Local Reductions for the Modal Cube

Cláudia Nalon[1], Ullrich Hustadt[2](✉), Fabio Papacchini[3], and Clare Dixon[4]

[1] Department of Computer Science, University of Brasília, Brasília, Brazil
nalon@unb.br
[2] Department of Computer Science, University of Liverpool, Liverpool, UK
U.Hustadt@liverpool.ac.uk
[3] School of Computing and Communications, Lancaster University in Leipzig, Leipzig, Germany
f.papacchini@lancaster.ac.uk
[4] Department of Computer Science, University of Manchester, Manchester, UK
clare.dixon@manchester.ac.uk

Abstract. The modal logic K is commonly used to represent and reason about necessity and possibility and its extensions with combinations of additional axioms are used to represent knowledge, belief, desires and intentions. Here we present local reductions of all propositional modal logics in the so-called modal cube, that is, extensions of K with arbitrary combinations of the axioms B, D, T, 4 and 5 to a normal form comprising a formula and the set of modal levels it occurs at. Using these reductions we can carry out reasoning for all these logics with the theorem prover K$_S$P. We define benchmarks for these logics and experiment with the reduction approach as compared to an existing resolution calculus with specialised inference rules for the various logics.

1 Introduction

Modal logics have been used to represent and reason about mental attitudes such as knowledge, belief, desire and intention, see for example [17,20,31]. These can be represented using extensions of the basic modal logic K with one or more of the axioms B (symmetry), D (seriality), T (reflexivity), 4 (transitivity) and 5 (Euclideaness). The logic K and these extensions form the so-called *modal cube*, see Fig. 1. In the diagram, a line from a logic L_1 to a logic L_2 to its right and/or above means that all theorems of L_1 are also theorems of L_2, but not vice versa. As indicated in Fig. 1, some of the logics have the same theorems, e.g., KB5 and KB4. Also, all logics not explicitly listed have the same theorems as KT5 aka S5. In total there are 15 distinct logics.

While these modal logics are well-studied and a multitude of calculi and translations to other logics exist, see, e.g., [1,3–6,9,13,14,16,18,22,41], fully

C. Dixon was partially supported by the EPSRC funded RAI Hubs FAIR-SPACE (EP/R026092/1) and RAIN (EP/R026084/1), and the EPSRC funded programme Grant S4 (EP/N007565/1).

J. Blanchette et al. (Eds.): IJCAR 2022, LNAI 13385, pp. 486–505, 2022.
https://doi.org/10.1007/978-3-031-10769-6_29

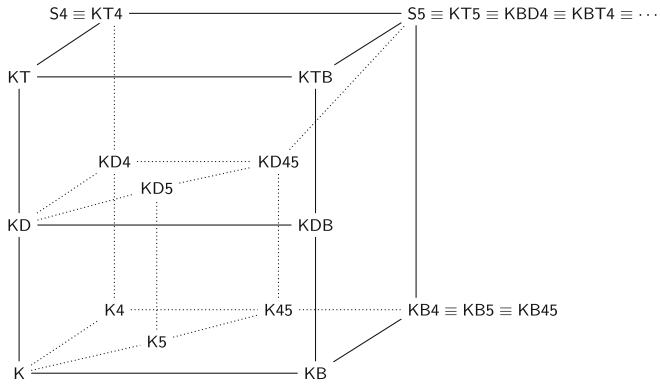

Fig. 1. Modal Cube: Relationships between modal logics

automatic support by provers is still lacking. Early implementations covering the full modal cube, such as Catach's TABLEAUX system [7], are no longer available. LoTREC 2.0 [10] supports a wide range of logics but is not intended as an automatic theorem prover. MOIN [11] supports all the logics but the focus is on producing human-readable proofs and countermodels for small formulae. Other provers that go beyond just K, like MleanCoP [28] and CEGARBox [15] only support a small subset of the 15 logics. There are also a range of translations from modal logics to first-order and higher-order logics [13,18,19,27,33]. Regarding implementations of those, SPASS [33,43] is limited to a subset of the 15 logics, while LEO-III [13,36] supports all the logics in the modal cube, but can only solve very few of the available benchmark formulae.

K$_S$P [23] is a modal logic theorem prover that implements both the modal-layered resolution (MLR) calculus [25] for the modal logic K and the global resolution (GMR) calculus [24] for all the 15 logics considered here. It also supports several refinements of resolution and a range of simplification rules. In this paper, we give reductions of all logics of the modal cube into a normal form for the basic modal logic K. We then compare the performance of the combination of these reductions with the modal-layered resolution calculus to that of the global resolution calculus on a new benchmark collection for the modal cube.

In [29] we have presented new reductions[1] of the propositional modal logics KB, KD, KT, K4, and K5 to Separated Normal Form with Sets of Modal Levels SNF_{sml}. SNF_{sml} is a generalisation of the Separated Normal Form with Modal Level, SNF_{ml}. In the latter, labelled modal clauses are used where a natural number label refers to a particular level within a tree Kripke structure at which a modal clause holds. In the former, a finite or infinite set of natural numbers labels each modal clause with the intended meaning that such a modal clause is true at every level of a tree Kripke structure contained in that set. As our prover K$_S$P and the modal-layered resolution calculus it implements currently only support sets of modal clauses in SNF_{ml}, we then use a further reduction from SNF_{sml}

[1] A *reduction* here is a satisfiability preserving mapping between logics.

to SNF_{ml} to obtain an automatic theorem prover for these modal logics. Where all modal clauses are labelled with finite sets, this reduction is straightforward. This is the case for KB, KD and KT. For $\mathsf{K4}$ and $\mathsf{K5}$, characterised by the axioms $\Box\varphi \to \Box\Box\varphi$ and $\Diamond\varphi \to \Box\Diamond\varphi$, modal clauses are in general labelled with infinite sets. However, using a result by Massacci [21] for $\mathsf{K4}$ and an analogous result for $\mathsf{K5}$ by ourselves, we are able to bound the maximal level occurring in those labelling sets which in turn makes a reduction to SNF_{ml} possible.

Also in [29], we have shown experimentally that these reductions allow us to reason effectively in these logics, compared to the global modal resolution calculus [24] and to the relational and semi-functional translation built into the first-order theorem prover SPASS 3.9 [33,38,42]. The reason that the comparison only included a rather limited selection of provers is that these are the only ones with built-in support for all six logics our reductions covered.

Unfortunately, we cannot simply combine our reductions for single axioms to obtain satisfiability preserving reductions for their combinations. There are two main reasons for this. First, our calculus does not use an explicit representation of the accessibility relationship within a Kripke structure, which would make it possible to reflect modal axioms via corresponding properties of that accessibility relationship. Instead, we add labelled modal clauses based on instances of the modal axioms for $\Box$-formulae occurring in the modal formula we want to check for satisfiability. However, if we deal with multiple modal axioms, then these axioms might interact making it necessary to add instances that are not necessary for each individual axiom. For instance, consider, the converse of axiom B, $\Diamond\Box\varphi \to \varphi$, and axiom $\mathsf{4}$, $\Box\varphi \to \Box\Box\varphi$. Together they imply $\Diamond\Box\varphi \to \Box\varphi$. Instances of this derived axiom are necessary for completeness of a reduction from $\mathsf{KB4}$ to K, but are unsound for KB and $\mathsf{K4}$ separately.

Second, our reductions attempt to keep the labelling sets minimal in size in order to decrease the number of inferences that can be performed. Again, taking axioms B and $\mathsf{4}$ as examples, in KB, a $\Box$-formula $\Box\psi$ true at level ml in a tree-like Kripke structure M forces ψ to be true at level $ml - 1$, while in $\mathsf{K4}$, $\Box\psi$ true at level ml in M forces ψ to be true all levels ml' with $ml' > ml$. This is reflected in the labelling sets we use for these two logics. However, for $\mathsf{KB4}$, $\Box\psi$ true at level ml forces ψ to be true at every level in a tree-like Kripke structure M (unless M consists only of a single world).

Since we intend to maintain these two properties of our reductions, we have to consider each modal logic individually. As we will see, for some logics a reduction can be obtained as the union of the existing reductions while for others we need a logic-specific reduction to accommodate the interaction of axioms.

The structure of the paper is as follows. In Sect. 2 we recall common concepts of propositional modal logic and the definition of our normal form SNF_{ml}. Section 3 introduces our reduction for extensions of the basic modal logic K with combinations of the axioms B, D, T, $\mathsf{4}$, and $\mathsf{5}$. Section 4 presents a transformation from SNF_{sml} to SNF_{ml} which allows us to use the modal resolution prover KSP to reason in all the modal logics. In Sect. 5 we compare the performance of a combination of our reductions and the modal-layered resolution calculus implemented in the prover KSP with resolution calculi specifically designed for the logics under consideration as well as the prover LEO-III.

