Solving the $100 Modal Logic Challenge

Florian Rabe [a,1] Petr Pudlák [b,2] Geoff Sutcliffe [c] Weina Shen [c]

[a] *Department of Computer Science, Carnegie Mellon University, USA*

[b] *Department of Theoretical Computer Science and Mathematical Logic Charles University, Czech Republic*

[c] *Department of Computer Science, University of Miami, USA*

Abstract

We present the theoretical foundation, design, and implementation, of a system that automatically determines the subset relation between two given axiomatizations of propositional modal logics. This is an open problem for automated theorem proving. Our system solves all but six out of 121 instances formed from 11 common axiomatizations of seven modal logics. Thus, although the problem is undecidable in general, our approach is empirically successful in practically relevant situations.

Key words: Modal Logic, $100 challenge, subset relationship

1 Introduction and Related Work

Modal logics are extensions of classical logic that handle the concept of modalities. Modern modal logic was founded by Clarence Irving Lewis in his 1910 Harvard thesis, and further developed in a series of scholarly articles beginning in 1912. In his book Symbolic Logic (with C. H. Langford), he introduced the five well-known modal logics S1 through S5 (Lewis and Langford, 1932). The contemporary era in modal logic began in 1959 when Saul Kripke introduced semantics for modal logics (Kripke, 1963). The mathematical structures of modal logics are modal algebras – Boolean algebras augmented with unary operations. Their study began to emerge with McKinsey's proof that S2 and S4 are decidable (McKinsey, 1941). Today plain propositional modal logic

[1] The author was supported by a fellowship for Ph.D. research of the German Academic Exchange Service.

[2] The author was supported by the Grant Agency of Charles University (grant no. 377/2005/A INF).

is standard knowledge, see, e.g., Hughes and Cresswell (1996), and first order modal logic has been thoroughly studied, e.g., Fitting and Mendelsohn (1998). Henceforth in this paper attention is limited to propositional modal logic with the standard modalities *possibility* and *necessity*.

Many modal logics have multiple axiomatizations that are *equivalent*, in the sense that they generate the same *theory* - the same set of theorems. Similarly, one modal logic may be *stronger* than another in the sense that the stronger logic's theory is a strict superset of that of the weaker logic. Finally, two modal logic may be *incomparable* with one another, because each has theorems that the other does not. Such relationships between different axiomatizations of individual modal logics and between different modal logics, are well known (Hughes and Cresswell, 1996).

The Modal Logic $100 Challenge (Sutcliffe, 2006) calls for a program that can determine the relationships between common Hilbert-style axiomatizations of the modal logics K, T, $\mathsf{S1}$, $\mathsf{S1}^\circ$, $\mathsf{S3}$, $\mathsf{S4}$ and $\mathsf{S5}$ (see Halleck (2006) for an overview and a list of references to various modal logics and axiomatizations). The program cannot simply encode known relationships. Rather, it must use logical reasoning from input axiomatizations to establish the relationships as if they were unknown. The syntactic representation of the axiomatizations can be anything reasonable, but the use of the TPTP syntax is encouraged. The challenge was sponsored by John Halleck, who has a practical need for such a program as an aid to maintaining his overview (Halleck, 2006).

Determining the relationship between two axiomatizations is undecidable in general (Blackburn et al., 2001). Decidability can be established separately for some modal logics. Historically, finding a complete Kripke semantics (Kripke, 1963) and establishing the finite model property by filtration (Lemmon, 1966) were used to obtain decidability of theoremhood. However, checking the subset relation between modal logics also involves checking the admissibility of rules. Here, decidability has been shown for a few cases, including $\mathsf{K4}$ and $\mathsf{S4}$ (Rybakov, 1997). Recently a framework has been developed in which admissibility is reduced to terminating analytic proofs for a variety of modal logics (Iemhoff and Metcalfe, 2007). In (Kracht, 1990) splittings are used to decide the admissibility of a rule for some logics, which correspond to transitive Kripke frames, but no algorithm is known for obtaining a splitting for an arbitrary logic. None of these methods is applicable to the full range of differently axiomatized modal logics. So far sophisticated implementations have focused on deriving theoremhood. (Goré et al., 1997) describe the *Logics Work Bench* program that is capable of reasoning about the modal systems K, KT, $\mathsf{KT4}$, $\mathsf{KT45}$ and KW. (Giunchiglia et al., 2002) use a SAT solver to decide a few classical systems. (Schmidt and Hustadt, 2006) give an overview over various methods based on translations of modal logic into first-order logic (e.g. Ohlbach and Schmidt, 1997). (Hustadt and Schmidt, 2000) extends the first-order theorem prover SPASS with the ability to apply such translations to its input.

This paper describes an implemented and tested system within which relationships between modal logics can be determined. The system has been applied successfully to the \$100 challenge. A partial preliminary version of this work has been presented as (Rabe, 2006b). The core idea is to use a simple translation to first-order logic (described in Section 3 of (Schmidt and Hustadt, 2006)), which encodes modal logic formulae as first-order terms, modal logic axioms and theorems as first-order atoms, and modal logic rules as first-order implications. This translation comes at the price of efficiency (McCune and Wos, 1992). We use it because it is applicable to any modal logic, in particular non-normal logics. Since we do not presuppose any semantics, the applicability of any other translation would itself have to be established automatically (which we do in the Kripke-based strategy described in Section 3.4). With this encoding, reasoning is performed using several modularly implemented (possibly incomplete) strategies, using first-order automated reasoning tools to prove or disprove the subset relationship: direct strategies, strategies based on Kripke semantics, and algebraic strategies that represent modal logics as modal algebras.

Section 2 provides the necessary background in modal logic and the encoding in first-order logic. Section 3 describes the implementation of our system, and the theoretical basis and implementation of the strategies. Section 4 documents and analyzes the results achieved by the system in attacking the \$100 challenge. Section 5 concludes and provides directions for future research.

2 Modal Logics

2.1 Formulas and Rules

Modal formulae $F, G, \ldots$ are defined as the elements of the languages generated from the atomic formulae and connectives given in Figure 1 (note that prefix notation is used for the binary connectives). Rules are of the form

$$\frac{H_1 \quad \ldots \quad H_n}{C}$$

where the H_i and C are modal formulae. The semantics of a rule is that if for any substitution σ of formulae for the propositional variables, all $\sigma(H_i)$ are derivable, then so is $\sigma(C)$. The case $n = 0$ means that C is an axiom. The rules with $n \neq 0$ relevant for the \$100 challenge are given in Figure 2.

$p, q, \ldots$	a countably infinite set of propositional variables	
$\neg F$	negation	(primitive)
$\wedge FG$	conjunction	(primitive)
$\Diamond F$	possibility	(primitive)
$\vee FG$	disjunction	$=_{defn} \neg\wedge\neg F\neg G$
$\rightarrow FG$	implication	$=_{defn} \neg\wedge F\neg G$
$\leftrightarrow FG$	equivalence	$=_{defn} \wedge\rightarrow FG\rightarrow GF$
$\Rightarrow FG$	strict implication	$=_{defn} \Box\rightarrow FG$
$\Leftrightarrow FG$	strict equivalence	$=_{defn} \wedge\Rightarrow FG\Rightarrow GF$
$\Box F$	necessity	$=_{defn} \neg\Diamond\neg F$

Fig. 1. Atomic formulae and connectives

An *axiomatization* of a modal logic is a set of axioms and rules, from which the theory is generated by finitely many (including no) rule applications. A modal logic is characterized by its theory. For an axiomatization $\mathcal{L}$ and a formula F, $\mathcal{L} \vdash F$ denotes that F is a theorem of $\mathcal{L}$. A rule R is an admissible rule of $\mathcal{L}$ if $\mathcal{L}$ and $\mathcal{L} \cup \{R\}$ are equivalent; this is denoted by $\mathcal{L} \vdash R$.

2.2 The Modal Logics of the \$100 Challenge

The relationships between the modal logics to be compared in the \$100 challenge are shown in Figure 3. A solid arrow shows that an axiomatization of the logic at the head can be constructed by adding axioms or rules to an axiomatization of the logic at the tail. A dashed arrow shows that the logic at the head is stronger than the logic at the tail, but the axiomatizations have different heritages, e.g., the axiomatization of T is built by adding to K rather than by adding to $\mathsf{S1}$. Regardless of the type of arrow, any path from one logic to another shows that the logic at the head of the path is stronger than the logic at the tail.

The axiomatizations used for the logics are given in Figure 4. There are two starting points for their construction - the propositional calculus PC and the strict system $\mathsf{S1}^{\circ}$. Since different axiomatizations can generate the same theory, some axiomatizations are equivalent, e.g., all four axiomatizations of $\mathsf{S5}$ are equivalent. Different axiomatizations of the same logic are differentiated by Greek subscripts.

PC is defined by the Hilbert and Bernays (Hilbert and Bernays, 1934), Łuka-

US	uniform substitution	$\dfrac{F}{\sigma(F)}$
MP	modus ponens	$\dfrac{F \quad \to FG}{G}$
SMP	strict modus ponens	$\dfrac{F \quad \Rightarrow FG}{G}$
AD	adjunction	$\dfrac{F \quad G}{\wedge FG}$
EQ	substitution of equivalents	$\dfrac{\leftrightarrow FF' \quad G}{G[F \rightsquigarrow F']}$
EQS	substitution of strict equivalents	$\dfrac{\Leftrightarrow FF' \quad G}{G[F \rightsquigarrow F']}$

σ: a substitution of propositional with formulae
$G[F \rightsquigarrow F']$: formed from G by replacing some occurrences of F with F'

Fig. 2. Rules

siewicz (Łukasiewicz, 1963), Rosser (Rosser, 1953), and Principia (Russell and Whitehead, 1910) axiomatizations as follows.

