Towards Determining the Subset Relation between Propositional Modal Logics

Florian Rabe[1,*]
[1]Carnegie Mellon University, International University Bremen

July 3, 2006

Abstract

We present design and implementation of a system that tries to automatically determine the subset relation between two given axiomatizations of almost arbitrary propositional modal logics, which is an open challenge problem for automated theorem proving. A test suite shows that relatively simple strategies can lead to satisfactory results, but also that certain subproblems are hard for current automated theorem provers.

1 Introduction

Plain propositional modal logic is standard knowledge (see, e.g., [HC96]). However, there is a fundamental problem that has not been addressed in a focussed way, namely the automated decision whether among two given axiomatizations one is weaker or stronger than the other one. For example, the Modal Logic \$ 100 Challenge ([Sut]) calls for a program that can answer this problem for the most common Hilbert-style axiomatizations of K, T, S1, $S1_0$, S3, S4 and S5 (see [Hal] for an overview and a list of references to various modal logics and axiomatizations).

In general, this problem is undecidable (see [HV89] for an example). But decidability can be established separately for lots of modal logics (e.g., [HM92]), e.g., using a complete Kripke semantics ([Kri63]) and methods like filtration ([Lem66]). There is a variety of implementations (see, e.g., [HS00], [GTG02]) of theorem provers for specific modal logics.

However, such separate implementations are relative to a specific modal logic. Sometimes the set of axioms can be varied, in particular by imposing additional restrictions on the used Kripke frames like reflexivity and transitivity. But we are not aware of a modal theorem prover permitting an arbitrary set of inference rules. Furthermore, to decide whether two distinct axiomatizations generate the same logic, a decision procedure for that logic is difficult to use since it is not always clear which derived rules it uses implicitly. Also, checking the subset relation between modal logics requires to derive rules whereas theorem provers focus on deriving axioms.

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

The presented work describes a general approach to the above-mentioned challenge problem. We provide a modular framework into which different strategies can be plugged in. Thus incomplete strategies covering different cases can be combined. Experimental results show that in certain cases, this works surprisingly well and identify the subproblems that still present the greatest challenges.

Sect. 2 gives the main definitions, Sect. 3 and 4 describe the currently implemented strategies, Sect. 5 introduces the system, and Sect. 6 analyzes its current and potential future strength.

2 Definitions

We encode modal logic in first-order logic, i.e., every modal formula is a first-order term. We use the following symbols for the first-order meta logic: $\forall$, $\exists$, $\neg$, $\wedge$, $\Rightarrow$, $\doteq$. The theory $\mathcal{T}$ of first-order logic with equality is defined as follows:

- primitive function symbols *or* (disjunction, binary), *not* (negation) and *box* (necessity, both unary),
- further function symbols *and* (conjunction), *imp* (implication), *eqv* (equivalence), *simp* (strict implication), *seqv* (strict equivalence, all binary) and *dia* (possibility, unary) along with equality axioms that (classically) define these symbols in terms of the primitive ones, in particular strict implication and strict equivalence as
  - $\forall x, y\ \mathit{simp}(x, y) \doteq \mathit{box}(\mathit{imp}(x, y))$,
  - $\forall x, y\ \mathit{seqv}(x, y) \doteq \mathit{and}(\mathit{box}(\mathit{imp}(x, y)), \mathit{box}(\mathit{imp}(y, x)))$,
- a unary predicate symbol p (expressing provability).

A modal formula is a term of this first-order language, in particular propositional variables of modal logic are identified with first-order variables, and henceforth, just called variables. Provability of $f(X)$ is encoded as $\forall X\ p(f(X))$ (see below for the implicit use of uniform substitution). Here, and throughout the paper, X abbreviates $x_1, \dots, x_m$.

