







# A Modular Framework for Proof-Search via Formalised Modal Completeness in HOL Light

Antonella Bilotta   

Scuola Normale Superiore di Pisa, Italy

Marco Maggesi   

University of Florence, Italy

Cosimo Perini Brogi   

IMT School for Advanced Studies Lucca, Italy

---

## Abstract

We extend the existing HOL Light Library for Modal Systems (HOLMS) to support a modular implementation of modal reasoning within the HOL Light proof assistant. We deeply embed axiomatic calculi and relational semantics for seven normal modal logics (K, T, B, K4, S4, S5, GL) and formalise modal adequacy theorems for these systems. We then leverage those formalisations to implement a mechanism for automated reasoning via proof-search in the associated labelled sequent calculi, which we shallowly embed in HOL Light’s goal-stack mechanism. This way, we equip the general-purpose proof assistant with (semi)decision procedures for these logics that, in case of failure to construct a proof for the input formula, return a certified countermodel within the appropriate class for the logic under consideration. On the methodological side, we propose a precise measure of the modularity of our approach by systematically adopting Christopher Strachey’s distinction between ad hoc and parametric polymorphism throughout the library.

**2012 ACM Subject Classification** Theory of computation → Modal and temporal logics; Theory of computation → Higher order logic; Theory of computation → Automated reasoning; Theory of computation → Type theory; Theory of computation → Proof theory; Theory of computation → Logic and verification

**Keywords and phrases** Modal logic, HOL Light, Labelled sequent calculi, Logical verification, Interactive theorem proving, Automated proof-search

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2026.18

**Supplementary Material** *Software (Source code)*: <https://github.com/HOLMS-lib/HOLMS> [19]  
archived at `swh:1:dir:36edca29f53f3fdd0a7fc675702990743eda4c8d`

*Software (Project Documentation Website)*: <https://holms-lib.github.io/> [21]

**Funding** This work was partially funded by the project SERICS – Security and Rights in the CyberSpace PE0000014, financed within PNRR, M4C2 I.1.3, funded by the European Union – NextGenerationEU (MUR Code: 2022CY2J5S, CUP: D67G22000340001); the Istituto Nazionale di Alta Matematica – INdAM group GNSAGA; the EC-COST Action (CA20111) “EuroProofNet”; the research project “Differential, Algebraic, Complex and Arithmetic Geometry (2025)”; The International Research Network “Logic and Interaction”.

*Marco Maggesi*: Istituto Nazionale di Alta Matematica – INdAM group GNSAGA.

*Cosimo Perini Brogi*: The project SERICS – Security and Rights in the CyberSpace PE0000014, financed within PNRR, M4C2 I.1.3, funded by the European Union – NextGenerationEU (MUR Code: 2022CY2J5S); The International Research Network “Logic and Interaction”.

**Acknowledgements** We thank three anonymous reviewers for their valuable and constructive feedback on our original submission.



© Antonella Bilotta, Marco Maggesi, and Cosimo Perini Brogi;  
licensed under Creative Commons License CC-BY 4.0

34th EACSL Annual Conference on Computer Science Logic (CSL 2026).

Editors: Stefano Guerrini and Barbara König; Article No. 18; pp. 18:1–18:29

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

**Modal Reasoning and Applications.** Since at least the 1970s, modal logic has been a central tool in various technical and scientific disciplines [50, 24], including knowledge representation, reasoning under uncertainty, multi-agent systems, computational processes, verification of normative consistency, and decision-making models [24, 23, 111, 3]. In computer science, modal operators capture abstract properties across domains: computation with side effects [96, 9, 99], recursive calls [81, 6], formal provability [25, 117], information flow in communication protocols [72, 100, 64, 75]. Furthermore, model checking is a key hardware and software verification technique that leverages modal and temporal logics [32, 31, 111].

The broad applicability of modal logics, as captured by their semantics, has made it essential in both academic and industrial contexts. Progressively, the syntactic characterisations have received scientific relevance, driven by enriched Gentzen-style proof systems and new deductive calculi [105, 8, 118, 90, 98, 78, 28]. These calculi have proven well-behaved structurally and computationally, leading to efficient automation of non-classical reasoning [34, 91, 52, 57, 53].<sup>1</sup> More recently, proof cycles have enabled the development of proof systems for modal logics with fix-point operators [5, 101, 4], whose semantics were already used in model checking for software and hardware verification.

**Our Library.** There exists a well-established research line in formalising and automating modal reasoning from both theoretical and more applicative viewpoints.<sup>2</sup> We have contributed to that trend by **HOLMS** [21], which stands for **HOL** Light **Library** for **Modal Systems**. It endows the HOL Light proof assistant [68, 66] with a general mechanism for automated theorem proving and countermodel generation for modal formulas.

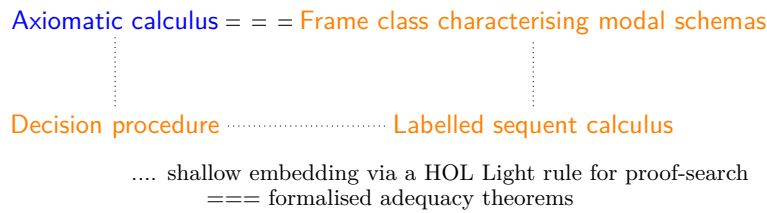
This library builds on preliminary work on the mechanisation of Gödel-Löb logic [76, 77]. In the first paper, we formalised a proof of completeness for GL w.r.t. its standard relational models. In the second, we used this result to design a shallow embedding of a cut-free labelled sequent calculus for GL [86], and we automated root-first proof-search in that deductive system through a new tactic of HOL Light dedicated to that. This tactic produces a HOL Light theorem asserting the provability in GL of the input modal formula, or identifies a countermodel to it.