Definition 1 (PC)**.** The axiomatizations of PC are defined by the rules *US*, *MP*, and the following axioms:

For PCH (Hilbert-style):
$\mathit{MT}: \to\, \to \neg p \neg q \to qp$
$\mathit{A1}: \to \wedge pqp$
$\mathit{A2}: \to \wedge pqq$
$\mathit{A3}: \to p \to q \wedge pq$
$\mathit{I1}: \to p \to qp$
$\mathit{I2}: \to\, \to p \to pq \to pq$
$\mathit{I3}: \to\, \to pq \to\, \to qr \to pr$
$\mathit{O1}: \to p \vee pq$
$\mathit{O2}: \to q \vee pq$
$\mathit{O3}: \to\, \to pr \to\, \to qr \to \vee pqr$
$\mathit{E1}: \to \leftrightarrow pq \to pq$
$\mathit{E2}: \to \leftrightarrow pq \to qp$
$\mathit{E3}: \to\, \to pq \to\, \to qp \leftrightarrow pq$

For PCL (Łukasiewicz-style):
$\mathit{CN1}: \to\, \to pq \to\, \to qr \to pr$
$\mathit{CN2}: \to p \to \neg pq$
$\mathit{CN3}: \to\, \to \neg ppp$

For PCR (Rosser-style):
$\mathit{KN1}: \to p \wedge pp$
$\mathit{KN2}: \to \wedge pqp$
$\mathit{KN3}: \to\, \to pq \to \neg \wedge qr \neg \wedge rp$

For PCP (Principia-style):[3]
$\mathit{R1}: \to \vee ppp$
$\mathit{R2}: \to q \vee pq$
$\mathit{R3}: \to \vee pq \vee qp$
$\mathit{R4}: \to \vee p \vee qr \vee q \vee pr$
$\mathit{R5}: \to\, \to qr \to \vee pq \vee pr$

S1° is axiomatized in the Lewis-style, as taken from Zeman (1973). The Lemmon-style axiomatization of S1°, which is an extension of PC, was not used because it requires the weakened necessitation rule "if A is a PC theorem then $\Box A$ is a theorem", which is unreasonable to encode using the single-sorted

[3] Note that the axioms include the redundant *R4*, which can be proved from the others (Bernays, 1926).

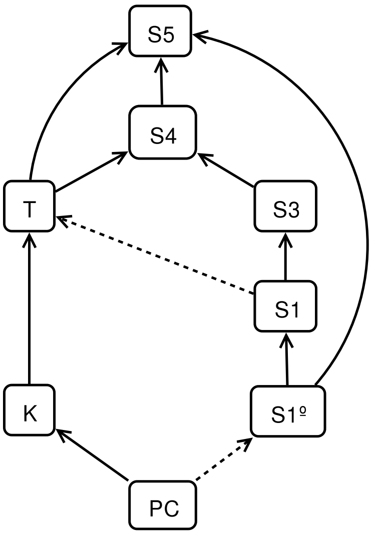

Fig. 3. The Hierarchy of the Modal Logics

first-order approach taken in this work.

Definition 2 (S1° – Lewis-style)**.** The axiomatization S1° is defined by the rules *US*, *SMP*, *AD*, and *EQS*, and the axioms

$$\begin{aligned}
\mathit{M1}:&\ \Rightarrow\wedge pq\wedge qp\\
\mathit{M2}:&\ \Rightarrow\wedge pqp\\
\mathit{M3}:&\ \Rightarrow\wedge\wedge pqr\wedge p\wedge qr\\
\mathit{M4}:&\ \Rightarrow p\wedge pp\\
\mathit{M5}:&\ \Rightarrow\wedge\Rightarrow pq\Rightarrow qr\Rightarrow pr
\end{aligned}$$

Note that all the axiomatizations include the structural rule *US*, which is crucial for the soundness of the first-order encoding described in Section 2.3.

2.3 First-order Encoding

Modal formulae are encoded as first-order formulae with equality, the first-order connectives are written as $\neg$, $\wedge$, and $\rightarrow$, the universal quantifier as $\forall X, Y, \ldots$, and equality as $=$. $T \vdash^{FOL} F$ denotes that F is a first-order theorem of the theory T.

The first-order signature used for encoding modal formulae consists of the following symbols:

- unary function symbols: `not`, `poss`, `necess`,
- binary function symbols: `and`, `or`, `impl`, `equiv`, `s_impl`, `s_equiv`,
- unary predicte symbol: `thm`.

K	$=$	PC	$+$	*Nec* :	$\frac{F}{\Box F}$
			$+$	*K* :	$\to\Box\to pq\to\Box p\Box q$
T	$=$	K	$+$	*M* :	$\to\Box pp$
S1	$=$	S1°	$+$	*M6* :	$\Rightarrow p\Diamond p$
S3	$=$	S1	$+$	*S3* :	$\Rightarrow\ \Rightarrow pq\Rightarrow\neg\Diamond q\neg\Diamond p$
S4$_\alpha$	$=$	S3	$+$	*M9* :	$\Rightarrow\Diamond\Diamond p\Diamond p$
S4$_\beta$	$=$	T	$+$	*4* :	$\to\Box p\Box\Box p$
S5$_\alpha$	$=$	S4$_\alpha$	$+$	*B* :	$\to p\Box\Diamond p$
S5$_\beta$	$=$	S4$_\beta$	$+$	*B* :	$\to p\Box\Diamond p$
S5$_\gamma$	$=$	S1°	$+$	*M10* :	$\Rightarrow\Diamond p\Box\Diamond p$
S5$_\delta$	$=$	T	$+$	*5* :	$\to\Diamond p\Box\Diamond p$

Fig. 4. Axiomatizations to be Compared

Then the encoding $\mathcal{E}(\cdot)$ is defined as follows:

(1) for an axiomatization $\mathcal{L} = \{R_1, \ldots, R_n\}$:

$$\mathcal{E}(\mathcal{L}) = \mathit{Def} \cup \{\mathcal{E}(R_1), \ldots, \mathcal{E}(R_n)\}$$

is a first-order theory over the above signature where *Def* consists of the following axioms:

- $\forall X, Y\ \mathtt{or}(X, Y) = \mathtt{not}(\mathtt{and}(\mathtt{not}(X), \mathtt{not}(Y)))$,
- $\forall X, Y\ \mathtt{impl}(X, Y) = \mathtt{not}(\mathtt{and}(X, \mathtt{not}(Y)))$,
- $\forall X, Y\ \mathtt{equiv}(X, Y) = \mathtt{and}(\mathtt{impl}(X, Y), \mathtt{impl}(Y, X))$,
- $\forall X, Y\ \mathtt{s\_impl}(X, Y) = \mathtt{necess}(\mathtt{impl}(X, Y))$,
- $\forall X, Y\ \mathtt{s\_equiv}(X, Y) = \mathtt{and}(\mathtt{s\_impl}(X, Y), \mathtt{s\_impl}(Y, X))$,
- $\forall X\ \mathtt{necess}(X) = \mathtt{not}(\mathtt{poss}(\mathtt{not}(X)))$.

(2) for a rule $R = \dfrac{H_1 \quad \ldots \quad H_n}{C}$ with propositional variables $p_1, \ldots, p_m$:

$$\mathcal{E}(R) = \forall X_1, \ldots, X_m \Big((\mathcal{E}(H_1)\ \wedge\ \ldots\ \wedge\ \mathcal{E}(H_n))\ \to\ \mathcal{E}(C)\Big),$$

(3) for a modal formula F: $\mathcal{E}(F) = \mathtt{thm}(\varepsilon(F))$, where $\varepsilon(F)$ encodes every formula F as a first-order term by

- $\varepsilon(\wedge FG) = \mathtt{and}(\varepsilon(F), \varepsilon(G))$ and similarly for $\vee, \to, \leftrightarrow, \Rightarrow$, and $\Leftrightarrow$,
- $\varepsilon(\Box F) = \mathtt{necess}(\varepsilon(F))$ and similarly for $\neg$ and $\Diamond$,
- $\varepsilon(p_i) = X_i$ for a propositional variable p_i.

For example, for the rule *MP*, we have

$$\mathcal{E}(\mathit{MP}) = \forall X_1, X_2 \left(\mathtt{thm}(X_1) \ \wedge \ \mathtt{thm}(\mathtt{impl}(X_1, X_2))) \ \rightarrow \ \mathtt{thm}(X_2) \right)$$

Note that the rules *US*, *EQ* and *EQS* cannot be encoded in this way. *US* is inherent in the encoding as Theorem 3 shows. Section 3.2.2 shows how *EQ* and *EQS* are replaced by congruence rules, and Section 3.2.3 shows how the congruence rules can be replaced by formulae allowing use of efficient first-order equality reasoning.

The following soundness result guarantees that reasoning about the first-order encoding is equivalent to reasoning about the encoded axiomatization.

Theorem 3. *Let $\mathcal{L}$ be an axiomatization. Then for modal rules R*

$$\mathcal{E}(\mathcal{L}) \vdash^{FOL} \mathcal{E}(R) \text{ if and only if } \mathcal{L} \vdash R.$$

Proof. Since $\mathcal{E}(\mathcal{L})$ contains only Horn formulae, there is a free first-order model M of $\mathcal{E}(\mathcal{L})$ such that M is term-generated and $\mathcal{E}(R)$ holds in M iff $\mathcal{E}(\mathcal{L}) \vdash^{FOL} \mathcal{E}(R)$. The universe of M can be constructed by taking the set of equivalence classes generated by equality axiomatized by reflexivity, symmetry, transitivity, congruence and the equality axioms. Let $[t]$ denote the equivalence class of t. Clearly, two terms are equal in M iff the modal formulae they represent can be transformed into each other by eliminating and introducing abbreviations of modal formulae.

Function symbols are interpreted in M as induced by the equivalence relation. And $\mathtt{thm}^M$ is the smallest fixed point of the following operation: $[t] \in \mathtt{thm}^M$ iff there is a rule in $\mathcal{L}$ encoded as

$$\forall X_1, \ldots, X_m \left((\mathtt{thm}(h_1) \ \wedge \ \ldots \ \wedge \ \mathtt{thm}(h_n)) \ \rightarrow \ \mathtt{thm}(c) \right)$$

and a substitution α for the variables $X_1, \ldots, X_m$ such that $[\alpha(h_i)] \in \mathtt{thm}^M$ for $i = 1, \ldots, m$ and $\alpha(c) = t$.