Axiom/rule	Encoding
$\Box(A \to B) \to (\Box A \to \Box B)$	$\forall x, y\ p\big(\mathit{imp}(\mathit{box}(\mathit{imp}(x, y)), \mathit{imp}(\mathit{box}(y), \mathit{box}(x)))\big)$
$\dfrac{A \quad A \to B}{B}$	$\forall x, y\ \big((p(y) \wedge p(\mathit{imp}(x, y))) \Rightarrow p(y)\big)$
$\dfrac{A}{\Box A}$	$\forall x\ (p(x) \Rightarrow p(\mathit{box}(x)))$

Figure 1: Encoding of K

Furthermore, we define:

- A rule is an additional axiom for $\mathcal{T}$ that is in Horn form and does not contain equality. In other words, the rule scheme

  $$\frac{h_1(X) \quad \ldots \quad h_n(X)}{c(X)}$$

  is encoded as

  $$\forall X\big((p(h_1(X) \wedge \ldots \wedge p(h_n(X))) \Rightarrow p(c(X))\big).$$

  If $n = 0$, the rule is called an axiom.

- A modal logic is an extension of $\mathcal{T}$ with a finite set of rules.

- A theorem of the modal logic L is a formula $f(X)$ such that $\forall X\ p(f(X))$ is a consequence of L. We write this as $f \in L$.

- A modal logic M is a subset of the modal logic L, written $M \subseteq L$, iff all theorems of M are theorems of L.

As an example, Fig. 1 gives the common and the encoded axiomatization of the modal logic K.

Let US be the rule of uniform substitution, i.e., substitution of all occurrences of a variable with the same modal formula. Then the above encoding is sound in the following sense: If L is a modal logic in our sense arising as the encoding of a Hilbert-style set S of axioms and rules and if f is the encoding a modal formula ϕ, then $f \in L$ iff ϕ is a theorem of $S \cup \{US\}$.

We only consider modal logics with the following two properties.

Firstly, the set of theorems is closed under uniform substitution. This is fundamental for the soundness of the above encoding. We do not know of any used modal logic that does not have this property.

Secondly, the set of theorems contains all classical propositional tautologies. This assumption is connected to a principal simplification of the challenge problem: We do not assume specific, let alone minimal axiomatizations for the propositional part. This greatly increases performance without dropping too much of the original challenge. In fact, it can be argued that proving propositional theorems from specific axiom sets should be another challenge independent from the modal logic challenge (see for example the problems in the Logic Calculi domain of TPTP ([SS98])). Furthermore, there is no discussion which propositional formulas are considered theorems (except for intuitionistic logic, but all modal logics considered in the challenge are classical) so that the choice of axioms becomes just a matter of implementation. For modal logic on the contrary, there are different applications and interpretations that naturally lead to different and not necessarily equivalent axiomatizations.

3 Positive Criteria

The most straightforward approach would be to let a first-order theorem prover prove the inclusion directly, e.g., by renaming the predicate p to q in one of two modal logics, say L_p and L_q and then trying to prove $\forall x(p(x) \Rightarrow q(x))$, which corresponds to $L_p \subseteq L_q$.

This is far from possible. Instead we break the problem down into subproblems treating every rule of L_q separately. The resulting problems can be attacked by existing tools.

In the sequel, we present simple criteria how to do that. These criteria are not necessarily complete and cover different situations. For the remainder of this section, let L and M be modal logics and let M' be as M but with an additional rule r.

Let $n = 0$, i.e., r is an axiom, say $r = \forall X p(c(X))$. Clearly, we have

Lemma 1. *$M' \subseteq L$ iff $M \subseteq L$ and $c \in L$*

This criterion can be used quite well by theorem provers. It can fail if r is complex. The following more sophisticated criterion using relational semantics and correspondence results can be applied less often but is successful in some cases in which Lem. 1 fails in practice.

Lemma 2. *Let*

1. *L be normal, i.e., $K \in L$ and L is closed under MP (modus ponens) and N (necessitation) (see Fig. 2 for the formulas; most of the time we use the same names as [Hal]),*
2. *F be a set of axioms of L that are Sahlqvist formulas,*
3. *P be the first-order property of Kripke frames completely characterized by F,*
4. *c' be the standard first-order translation of the modal formula c by making worlds explicit,*
5. *c' be first-order provable from P.*

Then $c \in L$.

By standard first-order translation, we mean the relational semantics of modal logics, e.g., $\Box P$ is translatated to $\forall w \forall x(Acc(w, x) \to P(x))$ for an accessibility realtion *Acc*.