The resulting method is not *per se* limited to that specific modal system, and can be extended – with some additional effort – to cover any logic with well-behaved labelled sequent calculi, even beyond the class of normal and classical modal logics [50]. Figure 1 illustrates the implementation structure underlying HOLMS. Its generality highlights the modular design of the framework and shows the approach can be extended to other modal logics by appropriately instantiating their axiomatic calculi, semantics, and associated labelled calculi.

Here we extend the preliminary results reported in two communication papers. In [22], we introduced HOLMS as an embryonic framework for covering normal modal logics in a modular way within the HOL Light proof assistant. In [20], we built on those results by presenting a modularisation of the code covering the completeness proofs for the minimal system K and its extensions T, K4, S4, B, S5, as well as a reshaping of the original completeness proof for GL.

<sup>1</sup> Structural properties of calculi play a key role in mechanising proof-search: see [115, 70, 89].

<sup>2</sup> We refer to Section 6 below for an overview.



■ **Figure 1** The implementation methodology behind HOLMS. Colours differentiate ad hoc vs parametric modularity as discussed below and in Appendix B.

**Paper Contributions.** Our paper pursues two complementary directions, targeting both implementation concerns and foundational aspects of HOLMS.

From an **implementation standpoint**, we further *augment* HOL Light’s *automated modal reasoning functionality*. The present implementation expands the proof assistant’s deductive framework through the introduction of a specialised inference rule (`HOLMS_RULE`). It evaluates whether a provided modal formula represents a theorem within a designated modal logic chosen from *seven* normal modal logics – namely, the minimal system K and its extensions T, K4, S4, B, S5, as well as GL. The new rule endows HOLMS with a principled decision procedure based on automated proof-search in the associated labelled sequent calculi. Furthermore, `HOLMS_RULE` provides a substantial improvement of the original tactic for GL (from [77]) in terms of performance.

Behind the decision procedure, there is a substantial formalisation work concerning a fully modular proof of completeness for normal modal systems w.r.t. relational semantics. Our formal proof generalises the previous for GL by leveraging known results in modal correspondence theory and identifying the precise parameters used in e.g. [25] to sketch completeness proofs of several modal logics uniformly.

In our codebase we have three interconnected presentations of (normal) modal logics: (i) axiomatic calculi; (ii) relational semantics – both deeply embedded; and (iii) labelled sequent calculi – which we shallow embed in the goal-stack mechanism of HOL Light as a certificate of correctness of the decision procedure behind `HOLMS_RULE`. The formalised adequacy theorem lets us safely reduce the provability task for a given formula into the task of proving its validity in the corresponding class of frames; the latter goal is solved (or refuted) by performing a root-first proof-search in the corresponding labelled sequent calculus – as in [88, 49] – shallowly embedded in HOL Light. In case of failure, the rule extracts from the top-most goal-stack of HOL Light a candidate countermodel, that is *formally proven* by the proof assistant to falsify the input formula at its root.

From a **foundational perspective**, this work *extends and reshapes* HOLMS according to a precisely defined *modular architecture*. Our approach centres on a scalable and uniform proof of semantic adequacy. To quantify proof modularity precisely – along with the corresponding formalisation counterpart – we adopted a strict coding methodology, drawing inspiration from [113]. Throughout development, we implemented a differentiation between: *parametric polymorphic* code, completely independent of particular parameter instantiations; and *ad hoc polymorphic* code, whose elements are customised for the specific modal logic being considered. This colour coding scheme is maintained consistently across subsequent sections. We provide a detailed discussion of this distinction and coding discipline in Appendix B below.

In parallel, we emphasise here that HOL Light – true to its name – features a minimal logical kernel, supporting more trustworthy verification and greater transparency [1]. While these advantages may be less of a concern in purely theoretical contexts, they are crucial for real-world applications of modal logics, particularly for the growing use of labelled systems in proof-based verification.

**Paper Structure.** We implement modal systems through a parametric polymorphic formal deducibility relation in Section 2, and prove the associated deduction theorem. In Section 3, we prepare the stage for establishing modal adequacy results formalising relational semantics and known results in correspondence theory, as well as soundness and consistency lemmas. We then present our ad hoc polymorphic completeness theorem in Section 4, providing code snippets to illustrate key steps for each logic considered here. In Section 5, we explain the (semi)decision procedures based on the shallow embedding of root-first proof-search in labelled sequent calculi.<sup>3</sup> Related and future work are outlined in Section 6. Appendix A provides a brief overview of our current codebase and its structure. Appendix B discusses the ad hoc vs parametric coding discipline we implemented. Appendix C provides additional details of the proof sketch in Section 4.

**Source Code.** The latest version of HOLMS is archived on Software Heritage [↗](#). The readme file from the git repository [↗](#) and the project webpage [↗](#) provide some pointers to the main contents of the library. Appendix A provides an overview of our repository. In the following pages, we provide hyperlinks – marked by the symbol [↗](#) – pointing to the code snippets we discuss.

## 2 Normal Modal Systems: Syntactic Presentation

HOLMS is a library for formal reasoning *with and about* modal logics, using HOL Light as the formal metatheory for studying modal logics as object theories.