Then because *US* is admissible in $\mathcal{L}$, we have for every modal formula F and every substitution α:

$$[\alpha(\varepsilon(F))] \in \mathtt{thm}^M \text{ if and only if } \mathcal{L} \vdash F^\alpha$$

where F^α denotes the uniform substitution instance of F under α. Therefore, the definition of $\mathcal{L} \vdash R$ is equivalent to saying that $\mathcal{E}(R)$ holds in M, which completes the proof. □

3 Solution

In this section our solution to the \$100 challenge is presented. This section is organized as follows. In Section 3.1, we give an overview over our system, and in the remaining sections, we present its theoretical basis and the implementation. In particular, Section 3.2 describes the preprocessing phase, and Sections 3.3 to 3.5 describe the comparison strategies used.

3.1 System Architecture and Process

Our system is implemented in Standard ML of New Jersey (SML, 2007). The source code can be obtained from (Rabe, 2006a). After loading the sources into the SML top-level, the user can call a function `compare : string * string -> unit`. This function takes the filenames of the logics to be compared as arguments, and prints the results of the comparison.

The input files must contain two axiomatizations, $\mathcal{L}$ and $\mathcal{M}$, in the TPTP format (Sutcliffe and Suttner, 1998). In addition to the encoded axioms and rules, the input files can contain *special rules* of the form:

$$\texttt{fof}(\mathit{name}, \texttt{special\_rule}, \texttt{ignored}).$$

where *name* identifies the special rule and the rest is ignored. Special rules are used to import PC axiomatizations into PC based modal logics, and to represent aspects that require special processing, e.g., rules for substitution of equivalents. After reading in the files, two phases can be distinguished.

The *preprocessing* phase, described in Section 3.2, includes the expansion of special rules into sets of normal rules, and optimizations related to congruence relations. We also try to establish certain properties of the logics, like normality, so that these properties can be reused later. The preprocessing returns two different but equivalent axiomatizations for every logic. For $\mathcal{L}$ and $\mathcal{M}$, we obtain $\mathcal{L}^b$ & $\mathcal{L}^s$ and $\mathcal{M}^b$ & $\mathcal{M}^s$. The b axiomatizations are "big", containing redundant axioms, useful lemmas, etc., and are used when proving *from* the logic. The s axiomatizations are "small" and used when proving *to* the logic.

The *comparison* phase, described in Sections 3.3 to 3.5, attempts to determine the relationship between the two input axiomatizations. First the system checks whether $\mathcal{L}^b$ is stronger than $\mathcal{M}^s$, and then it checks whether $\mathcal{M}^b$ is stronger than $\mathcal{L}^s$. In both directions the following happens: The system tries to prove every axiom and rule of the s logic from the b logic. Several proving strategies are available for proving each axiom and rule. The strategies are tried in turn until one succeeds or all have failed. Axioms and rules that fail to be proved are passed to disproving strategies. The disproving strategies try to find a counterexample for each axiom and rule, establishing that the axiom or rule cannot be proved.

Three kinds of strategies are used in the comparison phase: direct strategies are described in Section 3.3, strategies based on Kripke semantics in Section 3.4, and strategies based on algebraic encodings in Section 3.5. All strategies are parametric in the specific first-order prover or model finder that is used.

If both directions succeed, whether by proving or by disproving, the relationship between the logics is decided, and $\mathcal{L} \subset \mathcal{M}$, $\mathcal{M} \subset \mathcal{L}$, $\mathcal{L} = \mathcal{M}$ or $\mathcal{L}$ incomparable to $\mathcal{M}$ is printed. If only one direction succeeds, a partial result is printed.

3.2 Preprocessing

3.2.1 Special Rule pc

The special rule pc is expanded into an axiomatization of PC. The four axiomatizations of PC defined in Section 2.2 are equivalent (see also McCune et al. (2002)). This can be demonstrated automatically by proving the axioms of each from the axiomatizations of each other (as all axiomatizations use the same rules, the rules do not need to be proved), which was done using the ATP system Vampire 8.1 with a 180s CPU time limit, on a 2.8GHz PC with 1GB memory and running Linux 2.6. The results are summarized in Fig. 5, which gives the CPU times in seconds for the proofs of the axioms from the named axiomatizations, or TO for proof attempts that timed out at 180s. The results show that the Hilbert axiomatization can prove the Łukasiewicz and Principia axioms, the Łukasiewicz axiomatization can prove the Rosser axioms, and the Principia axiomatization can prove the Łukasiewicz axioms. While the results are not all positive, the results are useful: (i) if the Hilbert axiomatization can be proved, that is sufficient for claiming that all four axiomatizations have been proved, and (ii) if the Hilbert axiomatization is used as a basis for constructing modal logics, then it is possible to add the other three axiomatizations' axioms as lemmas.

Due to these results, the pc special rule is expanded into the Hilbert axioms when computing an s axiomatization, and into the union of all four PC axiomatizations when computing a b axiomatization. For simplicity, the proofs justifying this treatment are not executed explicitly every time.

Prove→	PCH												
From↓	*MT*	*A1*	*A2*	*A3*	*O1*	*O2*	*O3*	*I1*	*I2*	*I3*	*E1*	*E2*	*E3*
PCL	124	3	68	0	TO	TO	TO	0	3	TO	69	72	TO
PCR	110	5	11	TO	0	5	132	2	5	TO	0	4	TO
PCP	0	0	55	16	2	0	4	2	0	TO	2	0	TO

Prove→	PCL			PCR			PCP				
From↓	*CN1*	*CN2*	*CN3*	*KN1*	*KN2*	*KN3*	*R1*	*R2*	*R3*	*R4*	*R5*
PCH	0	0	1	0	0	TO	1	0	2	4	4
PCL	-	-	-	58	58	59	112	TO	TO	TO	TO
PCR	TO	4	3	-	-	-	1	5	1	TO	TO
PCP	16	1	4	0	2	TO	-	-	-	-	-

Fig. 5. Relationships between PC Axiomatizations

3.2.2 Special Rules eq *and* eqs

The special rules eq represents the *EQ* rules, which cannot be expressed directly using the first-order encoding. The first step around this is to use the following rules that define $\leftrightarrow$ to be a congruence relation:

$$\frac{\leftrightarrow FG}{\leftrightarrow \neg F \neg G}\,(EQ1) \qquad \frac{\leftrightarrow FF' \quad \leftrightarrow GG'}{\leftrightarrow \wedge FG \wedge F'G'}\,(EQ2) \qquad \frac{\leftrightarrow FG}{\leftrightarrow \Diamond F \Diamond G}\,(EQ3)$$

$$\frac{\leftrightarrow FG \quad F}{G}\,(EQ4) \qquad \frac{}{\leftrightarrow FF}\,(EQ5)$$

The following lemma then relates *EQ* to $\leftrightarrow$ being a congruence relation in the context of the modal logic under consideration.

Lemma 4. *If $\mathcal{L} \vdash EQ5$, then $\mathcal{L} \cup \{EQ\}$ and $\mathcal{L} \cup \{EQ1, EQ2, EQ3, EQ4\}$ are equivalent.*

Proof. If $\mathcal{L} \cup \{EQ1, EQ2, EQ3, EQ4\}$ is given, we need to derive *EQ*. Let F, F' and G be as in the definition of *EQ*, where we can assume without loss of generality that no defined connective occurs in them. We need to derive $G[F \rightsquigarrow F']$. We construct a backwards proof, firstly applying *EQ4*, to reduce to $\leftrightarrow G(G[F \rightsquigarrow F'])$. This can be derived by repeated application of *EQ1*-*EQ3* along the structure of G until all open proof goals are $\leftrightarrow FF'$ or are instances of *EQ5* under *US*. Conversely, let $\mathcal{L} \cup \{EQ\}$ be given. We need to derive the rules *EQ1*-*EQ4*. *EQ1*-*EQ3* are special cases of *EQ* with, e.g., $G = \leftrightarrow \neg p \neg p$, and *EQ4* is the special case of *EQ* where $F = G$. □

Given Lemma 4, when the special rule `eq` is found, an attempt is made to prove *EQ5*. If this succeeds the special rule is expanded to *EQ1*-*EQ4*, and the proved *EQ5* is added to the axiomatization. The analogue of Lemma 4 for strict equivalence can be proved, and the rule *EQS* is handled correspondingly by a special rule `eqs`.

3.2.3 Congruences

None of the axiomatizations of the challenge is defined to include the rule *EQ*. However, this rule is extremely powerful, and is necessary for success when proving relationships between modal logics. Section 3.2.2 explains that *EQ* is represented in input files as a special rule, and is expanded to *EQ1*-*EQ4* if *EQ5* can be proved. The congruence rules are inefficient in implementing substitution. A much more efficient approach is to exploit the equational reasoning of a first-order theorem prover. If the relation $\mathcal{L} \vdash \leftrightarrow FG$ is a congruence relation on the set of modal formulas, the rule

$$\forall X, Y \left(\texttt{thm}(\texttt{equiv}(X, Y)) \;\rightarrow\; X = Y\right) \quad (*)$$

is added to $\mathcal{L}^b$. The soundness of this addition is given by the following lemma:

Lemma 5. *If $\mathcal{L} \vdash$ EQi for all $i = 1, \ldots, 5$, then adding $(*)$ to the first-order encoding of $\mathcal{L}$ does not destroy the soundness of the encoding.*

Proof. Let M be the free model constructed in the proof of Theorem 3. Because $\mathcal{L}$ has the rules mentioned above and due to Lemma 4, if $\leftrightarrow FG$ is derivable in $\mathcal{L}$, either both F and G are derivable in $\mathcal{L}$ or none. Then, by induction on the construction of M, it follows that adding the above rule will never identify two terms in the term model of which only one corresponds to a derivable modal formula. Therefore, the terms that are in the equivalence classes in the interpretation of `thm` stay the same, and soundness is preserved. □

For a PC based axiomatization $\mathcal{L}$, proving $\mathcal{L} \vdash$ *EQi* for all $i = 1, \ldots, 5$ can be done in parts. The proofs of $\mathsf{PC} \vdash \{$*EQ1*, *EQ2*, *EQ4*, *EQ5*$\}$, which do not mention the modal operators, can be done offline in advance. This is described below. Then given a PC based axiomatization $\mathcal{L}$ it is necessary to prove only $\mathcal{L} \vdash$ *EQ3*.

