supset \Box_r \Diamond_r \phi & (15)
\end{array}
\qquad
\begin{array}{lr}
\forall\phi.\, \Diamond_r \phi \supset \Box_r \phi & (16) \\
\forall\phi.\, \Diamond_r \phi \Leftrightarrow \Box_r \phi & (17) \\
\forall\phi.\, \Box_r \Box_r \phi \supset \Box_r \phi & (18) \\
\forall\phi,\psi.\, \Box_r ((\phi \wedge \Box_r \phi) \supset \psi) \vee & \\
\qquad \Box_r ((\psi \wedge \Box_r \psi) \supset \phi) & (19) \\
\forall\phi.\, \Diamond_r \Box_r \phi \supset \Box_r \Diamond_r \phi & (20)
\end{array}
$$

Problem 1 For $(k) = (1), \ldots, (10)$ we can now easily formulate the well known correspondence theorems $(k) \Rightarrow (k+10)$ and $(k+10) \Rightarrow (k)$:

$$(1) \Rightarrow (11): \quad \models^{\mathcal{STT}} \forall R.\, (\text{reflexive } R) \Rightarrow (\text{valid } \forall\phi.\, \Box_R \phi \supset \phi) \tag{1.1}$$

$$(2) \Rightarrow (12): \quad \models^{\mathcal{STT}} \forall R.\, (\text{symmetric } R) \Rightarrow (\text{valid } \forall\phi.\, \phi \supset \Box_R \Diamond_R \phi) \tag{1.2}$$

$$\ldots$$

$$(11) \Rightarrow (1): \quad \models^{\mathcal{STT}} \forall R.\, (\text{valid } \forall\phi.\, \Box_R \phi \supset \phi) \Rightarrow (\text{reflexive } R) \tag{1.11}$$

$$(12) \Rightarrow (2): \quad \models^{\mathcal{STT}} \forall R.\, (\text{valid } \forall\phi.\, \phi \supset \Box_R \Diamond_R \phi) \Rightarrow (\text{symmetric } R)) \tag{1.12}$$

$$\ldots$$

$$(20) \Rightarrow (10): \quad \models^{\mathcal{STT}} \forall R.\, (\text{valid } \forall\phi.\, \Diamond_R \Box_R \phi \supset \Box_R \Diamond_R \phi) \Rightarrow (\text{weakly\_direct. } R)) \tag{1.20}$$

Problem 2 There are well known relationships between different modal logics and there exist alternatives for their axiomatization (cf. the relationship map in [34]). For example, for modal logic S5 we may choose 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:

$$
\begin{array}{rclr}
\text{S5} = \text{M5} & \Leftrightarrow & \text{MB5} & (2.1) \\
 & \Leftrightarrow & \text{M4B5} & (2.2) \\
 & \Leftrightarrow & \text{M45} & (2.3) \\
 & \Leftrightarrow & \text{M4B} & (2.4) \\
 & \Leftrightarrow & \text{D4B} & (2.5) \\
 & \Leftrightarrow & \text{D4B5} & (2.6) \\
 & \Leftrightarrow & \text{DB5} & (2.7)
\end{array}
\qquad
\begin{array}{rclr}
\text{KB5} & \Leftrightarrow & \text{K4B5} & (2.8) \\
 & \Leftrightarrow & \text{K4B} & (2.9)
\end{array}
$$

Exploiting the correlations from Problem 1, these problems can be formulated as follows; we give the case for (2.5):

$$\models^{\mathcal{STT}} \forall R.\, ((\text{reflexive } R) \wedge (\text{euclidean } R)) \\ \Leftrightarrow ((\text{serial } R) \wedge (\text{transitive } R) \wedge (\text{symmetric } R)) \tag{2.5}$$

Extending the above ideas, we can in fact employ our approach to effectively verify the entire modal logic cube [12].

Problem 3 We can also encode the Barcan formula and its converse. They are theorems in our approach, which confirms that we are 'constant domain'.

$$BF: \qquad \models^{\mathcal{STT}} \forall R.\forall P. \text{valid } \forall X. \Box_R (P\,X) \supset \Box_R \forall X_\iota. (P\,X) \tag{3.1}$$

$$BF^{-1}: \qquad \models^{\mathcal{STT}} \forall R.\forall P. \text{valid } \Box_R \forall X. (P\,X) \supset \forall X_\iota. \Box_R (P\,X) \tag{3.2}$$

Problem 4 An interesting meta-property for combined logics with modalities $\Diamond_i, \Box_j, \Box_k$, and $\Diamond_l$ is the correspondence between the following axiom and the (i, j, k, l)-confluence property:

$$\begin{aligned} &\models^{\mathcal{STT}} (\text{valid } \forall \phi. (\Diamond_i \Box_j \phi) \supset \Box_k \Diamond_l \phi) \\ &\qquad \Leftrightarrow (\forall A. \forall B. \forall C. (((i\,A\,B) \wedge (k\,A\,C)) \Rightarrow \exists D. ((j\,B\,D) \wedge (l\,C\,D)))) \end{aligned} \tag{4.1}$$

Problem 5 Segerberg [50] discusses a 2-dimensional logic providing two epistemic S5 modalities $\Box_a$ and $\Box_b$. He adds further axioms stating that these modalities are commutative and orthogonal. It actually turns out that orthogonality is already implied in this context. Exploiting the correspondences from Problem 1, this statement can be encoded in our framework as follows:

$$\begin{aligned} &(\text{reflexive } a), (\text{transitive } a), (\text{euclidean } a), (\text{reflexive } b), (\text{transitive } b), (\text{euclidean } b), \\ &(valid\, \forall \phi. \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi) \\ &\models^{\mathcal{STT}} (valid\, \forall \phi, \psi. \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi)) \wedge \\ &\qquad (valid\, \forall \phi, \psi. \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)) \end{aligned} \tag{5.1}$$

Problem 6 Suppose we work with a 2-dimensional logic combining an epistemic S5 modality $\Box_k$ (knowledge) with an doxastic D45 modality $\Box_b$ (belief). Moreover, suppose we add two axioms, as given below, characterizing their relationship. We may then want to check whether knowledge includes belief, that is, whether $\Box_k$ and $\Box_b$ coincide:

$$\begin{aligned} &(\text{reflexive } k), (\text{transitive } k), (\text{euclidean } k), (\text{serial } b), (\text{transitive } b), (\text{euclidean } b), \\ &(valid\, \forall \phi. \Box_k \phi \supset \Box_b \phi), (valid\, \forall \phi. \Box_b \phi \supset \Box_b \Box_k \phi) \\ &\models^{\mathcal{STT}} (valid\, \forall \phi. \Box_b \phi \supset \Box_k \phi) \end{aligned} \tag{6.1}$$

Problem 7 Nguyen [44,45] studies doxastic multimodal logics. The logics he considers are built up from base logic K by adding axioms $A \in \{D, I, 4, 4s, 5, 5s\}$. Axioms D and 4 are defined as at the beginning of this section. However, instead of axiom 5 as given earlier, Nguyen uses the variant $\forall \phi. \neg \Box_i \phi \supset \Box_i \neg \Box_i \phi$. The equivalence of these two variants of axiom 5 can easily be formalized:

$$\begin{aligned} &\models^{\mathcal{STT}} \forall R. (\text{valid } \forall \phi. \neg \Box_R \phi \supset \Box_R \neg \Box_R \phi) \\ &\qquad \Leftrightarrow (\text{valid } \forall \phi. \Diamond_R \phi \supset \Box_R \Diamond_R \phi) \end{aligned} \tag{7.1}$$

Exploiting the correspondence results from before, we may alternatively state:

$$\models^{\mathcal{STT}} \forall R. (\text{valid } \forall \phi. \neg \Box_R \phi \supset \Box_R \neg \Box_R \phi) \Leftrightarrow (\text{euclidean } R) \tag{7.2}$$

Axioms I, 4s, and 5s are given next:

$$
\begin{array}{ll}
I: & \forall\phi_\centerdot \Box_i\,\phi \supset \Box_j\,\phi \quad \text{if } i > j \\
4s: & \forall\phi_\centerdot \Box_i\,\phi \supset \Box_j\,\Box_i\,\phi \\
5s: & \forall\phi_\centerdot \neg\,\Box_i\,\phi \supset \Box_j\,\neg\,\Box_i\,\phi
\end{array}
$$

The inclusion axioms I assume a total ordering $>$ for the considered accessibility relations; they express that whatever agent i beliefs is also believed by agent j. In the doxastic context, axioms 4 and 4s express weak and strong positive introspection. Similarly, axioms 5 and 5s express weak and strong negative introspection.