In particular – by following the deep embedding provided by Harrison’s tutorial [66, §20] – our framework formalises the syntax of propositional modal logic together with the basics of Kripke semantics to distinguish the modal syntax from its semantic counterpart, and from HOL Light’s logical operators as well. We then define a notion of deducibility from hypotheses in an axiomatic system for normal modal logics which is parametric on a set of axiom schemas, instantiated case by case into the calculi corresponding to each of the logic we investigate. At the end of this section, we prove a deduction theorem for such a parametric deducibility relation.

### 2.1 Modal Language and Syntax

Let  $\mathcal{L}_\Box$  be a language containing *all* propositional connectives of classical logic, plus a unary modal operator “ $\Box$ ”. The set of modal formulas  $\mathbf{Form}_\Box$  is defined<sup>4</sup> by the following grammar:  $A, B \in \mathbf{Form}_\Box ::= \perp \mid \top \mid a \mid \neg A \mid A \wedge B \mid A \vee B \mid A \rightarrow B \mid A \leftrightarrow B \mid \Box A$ .

To distinguish the logical operators of HOL Light from those of the modal object language, we introduce specific notations and a new inductive type `:form ↗` for the latter. The table in Figure 2 provides a glossary making this distinction clear.

<sup>3</sup> We explain the need for the “semi” prefix in Remark 12 below.

<sup>4</sup> We use lowercase letters as metavariables for propositional atoms and capital letters for modal formulas. In HOL Light source code both denote modal formulas.

HOL Light notation		Natural language	
HOL Light	HOLMS	Metalinguage	Object Language
T, F	True, False	true, false	$\top, \perp$
$\sim$	Not	not	$\neg$
$\wedge, \vee$	$\&\&, \ \ $	and, or	$\wedge, \vee$
$\implies, \iff$	$\dashv\vdash, \iff$	if, iff	$\rightarrow, \leftrightarrow$
!	Box		$\square$
?		for all ( $\forall$ ), exists ( $\exists$ )	

■ **Figure 2** A glossary for HOLMS logical notation.

## 2.2 Axiomatic Calculi

To circumvent code duplication in the formalisation of deductive systems based on axiomatic calculi, we define a ternary deducibility relation  $\mathcal{S}.\mathcal{H} \vdash A$  that is parametric to a set of axiom schemas  $\mathcal{S}$  and considers a set of hypotheses  $\mathcal{H}$ . Such a formal predicate conceptualises the abstract notion of deducibility close to the one discussed in [46], based on a minimal deductive system modularly extended by instantiating  $\mathcal{S}$ .

As the current version of HOLMS only deals with normal modal systems, it is natural to take the logic K as the minimal deductive engine of the deducibility relation. For K, we take the standard axiomatization ( **KAXIOM**  $\varnothing$  ) extending the classical propositional calculus by the distribution schema  $K := \square(A \rightarrow B) \rightarrow (\square A \rightarrow \square B)$  and the necessitation rule **RN**  $\frac{\mathcal{S}.\varnothing \vdash A}{\mathcal{S}.\varnothing \vdash \square A}$ . The table in Figure 3 displays HOLMS formalisation of the sets of axiom schemas representing the seven normal logics under analysis.

Logic	Axiom Schema	HOL Light Constant
K	$\emptyset$	$\{\}$
T	$\{T := \square A \rightarrow A\}$	$T\_AX \varnothing$
K4	$\{4 := \square A \rightarrow \square \square A\}$	$K4\_AX \varnothing$
S4	$\{T; 4\}$	$S4\_AX \varnothing$
B	$\{T; B := A \rightarrow \square \diamond A\}$	$B\_AX \varnothing$
S5	$\{T; 5 := \diamond A \rightarrow \square \diamond A\}$	$S5\_AX \varnothing$
GL	$\{GL := \square(\square A \rightarrow A) \rightarrow \square A\}$	$GL\_AX \varnothing$

■ **Figure 3** HOLMS formalisations of the seven modal logics.

Our general deducibility relation is then defined as follows:

► **Definition 1 (Proof system)**. **MODPROVES**  $\varnothing$  The ternary predicate  $\mathcal{S}.\mathcal{H} \vdash A$ , denoting the deducibility of a formula  $A$  from a set of hypotheses  $\mathcal{H}$  in an axiomatic extension of logic  $K$  via schemas in the set  $\mathcal{S}$ , is inductively defined by the following conditions:

- For every instance  $A$  of axiom schemas for the calculus  $K$ ,  $\mathcal{S}.\mathcal{H} \vdash A$ ;
- For every instance  $A$  of schemas in  $\mathcal{S}$ ,  $\mathcal{S}.\mathcal{H} \vdash A$ ;
- For every  $A \in \mathcal{H}$ ,  $\mathcal{S}.\mathcal{H} \vdash A$ ;
- If  $\mathcal{S}.\mathcal{H} \vdash B \rightarrow A$  and  $\mathcal{S}.\mathcal{H} \vdash B$ , then  $\mathcal{S}.\mathcal{H} \vdash A$ ;
- If  $\mathcal{S}.\varnothing \vdash A$ , then  $\mathcal{S}.\mathcal{H} \vdash \square A$  for any set of formulas  $\mathcal{H}$ .

Notice that HOLMS source code now deals with two distinct symbols of derivability:  $\vdash$  denotes a theorem formally proven in HOL Light; the notation  $[\mathcal{S}.\{\}\mid\sim]$  denotes a term of HOL Light metalanguage representing a theorem of the object modal logic axiomatised by the schemas in  $\mathcal{S}$ .