The proofs of $\mathsf{PC} \vdash \{$*EQ1*, *EQ2*, *EQ4*, *EQ5*$\}$ were done using the combined axiomatization $\mathsf{PC} = \mathsf{PCH} \cup \mathsf{PCL} \cup \mathsf{PCR} \cup \mathsf{PCP}$ (whose combination is justified above), and the same hardware and software environment as above. *EQ1* was proved in 50s, *EQ4* in 0s, and *EQ5* in 1s. However, *EQ2* could not be proved, and two lemmas were used as stepping stones:

$$\frac{\leftrightarrow pp'}{\leftrightarrow \wedge pq \wedge p'q}\ (EQ2a), \qquad \frac{\leftrightarrow pp'}{\leftrightarrow \wedge qp \wedge qp'}\ (EQ2b)$$

EQ2a was proved in 79s and *EQ2b* in 95s. Attempts to prove *EQ2* from the combined axiomatization augmented with the two lemmas were not successful. However using only PCH augmented with the two lemmas produced a proof of *EQ2* in 6s (the redundancy in the combined axiomatization clearly affected VAMPIRE's search in this case).

The rule *EQS* is used in $\mathsf{S1}^\circ$ (based) axiomatizations. The analogue of Lemma 5 for strict equivalence can be proved, and the rule *EQS* is handled correspondingly. The proofs of the analogues of $\mathcal{L} \vdash EQi$ are all trivial, because $\mathsf{S1}^\circ$ based axiomatizations include the `eqs` special rule, which would have been expanded to those rules beforehand.

3.2.4 Testing the Applicability of Advanced Strategies

We use advanced strategies the applicability of which has to be proved itself. Those parts of these proofs that depend only on the modal logic we are proving from (and not on the axiom or rule to be proved or disproved) are executed in the preprocessing phase, and the results of the computations are stored along with $\mathcal{L}^b$ and $\mathcal{L}^s$ to represent $\mathcal{L}$. The details of these preprocessing steps are given in Sections 3.4 and 3.5 when describing these advanced strategies, namely the strategies `kripke_pos` and `s1O_pos`.

The strategy `kripke_pos` uses a relational translation into first-order logic, which depends on the normality of the logic $\mathcal{L}$. Therefore, we try to prove that $\mathcal{L}$ is normal, i.e., closed under the rules of K. If so, those rules are added to $\mathcal{L}^b$. This translation can be further improved by finding a property of Kripke frames that characterizes $\mathcal{L}$. Therefore, we identify the Sahlqvist axioms of $\mathcal{L}$ and find their corresponding frame properties.

The strategy `s1O_pos`, which uses an algebraic encoding of $\mathsf{S1}^\circ$, requires an axiomatization of $\mathcal{L}$ that consists of the axioms and rules of $\mathsf{S1}^\circ$ and additional axioms. Therefore, we try to find such an axiomatization. We also try to bring the additional axioms into a certain form to enhance the algebraic encoding.

3.3 Direct Strategies

In this section, the direct strategies are presented. The two proving strategies are purely syntactic, and the disproving strategy uses a first-order model finder. All the direct strategies are always applicable and do not require additional knowledge about the logics.

3.3.1 Proving

Let $\mathcal{L}$ be an axiomatization produced by the preprocessing and let $\mathcal{M}'$ be as $\mathcal{M}$ but with an additional rule R. Obviously, we have:

Lemma 6. *If R is an axiom,*

$$\mathcal{M}' \subseteq \mathcal{L} \text{ if and only if } \mathcal{M} \subseteq \mathcal{L} \text{ and } \mathcal{L} \vdash R,$$

and if R is not an axiom,

$$\mathcal{M}' \subseteq \mathcal{L} \text{ if } \mathcal{M} \subseteq \mathcal{L} \text{ and } \mathcal{L} \vdash R.$$

Lemma 6 is used to implement the strategy `direct_pos`. It takes a logic $\mathcal{L}$ and a rule R as input and calls a first-order theorem prover to prove $\mathcal{L} \vdash R$.

In Lemma 6, the “only if” direction does not hold for rules. This is because deriving R from $\mathcal{L}$ requires showing that whenever $\mathcal{L}$ contains instances of the hypotheses of R, it also contains the appropriate instance of the conclusion. For the “only if” direction to hold, we would need the weaker condition that whenever $\mathcal{M}'$ (which is a subset of $\mathcal{L}$) contains instances of the hypotheses of R, then $\mathcal{L}$ contains the appropriate instance of the conclusion. For a trivial example, let $\mathcal{M}$ be the empty axiomatization, R be the rule

$$\frac{p}{\neg p}$$

and $\mathcal{L}$ be any consistent non-empty axiomatization. Clearly, R is not admissible in $\mathcal{L}$ because $\mathcal{L}$ is consistent. But $\mathcal{M}'$ is still empty because an axiomatization without axioms has no theorems even if it contains an inconsistent rule, and therefore, $\mathcal{M}$ is a subset of $\mathcal{L}$.

Furthermore, a theorem prover will often not even find a proof of $\mathcal{L} \vdash R$, in particular if R is a rule that is admissible in $\mathcal{L}$ but not derivable. The simplest such case arises when R is the necessitation rule and $\mathcal{L}$ is an $\mathsf{S1}°$-based axiomatization of $\mathsf{S4}$ or $\mathsf{S5}$. The following lemma gives an inductive admissibility criterion.

Lemma 7. *Let R be of the form*

$$\frac{p}{F(p)}$$

for some formula F in one propositional variable p. We write $F(G)$ for substituting p in F with G. Then $\mathcal{M}' \subseteq \mathcal{L}$ if

- *$\mathcal{M} \subseteq \mathcal{L}$ and*
- *for every rule of $\mathcal{L}$ with hypotheses $H_1, \ldots, H_n$ and conclusion C, the rule*

$$\frac{H_1 \quad \ldots \quad H_n \quad F(H_1) \quad \ldots \quad F(H_n)}{F(C)}$$

is derivable from $\mathcal{L}$.

Proof. We need to show that R is admissible in $\mathcal{L}$, i.e., whenever a formula G is derivable, then so is $F(G)$. This is proved by a straightforward induction over the theorems of $\mathcal{L}$. The base case means that $\mathcal{L} \vdash F(A)$ for every axiom A. This holds due to the above condition (here $n = 0$). The induction step is a rule application leading from $H_1, \ldots, H_n$ to C: Under the induction hypothesis that $F(H_i)$ is a theorem for $i = 1, \ldots, n$, $F(C)$ must be a theorem. This is exactly what the above condition states. □

The necessitation rule arises in the special case where $F(p) = \Box p$. Lemma 7 is used to implement the strategy `direct_ind_pos`, which takes $\mathcal{L}$ and R as input and calls a first-order theorem prover to prove every induction step. Note that it would also be sufficient if the second condition quantified over the rules of $\mathcal{M}'$ instead of those of $\mathcal{L}$. But since these rules include R, it is less successful in practice.

3.3.2 Disproving

The direct strategy to disprove the subset relation $\mathcal{M} \subseteq \mathcal{L}$ is to show a certain satisfiability.

Lemma 8. *If R is an axiom or rule of $\mathcal{M}$, and if there is a first-order model M of $\mathcal{E}(\mathcal{L}) \cup \{\neg\, \mathcal{E}(R)\}$, then $\mathcal{M} \not\subseteq \mathcal{L}$.*

This approach is implemented in the strategy `direct_neg`, which calls a first-order model finder to search for a model of $\mathcal{E}(\mathcal{L}) \cup \{\neg\, \mathcal{E}(R)\}$ if R could not be proved by any positive strategy. This criterion is not complete since we only check finite models; see Section 4 for a discussion.

3.4 Strategies using Kripke semantics

This subsection presents a proving and a disproving strategy using relational translations, which we call Kripke-based strategies.

3.4.1 Proving

By standard first-order translation, we mean the translation based on the relational semantics of modal logics by making worlds explicit, e.g., $\Box p$ is translated to $\forall w\, \forall x\, (Acc(w,x) \;\rightarrow\; p(x))$ for an accessibility relation Acc (see Section 4.1 in Schmidt and Hustadt (2006)). Then we have:

Lemma 9. *Let*

(1) $\mathcal{L}$ be normal,
(2) $\mathcal{F}$ be a set of theorems of $\mathcal{L}$ that are Sahlqvist formulas,
(3) P be the first-order property of Kripke frames completely characterized by $\mathcal{F}$,
(4) R' be the standard first-order translation of the modal formula R,
(5) R' be first-order provable from P.

Then $\mathcal{L} \vdash R$.

This result follows from Sahlqvist's theorem (Sahlqvist, 1975). Lots of practically relevant axioms are Sahlqvist formulas, e.g., any formula of the form $F \rightarrow G$ where G is a positive formula and F is constructed by applying conjunction, disjunction and possibility to boxed atoms and negative formulas. To compute P from $\mathcal{F}$, we use the SCAN algorithm (Gabbay and Ohlbach, 1992; Goranko et al., 2004) for second-order quantifier elimination, for which an implementation is available. [4]

In our implementation, the first three steps, i.e., proving normality of $\mathcal{L}$ and computing P, are done in the preprocessing phase. The direct strategies are used for the normality proof. Then the strategy `kripke_pos` computes R' from R, where R is the rule or axiom that is to be proved from $\mathcal{L}$, and calls a first-order theorem prover to prove R' from P.

Note that we cannot use relational semantics in general, because Kripke semantics may not be sound (e.g., for `S1`) or not be complete (see, e.g., Thomason (1974)) for a given modal logic. It is necessary to find a set of Kripke frames that corresponds to the modal logic and show that this set of frames is complete for it. Lemma 9 gives the most important class of modal logics for which this has been proved.

[4] Technically, a SCAN implementation is only available for SunOS. Our Linux system outputs SCAN command lines, and the user has to run them on a SunOS machine and submit the result to a database.