For I, 4s and 5s the following correspondence theorems hold:

$$\models^{\mathcal{STT}} \forall I_\centerdot \forall J_\centerdot (\text{valid}\ \forall\phi_\centerdot \Box_I\,\phi \supset \Box_J\,\phi) \Leftrightarrow \forall U_\centerdot \forall V_\centerdot (J\ U\ V) \Rightarrow (I\ U\ V) \tag{7.3}$$

$$\models^{\mathcal{STT}} \forall I_\centerdot \forall J_\centerdot (\text{valid}\ \forall\phi_\centerdot \Box_I\,\phi \supset \Box_J\,\Box_I\,\phi) \Leftrightarrow \forall U_\centerdot \forall V_\centerdot \forall W_\centerdot ((J\ U\ V) \wedge (I\ V\ W)) \Rightarrow (I\ U\ W) \tag{7.4}$$

$$\models^{\mathcal{STT}} \forall I_\centerdot \forall J_\centerdot (\text{valid}\ \forall\phi_\centerdot \neg\,\Box_I\,\phi \supset \Box_J\,\neg\,\Box_I\,\phi) \Leftrightarrow \forall U_\centerdot \forall V_\centerdot \forall W_\centerdot ((J\ U\ V) \wedge (I\ U\ W)) \Rightarrow (I\ V\ W) \tag{7.5}$$

Different doxastic logics can be defined from K. For instance, the logic KDI4s5 adds the axioms D, I, 4s, and 5 to base logic K. An interesting observation (cf. Footnote 2 in [45]) is that axioms 5s are already implied in logic KDI4s5. We can easily formalize this claim for particular instances of KDI4s5. For example, for $i, j \in \{r1, r2, r3\}$ with $r3 > r2 > r1$ we get:

$$
\begin{array}{l}
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r1}\,\phi \supset \Diamond_{r1}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r2}\,\phi \supset \Diamond_{r2}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r3}\,\phi \supset \Diamond_{r3}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r2}\,\phi \supset \Box_{r1}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r3}\,\phi \supset \Box_{r1}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r3}\,\phi \supset \Box_{r2}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r1}\,\phi \supset \Box_{r1}\,\Box_{r1}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r1}\,\phi \supset \Box_{r2}\,\Box_{r1}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r1}\,\phi \supset \Box_{r3}\,\Box_{r1}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r2}\,\phi \supset \Box_{r1}\,\Box_{r2}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r2}\,\phi \supset \Box_{r2}\,\Box_{r2}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r2}\,\phi \supset \Box_{r3}\,\Box_{r2}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r3}\,\phi \supset \Box_{r1}\,\Box_{r3}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \Box_{r3}\,\phi \supset \Box_{r2}\,\Box_{r3}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \Box_{r3}\,\phi \supset \Box_{r3}\,\Box_{r3}\,\phi), \\
(\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r1}\,\phi \supset \Box_{r1}\,\neg\,\Box_{r1}\,\phi), (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r2}\,\phi \supset \Box_{r2}\,\neg\,\Box_{r2}\,\phi) \\
(\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r3}\,\phi \supset \Box_{r3}\,\neg\,\Box_{r3}\,\phi) \\
\models^{\mathcal{STT}} (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r1}\,\phi \supset \Box_{r1}\,\neg\,\Box_{r1}\,\phi) \wedge (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r1}\,\phi \supset \Box_{r2}\,\neg\,\Box_{r1}\,\phi) \wedge \\
\quad (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r1}\,\phi \supset \Box_{r3}\,\neg\,\Box_{r1}\,\phi) \wedge (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r2}\,\phi \supset \Box_{r1}\,\neg\,\Box_{r2}\,\phi) \wedge \\
\quad (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r2}\,\phi \supset \Box_{r2}\,\neg\,\Box_{r2}\,\phi) \wedge (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r2}\,\phi \supset \Box_{r3}\,\neg\,\Box_{r2}\,\phi) \wedge \\
\quad (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r3}\,\phi \supset \Box_{r1}\,\neg\,\Box_{r3}\,\phi) \wedge (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r3}\,\phi \supset \Box_{r2}\,\neg\,\Box_{r3}\,\phi) \wedge \\
\quad (\text{valid}\ \forall\phi_\centerdot\ \neg\,\Box_{r3}\,\phi \supset \Box_{r3}\,\neg\,\Box_{r3}\,\phi)
\end{array} \tag{7.6}
$$

In the remainder we refer to the formulas left/above of the $\models^{\mathcal{STT}}$ symbol as axioms for KDI4s5.

The example problems above can be solved automatically by general purpose higher-automated theorem provers (except for (1.19), which was not solved in our experiments). Further details on our experiments and the provers performances will be presented in Section 7.

5 Epistemic and Doxastic Reasoning in Multi-Agent Scenarios

This section illustrates how our approach supports reasoning *within* combined logics. We present example problems that address aspects of epistemic and doxastic reasoning for rational agents. In these examples the knowledge or belief of different agents a_i is modeled by different modalities $\Box_{a_i}$.

First, two example problems on epistemic reasoning are presented. The modeling in both cases adapts Baldoni's work [8].

Problem 8 (Epistemic reasoning: The friends puzzle) *(i) Peter is a friend of John, so if Peter knows that John knows something, then John knows that Peter knows the same thing. (ii) Peter is married, so if Peter's wife knows something, then Peter knows the same thing. John and Peter have an appointment. Let us consider the following situation: (a) Peter knows the time of their appointment. (b) Peter also knows that John knows the place of their appointment. Moreover, (c) Peter's wife knows that if Peter knows the time of their appointment, then John knows that too (since John and Peter are friends). Finally, (d) Peter knows that if John knows the place and the time of their appointment, then John knows that he has an appointment. From this situation we want to prove (e) that each of the two friends knows that the other one knows that he has an appointment.*

For modeling the knowledge of Peter, Peter's wife, and John we consider a 3-dimensional logic combining the modalities $\Box_{\mathrm{p}}$, $\Box_{(\mathrm{w\,p})}$, and $\Box_{\mathrm{j}}$. Actually modeling them as S4 modalities turns out to be sufficient for this example. Hence, we introduce three corresponding accessibility relations j, p, and (w p). The S4 axioms for $x \in \{\mathrm{j}, \mathrm{p}, (\mathrm{w\,p})\}$ are

$$\text{valid}\ \forall\phi\text{.}\ \Box_x\,\phi \supset \phi \qquad (21) \qquad\qquad \text{valid}\ \forall\phi\text{.}\ \Box_x\,\phi \supset \Box_x\,\Box_x\,\phi \qquad (22)$$

As done before, we could alternatively postulate that the accessibility relations are reflexive and transitive. Next, we encode the facts from the puzzle. For (i) we provide a persistence axiom and for (ii) an inclusion axiom:

$$\text{valid}\ \forall\phi\text{.}\ \Box_{\mathrm{p}}\,\Box_{\mathrm{j}}\,\phi \supset \Box_{\mathrm{j}}\,\Box_{\mathrm{p}}\,\phi \qquad (23) \qquad\qquad \text{valid}\ \forall\phi\text{.}\ \Box_{(\mathrm{w\,p})}\,\phi \supset \Box_{\mathrm{p}}\,\phi \qquad (24)$$