This result is due to Sahlqvist [Sah75]. Lots of practically relevant axioms are Sahlqvist formulas, e.g., any formula of the form $A \to B$ where B is a positive formula and A is constructed by applying conjunction, disjuncton and possiblity to boxed atoms and negative formulas. P can be computed from F using the SCAN algorithm ([GO92]) for second-order quantifier elimination, for which an implementation is available. We are currently working on implementing this strategy, so far only the special case where P does not exclude any Kripke frames is used.

Note that we cannot use relational semantics in general since Kripke semantics may be not sound (e.g., for S1) or not complete (see, e.g., [Tho74]) 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. The above criterion gives the most important class of modal logics where this has been done.

If r is not an axiom, the situation is more complicated. The obvious naive approach is given by

Lemma 3. *$M' \subseteq L$ if $M \subseteq L$ and r is a first-order consequence of L.*

Axioms	
K	$\forall X, Y\ p(imp(box(imp(X,Y)), imp(box(X), box(Y))))$
T	$\forall X\ p(imp(box(X), X))$
4	$\forall X\ p(imp(box(X), box(box(X))))$
B	$\forall X\ p(imp(X, box(dia(X))))$
5	$\forall X\ p(imp(dia(X), box(dia(X))))$
T2_1	$\forall X, Y\ p(eqv(dia(or(X,Y)), or(dia(X), dia(Y))))$
T2_2	$\forall X\ p(imp(X, dia(X)))$
M1	$\forall X, Y\ p(simp(and(X,Y), and(Y,X)))$
M2, H2	$\forall X, Y\ p(simp(and(X,Y), X))$
M3	$\forall X, Y, Z\ p(simp(and(and(X,Y), Z), and(X, and(Y,Z))))$
M4, H1	$\forall X\ p(simp(X, and(X,X)))$
M5	$\forall X, Y, Z\ p(simp(and(simp(X,Y), simp(Y,Z)), simp(X,Z)))$
M6, H5	$\forall X\ p(simp(X, dia(X)))$
M8	$\forall X, Y\ p(simp(simp(X,Y), simp(dia(X), dia(Y))))$
M9, H7	$\forall X\ p(simp(dia(dia(X)), dia(X)))$
M10	$\forall X\ p(simp(dia(X), box(dia(X))))$
AS1_6	$\forall X, Y\ p(simp(and(X, simp(X,Y)), Y))$
H3	$\forall X, Y, Z\ p(simp(and(and(Z,X), not(and(Y,Z))), and(X, not(Y))))$
H4	$\forall X\ p(imp(not(dia(X)), not(X)))$
H6	$\forall X, Y\ p(simp(simp(X,Y), simp(not(dia(Y)), not(dia(X)))))$
Other Rules	
MP	$\forall X, Y\ ((p(X) \wedge p(imp(X,Y))) \Rightarrow p(Y))$
N	$\forall X\ (p(X) \Rightarrow p(box(X)))$
REM	$\forall X, Y\ (p(eqv(X,Y)) \Rightarrow p(eqv(dia(X), dia(Y))))$
SMP	$\forall X, Y\ ((p(X) \wedge p(simp(X,Y))) \Rightarrow p(Y))$
AD	$\forall X, Y\ ((p(X) \wedge p(Y)) \Rightarrow p(and(X,Y)))$

Figure 2: Rules

The opposite direction does not hold making this criterion incomplete. Essentially, two things can go wrong.

Firstly, deriving r from L means to show that whenever L contains instances of the hypotheses of r, it also contains the appropriate instance of the conclusion. The necessary and sufficient condition, however, is that whenever M' contains instances of the hypotheses, then L contains the appropriate instance of the conclusion. For a trivial example, let M be empty, r be the rule $\forall x(p(x) \Rightarrow p(not(x)))$, and L be any consistent non-empty modal logic. Clearly r is not derivable from L, but M' is empty (An axiomatization without axioms has no theorems.) and therefore, a subset of L.

Secondly, even if the opposite direction holds, Lem. 3 may be ineffective in practice, namely if r is only admissible in M but not derivable. For a simple special case of that, the following lemma gives an inductive admissibility criterion.

Lemma 4. *Let r be of the form $\forall x(p(x) \Rightarrow p(f(x)))$ for some formula f in one variable. Then $M' \subseteq L$ if*

- *$M \subseteq L$ and*
- *for every rule of L with hypotheses $h_1, \ldots, h_n$ and conclusion c,*

$$\forall X\big(\bigwedge_{i=1}^{n}\big(p(h_i(X)) \wedge p(f(h_i(X)))\big) \Rightarrow p(f(c(X)))\big)$$

  is a first-order consequence of L.