2 Preliminaries

The language of modal logic is an extension of the language of propositional logic with a unary modal operator $\Box$ and its dual $\Diamond$. More precisely, given a denumerable set of *propositional symbols*, $P = \{p, p_0, q, q_0, t, t_0, \ldots\}$ as well as propositional *constants* **true** and **false**, *modal formulae* are inductively defined as follows: constants and propositional symbols are modal formulae. If φ and ψ are modal formulae, then so are $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$, $\Box\varphi$, and $\Diamond\varphi$. We also assume that $\wedge$ and $\vee$ are associative and commutative operators and consider, e.g., $(p \vee (q \vee r))$ and $(r \vee (q \vee p))$ to be identical formulae. We often omit parentheses if this does not cause confusion. By $\mathsf{var}(\varphi)$ we denote the set of all propositional symbols occurring in φ. This function straightforwardly extends to finite sets of modal formulae. A *modal axiom (schema)* is a modal formula ψ representing the set of all instances of ψ.

A *literal* is either a propositional symbol or its negation; the set of literals is denoted by L_P. By $\neg l$ we denote the *complement* of the literal $l \in L_P$, that is, if l is the propositional symbol p then $\neg l$ denotes $\neg p$, and if l is the literal $\neg p$ then $\neg l$ denotes p. By $|l|$ for $l \in L_P$ we denote p if $l = p$ or $l = \neg p$. A *modal literal* is either $\Box l$ or $\Diamond l$, where $l \in L_P$.

A *(normal) modal logic* is a set of modal formulae which includes all propositional tautologies, the axiom schema $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$, called the *axiom* K, it is closed under modus ponens (if $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$) and the rule of necessitation (if $\vdash \varphi$ then $\vdash \Box\varphi$).

K is the weakest modal logic, that is, the logic given by the smallest set of modal formulae constituting a normal modal logic. By $\mathsf{K}\Sigma$ we denote an *extension* of K by a set Σ of axioms.

The standard semantics of modal logics is the *Kripke semantics* or *possible world semantics*. A *Kripke frame* F is an ordered pair $\langle W, R\rangle$ where W is a non-empty set of *worlds* and R is a binary (accessibility) relation over W. A *Kripke structure* M over P is an ordered pair $\langle F, V\rangle$ where F is a Kripke frame and the *valuation* V is a function mapping each propositional symbol in P to a subset $V(p)$ of W. A *rooted Kripke structure* is an ordered pair $\langle M, w_0\rangle$ with $w_0 \in W$. To simplify notation, in the following we write $\langle W, R, V\rangle$ and $\langle W, R, V, w_0\rangle$ instead of $\langle\langle W, R\rangle, V\rangle$ and $\langle\langle\langle W, R\rangle, V\rangle, w_0\rangle$, respectively.

Satisfaction (or truth) of a formula at a world w of a Kripke structure $M = \langle W, R, V\rangle$ is inductively defined by:

$$
\begin{array}{ll}
\langle M, w\rangle \models \mathbf{true}; & \langle M, w\rangle \not\models \mathbf{false}; \\
\langle M, w\rangle \models p & \text{iff } w \in V(p), \text{ where } p \in P; \\
\langle M, w\rangle \models \neg\varphi & \text{iff } \langle M, w\rangle \not\models \varphi; \\
\langle M, w\rangle \models (\varphi \wedge \psi) & \text{iff } \langle M, w\rangle \models \varphi \text{ and } \langle M, w\rangle \models \psi; \\
\langle M, w\rangle \models (\varphi \vee \psi) & \text{iff } \langle M, w\rangle \models \varphi \text{ or } \langle M, w\rangle \models \psi; \\
\langle M, w\rangle \models (\varphi \to \psi) & \text{iff } \langle M, w\rangle \models \neg\varphi \text{ or } \langle M, w\rangle \models \psi; \\
\langle M, w\rangle \models \Box\varphi & \text{iff for every } v, w\,R\,v \text{ implies } \langle M, v\rangle \models \varphi; \\
\langle M, w\rangle \models \Diamond\varphi & \text{iff there is } v, w\,R\,v \text{ and } \langle M, v\rangle \models \varphi.
\end{array}
$$

Table 1. Modal axioms and relational frame properties

Name	Axiom	Frame Property	
D	$\Box\varphi \to \Diamond\varphi$	Serial	$\forall v \exists w . v\, R\, w$
T	$\Box\varphi \to \varphi$	Reflexive	$\forall w . w\, R\, w$
B	$\varphi \to \Box\Diamond\varphi$	Symmetric	$\forall v w . v\, R\, w \to w\, R\, v$
4	$\Box\varphi \to \Box\Box\varphi$	Transitive	$\forall u v w . (u\, R\, v \wedge v\, R\, w) \to u\, R\, w$
5	$\Diamond\varphi \to \Box\Diamond\varphi$	Euclidean	$\forall u v w . (u\, R\, v \wedge u\, R\, w) \to v\, R\, w$

Table 2. Rewriting Rules for Simplification

$\varphi \wedge \varphi \Rightarrow \varphi$	$\varphi \wedge \neg\varphi \Rightarrow \mathbf{false}$	$\Box\mathbf{true} \Rightarrow \mathbf{true}$	$\neg\mathbf{true} \Rightarrow \mathbf{false}$	$\neg\neg\varphi \Rightarrow \varphi$
$\varphi \vee \varphi \Rightarrow \varphi$	$\varphi \vee \neg\varphi \Rightarrow \mathbf{true}$	$\Diamond\mathbf{false} \Rightarrow \mathbf{false}$	$\neg\mathbf{false} \Rightarrow \mathbf{true}$	
$\varphi \wedge \mathbf{true} \Rightarrow \varphi$	$\varphi \wedge \mathbf{false} \Rightarrow \mathbf{false}$	$\varphi \vee \mathbf{false} \Rightarrow \varphi$	$\varphi \vee \mathbf{true} \Rightarrow \mathbf{true}$	

If $\langle M, w\rangle \models \varphi$ holds then M is a *model* of φ, φ is *true at* w *in* M and M *satisfies* φ. A modal formula φ is *satisfiable* iff there exists a Kripke structure M and a world w in M such that $\langle M, w\rangle \models \varphi$.

We are interested in extensions of K with the modal axioms shown in Table 1 and their combinations. Each of these axioms defines a class of Kripke frames where the accessibility relation R satisfies the first-order property stated in the table. Combinations of axioms then define a class of Kripke frames where the accessibility relation satisfies the combination of their corresponding properties.

Given a normal modal logic L with corresponding class of frames $\mathfrak{F}$, we say a modal formula φ is *L-satisfiable* iff there exists a frame $F \in \mathfrak{F}$, a valuation V and a world $w \in F$ such that $\langle F, V, w\rangle \models \varphi$. It is *$L$-valid* or *valid in* L iff for every frame $F \in \mathfrak{F}$, every valuation V and every world $w \in F$, $\langle F, V, w\rangle \models \varphi$. A normal modal logic L_2 is *an extension* of a normal modal logic L_1 iff all L_1-valid formulae are also L_2-valid.

A rooted Kripke structure $M = \langle W, R, V, w_0\rangle$ is a *rooted tree Kripke structure* iff R is a tree, that is, a directed acyclic connected graph where each node has at most one predecessor, with *root* w_0. It is a *rooted tree Kripke model* of a modal formula φ iff $\langle W, R, V, w_0\rangle \models \varphi$. In a rooted tree Kripke structure with root w_0 for every world $w_k \in W$ there is exactly one path connecting w_0 and w_k, the length of that path is the *modal level of* w_k *(in M), denoted by* $\mathsf{ml}_M(w_k)$.

It is well-known [17] that a modal formula φ is K-satisfiable iff there is a finite rooted tree Kripke structure $M = \langle F, V, w_0\rangle$ such that $\langle M, w_0\rangle \models \varphi$.

For the reductions presented in the next section we assume that any modal formula φ has been simplified by exhaustively applying the rewrite rules in Table 2, and it is in Negation Normal Form (NNF). That is, a formula where only propositional symbols are allowed in the scope of negations. We say that such a formula is in *simplified NNF*.

The reductions produce formulae in a clausal normal form, called *Separated Normal Form with Sets of Modal Levels* SNF_{sml}, introduced in [29]. The language of SNF_{sml} extends that of the basic modal logic K with sets of modal levels as labels. Clauses in SNF_{sml} have one of the following forms:

$S : \bigvee_{i=1}^{n} l_i$	$S : l' \to \Box l$	$S : l' \to \Diamond l$
(literal clause)	(positive modal clause)	(negative modal clause)

where $S \subseteq \mathbb{N}$ and l, l', l_i are propositional literals with $1 \leq i \leq n$, $n \in \mathbb{N}$. We write $\star : \varphi$ instead of $\mathbb{N} : \varphi$ and such clauses are called *global clauses*. Positive and negative modal clauses are together known as *modal clauses*.