Finally, the facts (a)-(d) and the conclusion (e) are encoded as follows (time, place, and appointment are propositional constants, that is, constants of type $\mu \to o$ in our framework):

$$\text{valid}\ \Box_{\mathrm{p}}\,\text{time} \qquad (25)$$
$$\text{valid}\ \Box_{\mathrm{p}}\,\Box_{\mathrm{j}}\,\text{place} \qquad (26)$$
$$\text{valid}\ \Box_{(\mathrm{w\,p})}\,(\Box_{\mathrm{p}}\,\text{time} \supset \Box_{\mathrm{j}}\,\text{time}) \qquad (27)$$
$$\text{valid}\ \Box_{\mathrm{p}}\,\Box_{\mathrm{j}}\,(\text{place} \wedge \text{time} \supset \text{appointment}) \qquad (28)$$
$$\text{valid}\ \Box_{\mathrm{j}}\,\Box_{\mathrm{p}}\,\text{appointment} \wedge \Box_{\mathrm{p}}\,\Box_{\mathrm{j}}\,\text{appointment} \qquad (29)$$

The combined proof problem is

$$(21), \ldots, (28) \models^{\mathcal{STT}} (29) \qquad (8.1)$$

Problem 9 (Wise men puzzle) *Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

We employ a 4-dimensional quantified multimodal logic combining the modalities $\Box_{\mathrm{a}}$, $\Box_{\mathrm{b}}$, and $\Box_{\mathrm{c}}$, for encoding the individual knowledge of the three wise men, with a box operator $\Box_{\mathrm{fool}}$, for encoding common knowledge. The entire encoding consists of the following axioms for $X, Y, Z \in \{a, b, c\}$ and $X \neq Y \neq Z$:

$$\text{valid } \Box_{\mathrm{fool}}\,((\text{ws a}) \vee (\text{ws b}) \vee (\text{ws c})) \tag{30}$$
$$\text{valid } \Box_{\mathrm{fool}}\,((\text{ws } X) \supset \Box_Y\,(\text{ws } X)) \tag{31}$$
$$\text{valid } \Box_{\mathrm{fool}}\,(\neg\,(\text{ws } X) \supset \Box_Y \neg\,(\text{ws } X)) \tag{32}$$
$$\text{valid } \forall\phi_\blacksquare\, \Box_{\mathrm{fool}}\,\phi \supset \phi \tag{33}$$
$$\text{valid } \forall\phi_\blacksquare\, \Box_{\mathrm{fool}}\,\phi \supset \Box_{\mathrm{fool}}\,\Box_{\mathrm{fool}}\,\phi \tag{34}$$
$$\text{valid } \forall\phi_\blacksquare\, \Box_{\mathrm{fool}}\,\phi \supset \Box_{\mathrm{a}}\,\phi \tag{35}$$
$$\text{valid } \forall\phi_\blacksquare\, \Box_{\mathrm{fool}}\,\phi \supset \Box_{\mathrm{b}}\,\phi \tag{36}$$
$$\text{valid } \forall\phi_\blacksquare\, \Box_{\mathrm{fool}}\,\phi \supset \Box_{\mathrm{c}}\,\phi \tag{37}$$
$$\text{valid } \forall\phi_\blacksquare\, \neg\,\Box_{\mathrm{X}}\,\phi \supset \Box_{\mathrm{Y}}\,\neg\,\Box_{\mathrm{X}}\,\phi \tag{38}$$
$$\text{valid } \forall\phi_\blacksquare\, \Box_{\mathrm{X}}\,\phi \supset \Box_{\mathrm{Y}}\,\Box_{\mathrm{X}}\,\phi \tag{39}$$

$$\text{valid } \neg\,\Box_{\mathrm{a}}\,(\text{ws a}) \tag{40}$$
$$\text{valid } \neg\,\Box_{\mathrm{b}}\,(\text{ws b}) \tag{41}$$

From these assumptions we want to conclude that

$$\text{valid } \Box_{\mathrm{c}}\,(\text{ws c}) \tag{42}$$

Axiom (30) says that a, b, or c must have a white spot and that this information is known to everybody. Axioms (31) and (32) express that it is generally known that if someone has a white spot (or not), then the others see and hence know this. $\Box_{\mathrm{fool}}$ is axiomatized as an S4 modality in axioms (33) and (34). For $\Box_{\mathrm{a}}$, $\Box_{\mathrm{b}}$, and $\Box_{\mathrm{c}}$ it is sufficient to consider K modalities. The relation between those and common knowledge ($\Box_{\mathrm{fool}}$ modality) is axiomatized in inclusion axioms (35)–(38). Axioms (38) and (39) encode that whenever a wise man does (not) know something, the others know that he does (not) know this. Axioms (40) and (41) say that a and b do not know whether they have a white spot. Finally, conjecture (42) states that that c knows he has a white spot. The combined proof problem is

$$(30), \ldots, (41) \models^{\mathcal{STT}} (42) \tag{9.1}$$

The above version of the wise men puzzle, which has been adapted from Baldoni, does not take temporal aspects into account. This clearly makes the scenario a kind of unrealistic, at least unsatisfying. However, in our framework we can easily further refine the above formalization so that temporal aspects are also addressed.

This is what we sketch next. In our refined formalization, the axioms (30)–(39) remain the same. However, Axioms (40) and (41) are modified as follows:

$$\text{valid } \Box_{\text{priorG}} \neg \Box_{\text{a}} (\text{ws a}) \tag{43}$$

$$\text{valid } \Box_{\text{priorG}} \Box_{\text{priorG}} \neg \Box_{\text{b}} (\text{ws b}) \tag{44}$$

Axioms (43) and (44) employ a temporal modality $\Box_{\text{priorG}}$, which is intended to model Prior's G operator [47,55] (read $\Box_{\text{priorG}}$ as "It will always be that ..."). (43) expresses that at all future time points the wise man a does not know that he has a white spot. (44) expresses that one further moment later in time it holds that for all futures times the wise man b does not know that he has a white spot. From these assumptions we want to conclude that two time points later from now for all future times it holds that the wise man c does know that he has a white spot:

$$\text{valid } \Box_{\text{priorG}} \Box_{\text{priorG}} \Box_{\text{priorG}} \Box_{\text{c}} (\text{ws c}) \tag{45}$$

We may want to appropriately constrain the accessibility relation associated with the temporal modality $\Box_{\text{priorG}}$. Irreflexivity and transitivity are usually considered minimal requirements for relations intended to characterize the flow of time:

$$(\text{irreflexive priorG}) \tag{46}$$

$$(\text{transitive priorG}) \tag{47}$$

Transitivity is defined as before, and irreflexivity[7] as follows:

$$\text{irreflexive} \quad = \quad \lambda R_{\blacksquare} \forall S_{\blacksquare} \neg (R\, S\, S) \tag{48}$$

Further properties for the time relation, e.g. linearity, can easily be postulated. The extended, temporalized wise men problem is:

$$(30), \ldots, (38), (43), (44), (46), (47) \models^{\mathcal{STT}} (45) \tag{9.2}$$

Example problems (8.1), (9.1), and (9.2) can all be effectively solved by higher-order automated theorem provers.

Several example problems addressing different degrees of belief in agents are presented by Nguyen [44,45]. Nguyen argues for the direct approach and he has developed the modal logic programming system MProlog [43]. Below we adapt some of his example problems. They can be solved automatically by general purpose higher-order automated theorem provers (except for (12.6), which was not solved in our experiments).

Problem 10 (Multimodal logic program about two different degrees of beliefs) Nguyen presents a small multimodal logic program formulated for logic KDI4s5 (cf. Figure 1 in [45]). Translated into our framework, Nguyen's program is formalized as follows (p, q, r, and s are unary predicates, and $r1$ and $r2$ are agent

[7] The irreflexivity property of an accessibility relation cannot be axiomatized with the help of a corresponding modal logic axiom in the sense of Problem 1, since there exists no such formula. Hence, our extended, temporalized wise men problem well demonstrates the flexibility and expressive power of our framework: we can easily directly model and postulate the irreflexivity property.

accessibility relations; we assume that $r2 > r1$, that is, whatever $r2$ believes is also believed by $r1$):

$$\begin{aligned}
&\text{valid } \forall X_\bullet\, \Box_{r2}\, (\Diamond_{r2}\, (q\ X) \supset (p\ X)),\\
&\text{valid } \forall X_\bullet\, \Box_{r1}\, (((r\ X) \wedge (s\ X)) \supset (q\ X)),\\
&\text{valid } \forall X_\bullet\, \Box_{r1}\, ((s\ X) \supset \Box_{r1}\, (r\ X)),\\
&\text{valid } \forall X_\bullet\, \Diamond_{r1}\, (s\ a),\\
&\text{axioms for KDI4s5 (only those mentioning } r1 \text{ or } r2)\\
&\models^{\mathcal{STT}} \text{ valid } \exists X_\bullet\, \Box_{r1}\, (p\ X) \qquad (10.1)
\end{aligned}$$

This example problem is actually already solvable in logic KI4s. Hence, we introduce the following variation: Problem (10.1.KI4s) is identical to (10.1) except that the axioms D and 5 are omitted.