This can be proven by a straightforward induction over the theorems of L (The second assumption is just what is needed to make the induction step.). The most important application is the case where $f(x) = box(x)$, i.e., where r is the necessitation rule.

Not all rules of modal logics can be easily encoded in the above way since some rules have complicated side conditions. The most important examples are (We refer to rules by abbreviations of their names, see Fig. 2 for the definitions.)

- substitution of strict equivalents (EQS): if f and $seqv(a, b)$ are theorems then f' which is as f but with a subformula a replaced with b is a theorem,
- propositional necessitation (Ga): if a is a theorem with no occurrences of box or dia, then $box(a)$ is a theorem.

It is reasonable to assume that any meta-language allowing for a natural encoding of these rules will be so strong that efficient automated solutions are unlikely. Instead we use

Lemma 5. *If M contains the rules SMP, AD, M1, M2, M4 and M5, then M enriched with EQS has the same theorems as M enriched with the following rules:*

- $\forall x, y, z(p(seqv(x, y)) \Rightarrow p(seqv(or(z, x), or(z, y))))$,
- $\forall x, y, z(p(seqv(x, y)) \Rightarrow p(seqv(or(x, z), or(y, z))))$,

- $\forall x, y(p(seqv(x,y)) \Rightarrow p(seqv(not(x), not(y))))$,
- $\forall x, y(p(seqv(x,y)) \Rightarrow p(seqv(box(x), box(y))))$,
- $\forall x, y((p(x) \wedge p(seqv(x,y))) \Rightarrow p(y))$.

This can be proven by using the assumptions made about M (without EQS) to show that strict equivalence is a congruence relation in M. That is equivalent to M being closed under *EQS* (see [Por80], [Tho68]). This lemma covers the most important case, in which EQS occurs, namely Lewis's family S1 to S5 ([Lew18]). In other cases, we are not able to express EQS.

A similar criterion for Ga is more complicated. While the necessary axiomatization of a further predicate *prop* that expresses the side condition is trivial, this predicate would not be preserved under uniform substitution and therefore, cannot be used. Instead a second sort, say s_2, with function symbols for propositional logic must be used along with with a predicate symbol t and an axiomatization such that $\forall X : s_2.\ t(f(X))$ expresses that f is a propositional tautology. Furthermore, a function symbol $i : s_2 \to s_1$ and axioms of the form

$$\forall x_1, x_2 : s_1, y_1, y_2 : s_2.\ \big((i(y_1) = x_1 \wedge i(y_2) = x_2) \Rightarrow i(and_2(y_1, y_2)) = and(x_1, x_2)\big)$$

are necessary.

If all previous definitions are appropriately extended to this more general setting, we have

Lemma 6. *If L has all propositional tautologies as theorems, then L enriched with Ga has the same theorems as L enriched with the rule $\forall y : s_2.(t(y) \Rightarrow p(box(i(y))))$.*

Currently, this is not supported by the implementation.

4 Negative Criteria

The simplest approach to disprove a subset relation $M \subseteq L$ is to use a first-order model finder to show that a theorem of M does not follow from L.

Lemma 7. *If f is a theorem of M, and if there is a first-order model for L enriched with $\exists X \neg p(f(X))$, then $M \not\subseteq L$.*

This approach is essentially the same as using matrix models for modal logics (see, e.g., [McK41]). It is not complete since only finite models can be checked, but this is often sufficient in practice.

Kripke models provide another strong criterion. It is clearly not complete but successful in many cases. The idea of this approach is to find a Kripke model $m' = (U, R, \alpha)$ such that the formulas satisfied by m' include the theorems of L but not f for some $f \in M$; here U is the set of worlds, R the accessibility relation, and α an assignment of truth values to the variables of f. Firstly, m' must satisfy all rules of L, i.e., an instance of the conclusion of a rule holds in all worlds whenever the appropriate instances of all hypotheses of the rule hold in all worlds. And secondly m' must satisfy $not(f)$ in one world.