Given a rooted tree Kripke structure M and a set S of natural numbers, by $M[S]$ we denote the set of worlds that are at a modal level in S, that is, $M[S] = \{w \in W \mid \mathsf{ml}_M(w) \in S\}$. Then

$$M \models S : \varphi \text{ iff } \langle M, w\rangle \models \varphi \text{ for every world } w \in M[S].$$

The motivation for using a set S to label clauses is that in our reductions the formula φ may hold at several levels, possibly an infinite number of levels. It therefore makes sense to label such formulae not with just a single level, but a set of levels. The Separated Normal Form with Modal Level, SNF_{ml}, can be seen as the special case of SNF_{sml} where all labelling sets are singletons.

Note that if $S = \emptyset$, then $M \models S : \varphi$ trivially holds. Also, a Kripke structure M can satisfy S : **false** if there is no world w with $\mathsf{ml}_M(w) \in S$. On the other hand, S : **false** with $0 \in S$ is unsatisfiable as a rooted tree Kripke structure always has a world with modal level 0.

If $M \models S : \varphi$, then we say that $S : \varphi$ *holds in* M or *is true in* M. For a set Φ of labelled formulae, $M \models \Phi$ iff $M \models S : \varphi$ for every $S : \varphi$ in Φ, and we say Φ is K-*satisfiable*.

We introduce some notation that will be used in the following. Let $S^+ = \{l+1 \in \mathbb{N} \mid l \in S\}$, $S^- = \{l-1 \in \mathbb{N} \mid l \in S\}$, and $S^{\geq} = \{n \mid n \geq \min(S)\}$, where $\min(S)$ is the least element in S. Note that the restriction of the elements being in $\mathbb{N}$ implies that S^- cannot contain negative numbers.

3 Extensions of K

In this section we define reductions from all the logics in the modal cube to SNF_{sml}. We assume that the set P of propositional symbols is partitioned into two infinite sets Q and T such that Q contains the propositional symbols of the modal formula φ under consideration, and T surrogate symbols t_ψ for every subformula ψ of φ and supplementary propositional symbols. In particular, for every modal formula ψ we have $\mathsf{var}(\psi) \subset Q$ and there exists a propositional symbol $t_\psi \in T$ uniquely associated with ψ. These surrogate symbols serve the same purpose as Tseitin variables [40] and Skolem predicates [30,39] in the transformation of propositional and first-order formulae, respectively, to clausal form via structural transformation.

It turns out that given a reduction $\rho_{\mathsf{K}\Sigma}$ for $\mathsf{K}\Sigma$ with $\{\mathsf{D}, \mathsf{T}\} \cap \Sigma = \emptyset$, there is a uniform and straightforward way we can obtain a reduction for $\mathsf{KD}\Sigma$ and $\mathsf{KT}\Sigma$ from $\rho_{\mathsf{K}\Sigma}$. Also, the valid formulae of $\mathsf{KDT}\Sigma$ are the same as those of

Table 3. Categorisation of modal logics in the modal cube

'Base logics'	K	KB	K4	K5	KB4	K45
Extensions with D	KD	KDB	KD4	KD5		KD45
Extensions with T	KT	KTB	KT4	KT5		

$\mathsf{KT}\Sigma$, so we do not need to consider the case of adding both axioms to $\mathsf{K}\Sigma$. Similarly, the logics KT45, KDB4, KTB4 and KT5 all have the same set of valid formulae. Therefore, as shown in Table 3, we can divide the 15 modal logics into three categories: Six 'base logics', five modal logics obtained by extending a 'base logic' with D, and a further four modal logics obtained by extending a 'base logic' with T. For four of the six 'base logics' (namely, K, KB, K4, and K5) we have already devised reductions in [29], so only two (i.e., KB4 and K45) remain.

Given a modal formula φ in simplified NNF and $L = \mathsf{K}\Sigma$ with $\Sigma \subseteq \{\mathsf{B},\mathsf{D},\mathsf{T},\mathsf{4},\mathsf{5}\}$, we can obtain a set Φ_L of clauses in SNF_{sml} such that φ is L-satisfiable iff Φ_L is K-satisfiable with $\Phi_L = \rho_L^{sml}(\varphi) = \{\{0\} : t_\varphi\} \cup \rho_L(\{0\} : t_\varphi \to \varphi)$, where ρ_L is defined as follows:

$$
\begin{aligned}
\rho_L(S : t \to \mathbf{true}) &= \emptyset\\
\rho_L(S : t \to \mathbf{false}) &= \{S : \neg t\}\\
\rho_L(S : t \to (\psi_1 \wedge \psi_2)) &= \{S : \neg t \vee \eta(\psi_1), S : \neg t \vee \eta(\psi_2)\} \cup \delta_L(S,\psi_1) \cup \delta_L(S,\psi_2)\\
\rho_L(S : t \to \psi) &= \{S : \neg t \vee \psi\}\\
&\quad\ \text{if } \psi \text{ is a disjunction of literals}\\
\rho_L(S : t \to (\psi_1 \vee \psi_2)) &= \{S : \neg t \vee \eta(\psi_1) \vee \eta(\psi_2)\} \cup \delta_L(S,\psi_1) \cup \delta_L(S,\psi_2)\\
&\quad\ \text{if } \psi_1 \vee \psi_2 \text{ is not a disjunction of literals}\\
\rho_L(S : t \to \Diamond\psi) &= \{S : t \to \Diamond\eta(\psi)\} \cup \delta_L(S^+,\psi)\\
\rho_L(S : t \to \Box\psi) &= P_L(S : t \to \Box\psi) \cup \Delta_L(S : t \to \Box\psi)
\end{aligned}
$$

η and δ_L are defined as follows:

$$
\eta(\psi) = \begin{cases} \psi, & \text{if } \psi \text{ is a literal}\\ t_\psi, & \text{otherwise}\end{cases}
\qquad
\delta_L(S,\psi) = \begin{cases} \emptyset, & \text{if } \psi \text{ is a literal}\\ \rho_L(S : t_\psi \to \psi), & \text{otherwise}\end{cases}
$$

and functions P_L and Δ_L, are defined as shown in Table 4.

We can see in Table 4 that the reduction for KB4 has an additional SNF_{sml} clause $\star : t_{\Box\psi} \vee t_{\Box\neg t_{\Box\psi}}$ that occurs neither in the reduction for KB nor in that for K4. It can be seen as an encoding of the derived axiom $\Diamond\Box\psi \to \Box\psi$ that follows from the contrapositive $\Diamond\Box\psi \to \psi$ of B and 4 $\Box\psi' \to \Box\Box\psi'$.

For K45 we see that all the SNF_{sml} clauses in the reduction for K5 carry over. These clauses are already sufficient to ensure that, semantically, if $t_{\Box\psi}$ is true at any world at a level other than 0, then $t_{\Box\psi}$ is true at every world. Consequently, to accommodate axiom 4, it suffices to add the SNF_{sml} clause $\{0\} : t_{\Box\psi} \to \Box t_{\Box\psi}$ to ensure that this also holds for the root world at level 0.

L	$P_L(S : t_{\Box\psi} \to \Box\psi)$	$\Delta_L(S : t_{\Box\psi} \to \Box\psi)$
K	$S : t_{\Box\psi} \to \Box\eta(\psi)$	$\delta_L(S^+, \psi)$
KB	$S : t_{\Box\psi} \to \Box\eta(\psi)$, $S^- : \eta(\psi) \vee t_{\Box\neg t_{\Box\psi}}$, $S^- : t_{\Box\neg t_{\Box\psi}} \to \Box\neg t_{\Box\psi}$	$\delta_L(S^- \cup S^+, \psi)$
K4	$S^{\geq} : t_{\Box\psi} \to \Box\eta(\psi)$, $S^{\geq} : t_{\Box\psi} \to \Box t_{\Box\psi}$	$\delta_L((S^+)^{\geq}, \psi)$
K5	$\star : t_{\Box\psi} \to \Box\eta(\psi)$, $\star : \neg t_{\Diamond t_{\Box\psi}} \vee t_{\Box\psi}$, $\star : t_{\Diamond t_{\Box\psi}} \to \Diamond t_{\Box\psi}$, $\star : \neg t_{\Diamond t_{\Box\psi}} \to \Box\neg t_{\Box\psi}$, $\star : t_{\Diamond t_{\Box\psi}} \to \Box t_{\Diamond t_{\Box\psi}}$	$\delta_L(\star, \psi)$
KB4	$\star : t_{\Box\psi} \to \Box\eta(\psi)$, $\star : \eta(\psi) \vee t_{\Box\neg t_{\Box\psi}}$, $\star : t_{\Box\psi} \vee t_{\Box\neg t_{\Box\psi}}$, $\star : t_{\Box\neg t_{\Box\psi}} \to \Box\neg t_{\Box\psi}$, $\star : t_{\Box\psi} \to \Box t_{\Box\psi}$	$\delta_L(\star, \psi)$
K45	$\star : t_{\Box\psi} \to \Box\eta(\psi)$, $\{0\} : t_{\Box\psi} \to \Box t_{\Box\psi}$ iff $0 \in S$, $\star : \neg t_{\Diamond t_{\Box\psi}} \vee t_{\Box\psi}$, $\star : t_{\Diamond t_{\Box\psi}} \to \Diamond t_{\Box\psi}$, $\star : \neg t_{\Diamond t_{\Box\psi}} \to \Box\neg t_{\Box\psi}$, $\star : t_{\Diamond t_{\Box\psi}} \to \Box t_{\Diamond t_{\Box\psi}}$	$\delta_L(\star, \psi)$
KDΣ	$\{lb^P_{\mathsf{K}\Sigma}(S) : t_{\Box\psi} \to \Diamond\eta(\psi)\} \cup P_{\mathsf{K}\Sigma}(S : t_{\Box\psi} \to \Box\psi)$	$\delta_L(lb^\delta_{\mathsf{K}\Sigma}(S), \psi)$
KTΣ	$\{lb^P_{\mathsf{K}\Sigma}(S) : \neg t_{\Box\psi} \vee \eta(\psi)\} \cup P_{\mathsf{K}\Sigma}(S : t_{\Box\psi} \to \Box\psi)$	$\delta_L(lb^\delta_{\mathsf{K}\Sigma}(S) \cup S, \psi)$