Problem 11 (Multimodal logic program about five different degrees of beliefs) For agents $i \in \{r1, \ldots, r5\}$ the converted clauses and queries of this self-explaining program of Nguyen [44] are:

$$\begin{aligned}
&\text{valid } \forall X_\bullet\, (\text{maths_teacher } X) \supset \Box_{r4}\, (\text{good_in_maths } X),\\
&\text{valid } \forall X_\bullet\, \Box_{r5}\, (\Box_i\, (\text{mathematician } X) \supset \Box_i\, (\text{good_in_maths } X)),\\
&\text{valid } \forall X_\bullet\, \Box_{r3}\, ((\text{maths_student } X) \supset \Diamond_i\, (\text{good_in_maths } X)),\\
&\text{valid } \forall X_\bullet\, \Box_{r3}\, ((\text{physics_student } X) \supset \Diamond_i\, (\text{good_in_physics } X)),\\
&\text{valid } \forall X_\bullet\, \Box_{r2}\, ((\text{good_in_physics } X) \supset \Diamond_{r2}\, (\text{good_in_maths } X)),\\
&\text{valid } (\text{maths_teacher john}),\\
&\text{valid } \Box_{r2}\, (\text{mathematician tom}),\\
&\text{valid } \Box_{r5}\, (\text{maths_student peter}),\\
&\text{valid } \Box_{r5}\, (\text{physics_student mike}),\\
&\text{axioms for KDI4s5 (for } r1, \ldots, r5)\\
&\models^{\mathcal{STT}} \text{ valid } \exists X_\bullet\, \Box_{r4}\, (\text{good_in_maths } X) \qquad (11.1)\\
&\ldots \models^{\mathcal{STT}} \text{ valid } \exists X_\bullet\, \Box_{r2}\, (\text{good_in_maths } X) \qquad (11.2)\\
&\ldots \models^{\mathcal{STT}} \text{ valid } \exists X_\bullet\, \Diamond_{r1}\, (\text{good_in_maths } X) \qquad (11.3)
\end{aligned}$$

The query (11.1) already has a solution in base logic K. We therefore define the problem variant (11.1.K), which is identical to (11.1), except that the axioms for KDI4s5 are omitted.

Problem 12 (A company with different branches) This example problem of Nguyen [44] models three different branches of a company as different agents $a1$, $a2$, and $a3$. The data and knowledge of each branch (which may contain noise and which may not be highly recognized by other branches) is modeled by Nguyen as belief rather than knowledge. Hence, $\Box_{a1}\, \phi$ and $\Diamond_{a1}\, \psi$ express that branch $a1$ beliefs ϕ and considers ψ possible. The logic employed by Nguyen is KDI4s5s, which is equivalent to KDI4s5 (cf. Problem (7.6)). In addition to the different company branches, the example assumes a central database (modeled as agent $a4$), which in a sense hides information stemming from the other branches, and which is used for communication with the user(s). Translated into our framework the problem's clauses are formalized as follows:

$$\mathrm{B1} = \begin{cases} \text{valid } \Box_{a1}\,(\text{likes jan cola}), \\ \text{valid } \Box_{a1}\,(\text{likes piotr pepsi}), \\ \text{valid } \forall X\centerdot\, \Box_{a1}\,((\text{likes } X \text{ pepsi}) \supset (\Diamond_{a1}\,(\text{likes } X \text{ cola}))), \\ \text{valid } \forall X\centerdot\, \Box_{a1}\,((\text{likes } X \text{ cola}) \supset (\Diamond_{a1}\,(\text{likes } X \text{ pepsi}))) \end{cases}$$

$$\mathrm{B2} = \begin{cases} \text{valid } \Box_{a2}\,(\text{likes jan pepsi}), \\ \text{valid } \Box_{a2}\,(\text{likes piotr cola}), \\ \text{valid } \Box_{a2}\,(\text{likes piotr beer}), \\ \text{valid } \forall X\centerdot\, \Box_{a2}\,((\text{likes } X \text{ pepsi}) \supset (\text{likes } X \text{ cola})), \\ \text{valid } \forall X\centerdot\, \Box_{a2}\,((\text{likes } X \text{ cola}) \supset (\text{likes } X \text{ pepsi})) \end{cases}$$

$$\mathrm{B3} = \begin{cases} \text{valid } \Box_{a3}\,(\text{likes jan cola}), \\ \text{valid } \Box_{a3}\,(\text{likes piotr pepsi}), \\ \text{valid } \Box_{a3}\,(\text{likes piotr beer}), \\ \text{valid } \forall X\centerdot\, \forall Y\centerdot \\ \qquad \Box_{a3}\,(((\text{likes } X\ Y) \wedge (\Box_{a1}\,(\text{likes } X\ Y)) \wedge (\Box_{a2}\,(\text{likes } X\ Y))) \\ \qquad\quad \supset (\text{very\_much\_likes } X\ Y)) \end{cases}$$

$$\mathrm{B4} = \begin{cases} \text{valid } \forall X\centerdot\, \forall Y\centerdot\, \Box_{a3}\,(\text{very\_much\_likes } X\ Y) \supset (\text{very\_much\_likes } X\ Y), \\ \text{valid } \forall X\centerdot\, \forall Y\centerdot\, \Diamond_{a3}\,(\text{very\_much\_likes } X\ Y) \supset (\text{likes } X\ Y), \\ \text{valid } \forall X\centerdot\, \forall Y\centerdot\, \Diamond_{a1}\,(\text{likes } X\ Y) \supset (\text{possibly\_likes } X\ Y), \\ \text{valid } \forall X\centerdot\, \forall Y\centerdot\, \Diamond_{a2}\,(\text{likes } X\ Y) \supset (\text{possibly\_likes } X\ Y), \\ \text{valid } \forall X\centerdot\, \forall Y\centerdot\, \Diamond_{a3}\,(\text{likes } X\ Y) \supset (\text{possibly\_likes } X\ Y) \end{cases}$$

We formalize various queries (queries (12.4)–(12.11) verify possible answer instantiations for the query variables in (12.1)–(12.3)):

$$\mathrm{B1}, \mathrm{B2}, \mathrm{B3}, \mathrm{B4}, \text{axioms of KBDI4s5 (for } a1,\ a2,\ a3,\ a4)$$

$$\models^{\mathcal{STT}} \text{ valid } \exists X\centerdot\, \exists Y\centerdot\, (\text{very\_much\_likes } X\ Y) \tag{12.1}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid } \exists X\centerdot\, \exists Y\centerdot\, (\text{likes } X\ Y) \tag{12.2}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid } \exists X\centerdot\, \exists Y\centerdot\, (\text{possibly\_likes } X\ Y) \tag{12.3}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (very\_much\_likes jan cola)} \tag{12.4}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (likes jan cola)} \tag{12.5}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (likes piotr pepsi)} \tag{12.6}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (possibly\_likes jan cola)} \tag{12.7}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (possibly\_likes piotr pepsi)} \tag{12.8}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (possibly\_likes jan pepsi)} \tag{12.9}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (possibly\_likes piotr cola)} \tag{12.10}$$

$$\ldots \models^{\mathcal{STT}} \text{ valid (possibly\_likes piotr beer)} \tag{12.11}$$

Some queries are already solvable in K, and therefore we introduce the following problem variations, where the axioms for KDI4s5 are omitted: (12.1.K), (12.2.K), (12.3.K), (12.4.K), (12.5.K), (12.7.K).

6 A Combination of Spatial and Epistemic Reasoning

In this section we illustrate that our approach is not limited to combinations of modal logics. For this, we combine the region connection calculus [48] with an epistemic logic.