We can prove a deduction theorem for our derivability relation as a minimal sanity check for Def. 1. This reduces the common notion of derivability of  $A$  in a calculus characterised by the schemas in  $\mathcal{S}$  to the relation  $\mathcal{S}.\emptyset \vdash A$ . Its formal proof is loosely inspired by the analogous strategy in [46, §4] and we can thus omit it from here.

► **Theorem 2 (Deduction theorem).** `MODPROVES_DEDUCTION_LEMMA`  $\varnothing$  *For any modal formulas  $A, B$  and any sets of formulas  $\mathcal{S}, \mathcal{H}$ , the following equivalence holds:  $\mathcal{S}.\mathcal{H} \cup \{B\} \vdash A$  iff  $\mathcal{S}.\mathcal{H} \vdash B \rightarrow A$ .*

### 3 Normal Modal Systems: Semantic Presentation

In this section, we formally define Kripke semantics, following the approach in [66, §20] and displaying its deep embedding in HOL Light. Next, we formalise results in correspondence theory and prove that the axiomatic systems are sound w.r.t. the class of frames corresponding to their schemas. We also derive a semantic proof of consistency for the seven logics.

#### 3.1 Relational Semantics

For the sake of completeness, we report below some basic concepts of classical relational semantics embedded in HOLMS. In particular, we define relational frames and the *forcing relation*. Such a relation computes the truth of a modal formula in a given world  $w$  of a frame-based model, and it is standardly defined by induction on the complexity of modal formulas.

► **Definition 3 (Relational model).** *A relational model is a triple  $\langle W, R, V \rangle$  such that:*

- $\langle W, R \rangle$  is a relational frame (`FRAME`  $\varnothing$ ), i.e. an ordered pair consisting of
  - $W$  is a nonempty set of “possible worlds”;
  - $R$  is a binary accessibility relation over worlds  $R \subseteq W \times W$ .
- $V$  is a valuation on  $W$ ; i.e. a binary relation  $V \subseteq \Phi \times W$ , where  $\Phi$  is the set of variables. We say that the model  $\mathcal{M} = \langle W, R, V \rangle$  is  $\langle W, R \rangle$ -based.

► **Definition 4 (Forcing relation).** `holds`  $\varnothing$  *The forcing relation  $\Vdash$  between a model  $\mathcal{M} = \langle W, R, V \rangle$ , a world  $x \in W$ , and a formula  $A$  is inductively defined on the structure of  $A$  as follows:*

- $x \Vdash_{\mathcal{M}} a \iff aVx$ ,
- $x \not\Vdash_{\mathcal{M}} \perp$  for any  $x, \mathcal{M}$ ,
- $x \Vdash_{\mathcal{M}} A \wedge B \iff x \Vdash_{\mathcal{M}} A$  and  $x \Vdash_{\mathcal{M}} B$ ,
- $x \Vdash_{\mathcal{M}} \Box A \iff \forall y \in W, \text{ if } xRy \text{ then } y \Vdash_{\mathcal{M}} A$ .

*Forcing conditions for the remaining classical connectives are deduced from the clauses for conjunction and bottom as usual. Similarly, we define the diamond operator as  $\Diamond A := \neg \Box \neg A$ .*

*In words, when  $x \Vdash_{\mathcal{M}} A$ , we say that “ $A$  is forced by  $x$  in  $\mathcal{M}$ ”.*

Listing 1 presents the definition in HOLMS of the predicate formalising this relation.

■ **Listing 1** Implementation in HOLMS of a forcing relation `holds`.

```
(holds WR V False (w:W) <=> F) /\
(holds WR V (Atom s) w <=> V s w) /\
(holds WR V (p && q) w <=> holds WR V p w /\ holds WR V q w) /\
(holds WR V (Box p) w <=>
 !w'. w' IN FST WR /\ SND WR w w' ==> holds WR V p w')
```

In addition to forcing, Kripke semantics develops three notions for validity of modal formulas, ordered in increasing level of generality.

► **Definition 5 (Validity of modal formulas)**. *Given a formula  $A$ :*

$A$  is valid in a model  $\mathcal{M}$  iff it is forced by every world  $w$  in  $\mathcal{M}$ . Formally, we write  $\mathcal{M} \models A$  if  $w \Vdash_{\mathcal{M}} A$  for any possible world  $w$  of  $\mathcal{M}$ .

$A$  is valid in a frame  $\mathcal{F}$  (`holds_in`) iff it is true in any model based on that frame, i.e., if any world forces it for any valuation on  $\mathcal{F}$ . Formally, we write  $\mathcal{F} \models A$  if  $\mathcal{M} \models A$  for any  $\mathcal{M}$  based on  $\mathcal{F}$ .

$A$  is valid in a class of frames  $\mathfrak{S}$  (`valid`) iff  $A$  is valid in any frame in  $\mathfrak{S}$ . Formally, we write  $\mathfrak{S} \models A$  if  $\mathcal{F} \models A$  for any frame  $\mathcal{F} \in \mathfrak{S}$ .

### 3.2 Correspondence Theory

Correspondence theory analyses and searches for connections between properties of relational frames and modal formulas that hold in exactly the relational frames with those properties [116]. Lemma 6 recaps the correspondence results for the logics currently developed in our codebase.