where $lb^P_{\mathsf{K}\Sigma}$ and $lb^\delta_{\mathsf{K}\Sigma}$ are defined as follows

Table 4. Reduction of $\Box$-formulae, $\Sigma \subseteq \{\mathsf{B}, \mathsf{4}, \mathsf{5}\}$.

L	K	KB	K4	K5	KB4	K45
$lb^P_L(S)$	S	S	$S^{\geq}$	$\star$	$\star$	$\star$
$lb^\delta_L(S)$	S^+	$S^- \cup S^+$	$(S^+)^{\geq}$	$\star$	$\star$	$\star$

For reductions of KDΣ and KTΣ we have favoured the reuse of reductions for KΣ, KD and KT over optimisation for specific logics. For example, take KBD. Given that in a symmetric model, every world w except the root world w_0 has an R-successor, the axiom D only 'enforces' that w_0 also has an R-successor. So, instead of adding a clause $S : t_{\Box\psi} \to \Diamond\psi$ for every clause $S : t_{\Box\psi} \to \Box\eta(\psi)$ we could just add $\{0\} : t_{\Box\psi} \to \Diamond\psi$ iff $0 \in S$. Similarly, in KT5, because of 5, for all worlds w except w_0 we already have $w\,R\,w$. So, we could again $\{0\} : \neg t_{\Box\psi} \vee \eta(\psi)$ for every clause $S : t_{\Box\psi} \to \Box\eta(\psi)$ iff $0 \in S$.

For the KB4-unsatisfiable formula $\psi_1 = (\neg p \wedge \Diamond\Diamond\Box p)$, if we were to independently apply the reductions for KB and K4, that is, we compute $\{\{0\} : t_{\psi_1}\} \cup \rho_{\mathsf{KB}}(\{0\} : t_{\psi_1} \to \psi_1) \cup \rho_{\mathsf{K4}}(\{0\} : t_{\psi_1} \to \psi_1)$, then the result is the following set of clauses Φ_1:

(1) $\{0\} : t_{\psi_1}$
(2) $\{0\} : \neg t_{\psi_1} \vee \neg p$
(3) $\{0\} : \neg t_{\psi_1} \vee t_{\Diamond\Diamond\Box p}$
(4) $\{0\} : t_{\Diamond\Diamond\Box p} \to \Diamond t_{\Diamond\Box p}$
(5) $\{1\} : t_{\Diamond\Box p} \to \Diamond t_{\Box p}$
(6) $\{2\}^{\geq} : t_{\Box p} \to \Box p$
(7) $\{2\}^{\geq} : t_{\Box p} \to \Box t_{\Box p}$
(8) $\{1\} : p \vee t_{\Box\neg t_{\Box p}}$
(9) $\{1\} : t_{\Box\neg t_{\Box p}} \to \Box\neg t_{\Box p}$

Clauses (1) to (5) stem from the transformation of ψ_1 to SNF_{sml} for K, Clauses (6) and (7) stem from the reduction for 4 and Clauses (8) and (9) stem from the reduction for B. This set of SNF_{sml} clauses is K-satisfiable. The clauses imply $\{1\} : p$, but neither $\{1\} : \Box p$ nor $\{0\} : p$ which we need to obtain a contradiction. Part of the reason is that we would need to apply the reduction for 4 and B recursively to newly introduced surrogates for $\Box$-formulae which in turn leads to the introduction of further surrogates and problems with the termination of the reduction.

In contrast, the clause set Φ_2 obtained by our reduction for KB4 is:

(10) $\{0\} : t_{\psi_1}$
(11) $\{0\} : \neg t_{\psi_1} \vee \neg p$
(12) $\{0\} : \neg t_{\psi_1} \vee t_{\Diamond\Diamond\Box p}$
(13) $\{0\} : t_{\Diamond\Diamond\Box p} \rightarrow \Diamond t_{\Diamond\Box p}$
(14) $\{1\} : t_{\Diamond\Box p} \rightarrow \Diamond t_{\Box p}$
(15) $\star : t_{\Box p} \rightarrow \Box p$
(16) $\star : t_{\Box p} \rightarrow \Box t_{\Box p}$
(17) $\star : p \vee t_{\Box\neg t_{\Box p}}$
(18) $\star : t_{\Box\neg t_{\Box p}} \rightarrow \Box\neg t_{\Box p}$
(19) $\star : t_{\Box p} \vee t_{\Box\neg t_{\Box p}}$
(20) $\star : t_{\Box\neg t_{\Box p}} \rightarrow \Box t_{\Box\neg t_{\Box p}}$

Note Clauses (19) and (20) in Φ_2 for which there are no corresponding clauses in Φ_1. Also, the set of labels of Clauses (15) to (18) are strict supersets of those of the corresponding Clauses (6) to (9). Φ_2 implies both $\{1\} : \Box p$ and $\{0\} : p$. The latter, together with Clauses (10) and (11), means Φ_2 is K-unsatisfiable.

Theorem 1. *Let φ be a modal formula in simplified NNF, $\Sigma \subseteq \{\mathsf{B}, \mathsf{D}, \mathsf{T}, \mathsf{4}, \mathsf{5}\}$, and $\Phi_{\mathsf{K}\Sigma} = \rho^{sml}_{\mathsf{K}\Sigma}(\varphi)$. Then φ is $\mathsf{K}\Sigma$-satisfiable iff $\Phi_{\mathsf{K}\Sigma}$ is K-satisfiable.*

Proof (Sketch). For $|\Sigma| \leq 1$ this follows from Theorem 5 in [29].

For K45, KB4, $\mathsf{KD}\Sigma'$, and $\mathsf{KT}\Sigma'$ with $\Sigma' \subseteq \{\mathsf{B}, \mathsf{4}, \mathsf{5}\}$ we proceed in analogy to the proofs of Theorems 3 and 4 in [29]. Let L be one of these logics.

To show that if φ is L-satisfiable then Φ_L is K-satisfiable, we show that given a rooted L-model M of φ a small variation of the unravelling of M is a rooted tree K-model $\vec{M}_L$ of Φ_L. The main step is to define the valuation of the additional propositional symbols t_ψ so that we can prove that all clauses in Φ_L hold in $\vec{M}_L$. To show that if Φ_L is K-satisfiable then φ is L-satisfiable, we take a rooted tree K-model $M = \langle W, R, V, w_0 \rangle$ of Φ_L and construct a Kripke structure $M_L = \langle W, R^L, V, w_0 \rangle$. The relation R^L is the closure of R under the relational properties associated with the axioms of L. The proof that M_L is a model of φ relies on the fact that the clauses in Φ_L ensure that for subformulae $\Box\psi$ of φ, ψ will be true at all worlds reachable via R^L from a world where $\Box\psi$ is true. $\Box$

4 From SNF_{sml} to SNF_{ml}

As K$_\mathrm{S}$P does not support SNF_{sml}, in our evaluation of the effectiveness of the reductions defined in Sect. 3, we have used a transformation from SNF_{sml} to SNF_{ml}. An alternative approach would be to reflect the use of SNF_{sml} in the calculus and re-implement the prover. Whilst we believe that redesigning the calculus presents few problems, re-implementing K$_\mathrm{S}$P needs more thought in particular how to represent infinite sets. The route we adopt here allows us to experiment with the approach in general without having to change the prover. For extensions of K with one or more of the axioms B, D, T such a transformation

Table 5. Bounds on the length of prefixes in SST tableaux

Logic L	Bound db^{φ}_{L}
K,KD,KT, KB,KDB,KTB	$1 + d^{\varphi}_{m}$
K4,S4	$2 + d^{\varphi}_{\Diamond} + n^{\varphi}_{\Diamond} \times n^{\varphi}_{\Box}$
KD4	$2 + d^{\varphi}_{\Diamond} + (\max(1, n^{\varphi}_{\Diamond}) \times n^{\varphi}_{\Box})$
KB4,KTB4, K5,S5,K45	$2 + d^{\varphi}_{\Diamond} + n^{\varphi}_{\Diamond}$
KD5	$2 + d^{\varphi}_{\Diamond} + \max(1, n^{\varphi}_{\Diamond})$

is straightforward as the sets of modal levels occurring in the normal form of modal formulae are all finite. Thus, instead of a single SNF_{sml} clause $S : \neg t_{\psi} \vee \eta_f(\psi)$ we can use the finite set of SNF_{ml} clauses $\{ml : \neg t_{\psi} \vee \eta_f(\psi) \mid ml \in S\}$.