Problem 13 A trivial example problem for the region connection calculus is (adapted from [32], p. 80):

$$\begin{aligned}
&(\mathit{tpp}\ \text{catalunya spain}),\\
&(\mathit{ec}\ \text{spain france}),\\
&(\mathit{ntpp}\ \text{paris france}),\\
&\models^{\mathcal{STT}}\ (\mathit{dc}\ \text{catalunya paris}) \wedge (\mathit{dc}\ \text{spain paris})
\end{aligned} \tag{13.1}$$

The assumptions express that (i) Catalunya is a border region of Spain, (ii) Spain and France are two different countries with a common border, and (iii) Paris is an inner region of France. The conjecture is that (iv) Catalunya and Paris are disconnected as well as Spain and Paris.

Problem 14 Within our $\mathcal{STT}$ framework we can easily put such spatial reasoning problems in an epistemic context, that is, we can model common spatial knowledge and also the individual spatial knowledge of single agents. Similar to before, we use $\Box_{\text{fool}}$ for modeling common knowledge (fool) and $\Box_{\text{a}}$ for modeling the knowledge of agent a:

$$\begin{aligned}
&\text{valid}\ \forall \phi_{\blacksquare}\ \Box_{\text{fool}}\, \phi \supset \Box_{\text{a}}\, \phi,\\
&\text{valid}\ \Box_{\text{a}}\, \lambda W_{\blacksquare}\, (\mathit{tpp}\ \text{catalunya spain}),\\
&\text{valid}\ \Box_{\text{fool}}\, \lambda W_{\blacksquare}\, (\mathit{ec}\ \text{spain france}),\\
&\text{valid}\ \Box_{\text{a}}\, \lambda W_{\blacksquare}\, (\mathit{ntpp}\ \text{paris france})\\
&\models^{\mathcal{STT}}\ \text{valid}\ \Box_{\text{a}}\, \lambda W_{\blacksquare}\, (\mathit{dc}\ \text{catalunya paris}) \wedge (\mathit{dc}\ \text{spain paris})
\end{aligned} \tag{14.1}$$

We here express that (ii) from above, namely that Spain and France have a common border, is commonly known, while (i) and (iii) are not. Instead we assume that (i) and (iii) are known to the educated agent a. In this situation, conjecture (iv) is still known to agent a. But (iv) is not commonly known, and the following statement is countersatisfiable:

$$\ldots \models^{\mathcal{STT}}\ \text{valid}\ \Box_{\text{fool}}\, \lambda W_{\blacksquare}\, (\mathit{dc}\ \text{catalunya paris}) \wedge (\mathit{dc}\ \text{spain paris}) \tag{14.2}$$

We present some further example queries; they are all theorems:

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\neg(po \text{ catalunya paris}) \tag{14.3}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.(o \text{ catalunya spain}) \wedge (o \text{ france paris}) \tag{14.4}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\neg(eq \text{ catalunya paris}) \tag{14.5}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.(eq \text{ catalunya paris}) \Rightarrow (o \text{ france spain}) \tag{14.6}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\exists Z.\neg(o\ Z \text{ paris}) \wedge \neg(eq\ Z \text{ spain}) \tag{14.7}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\forall Z.(p\ Z \text{ catalunya}) \Rightarrow \neg(p\ Z \text{ paris}) \tag{14.8}$$

$$\begin{aligned}\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\forall Z.&((ntpp \text{ france } Z) \wedge (ntpp \text{ spain } Z))\\ &\Rightarrow ((pp \text{ paris } Z) \wedge (pp \text{ catalunya } Z))\end{aligned} \tag{14.9}$$

$$\begin{aligned}\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\forall Z.&((ntpp \text{ france } Z) \wedge (ntpp \text{ spain } Z))\\ &\Rightarrow ((ntpp \text{ paris } Z) \wedge (ntpp \text{ catalunya } Z))\end{aligned} \tag{14.10}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\exists Z.\exists Y.\neg(eq\ Z\ Y) \tag{14.11}$$

$$\begin{aligned}\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{fool}}\, \lambda W.\forall Z.\forall Y.&((p\ Z \text{ spain}) \wedge (p\ Y \text{ france}))\\ &\Rightarrow \neg(o\ Z\ Y)\end{aligned} \tag{14.12}$$

$$\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{fool}}\, \lambda W.\neg\exists Z.\forall Y.(ntpp\ Z\ Y) \tag{14.13}$$

$$\begin{aligned}\ldots \models^{\mathcal{STT}} \text{valid } \Box_{\mathrm{a}}\, \lambda W.\forall Z.&((o\ Z \text{ paris}) \wedge (o\ Z \text{ catalunya}))\\ &\Rightarrow ((o\ Z \text{ spain}) \wedge (o\ Z \text{ france}))\end{aligned} \tag{14.14}$$

In order to facilitate the combination of spatial and epistemic reasoning we have lifted the region connection calculus propositions (originally of type o) to modal propositions of type $\mu \to o$ by λ-abstraction. This way the region connection calculus statements can be applied to possible worlds — they evaluate statically though for all possible worlds, since the λ-abstracted variable W is fresh for the encapsulated region connection calculus propositions, for example:

$$\underbrace{(tpp \text{ catalunya spain})}_{\text{type } o} \longrightarrow \underbrace{(\lambda W.(tpp \text{ catalunya spain}))}_{\text{type } \mu\to o}$$

7 Experiments

In this section we apply off-the-shelf automated reasoning systems for $\mathcal{STT}$ in order to solve the problems presented in the previous sections. The particular theorem provers employed in our case study are:[8]

LEO-II (version v1.2.5). LEO-II [21], the successor of LEO [14], is an automated theorem prover for $\mathcal{STT}$ which is based on extensional higher-order resolution. More precisely, LEO-II employs a refinement of extensional higher-order RUE resolution [9]. LEO-II is designed to cooperate with specialist systems for fragments of higher-order logic. By default, LEO-II cooperates with the first-order

[8] The short system sketches given here have been extracted and adapted from `http://tptp.org/CASC/J5/SystemDescriptions.html` and [53]; see there for further information.

ATP systems E [49]. LEO-II is often too weak to find a refutation amongst the steadily growing set of clauses on its own. However, some of the clauses in LEO-II's search space attain a special status: they are first-order clauses modulo the application of an appropriate transformation function. The default transformation is Hurd's fully typed translation [39]. Therefore, LEO-II launches a cooperating first-order ATP system every n iterations of its (standard) resolution proof search loop (e.g., n = 10). If the first-order ATP system finds a refutation, it communicates its success to LEO-II, which causes LEO-II to terminate and to report overall success. Communication between LEO-II and the cooperating first-order ATP system uses the TPTP language and standards.

TPS (version 3.080227G1d). TPS is a fully automated version of the higher-order theorem proving system TPS [5,6]. TPS can be used to prove theorems of $\mathcal{STT}$ automatically, interactively, or semi-automatically. When searching for a proof automatically, TPS first searches for an expansion proof [42] or an extensional expansion proof [26] of the theorem. Part of this process involves searching for acceptable matings [3]. Using higher-order unification, a pair of occurrences of subformulae (which are usually literals) is mated appropriately on each vertical path through an expanded form of the theorem to be proved. The behavior of TPS is controlled by hundreds of flags. A set of flags, with values for them, is called a mode. Forty-nine modes have been found that collectively suffice for automatically proving virtually all the theorems that TPS has proved automatically thus far. As the modes have quite different capabilities, and it is expected that any proofs found by any mode will be found quickly, strategy scheduling the modes is a simple way of obtaining greater coverage. A Perl script has been used to do this, running each of the 49 modes for a specified amount of time.

Satallax (version 1.4). Satallax is a higher-order automated theorem prover with additional model finding capabilities. The system is based on a complete ground tableau calculus for $\mathcal{STT}$ with a choice operator [7]. An initial tableau branch is formed from the axioms of the problem and negation of the conjecture (if any is given). From this point on, Satallax tries to determine unsatisfiability or satisfiability of this branch. Satallax progressively generates higher-order formulae and corresponding propositional clauses. These formulae and propositional clauses correspond to instances of the tableau rules. Satallax uses the SAT solver MiniSat as an engine to test the current set of propositional clauses for unsatisfiability. If the clauses are unsatisfiable, then the original branch is unsatisfiable. If there are no quantifiers at function types, the generation of higher-order formulae and corresponding clauses may terminate. In such a case, if MiniSat reports the final set of clauses as satisfiable, then the original set of higher-order formulae is satisfiable (by a standard model in which all types are interpreted as finite sets).