3.4.2 Disproving

We cannot easily use the proving approach as a disproving strategy because, in general, it only gives us a sublogic of $\mathcal{L}$ that is characterized by the property of Kripke frames. But this is not necessary anyway because the following simpler and more general strategy is successful. We search for a Kripke model $m' = (U, Acc', \alpha)$ such that the formulas satisfied by m' include the theorems of $\mathcal{L}$ but not F, in order to prove $\mathcal{M} \not\subseteq \mathcal{L}$ for $\mathcal{M} \vdash F$; here U is the set of worlds, Acc' the accessibility relation, and α an assignment of truth values to the propositional variables of F. This means that, firstly, m' must satisfy all rules of $\mathcal{L}$, i.e., an instance of the conclusion of a rule must hold in all worlds whenever the appropriate instances of all hypotheses of the rule hold in all worlds. Secondly, m' must satisfy $\neg F$ in one world.

This is non-trivial to implement. If Kripke semantics is used to translate modal logic to first-order logic, the first-order language is not a meta-language anymore, i.e., modal formulas are translated to first-order formulas, not to terms. Therefore, the possibility of quantifying over all modal formulas is lost, which is necessary to express that a model satisfies a rule. To circumvent this problem, we fix the number of worlds in U, say n, and proceed as follows: We assume that all propositional variables are of the form p_j for some natural number j. We search for a first-order model m, from which we can construct the Kripke model m'. Let the first-order signature Σ contain the following symbols: constants $1, \ldots, n$ (intended semantics: one constant for every world of U), the constants t and f (intended semantics: truth values of truth and falsity), the binary predicate Acc (intended semantics: the accessibility relation Acc'), and one constant a_{ji} for every variable p_j occurring in F and for every $i = 1, \ldots, n$ (intended semantics: a_{ji} gives the value of the assignment α to p_j in world i). Now let $\tilde{\mathcal{E}}(\cdot)$ be the translation from modal logic rules and formulas to first-order logic over Σ defined as follows.

(1) A rule with hypotheses $H_1, \ldots, H_r$ and conclusion C containing the propositional variables $p_1, \ldots, p_s$ is translated to

$$\forall X_{11}, \ldots, X_{1n}, X_{21}, \ldots, X_{sn} \ : ((\tilde{\mathcal{E}}(H_1) \ \wedge \ \ldots \ \wedge \ \tilde{\mathcal{E}}(H_r)) \ \rightarrow \ \tilde{\mathcal{E}}(C))$$

(2) $\tilde{\mathcal{E}}(\Rightarrow FG)$ and $\tilde{\mathcal{E}}(\Leftrightarrow FG)$ are reduced to the other cases by replacing them with their definitions,

(3) A formula F is translated to $\tilde{\mathcal{E}}(F) = \bigwedge_{i=1}^{n} \tilde{\mathcal{E}}^i(F)$ where $\tilde{\mathcal{E}}^i(F)$ is given by

- $\tilde{\mathcal{E}}^i(\wedge FG) = \tilde{\mathcal{E}}^i(F) \ \wedge \ \tilde{\mathcal{E}}^i(G)$ and accordingly for the other binary propositional connectives,
- $\tilde{\mathcal{E}}^i(\neg F) = \neg \, \tilde{\mathcal{E}}^i(F)$,
- $\tilde{\mathcal{E}}^i(\Box F) = \bigwedge_{j=1}^{n} (Acc(i,j) \ \rightarrow \ \tilde{\mathcal{E}}^j(F))$,

- $\tilde{\mathcal{E}}^i(\Diamond(F)) = \bigvee_{j=1}^{n} (Acc(i,j) \ \wedge \ \tilde{\mathcal{E}}^j(F))$,
- $\tilde{\mathcal{E}}^i(p_j) = (X_{ji} = t)$ for a propositional variable p_j (where the first equality sign is a meta-operator and the second one the logical symbol).

Here, the intended semantics of $\tilde{\mathcal{E}}(F)$ for a formula F is that F holds in all worlds of m' and that of $\tilde{\mathcal{E}}^i(F)$ is that F holds in the world i. With these definitions, we have the following lemma.

Lemma 10. *If F is a theorem of $\mathcal{M}$ and there is an n such that a first-order Σ-model m exists satisfying the following axioms*

- *$\neg\ c = d$ for all constants c and d of Σ,*
- *$a_{ji} = t \ \vee \ a_{ji} = f$ for all constants a_{ji} of Σ,*
- *$\tilde{\mathcal{E}}(R)$ for every rule R of $\mathcal{L}$,*
- *$\neg\ F'$ where F' is as $\tilde{\mathcal{E}}^1(F)$ but with all variables X_{ji} replaced with a_{ji},*

then $\mathcal{M} \not\subseteq \mathcal{L}$.

Proof. From m, m' is constructed by

- U: the universe of m minus the interpretations of t and f,
- Acc': the restriction of the interpretation of Acc to U,
- for a variable p_j of F and a world i of U, $\alpha(p_j)(i)$ is true if $(a_{ji} = t)$ holds in the model, and false if $(a_{ji} = f)$ holds.

Let T be the set of modal formulas that hold in all worlds of m'. Then, we observe that the above translation indeed has the intended semantics, i.e., if for a rule R, $\tilde{\mathcal{E}}(R)$ holds in m, then if T contains the hypotheses of R, T also contains the conclusion of R. Therefore, $T \subseteq \mathcal{L}$. And also by the translation, since $\neg\ F'$ holds in m, F does not hold in world 1 of m', and therefore $F \notin \mathcal{L}$. Because F is a theorem of $\mathcal{M}$, we have $\mathcal{M} \not\subseteq \mathcal{L}$. □

This criterion can be applied regardless of whether $\mathcal{L}$ has a complete Kripke semantics, $\mathcal{L}$ does not even have to be normal. Whereas for the proving case, the lack of a complete Kripke semantics threatens soundness, for a disproving strategy, it only threatens completeness, which is harmless.

Lemma 10 is used to implement the strategy `kripke_neg`, which executes the above translation and calls a first-order model finder to search for the model m. Experiments showed that very low values of n, e.g., $n = 3$, already lead to very satisfactory results. For example, when trying to show $\mathsf{S1} \not\subseteq \mathsf{S1}^\circ$ with $F = M6$, our test runs returned m' as $U = \{0, 1, 2\}$ and $Acc' := U^2 \setminus \{(1,1),(1,0)\}$ with the only constant p_1 being true in the worlds 0 and 1 and false in world 2.

Indeed, m' satisfies all rules of $\mathsf{S1}^\circ$ (This is always the case if all worlds of m' are a successor of some world.), and *M6* does not hold in world 0.

3.5 Algebraic Strategies

In this section, we describe an algebraic strategy for exploring extensions of $\mathsf{S1}^\circ$. For a modal logic $\mathcal{L}$ we construct a Boolean algebra $\Pi^{\mathcal{L}}$ such that we can convert reasoning about formulae in $\mathcal{L}$ to algebraic reasoning about $\Pi^{\mathcal{L}}$. In general, this procedure could be applied to any modal logic, but we focus on extensions of $\mathsf{S1}^\circ$, for which the other strategies are not very successful.

Originally, the idea of using algebraic means to analyze the structure of modal systems appeared in (McKinsey, 1941), and it was further developed by Tarski and Jónsson (Jónsson and Tarski, 1951, 1952).

Since the focus of the paper is on the empirical results, we will only present the main theorems that are required to describe the strategy and only sketches of the proofs. The complete derivation of the theoretical background can be found in (Pudlák, 2006).

3.5.1 Theoretical Basis

First, let us give definitions of a few concepts that we shall use often throughout this section.

Definition 11 (Strict formulae)**.** We shall call a formula *strict* if its topmost connective is $\Box$ or $\Rightarrow$.

One of the defining rules of $\mathsf{S1}^\circ$ is the substitution of strict equivalents *EQS* (recall Definition 2). Therefore, we can factor the set of formulae by strict equivalence and explore the constructed factor. The following lemma summarizes the main properties. This can be proved easily from basic properties of $\mathsf{S1}^\circ$, and therefore, we omit the proof.

Lemma 12. *Let $\mathcal{L}$ be an extension of $\mathsf{S1}^\circ$. If we construct the (Lindenbaum-Tarski) algebra of $\mathcal{L}$ by factoring the modal formulae by strict equivalence, then the algebra is a Boolean algebra defined by $F \cap G = \wedge FG$ and $\overline{F} = \neg F$. Its top element $\top$ is the class of propositional tautologies, and we write $\top$ to abbreviate any such tautology whose variables are used nowhere else.*

Furthermore, if we view the algebra as a lattice, the relation $\mathcal{L} \vdash \Rightarrow FG$ is the ordering of the lattice. In particular, $\mathcal{L} \vdash \Rightarrow FG$ if and only if $\mathcal{L} \vdash \Leftrightarrow F \wedge FG$ (which is the same as $\mathcal{L} \vdash \Leftrightarrow \rightarrow FG\top$).

Looking at extensions $\mathcal{L}$ of $\mathsf{S1}^\circ$, our aim is to express $\mathcal{L} \vdash F$ using strict equivalence. Then we are able to express it as an equality in the algebra. It is not difficult to express the trueness of strict formulae, which can again be proved easily using basic properties of $\mathsf{S1}^\circ$:

Lemma 13. $\mathsf{S1}^\circ \vdash \Box F$ *if and only if* $\mathsf{S1}^\circ \vdash \Leftrightarrow F \top$.

However, we would like to be able to express trueness of all formulae. Let us first examine the special case that the extension is formed by just strict axioms.