For extensions of K with at least one of the axioms 4 and 5, potentially together with other axioms, the sets of modal levels labelling clauses are in general infinite. For each logic L it is, however, possible to define a computable function that maps the modal formula φ under consideration onto a bound db^{φ}_{L} such that, restricting the modal levels in the normal form of φ by db^{φ}_{L}, preserves satisfiability equivalence.

To establish the bound and prove satisfiability equivalence, we need to introduce the basic notions of Single Step Tableaux (SST) calculi for a modal logic L [14,21], which uses sequences of natural numbers to prefix modal formulae in a tableau. The SST calculus consists of a set of rules, with the (π) rule being the only rule increasing prefixes' lengths (i.e., $\sigma : \Diamond\varphi/\sigma.n : \varphi$ with $\sigma.n$ new on the branch). For a logic L, an *L-tableau $\mathcal{T}$* in the SST calculus for a modal formula φ is a (binary) tree where the root of $\mathcal{T}$. is labelled with $1 : \varphi$, and every other node is labelled with a prefixed formula $\sigma : \psi$ obtained by application of a rule of the calculus. A *branch* $\mathcal{B}$ is a path from the root to a leaf. A *branch* $\mathcal{B}$ is *closed* if it contains either **false** or a propositional contradiction at the same prefix. A *tableau* "$\mathcal{T}$ is *closed* if all its branches are closed. A prefixed formula $\sigma : \psi$ is *reduced for rule* (r) *in* $\mathcal{B}$ if the branch $\mathcal{B}$ already contains the conclusion of such rule application. By a *systematic tableau construction* we mean an application of the procedure in [14, p. 374] adapted to SST rules.

For each logic L, we establish its bound by considering an L-SST calculus, where a modal level in an SNF_{sml} clause corresponds to the length of a prefix in an SST tableau. The bound then either follows from an already known bound on the length of prefixes in an SST tableau preserving correctness of the SST calculus, or we establish such a bound ourselves. To prove satisfiability equivalence, we show that, for a closed SST tableau with such a bound on the length of prefixes in place, we can construct a resolution refutation of a set of SNF_{sml} or SNF_{ml} clauses with a corresponding bound on modal levels in those clauses.

For a modal formula φ in simplified NNF let d^{φ}_{m} be the modal depth of φ, $d^{\varphi}_{\Diamond}$ be the maximal nesting of $\Diamond$-operators not under the scope of any $\Box$ operators in φ, $n^{\varphi}_{\Box}$ be the number of $\Box$-subformulae in φ, and $n^{\varphi}_{\Diamond}$ be the number of

$\Diamond$-subformulae below $\Box$-operators in φ. Our results for the bounds on the length of prefixes in SST tableaux can then be summarised by the following theorem.

Theorem 2. *Let $L = \mathsf{K}\Sigma$ with $\Sigma \subseteq \{\mathsf{B}, \mathsf{D}, \mathsf{T}, \mathsf{4}, \mathsf{5}\}$. A systematic tableau construction of an L-tableau for a modal formula φ in simplified NNF under the following Constraints (TC1) and (TC2)*

(TC1) a rule (r) of the SST calculus is only applicable to a prefixed formula $\sigma : \psi$ in a branch $\mathcal{B}$ if the formula is not already reduced for (r) in $\mathcal{B}$;
(TC2) rule (π) of the SST calculus is only applicable to prefixed formulae $\sigma : \Diamond\psi$ with $|\sigma| < db^{\varphi}_{L}$ for db^{φ}_{L} as defined in Table 5

terminates in one of following states:

(1) all branches of the constructed tableau are closed and φ is L-unsatisfiable or
(2) at least one branch $\mathcal{B}$ is not closed, no rule is still applicable to a labelled formula in $\mathcal{B}$, and φ is L-satisfiable.

The proof is analogous to Massacci's [21, Section B.2]. Note that for logics $\mathsf{KD4}$ and $\mathsf{KD5}$, we use $\max(1, n^{\varphi}_{\Diamond})$ in the calculation of the bound. That is, if $n^{\varphi}_{\Diamond} \geq 1$ then $\max(1, n^{\varphi}_{\Diamond}) = n^{\varphi}_{\Diamond}$ and the bound is the same as for $\mathsf{K4}$ and $\mathsf{K5}$. Otherwise $\max(1, n^{\varphi}_{\Diamond}) = 1$, that is, the bound is the same as for a formula with a single $\Diamond$-subformula below $\Box$-operators in φ.

For K, KD, KT, KB and KDB these bounds were already stated in [21, Tables III and IV]. The bound for KTB follows straightforwardly from that for KB and KDB. For $\mathsf{KD4}$, Massacci [21, Tables III and IV] states the bound to be the same as for $\mathsf{K4}$. However, this is not correct for the case that the formula φ contains no $\Diamond$-formulae, where its bound would simply be 2, independent of φ. For example, the formula $\Box\Box\Box\mathbf{false}$ which is $\mathsf{KD4}$-unsatisfiable, does not have a closed $\mathsf{KD4}$-tableau with this bound. For the other logics the bounds are new. As argued in [21], the bounds allow tableau decision procedures for extensions of K with axioms $\mathsf{4}$ and $\mathsf{5}$ that do not require a loop check and are therefore of wider interest.

Note that in $\mathsf{KT4}$, $\Box\Box\psi$ and $\Box\psi$ are equivalent and so are $\Box(\psi \wedge \Box\vartheta)$ and $\Box(\psi \wedge \vartheta)$. So, it makes sense to further simplify $\mathsf{KT4}$ formulae using such equivalences before computing the normal form and the bound with the benefit that it may not only reduce the bound but also the size of the normal form. Similar equivalences that can be used to reduce the number of modal operators in a formula also exist for other logics, see, e.g., [8, Chapter 4].

To establish a relationship between closed tableaux and resolution refutations of a set of SNF_{ml} clauses, we formally define the modal layered resolution calculus. Table 6 shows the inference rules of the calculus restricted to labels occurring in our normal form. For GEN1 and GEN3, if the modal clauses in the premises occur at the modal level ml, then the literal clause in the premises occurs at modal level $ml + 1$.

Let Φ be a set of SNF_{ml} clauses. A *(resolution) derivation from* Φ is a sequence of sets $\Phi_0, \Phi_1, \ldots$ where $\Phi_0 = \Phi$ and, for each $i > 0$, $\Phi_{i+1} = \Phi_i \cup \{D\}$, where $D \notin \Phi_i$ is the resolvent obtained from Φ_i by an application of one of the inference rules to premises in Φ_i. A *(resolution) refutation of* Φ is a derivation $\Phi_0, \ldots, \Phi_k$, $k \in \mathbb{N}$, where $0 : \mathbf{false} \in \Phi_k$.

To map a set of SNF_{sml} clauses to a set of SNF_{ml} clauses, using a bound $n \in \mathbb{N}$ on the modal levels, we define a function db_n on clauses and sets of clauses in SNF_{sml} as follows:

$$\mathrm{db}_n(S : \varphi) = \{ml : \varphi \mid ml \in S \text{ and } ml \leq n\}$$
$$\mathrm{db}_n(\Phi) = \bigcup_{S:\varphi \in \Phi} \mathrm{db}_n(S : \varphi)$$

Note that prefixes in SST-tableaux have a minimal length of 1 while the minimal modal level in SNF_{ml} clauses is 0. So, a prefix of length n in a prefixed formula corresponds to a modal level $n - 1$ in an SNF_{ml} clause.

The proof of the following theorem then takes advantage of the fact that we have surrogates and associated clauses for each subformula of φ and proceeds by induction over applications of rule (π).