IsabelleP (version 2009-2). The higher-order proof assistant Isabelle/HOL [46] is normally used interactively. In this mode it is possible to apply various automated tactics that attempt to solve the current goal without further user interaction. Examples of these tactics are *blast*, *auto*, and *metis*. It is also possible to run Isabelle from the command line, passing in a theory file containing a lemma to prove. Finally, Isabelle theory files can include ML code to be executed when the file is processed. While it was probably never intended to use Isabelle as a fully automatic system, these three features have been combined to implement a fully automatic Isabelle/HOL, called IsabelleP. The TPTP2X Isabelle format module outputs a THF problem in Isabelle/HOL syntax, augmented with ML code that runs tactics in sequence, each with a CPU time limit until one succeeds or all fail.

The reasoning systems described above are available online via the SystemOnTPTP tool [51] and they support the new TPTP THF infrastructure for typed higher-order logic [20]. Exploiting the TPTP World infrastructure [52], all experiment runs reported below were done remotely at the University of Miami on 2.80GHz computers with 1GB memory and running the Linux operating system.

The axiomatizations of $\mathcal{QML}^{\mathcal{STT}}$ and the region connections calculus are available as LCL013^0.ax and LCL014^0.ax in the TPTP library.[9] Satallax proves the satisfiability of LCL013^0.ax and LCL014^0.ax in 0.29 seconds and 0.28 seconds respectively. Countersatisfiability of problem (14.2) cannot be detected by any of the above reasoners.

Table 1 presents the further results of our experiments. All example problems in this table are actually theorems. Only three of these theorems cannot be solved by any of the above provers: (1.19), (12.6), and (14.10). For all other example problems at least one system finds a proof.

Probl. – TPTP id	LEO-II	TPS	Satallax	IsabelleP
Reasoning about Logics and Combined Logics				
(1.1)/LCL699^1	.03/.03	.36/.37	.28/.28	4.37/19.38
(1.2)/LCL700^1	.04/.04	.37/.37	.29/.29	5.40/28.67
(1.3)/LCL701^1	.04/.04	.37/.37	.32/.32	4.49/19.52
(1.4)/LCL702^1	.05/.05	.38/.38	.39/.39	5.40/35.49
(1.5)/LCL703^1	.04/.05	.38/.38	.39/.40	5.42/35.45
(1.6)/LCL704^1	.04/.04	.39/.39	.48/.52	4.48/19.49
(1.7)/LCL705^1	.12/.11	–/45.34	.57/.57	4.58/19.61
(1.8)/LCL706^1	.05/.05	.38/.38	.46/.46	4.50/19.54
(1.9)/LCL707^1	.07/.07	.48/.48	.52/.52	4.50/19.57
(1.10)/LCL708^1	.06/.05	.40/.49	.49/.49	4.44/19.58
(1.11)/LCL709^1	.04/.03	.36/.37	–/–	4.47/19.52
(1.12)/LCL710^1	.14/.14	.37/.37	–/–	9.46/84.58
(1.13)/LCL711^1	.04/.04	.38/.39	–/54.45	4.51/19.53
(1.14)/LCL712^1	.05/.05	.38/.38	–/–	4.51/19.56
(1.15)/LCL713^1	10.74/120.73	.39/.40	–/–	8.80/83.84
(1.16)/LCL714^1	.10/.10	.71/.70	–/–	6.65/51.69
(1.17)/LCL715^1	–/–	–/69.18	–/–	–/–
(1.18)/LCL716^1	–/–	6.67/6.69	–/–	–/–
(1.19)/LCL717^1	–/–	–/–	–/–	–/–
(1.20)/LCL718^1	.11/.11	.40/.40	–/–	5.43/23.42
(2.1)/LCL859^1	.11/.10	.50/.49	.52/.52	4.44/19.50
(2.2)/LCL860^1	.17/.17	–/21.71	2.93/2.92	4.53/19.54
(2.3)/LCL861^1	.12/.12	9.07/9.07	14.83/14.37	4.46/19.54
(2.4)/LCL862^1	.11/.11	1.29/1.29	3.04/3.04	4.61/19.68
(2.5)/LCL863^1	.11/.12	1.74/1.74	–/84.42	5.21/21.14
(2.6)/LCL864^1	.17/.17	–/41.62	–/84.44	10.67/21.14
(2.7)/LCL865^1	.11/.11	.65/.65	–/–	4.62/19.69
(2.8)/LCL866^1	.16/.16	2.35/2.35	3.48/3.48	4.45/19.50
(2.9)/LCL867^1	.11/.11	.95/.97	3.45/3.46	4.53/19.56
(3.1)/LCL606^1*	.04/.04	.37/.37	.29/.29	4.37/19.38
(3.2)/LCL611^1*	.03/.03	.34/.35	.28/.28	4.01/19.35

9 For the axiomatization of $\mathcal{IPL}^{\mathcal{STT}}$ see LCL010^0.ax and for Access Control Logic see the files SWV008^0.ax, SWV008^1.ax, and LCL008^0.ax.

Probl. – TPTP id	LEO-II	TPS	Satallax	IsabelleP
(4.1)/LCL872^1	.12/.12	.49/.49	–/–	4.94/19.60
(5.1)/LCL873^1	.14/.14	.96/.95	5.71/5.70	12.64/147.74
(6.1)/LCL874^1	1.74/1.73	–/–	–/–	–/–
(7.1)/LCL877^1	9.63/121.03	.63/.64	–/–	–/–
(7.2)/LCL877^2	.09/.09	.43/.44	–/–	5.49/35.66
(7.3)/LCL878^1	.07/.07	.39/.39	–/–	4.59/19.61
(7.4)/LCL879^1	.09/.09	.42/.41	–/–	4.64/19.69
(7.5)/LCL880^1	.09/.09	.41/.42	–/–	4.68/19.68
(7.6)/LCL881^1	10.63/10.66	–/–	–/–	–/–
Reasoning within Combined Logics				
(8.1)/PUZ086^1	.15/.15	–/–	.75/.76	12.87/147.95
(9.1)/PUZ087^1	3.27/4.17	–/–	–/–	–/–
(9.2)/PUZ087^2	.84/.83	–/–	–/–	–/–
(10.1)/AGT027^1	.87/.86	–/–	–/–	–/–
(10.1.KI4s)/AGT027^2	11.08/120.63	–/–	–/–	–/–
(11.1)/AGT028^1	.88/.92	–/–	–/31.15	–/150.03
(11.1.K)/AGT028^2	.24/.24	2.24/2.24	2.52/2.52	6.44/22.61
(11.2)/AGT029^1	.96/.96	–/–	–/–	–/–
(11.3)/AGT030^1	4.01/4.07	–/–	–/–	–/–
(12.1)/AGT031^1	3.35/40.42	–/–	–/–	–/–
(12.1.K)/AGT031^2	.28/.28	–/102.19	–/–	–/–
(12.2)/AGT032^1	3.35/40.42	–/–	–/42.85	–/–
(12.2.K)/AGT032^2	.29/.28	–/101.44	7.93/8.02	–/136.00
(12.3)/AGT033^1	.42/.42	–/96.27	5.18/5.20	12.56/132.66
(12.3.K)/AGT033^2	.18/.18	1.57/1.58	1.24/1.25	12.03/132.16
(12.4)/AGT034^1	3.40/40.49	–/–	–/84.91	–/–
(12.4.K)/AGT034^2	.34/.35	–/101.47	–/31.84	–/138.53
(12.5)/AGT035^1	3.38/40.49	–/–	–/–	–/–
(12.5.K)/AGT035^2	.33/.33	–/101.09	–/–	–/134.14
(12.6)/AGT036^1	–/–	–/–	–/–	–/–
(12.7)/AGT037^1	.78/.79	–/–	5.51/5.59	12.48/132.59
(12.7.K)/AGT037^2	.18/.18	1.47/1.48	1.40/1.40	12.03/132.18
(12.8)/AGT038^1	.39/.39	–/24.57	5.25/5.25	12.46/132.62
(12.9)/AGT039^1	.70/.69	–/–	5.41/5.44	12.47/132.69
(12.10)/AGT040^1	.65/.65	–/–	5.33/5.35	12.52/132.59
(12.11)/AGT031^1	.40/.41	–/24.61	5.25/5.24	12.50/132.64
(13.1)/GEG002^1	11.23/122.29	–/–	–/–	13.74/158.34
(14.1)/GEG003^1	–/138.29	–/–	–/–	–/–
(14.3)/GEG005^1	.56/.56	–/–	–/–	–/–
(14.4)/GEG006^1	6.60/44.81	–/–	12.63/12.74	13.23/148.24
(14.5)/GEG007^1	.50/.49	–/–	–/–	–/–
(14.6)/GEG008^1	.44/.44	–/–	3.57/3.67	–/–
(14.7)/GEG009^1	.69/.70	–/–	–/–	–/–
(14.8)/GEG010^1	.59/.59	–/–	–/–	–/–
(14.9)/GEG011^1	.90/.89	–/–	1.13/1.14	–/149.16
(14.10)/GEG012^1	–/–	–/–	–/–	–/–
(14.11)/GEG013^1	.36/.36	10.29/10.34	14.42/14.41	13.10/37.32
(14.12)/GEG014^1	.42/.42	–/33.62	2.92/2.98	–/–
(14.13)/GEG015^1	.45/.45	1.39/1.41	2.71/2.72	5.93/35.99
(14.14)/GEG016^1	7.21/81.78	–/–	–/–	–/–
Total solved	73/74	40/50	36/43	48/53