Lemma 14. *If $\mathcal{L}$ is an extension of $\mathsf{S1}^\circ$ that can be constructed from $\mathsf{S1}^\circ$ by adding only strict axioms, then the rule $\frac{F}{\Rightarrow \Box \top F}$ (or equivalently $\frac{F}{\Leftrightarrow \to \Box \top F \top}$) is an admissible rule of $\mathcal{L}$. In other words, $\Box\top$ is the weakest true formula of the extension with respect to strict implication.*

Proof sketch. This is shown by induction on the proof of F. Since all the axioms are strict, the base case follows from Lemma 13, and we omit the induction step. □

The next lemma shows that if the extension is formed by adding arbitrary axioms, we can add a new logical constant π (a connective of arity 0) that will represent the weakest true formula:

Lemma 15. *Let $\mathcal{L}$ be the logic $\mathsf{S1}^\circ$ extended by the axioms $H_1, \ldots, H_n$. Let us construct an extension $\mathcal{L}_\pi$ of this logic by adding a new symbol π to the language of $\mathcal{L}$ and by adding the axiom and the rule*

$$\begin{aligned} &A_\pi : \pi \\ &R_\pi : \frac{F}{\Rightarrow \pi F} \quad \left(\textit{or equivalently} \quad \frac{F}{\Leftrightarrow \to \pi F \top} \right) \end{aligned}$$

Then, $\mathcal{L}_\pi$ is a conservative extension of $\mathcal{L}$, that is if F does not contain π, then $\mathcal{L}_\pi \vdash F$ if and only if $\mathcal{L} \vdash F$. Moreover, $\mathcal{L}_\pi \vdash F$ if and only if $\mathcal{L}_\pi \vdash \Rightarrow \pi F$, which is the same as $\mathcal{L}_\pi \vdash \Leftrightarrow \to \pi F \top$.

Proof. In the proof we shall often replace π in a formula F by another formula G. $F[\pi \rightsquigarrow G]$ is the formula obtained by replacing all occurrences of the symbol π in F by G.

We shall first prove two auxiliary propositions and then use them to prove the main statement.

(1) If $\mathcal{L}' \vdash F$ in an extension $\mathcal{L}'$ of $\mathsf{S1}^\circ$ (F may or may not contain π) then there is a proof[5] of F such that the first part of the proof consists only of applications of the rule of substitution for propositional variables to axioms, and the rest of the proof uses only the remaining three rules (substitution of strict equivalents, strict detachment and adjunction).

Proof. If we examine the three remaining rules, we see that the rules are closed under substitution for propositional variables. Instead of deriving F by one of the three rules and then substituting for variables, we can first substitute for variables and then apply the particular rule. Hence, we can propagate all uses of the rule of substitution for variables backwards, until the substitution is performed on only the axioms.

(2) If we can prove a formula F in $\mathcal{L}_\pi$ without using the rule R_π, then there is a formula E (not containing π) such that $\mathcal{L}_\pi \vdash E$ and such that we can prove $\mathcal{L}_\pi \vdash \Rightarrow \wedge E\pi F[\pi \rightsquigarrow \wedge E\pi]$ without using R_π.

Proof. By (1) we can construct a proof of F of the form

$$\underbrace{G_1, \ldots, G_k}_{\text{instances of the axioms}} \quad , \quad \underbrace{H_1, \ldots, H_n}_{\text{only the three remaining rules being used}}$$

where $H_n = F$. Let E be the formula $\wedge \ldots \wedge G_1 \ldots G_k$. This formula is surely true. We shall prove by induction on the length of the proof of H_i that $\mathcal{L}_\pi \vdash \Rightarrow \wedge E\pi H_i$ for $1 \leq i \leq n$. And moreover, each of the proofs will not use R_π.

It is clear that $\mathcal{L}_\pi \vdash \Rightarrow \wedge E\pi G_j$ for all G_js. By examining the all possible rules we show that $\mathcal{L}_\pi \vdash \Rightarrow \wedge E\pi H_i$ assuming that it is true for all H_js $(1 \leq j < i)$. Now, since $\mathcal{L}_\pi \vdash \wedge E\pi$ and since we have never used the rule R_π, we can replace π by $\wedge E\pi$ and get a proof of the formula

$$\Rightarrow \wedge E \wedge E\pi F[\pi \rightsquigarrow \wedge E\pi] \equiv \Rightarrow \wedge \wedge EE\pi F[\pi \rightsquigarrow \wedge E\pi] \equiv \Rightarrow \wedge E\pi F[\pi \rightsquigarrow \wedge E\pi].$$

We now prove the main statement of the lemma. Let F be a formula proved inside $\mathcal{L}_\pi$ such that F does not contain the symbol π. We shall show that F can also be proved just inside $\mathcal{L}$.

First, we use induction on the number of applications of R_π to prove that if G is any formula provable inside $\mathcal{L}_\pi$ then there is a formula D such that we can construct a proof of $G[\pi \rightsquigarrow D]$ without using R_π. Let $H_1, \ldots, H_n$ be the proof of G and let H_i be the first application of the rule R_π. Thus, H_i is $\Rightarrow \pi H_j$

[5] By a proof of F, we mean a sequence $G_1, \ldots, G_n$ such that $G_n = F$ and each G_i is either the axiom of $\mathcal{L}$ or G_i is derived from some of $G_1, \ldots, G_{i-1}$ using one of the rules of $\mathcal{L}$.

where $1 \leq j < i \leq n$. By (2) we can find a formula E and construct a proof of $\Rightarrow \wedge E\pi H_j[\pi \rightsquigarrow \wedge E\pi]$ without using R_π. Then the sequence

$$\Big(\text{proof of} \Rightarrow \wedge E\pi H_j[\pi \rightsquigarrow \wedge E\pi]\Big),\ H_1[\pi \rightsquigarrow \wedge E\pi], \ldots, H_n[\pi \rightsquigarrow \wedge E\pi]$$

is a proof of $G[\pi \rightsquigarrow \wedge E\pi]$. If some H_k $(1 \leq k \leq n)$ is the axiom $A_\pi : \pi$, then $H_k[\pi \rightsquigarrow \wedge E\pi] \equiv \wedge E\pi$ is a provable formula, and if some $H_k = \Rightarrow \pi H_m$ is the result of the application of the rule R_π to some formula H_m, then $H_k[\pi \rightsquigarrow \wedge E\pi] = \Rightarrow \wedge E\pi H_m[\pi \rightsquigarrow \wedge E\pi]$. To prove it, we apply R_π to $H_m[\pi \rightsquigarrow \wedge E\pi]$, get $\mathcal{L}_\pi \vdash \Rightarrow \pi H_m[\pi \rightsquigarrow \wedge E\pi]$, and by combining it with $\mathcal{L}_\pi \vdash \Rightarrow \wedge E\pi\pi$ we get $\mathcal{L}_\pi \vdash \Rightarrow \wedge E\pi H_m[\pi \rightsquigarrow \wedge E\pi]$.

Recall that H_i is the first result of the application of R_π. Since $H_i[\pi \rightsquigarrow \wedge E\pi]$ is just $\Rightarrow \wedge E\pi H_j[\pi \rightsquigarrow \wedge E\pi]$, we have proved $G[\pi \rightsquigarrow \wedge E\pi]$ using one less application of R_π. By induction hypothesis, we can then prove $G[\pi \rightsquigarrow D]$ for some formula D without using R_π at all.

Thus, since the formula F, whose proof we are looking for, does not contain π, we can construct a proof $F_1, \ldots, F_n, F$ of F without using the rule R_π. Now we replace π by an arbitrary axiom (we choose *M4*) and the sequence

$$F_1[\pi \rightsquigarrow \Rightarrow p \wedge pp], \ldots, F_n[\pi \rightsquigarrow \Rightarrow p \wedge pp], F$$

is a proof of F without π at all, hence a proof within $\mathcal{L}$. □

The extensions with the added symbol π have one significant disadvantage – rules cannot be disproved. If we prove that a rule is an admissible rule of $\mathcal{L}_\pi$, then it is surely an admissible rule of $\mathcal{L}$. But the case where a rule is not an admissible rule of $\mathcal{L}_\pi$ is problematic. For example, the necessitation rule $\frac{F}{\Box F}$ is an admissible rule of $\mathsf{S4}$, but not a rule of $\mathsf{S4}_\pi$ since we know nothing about $\Box\pi$. Clearly, finding out that $\frac{F}{\Box F}$ is not a rule of $\mathsf{S4}_\pi$ gives no information about admissibility of the rule in $\mathsf{S4}$.

Therefore, we add π whenever possible and finally obtain the following theorem as the basis of our strategy.

Theorem 16. *Let $\mathcal{L}$ be the logic $\mathsf{S1}^\circ$ extended with the axioms $H_1, \ldots, H_n$. Let $\Pi^{\mathcal{L}}$ be the free algebra defined by the theory Def given in Section 2.3 extended with constants* `true` *and π, and the following axioms*[6] *(where, for brevity, we*

[6] The relation (1) is an implication of equations. Thus, these equational relations do not form a variety but a quasi-variety.

omit the universal quantifiers):

$$\begin{aligned}
&\texttt{and}(X,Y) = \texttt{and}(Y,X)\\
&\texttt{and}(X,\texttt{and}(Y,Z)) = \texttt{and}(\texttt{and}(X,Y),Z)\\
&\texttt{and}(X,\texttt{or}(X,Y)) = X\\
&\texttt{and}(X,\texttt{or}(Y,Z)) = \texttt{or}(\texttt{and}(X,Y),\texttt{and}(X,Z))\\
&\texttt{true} = \texttt{not}(\texttt{and}(X,\texttt{not}(X)))\\
&\texttt{impl}(\texttt{and}(\texttt{s\_impl}(X,Y),\texttt{s\_impl}(Y,Z)),\texttt{s\_impl}(X,Z)) = \texttt{true}\\
&\texttt{impl}(\pi,\texttt{necess}(\texttt{true})) = \texttt{true}\\
&\texttt{impl}(\pi,\texttt{necess}(X)) = \texttt{true} \ \rightarrow \ X = \texttt{true} && (1)\\
&\texttt{impl}(\pi,\varepsilon(H_1)) = \texttt{true}\\
&\vdots\\
&\texttt{impl}(\pi,\varepsilon(H_n)) = \texttt{true}
\end{aligned}$$

(1) If all the axioms $H_1,\ldots,H_n$ are strict, we also add the equation

$$\pi = \texttt{necess}(\texttt{true})$$

Then $\mathcal{L} \vdash A$ for a formula A if and only if

$$\Pi^{\mathcal{L}} \vdash^{FOL} \texttt{impl}(\texttt{necess}(\texttt{true}),\varepsilon(A)) = \texttt{true}$$

(2) If some of the axioms are not strict, then $\mathcal{L} \vdash A$ if and only if

$$\Pi^{\mathcal{L}} \vdash^{FOL} \texttt{impl}(\pi,\varepsilon(A)) = \texttt{true}$$

Proof sketch. The axiom $A_\pi : \pi$ and the rule $R_\pi : \frac{F}{\Rightarrow \pi F}$ in the extension $\mathcal{L}_\pi$ from Lemma 15 together with the rule of strict detachment guarantee that deriving $\mathcal{L}_\pi \vdash \Rightarrow \pi G$ is equivalent to deriving $\mathcal{L}_\pi \vdash G$ and hence equivalent to $\mathcal{L} \vdash G$, if G does not contain π. If in addition all the axioms $H_1,\ldots,H_n$ are strict then the conditions of Lemma 14 are satisfied and we can explicitly set $\pi = \texttt{necess}(\texttt{true})$.