Implementing that is less trivial than it sounds. If Kripke semantics is used to translate modal logic to first-order logic, the possibility to quantify over all formulas is lost, which is necessary to check whether a model satisfies a rule. And we are not aware of a model finder for (monadic) second-order logic. To circumvent this problem, we fix the number of worlds in U, say n, and proceed as follows.

We search for a first-order model m, from which m' can be generated. The signature for m contains constants $1, \ldots, n$ (intended semantics: one constant per world of U), the constant t (intended semantics: truth value of truth; we use any of the worlds as the truth value for falsity), and the binary predicate Acc (intended semantics: the accessibility relation R). m must satisfy that all worlds are different from each other and from t. Furthermore, the signature contains one constant x_i for every variable x occurring in f and for every $i = 1, \ldots, n$ (intended semantics: x_i gives the value of the assignment α to x in world i).

Let $\bar{\cdot}$ be the following translation from the language over $\mathcal{T}$ to this new signature:

- $\overline{\forall x\ F} = \forall x_1, \ldots, x_n\ \overline{F}$,
- the meta language connectives $\wedge$ and $\Rightarrow$ remain unchanged,
- $\overline{p(f(X))} = \bigwedge_{i=1}^{n} \overline{f(X)}(i)$ where $\overline{f(X)}(i)$ is given by the remaining cases,
- $\overline{and(f_1(X), f_2(X))}(i) = \overline{f_1(X)}(i) \wedge \overline{f_2(X)}(i)$ and accordingly for the other propositional connectives,
- $\overline{x}(i) = \ x_i \doteq t$ for a variable x,
- $\overline{box(f_1(X))}(i) = \bigwedge_{j=1}^{n} (Acc(i,j) \Rightarrow \overline{f_1(X)}(j))$,
- $\overline{dia(f_1(X))}(i) = \bigvee_{j=1}^{n} (Acc(i,j) \wedge \overline{f_1(X)}(j))$.

Here, the intended semantics of $\overline{\forall X\ p(f(X))}$ is that f holds in all worlds of m' and that of $\overline{f(X)}(i)$ is that f holds in the world i. Then the needed requirements for m are that it satisfies $\overline{r}$ for all rules r of L and $\overline{not(f(X))}(1)$.

These requirements can be sent to a first-order model finder. From the generated model m, m' is constructed by

- U: the universe of m minus the interpretation of t,
- R: the restriction of the interpretation of Acc to U,
- for a variable x of f and a world i of U, $\alpha(x)(i)$ is true if x_i is equal to t in m, and it is false otherwise.

Then we have

Lemma 8. *If f is a theorem of M and there is an n such that a model for L and $not(f)$ as described above can be found, then $M \not\subseteq L$.*

The follows directly from the above construction.

5 Implementation

The implementation of the presented criteria was done in SMLNJ ([SML]) and is available as [Rab]. It requires binary files for Vampire ([RV02]) and Paradox ([CS03]) to be present in the current directory because these are invoked to solve the generated subproblems.

The main function is invoked as `Main.compare(f1,f2)` where `f1` and `f2` are the names of files containing the modal logics in TPTP syntax ([SS98]). The content of the file `definitions.tptp` contains the propositional axiomatization and is added to each logic.

`compare` calls both `subsumes(f1,f2)` and `subsumes(f2,f1)`. Firstly, `subsumes(big,small)` calls `inclusion(big,small)` to prove `small` $\subseteq$ `big`. It tries all strategies given in the list `incl_strategies` trying to derive each rule of `small` as a consequence of `big`. Proven rules are added to `big`. If all rules are proven, a proof object is returned, otherwise the exception `fail` is raised.

In the latter case, the underived rules, say `l`, are passed to `noninclusion(big,l)`, which calls each disproving strategy given in the list `nonincl_strategies` with each element of `l`. If a counter-example is found a proof object is returned. `compare` collects these results and outputs the final result.

Optionally, whenever an external call is carried out the user is prompted for confirmation. It is possible to skip a call and choose whether the program should assume that the call has succeeded or failed.