Table 1: Performance results of $\mathcal{STT}$ provers for example problems.

The first column in Table 1 lists the problem numbers, and additionally a respective TPTP identifier.[10] The remaining columns present the performance results of our theorem provers in seconds for each example problem. The provers were applied twice to every problem. The first run was with a prover timeout of just 15 seconds and the second run with a prover timeout of 200 seconds. For instance, the result entry 4.37/19.38 for Problem (1.1) in the column of IsabelleP says that IsabelleP proved this problem in the first run in 4.37 seconds, while it needed 19.38 seconds in the second run. This effect is caused by the fact that IsabelleP — like LEO-II and TPS — applies a time slicing approach, in which the initially given time resource is split into slices, and in each time slice a different IsabelleP tactic is applied. Each problem that IsabelleP could solve in 15 seconds, it could also solve in 200 seconds. However, the performance of IsabelleP significantly decreased for all example problems in the second run. For LEO-II we only occasionally witnessed such a performance decrease in the second run, and TPS and Satallax did not show it all in our case study. There is only one problem, namely (14.1), which LEO-II could solve only in the second run.[11] For IsabelleP there are 5 such problems, for Satallax 7, and for TPS 10.

Overall, LEO-II was by far the strongest prover in our experiments. With a timeout of 200 seconds, LEO-II could solve 75 out of the 80 proof problems in Table 1. IsabelleP comes second with 53 solved problems, followed by TPS with 50 and Satallax with 43. The distance between LEO-II and the other provers was even bigger for the 15 seconds timeout setting: LEO-II still proved 74 problems. IsabelleP comes second with 48 solved problems, followed by TPS and Satallax with 40 and 36.

There were no problems in our experiments that could be solved by IsabelleP or Satallax but not by LEO-II. Hence, both provers are subsumed by LEO-II in our case study. There were 15 problems that only LE0-II could solve, and there were 2 problems that only TPS could solve.

In summary, all but three example problems could be solved by our off-the-shelf provers, and most results were obtained in less than second.

We are not aware of any other running system, in particular no prover in the direct approach, that can handle all of the above problems.

8 Conclusion

Our overall goal has been to show that prominent classical and non-classical logics and their combinations can be elegantly modeled as fragments of classical higher-order logic $\mathcal{STT}$ and (partly) automated with off-the-shelf higher-order theorem provers.

Our experiments are encouraging and they provide first evidence for our claim that $\mathcal{STT}$ is suited as a generic and flexible framework for combining classical and

[10] See `http://www.tptp.org/cgi-bin/SeeTPTP?Category=Problems`; the TPTP problem files marked with * are only similar and not identical to the problem files used in the experiments.

[11] Note, that version v1.1 of LEO-II, which did not yet employ time slicing, did in fact show a significantly better performance for this particular example [11]. Hence, (14.1) is one of those interesting cases where LEO-II got worse after introduction of the time slicing approach. Overall, time-slicing did significantly increase the number of problems that can be solved by LEO-II in CASC-J5.

non-classical logics. It is obvious, however, that $\mathcal{STT}$ provers should be significantly improved for fruitful application to more challenge problems in practice. The author is convinced that significant improvements — in particular for fragments of $\mathcal{STT}$ as illustrated in this article — are possible and that they will be fostered by the new TPTP infrastructure and the new yearly higher-order CASC competitions.

Note that when working with our provers from within a proof assistant such as Isabelle/HOL the user may also provide interactive help if the reasoning tasks are still to challenging, for example, by formulating some lemmas or by splitting proof tasks in simpler subtasks.

An advantage of our approach also is that provers such as our LEO-II are generally capable of producing verifiable proof output, though much further work is needed to make these proof protocols exchangeable between systems or to explain them to humans. Furthermore, it may be possible to formally verify the entire theory of our embedding(s) within a proof assistant.

Future work includes the study of the overall philosophical and computational characteristics of our approach, and the investigation of its range and its limitations. In particular, prominent notions for combining logics, such as fusions [54], products [50] and fibrings [31], will be re-investigated in the context of our framework. For this, note that the accessibility relations r associated with $\Box_r$ operators in this article all range over the same world type μ. In this sense the particular notion of logic combination employed here is related to that of a fusion. In order to model our example problems as products, different world types μ_i can be introduced and the modal connectives can be copied for each of those. Moreover, respective axioms can be postulated to model the desired product properties.

The work presented in this article has its roots in the LEO-II project (in 2006/2007 at University of Cambridge, UK) in which we first studied and employed the presented embedding of quantified and propositional multimodal logics in $\mathcal{STT}$ [15,17]. This research, amongst others, is currently continued in the DFG project ONTOLEO (BE 2501/6-1). In ONTOLEO we study whether our approach can be applied to automate modalities in ontology reasoning [19,18]. However, our work is obviously relevant also for many other application directions. Studying the scalability of our approach for a range of these applications is thus important future work.

Acknowledgment: The author is indebted to Larry Paulson, Geoff Sutcliffe, and Chad Brown. 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. Moreover, Chad Brown provided some early inspiration for this research with a presentation at Loria, Nancy in 2005. Further thanks goes to the developers of the $\mathcal{STT}$ provers employed in this article.