It can be easily proved that for all these equations the corresponding equivalences are true in the corresponding system $\mathcal{L}_\pi$. Rule (1) is just Lemma 13. The rule of substitution of strict equivalents justifies combining equivalences in extensions of $\mathsf{S1}^\circ$ just in the same way as equations, therefore anything we derive from the equations can be derived as an equivalence within $\mathcal{L}_\pi$ as well.

Now, let us prove the opposite, that if $\mathcal{L} \vdash G$ then we can derive

$$\texttt{impl}(\pi,\varepsilon(G)) = \texttt{true}$$

using the equations. We shall prove that by induction on the number of steps of the proof of a formula G. This is trivial for the additional axioms $H_1,\ldots,H_n$

of $\mathcal{L}$ and it can be also easily shown for the axioms *M1–M5* of $\mathsf{S1}^\circ$. We then complete the proof by examining the last rule from the proof of G and showing that if we can derive the equality for all preceding formulae in the proof then we can derive the equality for G. □

3.5.2 Implementation

The algebraic strategy for an axiomatization $\mathcal{L}$ with axioms $\mathcal{A}$ and other rules $\mathcal{R}$ is prepared by the following steps which are executed in the preprocessing phase:

(1) Try to prove all the axioms and all the rules of $\mathsf{S1}^\circ$ from $\mathcal{L}$. If successful, then $\mathcal{L} = \mathsf{S1}^\circ + \mathcal{A} + \mathcal{R}$.
(2) For every rule $R \in \mathcal{R}$ try to prove R from $\mathsf{S1}^\circ + \mathcal{A}$. If successful, then $\mathcal{L} = \mathsf{S1}^\circ + \mathcal{A}$.
(3) Try to prove that the rule $\frac{\Box F}{F}$ is admissible in $\mathsf{S1}^\circ + \mathcal{A}$.
(4) Construct $\mathcal{A}'$ from $\mathcal{A}$ as follows: For every every axiom $F \in \mathcal{A}$ that is not strict, prove $\mathsf{S1}^\circ + \mathcal{A} \vdash \Box F$ and replace F in $\mathcal{A}$ with $\Box F$. If successful, then $\mathcal{L} = \mathsf{S1}^\circ + \mathcal{A}'$.

The mentioned proofs are attempted using the direct proving strategy. Then Theorem 16 yields the soundness of the following strategy, which is called to prove or disprove R from $\mathcal{L}$:

- If steps 1 to 4 have been successful, construct the algebra $\Pi^{\mathsf{S1}^\circ + \mathcal{A}'}$ with the additional equation $\pi = \texttt{necess(true)}$. Call a first-order theorem prover or model finder to prove or disprove R, respectively.
- If only steps 1 and 2 have been successful, construct the algebra $\Pi^{\mathsf{S1}^\circ + \mathcal{A}}$ (without the additional equation). If R is an axiom, call a first-order theorem prover or model finder to prove or disprove R, respectively. If R is not an axiom, call a theorem prover to prove R (i.e., the strategy is not applicable for disproving rules).

4 Results

We ran our implementation on all 121 pairs of axiomatizations of the modal logic challenge on a machine with an 3.0GHz PC with 1GB memory, running Linux 2.6. For the proving strategies, we used the prover Vampire 7.45 (Riazanov and Voronkov, 2002) with a time limit of five minutes, and for the disproving strategies we used the model finder Paradox 1.3 (Claessen and Sorensson, 2003) with 8 elements per model for the direct and algebraic strategies and 3 worlds per Kripke-model for the Kripke-based strategy. All tools were used with default settings. To compare the strategies against each other, we repeated the experiment three more times switching off the Kripke-based or the algebraic strategies or both, respectively.

The results are given in Fig. 6. For the run with all strategies switched on, we took the run time. First all axiomatizations went through the preprocessing which was timed independently. Then for every pair $(\mathcal{L}, \mathcal{M})$ of axiomatizations, both directions of the comparison were run and timed separately. The results of (dis-)proving $\mathcal{M}$ from $\mathcal{L}$ are given in row $\mathcal{L}$, column $\mathcal{M}$. Remember that when (dis-)proving $\mathcal{M}$ from $\mathcal{L}$, the system tries to prove every rule or axiom of $\mathcal{M}$ from $\mathcal{L}$ trying every applicable proving strategy. The strategies were applied in the order Kripke-based, algebraic, direct. If that fails, it tries to disprove the relationship.

It can be seen that the system can solve all but six instances of the challenge. In all five attempted derivations of K-based axiomatizations from $\mathsf{S5}_\alpha$, which failed even when all strategies were used, the direct strategies failed only because the induction step for the axiom B in the derivation of *Nec* in the strategy `direct_ind_pos` failed. Thus normality could not be established either, and the Kripke-based strategies could not be applied. The algebraic strategy was not successful either because the $\mathsf{S1}^\circ$-based axiomatization involves a non-strict axiom, which makes it less efficient. Note that because $\mathsf{S4}_\alpha$ does not have the axiom B, we could prove more inclusions from $\mathsf{S4}_\alpha$ than from the stronger system $\mathsf{S5}_\alpha$. The sixth failing case is to disprove $\mathsf{S1}^\circ$ from K. Here the problem is that only those axioms and rules that could not be proved are used as potential counter-examples. A stronger strategy could apply the unproved rules to generate more formulae that may be counter-examples.

The preprocessing times are very high because the preprocessing already involves proving tasks, and every failed proving attempt takes five minutes. In particular, the ultimately failing attempts to establish normality lead to very high preprocessing times. On the other hand, this significantly reduces the run time spent in the comparison phases.

The execution time for the comparisons where the inclusion must be disproved is extremely high. This was to be expected because disproving is tried only after all proving strategies have failed. A reimplementation should switch between trying to prove and disprove the inclusion. Due to the preprocessing, when the inclusion can be proved, the execution time is either very small or medium. This mainly depends on how often a strategy is invoked that fails. For example, proving an $\mathsf{S1}^\circ$-based axiomatization from itself can take surprisingly long because the algebraic strategy may time out for one rule, which is then proved instantaneously by the direct strategy. All proved inclusions take less than 900 seconds, i.e., there are at most two failing proving attempts.

	P	K	S1°	S1	T	S3	S4$_\alpha$	S4$_\beta$	S5$_\alpha$	S5$_\beta$	S5$_\gamma$	S5$_\delta$
K	16	⊆ 0 ● ●● ●	⊈ ○ ○○ ○	⊈ 2110 ● ●○ ○	⊈ 301 ● ●○ ○	⊈ 2410 ● ●○ ○	⊈ 2710 ● ●○ ○	⊈ 602 ● ●○ ○	⊈ 3010 ● ●○ ○	⊈ 902 ● ●○ ○	⊈ 2115 ● ●○ ○	⊈ 602 ● ●○ ○
S1°	61	⊈ 3339 ● ○● ○	⊆ 319 ● ●● ●	⊈ 932 ● ●● ○	⊈ 3954 ● ●● ○	⊈ 1537 ● ●● ○	⊈ 2145 ● ●● ○	⊈ 4553 ● ●● ○	⊈ 2736 ● ●● ○	⊈ 5166 ● ●● ○	⊈ 934 ● ●● ○	⊈ 4551 ● ●● ○
S1	55	⊈ 2915 ● ○● ○	⊆ 349 ● ●● ●	⊆ 348 ● ●● ●	⊈ 2927 ● ○● ○	⊈ 963 ● ●● ○	⊈ 1564 ● ●● ○	⊈ 3535 ● ●● ○	⊈ 2164 ● ●● ○	⊈ 4136 ● ●● ○	⊈ 963 ● ●● ○	⊈ 3536 ● ●● ○
T	14	⊆ 0 ● ●● ●	⊆ 0 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 0 ● ●● ●	⊈ 301 ● ●○ ○	⊈ 602 ● ●○ ○	⊈ 301 ● ●○ ○	⊈ 902 ● ●○ ○	⊈ 602 ● ●○ ○	⊈ 301 ● ●○ ○	⊈ 301 ● ●○ ○
S3	52	⊈ 3118 ● ○● ○	⊆ 349 ● ●● ●	⊆ 349 ● ●● ●	⊈ 2909 ● ○● ○	⊆ 350 ● ●● ●	⊈ 968 ● ○● ○	⊈ 3524 ● ○● ○	⊈ 1569 ● ●● ○	⊈ 4124 ● ●● ○	⊈ 964 ● ●● ○	⊈ 3521 ● ●● ○
S4$_\alpha$	24	⊆ 312 ● ●● ○	⊆ 358 ● ●● ●	⊆ 358 ● ●● ●	⊆ 322 ● ●● ○	⊆ 359 ● ●● ●	⊆ 359 ● ●● ●	⊈ 334 ● ●● ○	⊈ 974 ● ●● ○	⊈ 949 ● ●● ○	⊈ 973 ● ●● ○	⊈ 938 ● ●● ○
S4$_\beta$	11	⊆ 0 ● ●● ●	⊆ 0 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 0 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊈ 417 ● ●○ ○	⊈ 419 ● ●○ ○	⊈ 431 ● ●○ ○	⊈ 429 ● ●○ ○
S5$_\alpha$	54	⊆ ○ ○○ ○	⊆ 345 ● ●● ●	⊆ 344 ● ●● ●	⊆ ○ ○○ ○	⊆ 358 ● ●● ●	⊆ 343 ● ●● ●	⊆ ○ ○○ ○	⊆ 343 ● ●● ●	⊆ ○ ○○ ○	⊆ 447 ● ●● ○	⊆ ○ ○○ ○
S5$_\beta$	18	⊆ 0 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●
S5$_\gamma$	27	⊆ 376 ● ○● ○	⊆ 340 ● ●● ●	⊆ 439 ● ●● ○	⊆ 475 ● ○● ○	⊆ 544 ● ●● ○	⊆ 646 ● ●● ○	⊆ 576 ● ○● ○	⊆ 747 ● ●● ○	⊆ 678 ● ○● ○	⊆ 338 ● ●● ●	⊆ 475 ● ○● ○
S5$_\delta$	15	⊆ 0 ● ●● ●	⊆ 0 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●	⊆ 1 ● ●○ ○	⊆ 1 ● ●● ●