Strategy	Criterion	Prover/Model finder	Remarks
`Direct`	Lem. 1, 3	Vampire	
`Inductive`	Lem. 4	Vampire	
`Sahlqvist`	Lem. 2	–	not implemented yet
`Containsk`	Lem. 2	Vampire	special case of Sahlqvist, implemented but not yet used in performance test
`Naivemodel`	Lem. 7	Paradox	implemented but not yet used in performance test
`Kripke`	Lem. 8	Paradox	

Figure 3: Strategies and provers

Strategies are functors that take a prover (or model finder) as an argument. Each strategy implements a sufficient criterion and tries to establish it by calling the prover. The provers are wrapper structures that invoke external first-order tools.

Strategies and provers are designed to be modular, which makes it easy to add further strategies and provers in the future. Currently only a minimal set of strategies is implemented, see Fig. 3. Testing showed that using the CASC competition ([PSS02]) winners Vampire and Paradox already gives quite satisfactory results.

$Log1$	Axiomatization	$Log2$	Axiomatization	R	R'	Pr. +	Pr. −
K	MP, N, K	M	MP, N, REM, T2_1, T2_2	$\subset$	$\not\supseteq$	K	
K	MP, N, K	S1_0	SMP, AD, EQS, M1-M5	$\neq$	$-$		inc.
S1_0	SMP, AD, EQS, M1-M5	S4	MP, N, K, T, 4	$\subset$	$\not\supseteq$	M5	
S1	SMP, AD, EQS, M1-M5, AS1.6	S4	MP, H1-H7	$\subset$	$\not\supseteq$	M5	
S3	SMP, AD, EQS, M1-M6, M8	S5	MP, N, K, T, B, 4	$\subset$	$\not\supseteq$	M5, M8	
S5	MP, N, K, T, 5	S5	SMP, AD, EQS, M1-M5, M10	$=$	$\subseteq$	M5	

Figure 4: Example instances and results

6 Experimental Results

The 100 $ challenge mentions eight logics with a total of about 25 axiomatizations that can be expressed in our encoding. This leads to about 600 combinations of logics. Fig. 4 shows the experimental results for a set of six examples. Running the program partially for other cases showed that this set is fairly representative for difficult instances. Here, the relationship between the logics (fifth column) is $Log1\ R\ Log2$ where $R \in \{\subset, \supset, =, \neq\}$. The program's answer R' is given in the sixth column: For partial success of the search, R' can also be in $\{\subseteq, \supseteq, \not\subseteq, \not\supseteq, -\}$. The symbol $\neq$ denotes incomparability, and $-$ denotes complete failure. Note that the strategies Containsk and Naivemodelwere not used for this test.

As can be seen, none of the cases could be solved completely. The seventh column gives the problems with positive criteria, which consisted of rules that could not be proven although they were provable. The eighth column gives the problems with negative criteria, which occurred only in one case where the used criterion was incomplete.

Comparing the run time shows that if a relation could be determined, this was done relatively fast: less than 30 minutes on a 3 GHz Intel Xeon machine. And this time was mainly spent waiting for the five-minute time limit of a failing Vampire call. Successful Vampire calls returned results within seconds, only sometimes minutes. Even more interesting, hardly any improvement could be observed by increasing Vampire's time limit: In the case of the third row, even 20 hours did not suffice to derive M5. Since M5 is a base axiom occurring in several natural axiomatizations, this becomes an interesting challenge problem for automated theorem proving. Calls to negative criteria did not contribute significantly to the run time.

This shows that the run time is essentially determined by the time limit for failing prover calls. Due to the nature of the problem, such calls occur frequently. This suggests that a more sophisticated switching between trying to prove inclusion or non-inclusion and varying the time limit for porver calls may lower the run time significantly.

The most interesting conclusion to be drawn about proving inclusion is that the complicated theoretical problems of rules that are only admissible but not derivable and of the incompleteness of the criterion in Lem. 3 are less relevant than expected: Both problems did not strike at all. All rules that could not be derived are axioms (K, M5 and M8). For these, however, the strategy `Direct` is complete. Apparently even a relatively small complexity of an axiom leads to de-facto unprovability.

As to proving non-inclusion, the incompleteness of the strategy `Kripke` hit twice: Both K and M1 to M5 hold in all Kripke models and their negation can never be satisfied. An obvious improvement of the current implementation is to apply rules that could not be derived to axioms in order to obtain more potential counter-examples. It is not clear if this will be successful in practice or if other approaches (like the recently implemented strategy `Naivemodel`) have to be considered.