► **Lemma 6 (Correspondence lemmas)**. *For any  $A \in \mathbf{Form}_{\square}$ , the following hold:*<sup>5</sup>

1. `MODAL_REFL`  $\varnothing \langle W, R \rangle \models \top : \Box A \rightarrow A$  iff  $R$  is reflexive.
2. `MODAL_TRANS`  $\varnothing \langle W, R \rangle \models 4 : \Box A \rightarrow \Box \Box A$  iff  $R$  is transitive.
3. `MODAL_SYM`  $\varnothing \langle W, R \rangle \models \mathbf{B} : A \rightarrow \Box \Diamond A$  iff  $R$  is symmetric.
4. `MODAL_EUCL`  $\varnothing \langle W, R \rangle \models 5 : \Diamond A \rightarrow \Box \Diamond A$  iff  $R$  is Euclidean.
5. `MODAL_TRANSNT`  $\varnothing \langle W, R \rangle \models \mathbf{GL} : \Box(\Box A \rightarrow A) \rightarrow \Box A$  iff  $R$  is transitive and Noetherian.<sup>6</sup>

To consider further development of our library, we found it convenient to rephrase specific results in a general setting and formally define a predicate `CHAR S`, representing the class of frames characteristic to  $S$ :

► **Definition 7 (Characteristic class)**. `CHAR`  $\varnothing$   $A$  frame property  $P$  is characteristic to a set of axiom schemas  $S$  when the following holds: every axiom schema in  $S$  is valid in a frame if and only if the frame has the given property  $P$ .

The characteristic class for  $S$  consists exactly of the frames satisfying the characteristic property to  $S$ .

■ **Listing 2** HOLMS definitions of sets of characteristic frames.

```
CHAR S = {(W:W->bool, R:W->W->bool) |
  (W,R) IN FRAME /\
  (!p. p IN S ==> holds_in (W:W->bool, R:W->W->bool) p)}
```

<sup>5</sup> Our proofs of these lemmas formalise the standard arguments presented in [25, §4], and we omit them.

<sup>6</sup> A binary relation  $R$  is said to be Noetherian if any non-empty subset of its domain does not have infinite ascending  $R$ -chains. See Figure 4 for a formalisation in HOL.

Following [25, § 11], we also introduce the notion of appropriate frame.

► **Definition 8 (Appropriate frame).** `APPR`  $\varnothing$  *The class of appropriate frames to a set of axiom schemas  $\mathcal{S}$  consists of those finite frames  $\mathcal{F}$  such that any theorem of the axiomatic calculus defined by  $\mathcal{S}$  is valid in  $\mathcal{F}$ .*

As the set of formulas valid in a frame is closed under the rules of our deducibility relation, it is relatively easy to prove that, for any systems currently in our codebase, the class of its appropriate frames coincides with its characteristic class *restricted to finite frames* (`APPR_EQ_CHAR_FINITE`  $\varnothing$ ).<sup>7</sup>

The table in Figure 4 summarises the known correspondence results that we prove for each system in our codebase.

$\mathcal{S}$	CHAR $\mathcal{S}$	APPR $\mathcal{S}$	Characteristic properties for $\mathcal{S}$
K	FRAME $\varnothing$	FINITE_FRAME $\varnothing$	-
T	REFL $\varnothing$	RF $\varnothing$	Reflexivity: $\forall w \in W. wRw$
K4	TRANS $\varnothing$	TF $\varnothing$	Transitivity: $\forall w, x, z \in W. wRx \text{ and } xRz \implies wRz$
S4	RTRANS $\varnothing$	RTF $\varnothing$	Reflexivity and transitivity
B	RSYM $\varnothing$	RSF $\varnothing$	Reflexivity and symmetry: $\forall w, x \in W. wRx \implies xRw$
S5	REUCL $\varnothing$	REF $\varnothing$	Reflexivity and Euclideanity: $\forall w, x, z \in W. wRx \text{ and } wRz \implies zRx$
GL	TRANSNT $\varnothing$	ITF $\varnothing$ <sup>8</sup>	Transitivity and Noetherianity: $\forall X \subseteq W. X \neq \emptyset \implies \exists w \in X. \nexists x \in W. wRx$

■ **Figure 4** Correspondence results in our HOLMS codebase.

### 3.3 Soundness and Consistency Proofs

Notice that the identification of appropriate classes for our logics provides per se – just because of Def. 8 – a proof of soundness for each axiomatic calculus w.r.t a corresponding class of *finite* relational frames. An analogous result for frames that are not necessarily finite is obtained because the characteristic classes are closed under the deduction rules of our axiomatic proof systems.<sup>9</sup> Soundness can then be stated and proven parametrically:

► **Theorem 9 (Soundness).**  $\varnothing$  *Let  $\mathcal{S}$  be a set of axiom schemas. Let  $\mathfrak{S}$  be the characteristic class of frames for  $\mathcal{S}$ . For every  $A \in \mathbf{Form}_{\square}$ , if  $\mathcal{S}.\emptyset \vdash A$  then  $\mathfrak{S} \models A$ .*

Soundness for any axiom system formalisable by our deducibility relation is then obtained by instantiating the parameters in `GEN_CHAR_VALID`  $\varnothing$  (which is the formal version of Theorem 9).

From soundness, we can now prove as usual that any modal system  $\mathcal{S}$  that is sound w.r.t. a non-empty class of frames  $\mathfrak{S}$  is also consistent. Our HOLMS codebase presents a proof of consistency `*_CONSISTENT` of  $* \in \{\text{K } \varnothing, \text{T } \varnothing, \text{K4 } \varnothing, \text{S4 } \varnothing, \text{B } \varnothing, \text{S5 } \varnothing, \text{GL } \varnothing\}$  by applying the specific soundness lemmas and proving an inhabitant of the respective class of appropriate frames.