P: Preprocessing time in minutes
⊆ or ⊈ in row $\mathcal{L}$, column $\mathcal{M}$: $\mathcal{M} \subseteq \mathcal{L}$ or $\mathcal{M} \not\subseteq \mathcal{L}$, respectively
number: (dis-)proving time in seconds
● or ○: system returned correct answer or failed, respectively
bottom symbol: only direct strategies used
middle left symbol: direct and Kripke-based strategies used
middle right symbols: direct and algebraic strategies used
top symbol: all strategies used

Fig. 6. Experimental Results

When comparing the strategies, we find that the Kripke-based and the algebraic strategy complement each other nicely. This is not surprising since the former is strong for normal logics and the latter for S1°-based axiomatizations. It cannot be seen from the table that the direct proving strategy was not superfluous: Apart from being needed to establish applicability of the other two more sophisticated strategies, it occasionally succeeded when the other ones failed, e.g., in the example above. Furthermore, the inductive direct strategy `direct_ind_pos` was often needed to prove the necessitation rule.

The disproving results show that the direct disproving strategy was never successful. The reason for the failure is that only finite models are considered, while the first-order universe of the model needs to contain an interpretation of every formula. Using Herbrand models promises to be a more successful strategy, which is possible using the Darwin model finder (Baumgartner et al., 2005). However, since the other two strategies were so successful, we have not pursued this. In general, we were surprised to find the disproving cases to be the much simpler than the proving cases.

5 Conclusion and Future Work

We have presented a system that approaches the open challenge problem of automatically determining the subset relationship between modal logics. The correctness of the system is based on theoretical development that in turn depends on successful proofs, in order to admit the various preprocessing steps (e.g., the proofs that show equivalence of the four axiomatizations of PC) and comparison strategies (e.g., proofs of the congruence rules to admit efficient equational reasoning). The full system has been tested on 121 pairs of 11 axiomatizations of 7 common modal logics. Only six cases could not be solved because of, in total, two failing subcases, thus obtaining a high degree of empirical success.

Future work will focus mainly on improving the efficiency and usability of the system. It may prove useful to develop heuristics that govern the order of strategy application. The system should switch between trying to prove and trying to disprove an inclusion. It is also promising to conduct experiments in order to further optimize the time limits for proving and the model sizes for disproving attempts or to change these values dynamically.

Only minor improvements of the underlying theoretical results are necessary. In particular, the strategy `direct_neg` should be improved to check infinite models. If there are rules that cannot be proved, they should be applied a few times to generate theorems which can serve as potential counter-examples for the disproving strategies. It may also be worthwhile to investigate whether an algebraic treatment of normal logics is more powerful than using Kripke semantics. Of course, it is generally interesting to consider integrating more strategies, e.g., the decidability results of Rybakov (Rybakov, 1997), if they can be formulated to apply to big classes of logics with decidable applicability conditions.

References

Baumgartner, P., Fuchs, A., Tinelli, C., 2005. Implementing the Model Evolution Calculus. In: Schulz, S., Sutcliffe, G., Tammet, T. (Eds.), Special Issue of the International Journal of Artificial Intelligence Tools (IJAIT). Vol. 15 of International Journal of Artificial Intelligence Tools.

Bernays, P., 1926. Axiomatische Untersuchungen des Aussagenkalküls der Principia Mathematica. Mathematiche Zeitschrift 25, 305–320.

Blackburn, P., de Rijke, M., Venema, Y., 2001. Modal logic. Cambridge University Press, New York, NY, USA.

Claessen, K., Sorensson, N., 2003. New techniques that improve MACE-style finite model finding. In: CADE-19 Workshop on Model Computation - Principles, Algorithms, Applications.

Fitting, M., Mendelsohn, R., 1998. First-Order Modal Logic. Kluwer.

Gabbay, D., Ohlbach, H., 1992. Quantifier elimination in second-order predicate logic. In: Nebel, B., Rich, C., Swartout, W. (Eds.), Principles of Knowledge Representation and Reasoning (KR92). Morgan Kaufmann, pp. 425–435.

Giunchiglia, E., Giunchiglia, F., Tacchella, A., 2002. SAT-Based Decision Procedures for Classical Modal Logics. Journal of Automated Reasoning 28 (2), 143–171.

Goranko, V., Hustadt, U., Schmidt, R., Vakarelov, D., 2004. SCAN is complete for all Sahlqvist formulae. In: Berghammer, R., Moller, B., Struth, G. (Eds.), Relational and Kleene-Algebraic Methods in Computer Science. pp. 149–162.

Goré, R., Heinle, W., Heuerding, A., 1997. Relations between propositional normal modal logics: An overview. Journal of Logic and Computation 7 (5), 649–658.

Halleck, J., 2006. Logic Systems. WWW pages, see `http://www.cc.utah.edu/~nahaj/logic/structures/systems/index.html`.

Hilbert, D., Bernays, P., 1934. Grundlagen der Mathematik. Julius Springer Verlag.

Hughes, G., Cresswell, M., 1996. A New Introduction to Modal Logic. Routledge.

Hustadt, U., Schmidt, R. A., 2000. MSPASS: Modal Reasoning by Translation and First-Order Resolution. In: Dyckhoff, R. (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods, International Conference (TABLEAUX 2000). pp. 67–71.

Iemhoff, R., Metcalfe, G., 2007. Proof theory for admissible rules. Preprint submitted to `http://www.phil.uu.nl/preprints/lgps/`.

Jónsson, B., Tarski, A., 1951. Boolean algebras with operators, I. Amer. J. Math. 73, 891–939.

Jónsson, B., Tarski, A., 1952. Boolean algebras with operators, II. Amer. J. Math. 74, 127–162.

Kracht, M., 1990. An almost general splitting theorem for modal logic. Studia

Logica 49 (4), 455–470.

Kripke, S., 1963. Semantical analysis of modal logic I. Normal modal propositional calculi. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 9, 67–96.

Lemmon, E., 1966. Algebraic Semantics for Modal Logics II. The Journal of Symbolic Logic 31, 191–218.

Lewis, C., Langford, C., 1932. Symbolic Logic. The Century Co, New York and London.

Łukasiewicz, J., 1963. Elements of Mathematical Logic. Pergamon Press.

McCune, W., Veroff, R., Fitelson, B., Harris, K., Feist, A., Wos, L., 2002. Short single axioms for boolean algebra. Journal of Automated Reasoning archive 29 (1), 1–16.

McCune, W., Wos, L., 1992. Experiments in automated deduction with condensed detachment. In: CADE-11: Proceedings of the 11th International Conference on Automated Deduction. Springer, pp. 209–223.

McKinsey, J. C., December 1941. A solution of the decision problem for the Lewis systems S2 and S4 with an application to topology. The Journal of Symbolic Logic 6 (4), 117–134.

Ohlbach, H., Schmidt, R., 1997. Functional translation and second-order frame properties of modal logics. Journal of Logic and Computation 7 (5), 581–603.

Pudlák, P., 2006. Verification of mathematical proofs. Ph.D. thesis, Charles University in Prague, Faculty of Mathematics and Physics, http://lipa.ms.mff.cuni.cz/~pudlak/pp-thesis.ps.gz.

Rabe, F., 2006a. Determining the Subset Relation between Propositional Modal Logics. See http://kwarc.eecs.iu-bremen.de/frabe/Research/moloss/index.html.

Rabe, F., 2006b. Towards Determining the Subset Relation between Propositional Modal Logics. In: Sutcliffe, G., Schmidt, R., Schulz, S. (Eds.), Proceedings of the FLoC 06 Workshop on Empirically Successful Computerized Reasoning, 3rd International Joint Conference on Automated Reasoning. Vol. 192 of CEUR Workshop Proceedings. pp. 126–140.

Riazanov, A., Voronkov, A., 2002. The design and implementation of Vampire. AI Communications 15, 91–110.

Rosser, J., 1953. Logic for Mathematicians. McGraw-Hill.

Russell, B., Whitehead, A., 1910. Principia Mathmatica. Cambridge University Press.

Rybakov, V., 1997. Admissibility of Logical Inference Rules. North-Holland.

Sahlqvist, H., 1975. Completeness and Correspondence in the First and Second Order Semantics for Modal Logic. In: Kanger, S. (Ed.), Proceedings of the Third Scandinavian Logic Symposium. North-Holland, pp. 110–143.

Schmidt, R., Hustadt, U., 2006. First-Order Resolution Methods for Modal Logics. To appear in Volume in memoriam of Harald Ganzinger.

SML, 2007. Standard ML of New Jersey. See http://www.smlnj.org.

Sutcliffe, G., 2006. The Modal Logic $100 Challenge. See http://www.cs.

miami.edu/~tptp/HHDC/.

Sutcliffe, G., Suttner, C., 1998. The TPTP Problem Library: CNF Release v1.2.1. Journal of Automated Reasoning 21 (2), 177–203.

Thomason, S., 1974. An incompleteness theorem in modal logic. Theoria 40, 30–34.

Zeman, J. J., 1973. Modal Logic, the Lewis-Modal systems. Oxford University Press, http://www.clas.ufl.edu/users/jzeman/modallogic/.