It is not clear whether the strategy `Sahlqvist` obeys the spirit of the 100 $ challenge: Since it uses an external completeness result, it can be argued that the strategy uses hidden knowledge about the problems and is not a pure theorem prover. However, the performance of this strategy provides an enticing argument: Already the very special case implemented in the strategy `Containsk` (which does not even use the Sahlqvist results) can derive M5 in all four critical cases within seconds, and M8 can be derived by `Sahlqvist`. Thus, although this strategy can never help to prove K itself or non-normal axioms, it solves the problems in the last four rows of Fig. 4.

7 Conclusion and Future Work

We are bringing together different techniques to implement and integrate them into a framework directed at a specific challenge problem of theorem proving. Clearly, the current implementation is not much more than a proof of concept, but the current version already brings promising results. Implementing the algorithm of Lem. 2 fully or even partially will strengthen it significantly.

We also found that pure proof search without using semantical correspondence results seems to be practically incomplete for relevant problems. And the question how to derive non-normal axioms or K is open. However, we expect that a larger set of strategies (e.g., splitting as described in [Kra99]) will cover almost all interesting cases.

Less sophisticated possible future work includes

- utilizing different provers to exploit their respective strengths,
- tweaking, dynamically assigning or iteratively increasing the search depth, which is now fixed (5 minutes for Vampire, 4 worlds for Kripke models),
- redesigning the main functions to minimize the time spent with failing calls, in particular dynamically switching between trying to prove inclusion or non-inclusion, and keeping better track of positive and negative partial results,
- parallelizing external calls.

By publishing this work in progress, we hope to gain support and feedback for the further attack of this problem.

References

[CS03] K. Claessen and N. Sorensson. New techniques that improve MACE-style finite model finding. In *CADE-19 Workshop on Model Computation - Principles, Algorithms, Applications*, 2003.

[GO92] D. Gabbay and H. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 1992.

[GTG02] E. Giunchiglia, A. Tacchella, and F. Giunchiglia. SAT-based decision procedures for classical modal logics. *Journal of Automated Reasoning*, 28(2):143–171, 2002.

[Hal] J. Halleck. Logic systems. See `http://www.cc.utah.edu/~nahaj/logic/structures/systems/index.html`.

[HC96] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[HM92] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[HS00] U. Hustadt and R. A. Schmidt. MSPASS: Modal reasoning by translation and first-order resolution. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference (TABLEAUX 2000)*, pages 67–71, 2000.

[HV89] J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

[Kra99] M. Kracht. *Tools and Techniques in Modal Logic*. Elsevier, 1999.

[Kri63] S. A. Kripke. Semantical analysis of modal logic I. Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[Lem66] E. J. Lemmon. Algebraic Semantics for Modal Logics II. *The Journal of Symbolic Logic*, 31:191–218, 1966.

[Lew18] C. Lewis. *A Survey of Symbolic Logic*. University of California Press, 1918.

[McK41] J. C. McKinsey. A solution of the decision problem for the lewis systems s2 and s4 with an application to topology. *The Journal of Symbolic Logic*, 6:117–134, 1941.

[Por80] J. Porte. Congruences in Lemmon's S0.5. *Notre Dame Journal of Formal Logic*, 21(4):672–678, 1980.

[PSS02] F. Pelletier, G. Sutcliffe, and C. Suttner. The Development of CASC. *AI Communications*, 15(2-3):79–90, 2002.

[Rab] F. Rabe. Determining the subset relation between propositional modal logics. `http://kwarc.eecs.iu-bremen.de/frabe/Research/moloss`.

[RV02] A. Riazanov and A. Voronkov. The design and implementation of Vampire. *AI Communications*, 15:91–110, 2002.

[Sah75] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, pages 110–143. North-Holland, 1975.

[SML] Standard ML of New Jersey. See `http://www.smlnj.org`.

[SS98] G. Sutcliffe and C. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.

[Sut] G. Sutcliffe. The modal logic $100 challenge. See `http://www.cs.miami.edu/~tptp/HHDC/`.

[Tho68] I. Thomas. Replacement in Some Modal Systems. *The Journal of Symbolic Logic*, 33(4):569–570, 1968.

[Tho74] S. K. Thomason. An incompleteness theorem in modal logic. *Theoria*, 40:30–34, 1974.