Theorem 3. *Let $L = \mathsf{K}\Sigma$ with $\Sigma \subseteq \{\mathsf{B}, \mathsf{D}, \mathsf{T}, \mathsf{4}, \mathsf{5}\}$, φ be a $\mathsf{K}\Sigma$-unsatisfiable formula in simplified NNF, db_L^φ be as defined in Table 5, and $\Phi_L = \rho_L^{ml}(\varphi) = \mathrm{db}_{db_L^\varphi - 1}(\rho_L^{sml}(\varphi))$. Then there is a resolution refutation of Φ_L.*

Regarding the size of the encoding, we note that, ignoring the labelling sets, the reduction ρ_L^{sml} into SNF_{sml} is linear with respect to the size of the original formula. The size including the labelling sets would depend on the exact representation of those sets, in particular, of infinite sets. As those are not arbitrary, there is still an overall polynomial bound on the size of the sets of SNF_{sml} clauses produced by ρ_L^{sml}. When transforming clauses from SNF_{sml} into SNF_{ml}, we may need to add every clause to all levels within the bounds provided by Theorem 3. The parameters for calculating those bounds, d_m^φ, $d_\Diamond^\varphi$, $n_\Diamond^\varphi$, and $n_\Box^\varphi$, are all themselves linearly bound by the size of the formula. Thus, in the worst case, which is $\mathsf{S4}$, the size of the clause set produced by ρ_L^{ml} is bounded by a polynomial of degree 3 with respect to the size of the original formula.

It is worth pointing out that both the reduction ρ_L^{sml} of a modal formula to SNF_{sml} and the reduction ρ_L^{ml} to SNF_{ml} are also reversible, that is, we can reconstruct the original formula from the SNF_{sml} and from the SNF_{ml} clause set obtained by ρ_L^{sml} or ρ_L^{ml}, respectively. This reconstruction can also be performed in polynomial time. Thus the reduction itself does not affect the complexity of the satisfiability problem. For instance, the satisfiability problem for $\mathsf{S5}$ is NP-complete and so is the satisfiability problem of the subclass $\mathbb{C}_{\mathsf{S5}}$ of SNF_{ml} clause sets that can be obtained as the result of an application of ρ_{S5}^{ml} to a modal formula. However, a generic decision procedure for K will not be a complexity-optimal decision procedure for $\mathbb{C}_{\mathsf{S5}}$.

Table 6. Inference rules of the MLR calculus

$$\mathsf{LRES}: \frac{\begin{array}{l} ml: D \lor l \\ ml: D' \lor \neg l \end{array}}{ml: D \lor D'} \qquad \mathsf{MRES}: \frac{\begin{array}{l} ml: l_1 \to \Box l \\ ml: l_2 \to \Diamond \neg l \end{array}}{ml: \neg l_1 \lor \neg l_2} \qquad \mathsf{GEN2}: \frac{\begin{array}{l} ml: l'_1 \to \Box l_1 \\ ml: l'_2 \to \Box \neg l_1 \\ ml: l'_3 \to \Diamond l_2 \end{array}}{ml: \neg l'_1 \lor \neg l'_2 \lor \neg l'_3}$$

$$\mathsf{GEN1}: \frac{\begin{array}{l} ml: l'_1 \to \Box \neg l_1 \\ \quad \vdots \\ ml: l'_m \to \Box \neg l_m \\ ml: l' \to \Diamond \neg l \\ ml+1: l_1 \lor \ldots \lor l_m \lor l \end{array}}{ml: \neg l'_1 \lor \ldots \lor \neg l'_m \lor \neg l'} \qquad \mathsf{GEN3}: \frac{\begin{array}{l} ml: l'_1 \to \Box \neg l_1 \\ \quad \vdots \\ ml: l'_m \to \Box \neg l_m \\ ml: l' \to \Diamond l \\ ml+1: l_1 \lor \ldots \lor l_m \end{array}}{ml: \neg l'_1 \lor \ldots \lor \neg l'_m \lor \neg l'}$$

5 Evaluation

An empirical evaluation of the practical usefulness of the reductions we presented in Sects. 3 and 4 faces the challenge that there is no substantive collection of benchmark formulae for the 15 logics of the modal cube except for basic modal logic. Catach [7] evaluates his prover on 31 modal formulae with a maximal length of 22 and maximal modal depth of 4. They are not sufficiently challenging. The QMLTP Problem Library for First-Order Modal Logics [32] focuses on quantified formulae and contains only a few formulae taken from the research literature that are purely propositional and were not written for the basic modal logic K. The Logics Workbench (LWB) benchmark collection [2] contains formulae for K, KT and S4 but not for any of the other logics we consider. For each of these three logics, the collection consists of 18 parameterised classes with 21 formulae each, plus scripts with which further formulae could be generated if needed. All formulae in 9 classes are satisfiable and all formulae in the other 9 classes are unsatisfiable in the respective logic.

In [29] we have used the 18 classes of the LWB benchmark collection for K to evaluate our approach for the six logics consisting of K and its extensions with a single axiom. One drawback of using these 18 classes for other modal logics is that formulae that are K-satisfiable are not necessarily KΣ-satisfiable for non-empty sets Σ of additional axioms. For example, for K5, only 60 out of 180 K-satisfiable formulae were K5-satisfiable. Another drawback is that while K-unsatisfiable formulae are also KΣ-unsatisfiable, a resolution refutation would not necessarily involve any of the additional clauses introduced by our reduction for KΣ. It may be that the additional clauses allow us to find a shorter refutation, but it may just be a case of finding the same refutation in a larger search space. It is also worth recalling that simplification alone is sufficient to determine that all formulae in the class `k_lin_p` are K-unsatisfiable while pure literal elimination can be used to reduce all formulae in `k_grz_p` to the same simple formula [26].

Table 7. Logic-specific modification of unsatisfiable benchmark formulae

Logic L	ψ_l^p	Logic L	ψ_l^p
K	**false**	KD4	$(\Box q_p \wedge \Diamond\Diamond\Box\Diamond\neg q_p)$
KB	$(\neg q_p \wedge \Diamond\Box q_p)$	K5	$(\Diamond\neg q_p \wedge \Diamond\Box q_p)$
KDB	$(\neg q_p \wedge \Diamond\Box((\Box\neg q'_p \wedge \Box q'_p) \vee q_p))$	KD5	$((\Box\neg q_p \wedge \Box q_p) \vee (\Diamond\Box q'_p \wedge \Diamond\neg q'_p)$
KTB	$(\neg q_p \wedge \Diamond\Box((\neg q'_p \wedge \Box q'_p) \vee q_p))$	K45	$(\Box q_p \wedge \Diamond\Box q'_p \wedge \Diamond\Diamond(\neg q_p \vee \neg q'_p))$
KD	$(\Box\neg q_p \wedge \Box q_p)$	KD45	$((\Box\neg q'_p \wedge \Box q'_p) \wedge$
KT	$(\neg q_p \wedge \Box q_p)$		$(\Box q_p \wedge \Diamond\Box q'_p \wedge \Diamond\Diamond(\neg q_p \vee \neg q'_p))$
K4	$(\Box q_p \wedge \Diamond\Diamond\neg q_p)$	S4	$(\neg q'_p \wedge \Box(\neg q'_p \vee \Box q_p) \wedge \Diamond\Diamond\neg q_p)$
K4B	$(\neg q_p \wedge \Diamond\Diamond\Box q_p)$	S5	$((\neg q_p \wedge \Box q_p) \vee (\neg q'_p \wedge \Diamond\Diamond\Diamond\Box q'_p)$

Thus, some of the classes evaluate the preprocessing capabilities of a prover but not the actual calculus and its implementation.

We therefore propose a different approach here. The principles underlying our approach are that (i) there should be the same number of formulae for each logic though not necessarily the same formulae across all logics; (ii) there should be an equal number of satisfiable and unsatisfiable formulae for each logic; (iii) a formula that is L-unsatisfiable should only be L'-unsatisfiable for every extension L' of L; (iv) a formula that is L'-satisfiable should be L-satisfiable for every extension L' of L; (v) the formulae should belong to parameterised classes of formulae of increasing difficulty. Note that Principles (iii) and (iv) are intentionally not symmetric. For L-unsatisfiable formulae it should be necessary for a prover to use the rules or clauses specific to L instead of being able to find a refutation without those. For L-satisfiable formulae we want to maximise the search space for a model.

For unsatisfiable formulae, we take the five LWB classes `k_branch_p`, `k_path_p`, `k_ph_p`, `k_poly_p`, `k_t4p_p` and for each logic L in the modal cube transform each formula in a class so that is L-unsatisfiable, but L'-satisfiable for any logic L' that is not an extension of L. The transformation proceeds by first converting a formula φ to simplified NNF. Then for each propositional literal l it replaces all its occurrences by $(l \vee \psi_L^p)$ where $|l| = p$ and ψ_L^p is a modal formula uniquely associated with p and L, resulting in a formula φ'. Finally, for logics KD4 and KDB we need to add a disjunct $(\Box q \wedge \Box\neg q)$ to φ', while for logics S4 and KTB we need to add a disjunct $(q \wedge \Box\neg q)$, where q is a propositional symbol not occurring in φ'. These disjuncts are unsatisfiable in the respective logics but satisfiable in logics where D, or T, do not hold. Table 7 shows the formulae ψ_L^p that we use in our evaluation. In the table, q_p and q'_p are propositional variables uniquely associated with p that do not occur in φ. The overall effect of this transformation is that the resulting classes of formulae satisfy Principles (iii) and (v).