<sup>7</sup> For practical reasons, we freely use such an equivalence in the code, even if in the following pages we will only refer to classes of finite characteristic frames.

<sup>8</sup> ITF is the frame of finite irreflexive-transitive frames: it is easy to prove that for every finite transitive frame, it is Noetherian iff it is irreflexive [25, § 4].

<sup>9</sup> The theorem of HOL Light proving this fact goes under the name `CHAR_CAR`  $\varnothing$ .

## 4 Completeness

This section addresses the formalisation of a modular proof of semantic completeness for seven normal modal logics. For K, T, K4, S4, B, and S5 it would be possible to prove completeness w.r.t. their characteristic classes by employing the well-established canonical model method [23, § 4]. For incompleteness-related issues, however, we cannot directly follow that strategy for GL [25, § 6].

The work in [76, 77] formalised a Henkin-style proof for GL w.r.t. finite frames by hacking the strategy outlined by Boolos in [25, §5]. That strategy suits a wide range of normal modal logics, including all the seven systems considered here. We present now a refined formal proof of Boolos' results, clarifying the modularity of his strategy. By avoiding code duplication as far as possible, we can condense the resulting completeness statement by the following (ad hoc polymorphic) theorem:

► **Theorem 10 (Completeness).** *Let  $\mathcal{S}$  be a set of axiom schemas. Let  $\mathfrak{S}$  be the characteristic class of finite frames for  $\mathcal{S}$ . For every  $A \in \mathbf{Form}_{\square}$ , if  $\mathfrak{S} \models A$ , then  $\mathcal{S}.\emptyset \vdash A$ .*

We provide a proof sketch of this theorem, as here we are more interested in its application for the automation procedure we propose in Section 5. We refer the reader to Appendix C for further details.

### Proof Sketch

By contraposition, we prove that, given a generic modal formula  $A$ : *If  $\mathcal{S}.\emptyset \not\vdash A$  then  $\mathfrak{S} \not\models A$ .*

This means that for each set of axioms  $\mathcal{S}$  and for each modal formula  $A$  such that  $\mathcal{S}.\emptyset \not\vdash A$ , we must find a countermodel  $\mathcal{M}_A^{\mathcal{S}}$  inhabiting  $\mathfrak{S}$ , and a “counterworld”  $w_A^{\mathcal{S}}$  inhabiting  $\mathcal{M}_A^{\mathcal{S}}$  such that  $w_A^{\mathcal{S}} \not\models_{\mathcal{M}_A^{\mathcal{S}}} A$ . To do so, we formalise the argument in [25, §5] and implement the following key-strategy.

#### Parametric part of the proof:

1. We identify a parametric notion of (counter)model  $\mathcal{M}_A^{\mathcal{S}}$  in  $\mathfrak{S}$  having: maximal consistent lists of modal formulas as possible worlds; an accessibility relation that verifies two given constraints; a valuation such that  $V_A^{\mathcal{S}}(q, w)$  iff  $q$  is a subformula of  $A$  and  $q \in w$ .  
(GEN\_STANDARD\_MODEL  $\checkmark$ )
2. We prove a *general truth-lemma* independent from the considered  $\mathcal{S}$  and  $A$ . This step allows the reduction of the notion of *forcing* ( $w \Vdash_{\mathcal{M}_A^{\mathcal{S}}} B$ ) to *membership* ( $B \in w$ ).  
(GEN\_TRUTH\_LEMMA  $\checkmark$ )
3. We identify a “counterworld”  $X$  in  $\mathcal{M}_A^{\mathcal{S}}$  such that  $A \notin X$  and thus  $X \not\models_{\mathcal{M}_A^{\mathcal{S}}} A$ .  
(GEN\_COUNTERMODEL\_ALT  $\checkmark$ )

#### Ad hoc polymorphic part of the proof:

- I. We identify for each system  $\mathcal{S}$  of its specific countermodel  $\mathcal{M}_A^{\mathcal{S}}$ , and in particular of its accessibility relation  $Rel_A^{\mathcal{S}}$ ;
- II. We verify that this model satisfies the two requirements for  $Rel_A^{\mathcal{S}}$ .

► **Remark 11.** For step 2, we can work on worlds that are simply sets of formulas. In HOLMS, we use worlds that are lists of formulas without repetitions for purely practical reasons: our implementation choice leverages pre-proven results for finite lists and a useful structural recursion rule, as in [76]. Because of that choice, the formal proof of Theorem 10 establishes completeness w.r.t. frames sitting on the type-theoretic domain of formula lists. We obtain the more general result by formally proving a type-theoretic version of the bisimulation

invariance lemma ([BISIMILAR\\_VALID](#)  $\square$ )<sup>10</sup> which allows us to prove completeness w.r.t. any infinite domain – including formula sets commonly used in this kind of constructions – thanks also to the auxiliary lemma [GEN\\_LEMMA\\_FOR\\_GEN\\_COMPLETENESS](#)  $\square$ .

## 5 Shallow Embedding of Root-First Proof-Search

We now leverage the formalised completeness theorems from Section 4 to support our automated reasoning procedure for the seven modal logics under analysis and showcase some examples of its concrete use in HOL Light.

The procedure is implemented by the `HOLMS_TAC`  $\square$  tactic, along with the corresponding rule `HOLMS_RULE`  $\square$ . `HOLMS_TAC` performs a bounded root-first proof-search for the input formula in the labelled sequent calculus associated to each logic we consider here. If the proof-search fails, the tactic generates a certified relational countermodel to the input formula.