References

1. Peter B. Andrews. General models and extensionality. *Journal of Symbolic Logic*, 37:395–397, 1972.

2. Peter B. Andrews. General models, descriptions, and choice in type theory. *Journal of Symbolic Logic*, 37:385–394, 1972.
3. Peter B. Andrews. Theorem Proving via General Matings. *Journal of the ACM*, 28(2):193–214, 1981.
4. Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Kluwer Academic Publishers, second edition, 2002.
5. Peter B. Andrews, Matt Bishop, Sunil Issar, Dan Nesmith, Frank Pfenning, and Hongwei Xi. TPS: A Theorem-Proving System for Classical Type Theory. *Journal of Automated Reasoning*, 16(3):321–353, 1996.
6. Peter B. Andrews and Chad Brown. TPS: A Hybrid Automatic-Interactive System for Developing Proofs. *Journal of Applied Logic*, 4(4):367–395, 2006.
7. Julian Backes and Chad E. Brown. Analytic tableaux for higher-order logic with choice. In Jürgen Giesl and Reiner Hähnle, editors, *Automated Reasoning: 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16-19, 2010, Proceedings*, volume 6173 of *Lecture Notes in Artificial Intelligence*, pages 76–90. Springer, 2010.
8. Matteo Baldoni. *Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension.* PhD thesis, Universita degli studi di Torino, 1998.
9. Christoph Benzmüller. Extensional higher-order paramodulation and RUE-resolution. In *In Proc. of CADE-16*, number 1632 in LNAI, pages 399–413. Springer, 1999.
10. Christoph Benzmüller. Automating access control logic in simple type theory with LEO-II. In Dimitris Gritzalis and Javier López, editors, *Emerging Challenges for Security, Privacy and Trust, 24th IFIP TC 11 International Information Security Conference, SEC 2009, Pafos, Cyprus, May 18-20, 2009. Proceedings*, volume 297 of *IFIP*, pages 387–398. Springer, 2009.
11. Christoph Benzmüller. Combining logics in simple type theory. In Jürgen Dix, Joao Leite, Guido Governatori, and Woitek Jamroga, editors, *Computational Logic in Multi-Agent Systems, 11th International Workshop, CLIMA XI, Lisbon, Portugal, August 16-17, 2010. Proceedings*, volume 6245 of *Lecture Notes in Artificial Intelligence*, pages 33–48, Lisbon, Portugal, 2010. Springer.
12. Christoph Benzmüller. Verifying the modal logic cube is an easy task (for higher-order automated reasoners). In Simon Siegler and Nathan Wasser, editors, *Verification, Induction, Termination Analysis - Festschrift for Christoph Walther on the Occasion of His 60th Birthday*, volume 6463 of *LNCS*, pages 117–128. Springer, 2010.
13. Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Higher order semantics and extensionality. *Journal of Symbolic Logic*, 69:1027–1088, 2004.
14. Christoph Benzmüller and Michael Kohlhase. LEO – a higher-order theorem prover. In Claude Kirchner and Hélène Kirchner, editors, *Automated Deduction - CADE-15, 15th International Conference on Automated Deduction, Lindau, Germany, July 5-10, 1998, Proceedings*, number 1421 in Lecture Notes in Artificial Intelligence, pages 139–143, Lindau, Germany, 1998. Springer.
15. Christoph Benzmüller and Lawrence C. Paulson. *Festschrift in Honor of Peter B. Andrews on His 70th Birthday*, chapter Exploring Properties of Normal Multimodal Logics in Simple Type Theory with LEO-II. Studies in Logic, Mathematical Logic and Foundations. College Publications, 2008.
16. 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.
17. Christoph Benzmüller and Lawrence C. Paulson. Multimodal and intuitionistic logics in simple type theory. *The Logic Journal of the IGPL*, 18:881–892, 2010.
18. Christoph Benzmüller and Adam Pease. Progress in automating higher-order ontology reasoning. In Boris Konev, Renate Schmidt, and Stephan Schulz, editors, *Proceedings of the Workshop on Practical Aspects of Automated Reasoning (PAAR-2010)*, Edinburgh, UK, 2010. CEUR Workshop Proceedings.
19. Christoph Benzmüller and Adam Pease. Reasoning with embedded formulas and modalities in SUMO. In A. Bundy, J. Lehmann, G. Qi, and I. J. Varzinczak, editors, *Proceedings of the ECAI-10 Workshop on Automated Reasoning about Context and Ontology Evolution (ARCOE-10)*, August 16-17, Lisbon, Portugal, 2010.
20. 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. Springer, 2008.

21. Christoph Benzmüller, Frank Theiss, Larry Paulson, and Arnaud Fietzke. LEO-II — A cooperative automatic theorem prover for higher-order logic. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, volume 5195 of *Lecture Notes in Artificial Intelligence*, pages 162–170. Springer, 2008.
22. Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*. Elsevier Science Inc., New York, NY, USA, 2006.
23. Patrick Blackburn and Maarten de Rijke. Why combine logics? *Studia Logica*, 59:5–27, 1995.
24. 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.
25. Torben Braüner. Natural deduction for first-order hybrid logic. *Journal of Logic, Language and Information*, 14(2):173–198, 2005.
26. Chad Brown. *Automated Reasoning in Higher-Order Logic: Set Comprehension and Extensionality in Church's Type Theory*. Number 10 in Studies in Logic: Logic and Cognitive Systems. College Publications, 2007.
27. Walter Carnielli and Marcelo Esteban Coniglio. Combining logics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2008 edition, 2008.
28. Walter A. Carnielli, Marcelo Coniglio, Dov M. Gabbay, Paula Gouveia, and Cristina Sernadas. *Analysis and synthesis of logics: how to cut and paste reasoning systems; electronic version*. Applied Logic Series. Springer, Dordrecht, 2008.
29. Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.
30. Melvin Fitting. Interpolation for first order S5. *Journal of Symbolic Logic*, 67(2):621–634, 2002.
31. Dov M. Gabbay. Fibred semantics and the weaving of logics, part 1: Modal and intuitionistic logics. *J. Symb. Log.*, 61(4):1057–1120, 1996.
32. Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic, 148. Elsevier Science, 2003.
33. Deepak Garg and Martin Abadi. A Modal Deconstruction of Access Control Logics. In R. Amadio, editor, *Proceedings of the 11th International Conference on the Foundations of Software Science and Computational Structures*, number 4962 in Lecture Notes in Computer Science, pages 216–230, 2008.
34. James Garson. Modal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2009 edition, 2009.
35. Kurt Gödel. Eine interpretation des intuitionistischen aussagenkalküls. *Ergebnisse eines Mathematischen Kolloquiums*, 8:39–40, 1933. Also published in Gödel [36], 296–302.
36. Kurt Gödel. *Collected Works, Volume I*. Oxford University Press, 1986.
37. Robert Goldblatt. *Logics of Time and Computation*. Number 7 in Center for the Study of Language and Information - Lecture Notes. Leland Stanford Junior University, 1992.
38. Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.
39. Joe Hurd. First-Order Proof Tactics in Higher-Order Logic Theorem Provers. In M. Archer, B. Di Vito, and C. Munoz, editors, *Proceedings of the 1st International Workshop on Design and Application of Strategies/Tactics in Higher Order Logics*, number NASA/CP-2003-212448 in NASA Technical Reports, pages 56–68, 2003.
40. 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, 2009.
41. John C. C. McKinsey and Alfred Tarski. Some theorems about the sentential calculi of lewis and heyting. *Journal of Symbolic Logic*, 13:1–15, 1948.
42. Dale Miller. A Compact Representation of Proofs. *Studia Logica*, 46(4):347–370, 1987.
43. Linh Anh Nguyen. MProlog: An extension of prolog for modal logic programming. In Bart Demoen and Vladimir Lifschitz, editors, *Logic Programming, 20th International Conference, ICLP 2004, Saint-Malo, France, September 6-10, 2004, Proceedings*, volume 3132 of *Lecture Notes in Computer Science*, pages 469–470. Springer, 2004.

44. Linh Anh Nguyen. Multimodal logic programming. *Theoretical Computer Science*, 360:247–288, 2006.
45. Linh Anh Nguyen. Modal logic programming revisited. *Journal of Applied Non-Classical Logics*, 19(2):167–181, 2009.
46. Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Number 2283 in Lecture Notes in Computer Science. Springer, 2002.
47. Arthur N. Prior. *Past, Present and Future*. Oxford: Clarendon Press, 1967.
48. David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proceedings 3rd International Conference on Knowledge Representation and Reasoning*, pages 165–176, 1992.
49. Stephan Schulz. E – A Brainiac Theorem Prover. *Journal of AI Communications*, 15(2/3):111–126, 2002.
50. Krister Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2(1):77–96, 1973.
51. 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, 2007.
52. Geoff Sutcliffe. The TPTP World - infrastructure for automated reasoning. In Edmund M. Clarke and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Dakar, Senegal, April 25-May 1, 2010, Revised Selected Papers*, volume 6355 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2010.
53. 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.
54. Richmond H. Thomason. Combinations of tense and modality. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol. 2: Extensions of Classical Logic*, pages 135–165. D. Reidel, 1984.
55. Yde Venema. In Lou Goble, editor, *The Blackwell Guide to Philosophical Logic*, chapter Temporal Logic. Blackwell Publishing, 2001.