For satisfiable formulae, we use the five classes `k_poly_n`, `s4_md_n`, `s4_ph_n`, `s4_path_n`, `s4_s5_n` without modification. Although the first of these classes was designed to be K-satisfiable and the other four to be S4-satisfiable, the formulae in those classes are satisfiable in all the logics we consider. `s4_ipc_n` also consists

Table 8. Benchmarking results

Logic	Status	Total	GMR (cneg)	GMR (cord)	GMR (cplain)	R+MLR (cneg)	R+MLR (cord)	R+MLR (cplain)	LEO-III+E
K	S	100	84	85	77	**100**	**100**	**100**	0
KD	S	100	84	85	77	96	**100**	93	0
KT	S	100	70	**81**	50	66	68	61	0
KB	S	100	58	58	29	51	**64**	51	0
K4	S	100	83	**85**	77	56	57	50	0
K5	S	100	**67**	60	45	36	37	26	0
KDB	S	100	63	70	40	56	**73**	55	0
KTB	S	100	58	**59**	38	52	57	31	0
KD4	S	100	83	**85**	77	52	53	46	0
KD5	S	100	**73**	70	61	46	47	38	0
K45	S	100	45	**53**	34	36	37	25	0
K4B	S	100	18	19	11	23	**38**	15	0
KD45	S	100	**67**	66	56	46	47	38	0
S4	S	100	66	**76**	48	45	44	33	0
S5	S	100	32	28	32	32	**35**	24	0
All	S	1500	951	**980**	752	793	857	686	0
K	U	100	74	76	71	**79**	78	77	21
KD	U	100	74	**76**	71	73	75	62	13
KT	U	100	74	**77**	70	71	74	67	30
KB	U	100	71	**78**	68	71	52	55	10
K4	U	100	55	52	**57**	41	29	35	4
K5	U	100	74	46	**75**	50	30	48	8
KDB	U	100	73	**77**	71	73	52	56	8
KTB	U	100	72	**77**	69	67	50	53	9
KD4	U	100	**70**	59	67	40	32	39	1
KD5	U	100	75	46	**77**	51	40	46	3
K45	U	100	**51**	37	49	16	12	8	3
K4B	U	100	47	52	46	**53**	30	49	5
KD45	U	100	**64**	43	55	33	22	28	1
S4	U	100	47	**68**	66	45	21	23	4
S5	U	100	47	51	**52**	36	13	29	2
All	U	1500	**968**	915	964	799	610	675	122

only of S5-satisfiable formulae but these appear to be insufficiently challenging and have not been included in our benchmark set. All other classes of the LWB benchmark classes for K and S4 are satisfiable in some of the logics, but not in all. The five classes satisfy Principles (iv) and (v). The benchmark collection consisting of all ten classes together then also satisfies Principles (i) and (ii).

Another challenge for an empirical evaluation is the lack of available fully automatic theorem provers for all 15 logics that we have already discussed in Sect. 1. This leaves us with just three different approaches we can compare (i) the higher-order logic prover LEO-III [12,37], with **E** 2.6 as external reasoner, *LEO-III+E for short*, that supports a wide range of logics via semantic embedding into higher-order logic (ii) the combination of our reductions with the modal-layered resolution (MLR) calculus for SNF_{ml} clauses [25], *R+MLR calculus for short*, implemented in the modal theorem prover K$_\mathsf{S}$P (iii) the global modal resolution (GMR) calculus, implemented in K$_\mathsf{S}$P, which has resolution rules for all 15 logics [24]. For R+MLR and GMR calculi, resolution inferences between literal clauses can either be unrestricted (`cplain` option), restricted by negative resolution (`cneg` option), or restricted by an ordering (`cord` option). It is worth pointing out that negative and ordered resolution require slightly different transformations to the normal form that introduce additional clauses (`snf+` and `snf++` options, respectively). Also, the ordering cannot be arbitrary [25]. For the experiments, we have used the following options: (i) input processing: prenexing, together with simplification and pure literal elimination (`bnfsimp`, `prenex`, `early_ple`); (ii) preprocessing of clauses: renaming reuses symbols (`limited_reuse_renaming`), forward and backward subsumption (`fsub`, `bsub`) are enabled; the usable is populated with clauses whose maximal literal is positive (`populate_usable, max_lit_positive`); pure literal elimination is set for GMR (`ple`) and modal level ple is set for MLR (`mlple`); (iii) processing: inference rules not required for completeness are also used (`unit, lhs_unit,mres`), the options for preprocessing of clauses are kept and clause selection takes the shortest clause by level (`shortest`).

For LEO-III we provide the prover with a modal formula in the syntax it expects plus a logic specification that tells the prover in which modal logic the formula is meant to be solved, for example, `$modal_system_S4`. LEO-III can collaborate with external reasoners during proof search and we have used **E** 2.6 [34,35] as external reasoner and restricted LEO-III to one instance of **E** running in parallel. LEO-III is implemented in Java and we have set the maximum heap size to 1 GB and the thread stack size to 64 MB for the JVM.

Table 8 shows our benchmarking results. The first three columns of the table show the logic in which we determine the satisfiability status of each formula, the satisfiability status of the formulae, and their number. The next six columns then show how many of those formulae were solved by K$_\mathsf{S}$P with a particular calculus and refinement. The last column shows the result for LEO-III. The highest number or numbers are highlighted in bold. A time limit of 100 CPU seconds was set for each formula. Benchmarking was performed on a PC with an AMD Ryzen 5 5600X CPU @ 4.60 GHz max and 64 GB main memory using Fedora release 34 as operating system.

While the R+MLR calculus is competitive with GMR on extensions of K with axioms D, T and, possibly, B, the GMR calculus has better performance on extensions with axioms $\mathsf{4}$ and $\mathsf{5}$.

On satisfiable formulae, where for all logics we use exactly the same formulae and both resolution calculi have to saturate the set of clauses up to redundancy, the number of formulae solved is directly linked to the number of inferences necessary to do so. The fact that we reduce SNF_{sml} clauses to SNF_{ml} clauses via the introduction of multiple copies of the same clausal formulae with different labels clearly leads to a corresponding multiplication of the inferences that need to be performed. LEO-III+E does not solve any of the satisfiable formulae. This can be seen as an illustration of how important the use of additional techniques is that can turn resolution into a decision procedure on embeddings of modal logics into first-order logic [18,33].

On unsatisfiable formulae, where we use different formulae for each logic, the number of formulae solved is linked to the number of inferences it takes to find a refutation. For instance, on K it takes the GMR calculus on average 6.2 times the number of inferences to find a refutation than the R+MLR calculus. However, for all other logics the opposite is true. On the remaining 14 logics, the R+MLR calculus on average requires 6.5 times the number of inferences to find a refutation than the GMR calculus. Given that the R+MLR calculus currently uses a reduction from a modal logic to SNF_{sml} followed by a transformation from SNF_{sml} to SNF_{ml}, it is difficult to discern which of the two is the major problem. It is clear that multiple copies of the same clausal formulae are also detrimental to proof search. LEO-III+E does reasonably well on unsatisfiable formulae and the results clearly show the impact that additional axioms have on its performance. It performs best for KT and K but for logics involving axioms $\mathsf{4}$ and $\mathsf{5}$ very few formulae can be solved. The external prover **E** finds the proof for 121 out of the 122 modal formulae LEO-III+E can solve.

6 Conclusions

We have presented novel reductions of extensions of the modal logic K with arbitrary combinations of the axioms B, D, T, $\mathsf{4}$, $\mathsf{5}$ to clausal normal forms SNF_{sml} and SNF_{ml} for K. The implementation of those reductions combined with K$_{\mathrm{S}}$P [26], allows us to reason in all 15 logics of the modal cube in a fully automatic way. Such support was so far extremely limited.

The transformation of sets of SNF_{sml} to sets of SNF_{ml} relies on new results that show that non-clausal closed tableaux in the Single Step Tableaux calculus [14,21] can be simulated by refutations in the modal-layered resolution (MLR) calculus for SNF_{ml} clauses [25].