Before describing the implementation of our decision procedure (Section 5.2) and some examples of its application (Section 5.3), we provide a concise overview (Section 5.1) of labelled sequent calculi.

### 5.1 Labelled Sequent Calculi

Labelled sequent calculi for modal logics have been extensively developed since [86]. They enrich the language of sequent calculi with syntactic elements that directly represent semantic components. In particular, labelled formulas have shape  $x : A$  – forcing atoms, to represent within the syntax the forcing relation  $\mathcal{M}, x \models A$  – or  $xRy$  – relational atoms, encoding accessibility relations between worlds. The rules for logic operators – such as those for the system G3K in Figure 5 – are designed to formalise the corresponding forcing conditions within the sequent calculus itself.

---

<b>Initial sequents:</b>	
$w : p, \Gamma \Rightarrow \Delta, w : p$	
<b>Propositional rules:</b>	
$\frac{}{w : \perp, \Gamma \Rightarrow \Delta} \text{L}\perp$	
$\frac{\Gamma \Rightarrow \Delta, w : A}{w : \neg A, \Gamma \Rightarrow \Delta} \text{L}\neg$	$\frac{w : A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, w : \neg A} \text{R}\neg$
$\frac{w : A, w : B, \Gamma \Rightarrow \Delta}{w : A \wedge B, \Gamma \Rightarrow \Delta} \text{L}\wedge$	$\frac{\Gamma \Rightarrow \Delta, w : A \quad \Gamma \Rightarrow \Delta, w : B}{\Gamma \Rightarrow \Delta, w : A \wedge B} \text{R}\wedge$
$\frac{w : A, \Gamma \Rightarrow \Delta \quad w : B, \Gamma \Rightarrow \Delta}{w : A \vee B, \Gamma \Rightarrow \Delta} \text{L}\vee$	$\frac{\Gamma \Rightarrow \Delta, w : A, w : B}{\Gamma \Rightarrow \Delta, w : A \vee B} \text{R}\vee$
$\frac{\Gamma \Rightarrow \Delta, w : A \quad w : B, \Gamma \Rightarrow \Delta}{w : A \rightarrow B, \Gamma \Rightarrow \Delta} \text{L}\rightarrow$	$\frac{w : A, \Gamma \Rightarrow \Delta, w : B}{\Gamma \Rightarrow \Delta, w : A \rightarrow B} \text{R}\rightarrow$
<b>Modal rules:</b>	
$\frac{x : A, wRx, w : \Box A, \Gamma \Rightarrow \Delta}{wRx, w : \Box A, \Gamma \Rightarrow \Delta} \text{L}\Box$	$\frac{wRx, \Gamma \Rightarrow \Delta, x : A}{\Gamma \Rightarrow \Delta, w : \Box A} \text{R}\Box \text{ } x \text{ does not occur in } \Gamma, \Delta$

---

■ **Figure 5** Labelled Sequent Calculus G3K associated to the modal system K.

<sup>10</sup>Stating that validity of modal formulas is preserved by any bisimulation; refer to e.g. [112].

As for its axiomatic counterpart K, G3K is the base for a modular family of calculi, and can be extended with additional rules to internalise the characteristic relational properties of other modal logics. Figure 6 displays the rules corresponding to the characteristic properties of the seven logics under analysis.

	Characteristic Property	Corresponding Rule
T	Reflexive: $\forall w(wRw)$	$\frac{wRw, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$ Ref
4	Transitive: $\forall w, x, z(wRx \wedge xRz \Rightarrow wRz)$	$\frac{wRz, wRx, xRz, \Gamma \Rightarrow \Delta}{wRx, xRz, \Gamma \Rightarrow \Delta}$ Trans
5	Euclidean: $\forall w, x, y(wRx \wedge wRy \Rightarrow yRx)$	$\frac{yRx, wRx, wRy, \Gamma \Rightarrow \Delta}{wRx, wRy, \Gamma \Rightarrow \Delta}$ Eucl
B	Symmetric: $\forall w, x(wRx \Rightarrow xRw)$	$\frac{xRw, wRx, \Gamma \Rightarrow \Delta}{wRx, \Gamma \Rightarrow \Delta}$ Sym
	Irreflexive: $\forall w(\neg wRw)$	$\overline{wRw, \Gamma \Rightarrow \Delta}$ Irref

■ **Figure 6** Geometric rules correspondent to relational characteristic properties.

By extending G3K with the appropriate rules, we obtain labelled sequent calculi for the six systems within the cube we discuss here. As detailed in [77, §6], the labelled calculus for GL additionally requires modifying the  $R\Box$  rule. Specifically, the resulting calculus, denoted by  $G3KGL := G3K - R\Box + R\Box_{L\delta b} + \text{Trans} + \text{Irrefl}$ , consists of the initial sequents, the propositional rules and  $L\Box$ , together with transitive and irreflexive inference rules and the new modal rule  $R\Box_{L\delta b}$  from [86]:

$$\frac{wRx, x : \Box A, \Gamma \Rightarrow \Delta, x : A}{\Gamma \Rightarrow \Delta, w : \Box A} R\Box_{L\delta b} \quad x \text{ does not occur in } \Gamma, \Delta.$$