We have also developed a new collection of benchmark formulae that covers all 15 logics of the modal cube. The collection consists of classes of parameterised and therefore scalable formulae. It contains an equal number of satisfiable and unsatisfiable formulae for each logic and the satisfiability status of each formula is known in advance. So far extensive collections of benchmark formulae were only available for K with smaller collections available for KT and $\mathsf{S4}$. A key feature of the approach is that it uses the systematic modification of K-unsatisfiable formulae to obtain unsatisfiable formulae in other logics. Thus, we could obtain a more extensive collection by applying this approach to further collections of benchmark formulae for K.

The evaluation we presented shows that on most of the 15 modal logics the combination of our reduction to SNF_{ml} with the MLR calculus does not perform as well as the global modal resolution (GMR) calculus, also implemented in KSP. This contrasts with the evaluation in [29], where we only considered six logics and used a different collection of benchmarks. We believe that the new benchmark collection more clearly indicates weaknesses in the current approach, in particular, the reduction from SNF_{sml} to SNF_{ml}. It is possible that the implementation of a calculus that operates directly on sets of SNF_{sml} clauses would perform considerably better as it avoids the repetition of clauses with different labels. However, it does so by using potentially infinite sets of labels which makes an implementation challenging. We intend to explore this possibility in future work.

References

1. Areces, C., de Rijke, M., de Nivelle, H.: Resolution in modal, description and hybrid logic. J. Log. Comput. **11**(5), 717–736 (2001)
2. Balsiger, P., Heuerding, A., Schwendimann, S.: A benchmark method for the propositional modal logics K, KT, S4. J. Autom. Reasoning **24**(3), 297–317 (2000). https://doi.org/10.1023/A:1006249507577
3. Basin, D., Matthews, S., Vigano, L.: Labelled propositional modal logics: theory and practice. J. Log. Comput. **7**(6), 685–717 (1997)
4. Blackburn, P., van Benthem, J., Wolter, F. (eds.): Handbook of Modal Logic. Elsevier (2006)
5. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (2002)
6. Brünnler, K.: Deep sequent systems for modal logic. Arch. Math. Log. **48**(6), 551–577 (2009). https://doi.org/10.1007/s00153-009-0137-3
7. Catach, L.: TABLEAUX: a general theorem prover for modal logics. J. Autom. Reason. **7**(4), 489–510 (1991). https://doi.org/10.1007/BF01880326
8. Chellas, B.F.: Modal Logic: An Introduction. Cambridge University Press, Cambridge (1980)
9. Fitting, M.: Prefixed tableaus and nested sequents. Ann. Pure Appl. Log. **163**(3), 291–313 (2012)
10. Gasquet, O., Herzig, A., Longin, D., Sahade, M.: LoTREC: logical tableaux research engineering companion. In: Beckert, B. (ed.) TABLEAUX 2005. LNCS (LNAI), vol. 3702, pp. 318–322. Springer, Heidelberg (2005). https://doi.org/10.1007/11554554_25
11. Girlando, M., Straßburger, L.: MOIN: a nested sequent theorem prover for intuitionistic modal logics (system description). In: Peltier, N., Sofronie-Stokkermans, V. (eds.) IJCAR 2020. LNCS (LNAI), vol. 12167, pp. 398–407. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51054-1_25
12. Gleißner, T., Steen, A.: LEO-III (2022). https://github.com/leoprover/Leo-III
13. Gleißner, T., Steen, A., Benzmüller, C.: Theorem provers for every normal modal logic. In: Eiter, T., Sands, D. (eds.) LPAR 2017. EPiC Series in Computing, vol. 46, pp. 14–30. EasyChair (2017). https://doi.org/10.29007/jsb9
14. Goré, R.: Tableau methods for modal and temporal logics. In: D'Agostino, M., Gabbay, D., Hähnle, R., Posegga, J. (eds.) Handbook of Tableau Methods, pp.

297–396. Springer, Heidelberg (1999). https://doi.org/10.1007/978-94-017-1754-0_6
15. Goré, R., Kikkert, C.: CEGAR-Tableaux: improved modal satisfiability via modal clause-learning and SAT. In: Das, A., Negri, S. (eds.) TABLEAUX 2021. LNCS (LNAI), vol. 12842, pp. 74–91. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_5
16. Governatori, G.: Labelled modal tableaux. In: Areces, C., Goldblatt, R. (eds.) AiML 2008, pp. 87–110. College Publications (2008)
17. Halpern, J.Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artif. Intell. **54**(3), 319–379 (1992)
18. Horrocks, I., Hustadt, U., Sattler, U., Schmidt, R.A.: Computational modal logic. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, chap. 4, pp. 181–245. Elsevier (2006)
19. Hustadt, U., de Nivelle, H., Schmidt, R.A.: Resolution-based methods for modal logics. Log. J. IGPL **8**(3), 265–292 (2000)
20. van Linder, B., van der Hoek, W., Meyer, J.J.C.: Formalising abilities and opportunities of agents. Fundamenta Informaticae **34**(1–2), 53–101 (1998)
21. Massacci, F.: Single step tableaux for modal logics. J. Autom. Reason. **24**, 319–364 (2000). https://doi.org/10.1023/A:1006155811656
22. Mayer, M.C.: Herbrand style proof procedures for modal logics. J. Appl. Non Class. Logics **3**(2), 205–223 (1993)
23. Nalon, C.: K_SP (2022). https://www.nalon.org/#software
24. Nalon, C., Dixon, C.: Clausal resolution for normal modal logics. J. Algorithms **62**, 117–134 (2007)
25. Nalon, C., Dixon, C., Hustadt, U.: Modal resolution: proofs, layers, and refinements. ACM Trans. Comput. Log. **20**(4), 23:1–23:38 (2019)
26. Nalon, C., Hustadt, U., Dixon, C.: K_SP: architecture, refinements, strategies and experiments. J. Autom. Reason. **64**(3), 461–484 (2020). https://doi.org/10.1007/s10817-018-09503-x
27. Ohlbach, H.J., Nonnengart, A., de Rijke, M., Gabbay, D.M.: Encoding two-valued nonclassical logics in classical logic. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, chap. 21, pp. 1403–1485. Elsevier (2001)
28. Otten, J.: MleanCoP: a connection prover for first-order modal logic. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) IJCAR 2014. LNCS (LNAI), vol. 8562, pp. 269–276. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08587-6_20
29. Papacchini, F., Nalon, C., Hustadt, U., Dixon, C.: Efficient local reductions to basic modal logic. In: Platzer, A., Sutcliffe, G. (eds.) CADE 2021. LNCS (LNAI), vol. 12699, pp. 76–92. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79876-5_5
30. Plaisted, D.A., Greenbaum, S.: A structure-preserving clause form translation. J. Symb. Comput. **2**(3), 293–304 (1986)
31. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: KR 1991, pp. 473–484. Morgan Kaufmann (1991)
32. Raths, T., Otten, J.: The QMLTP problem library for first-order modal logics. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR 2012. LNCS (LNAI), vol. 7364, pp. 454–461. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31365-3_35
33. Schmidt, R.A., Hustadt, U.: First-order resolution methods for modal logics. In: Voronkov, A., Weidenbach, C. (eds.) Programming Logics. LNCS, vol. 7797, pp. 345–391. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37651-1_15

34. Schulz, S.: E 2.6 (2022). http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/Download.html
35. Schulz, S., Cruanes, S., Vukmirović, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) CADE 2019. LNCS (LNAI), vol. 11716, pp. 495–507. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29436-6_29
36. Steen, A., Benzmüller, C.: The higher-order prover Leo-III. In: Giacomo, G.D., et al. (eds.) ECAI 2020. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2937–2938. IOS Press (2020). https://doi.org/10.3233/FAIA200462
37. Steen, A., Benzmüller, C.: Extensional higher-order paramodulation in Leo-III. J. Autom. Reason. **65**(6), 775–807 (2021). https://doi.org/10.1007/s10817-021-09588-x
38. The SPASS Team: SPASS 3.9 (2016). http://www.spass-prover.org/
39. Boy de la Tour, T.: An optimality result for clause form translation. J. Symb. Comput. **14**(4), 283–301 (1992)
40. Tseitin, G.S.: On the complexity of derivation in propositional calculus. In: Siekmann, J.H., Wrightson, G. (eds.) Automation of Reasoning: Classical Papers on Computational Logic 1967–1970, vol. 2, pp. 466–483. Springer, Heidelberg (1983). https://doi.org/10.1007/978-3-642-81955-1_28. Original paper (in Russian) appeared in 1968
41. Wansing, H.: Sequent calculi for normal modal proposisional logics. J. Log. Comput. **4**(2), 125–142 (1994)
42. Weidenbach, C.: Combining superposition, sorts and splitting. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 1965–2013. Elsevier and MIT Press (2001)
43. Weidenbach, C., Dimova, D., Fietzke, A., Kumar, R., Suda, M., Wischnewski, P.: SPASS version 3.5. In: Schmidt, R.A. (ed.) CADE 2009. LNCS (LNAI), vol. 5663, pp. 140–145. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02959-2_10