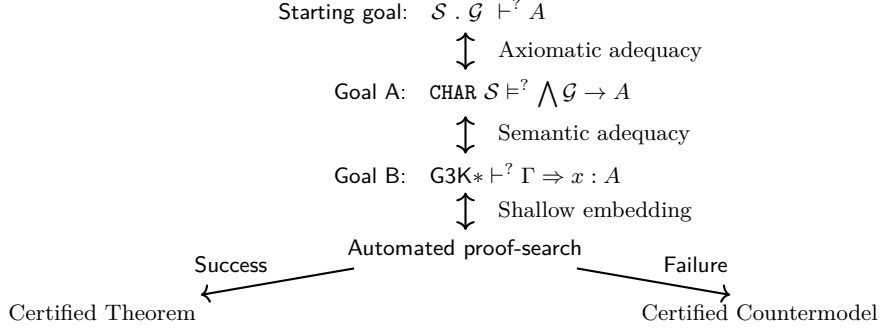
These calculi offer a systematic and structured approach to modal proof theory, ensuring all the desiderata for sequent calculi and hence the possibility to mechanise root-first proof-search as a decision procedure [88].

## 5.2 Implementation in HOL Light

To decide whether a formula is provable in a given modal system, the HOLMS framework leverages all the discussed characterisations: axiomatic, semantic and sequent-based, as shown in Figure 7.

To begin, the user provides a formula  $A$  of the modal language, a set of hypothesis  $\mathcal{G}$  and specifies one of the seven logics  $* \in \{K, T, K4, S4, B, S5, GL\}$ , each identified by a specific set  $\mathcal{S}$  of axiom schemas (**Starting Goal**). HOLMS\_TAC first translates the provability question into the semantical one of whether  $A$  is valid in the class of Kripke frames characteristic to  $\mathcal{S}$  (**Goal A**). This step is justified by the mechanised adequacy theorems. Through the shallow embedding of labelled sequent calculi and their internalisation of Kripke semantics, the problem is reframed once more: checking whether we can formally derive  $A$  in the labelled calculus associated to  $*$  (**Goal B**). For each of these calculi, we automate a root-first proof-search strategy – known to provide a decision algorithm for  $*$  [88]. If the search succeeds by reaching initial sequents, HOLMS\_TAC produces (a shallowly embedded) derivation of  $A$ , that is certified by HOL Light as a fully trusted proof of the validity of  $A$  (**Success**

branch). If it fails, `HOLMS_BUILD_COUNTERMODEL` extracts from HOL Light goal-stack a candidate countermodel;<sup>11</sup> finally by `HOLMS_CERTIFY_COUNTERMODEL`, HOL Light verifies that the candidate falsifies the input formula  $A$  at its root (*Failure* branch).



■ **Figure 7** Schema of the automated decision procedure implemented by `HOLMS_RULE` and relatives.

The following paragraphs describe how the shallow embedding, the `HOLMS_TAC`, and the countermodel extraction have been implemented, along with future improvements.

**Embedding Labelled Sequent Calculi.** Traditional approaches to modal logic decision procedures often rely on (counter)model finding algorithms that, while computationally efficient, provide no guarantee of correctness beyond the implementation’s reliability [32, 7].

Our method takes a radically different approach by embedding the entire countermodel extraction from a failed proof-search in labelled sequent calculi within HOL Light, thereby providing mathematical certification for every generated countermodel. This certification process transforms what would otherwise be a purely computational artefact into a machine-verified mathematical theorem. There are two possible approaches to achieving this goal, based on *deep* or *shallow embeddings*, respectively.

The *first* involves formalising a dedicated derivability predicate for each labelled sequent calculus corresponding to the logics under consideration – similarly to what was done for axiomatic calculi in Section 2.2. This would yield an additional deep embedding of the various presentations of these logics. One would formalise all structural and logical rules of the calculi, and subsequently reason about the provability of a given formula by working directly within the formal derivability predicate. This approach affords fine-grained control over the representation, but requires redeveloping the associated meta-theory from scratch.

The *second* exploits HOL Light’s native deductive engine – namely, its goal-stack mechanism – building upon our existing formalisation of the axiomatic and semantic presentations of the seven logics under exam. This second route would leverage the certified adequacy between these two presentations, as established by Theorems 9 and 10.

Following our [77], we opted for this second route. We use the previously defined datatype for modal formulas and the definition of the predicate `holds` from Listing 1 to capture the notion of forcing atoms in sequent calculi. This way, a generic labelled sequent  $\mathcal{R}, \Gamma \Rightarrow \Delta$  –

<sup>11</sup> This is analogous to the countermodel extraction from failed proof-search in (enriched) sequent calculi, as documented by wide proof-theoretic literature [115, 97, 88], which provides a more explicit approach than Henkin’s proof of completeness introduced, discussed, and analysed first for labelled sequent calculi in [87].

where  $\mathcal{R}$  is a multiset of relational atoms, and  $\Gamma, \Delta$  are multisets of forcing atoms – is encoded as a HOL Light goal whose assumptions are  $\bigwedge_{(xRy) \in \mathcal{R}} R\ x\ y \wedge \bigwedge_{(x:A) \in \Gamma} \text{holds}\ WR\ V\ A\ x$  and whose conclusion is the disjunction  $\bigvee_{(y:B) \in \Delta} \text{holds}\ WR\ V\ B\ y$ .<sup>12</sup>

Classical propositional connectives in the object logic are handled *shallowly* by standard HOL tactics – documented in [67]: `IMP_TAC`, `DISJ_TAC`, `CONJ_TAC`, `ASM_REWRITE_TAC`, and so on. Thus we can re-use the meta-logic’s proof machinery directly, without any bespoke encoding of boolean operators. Actually, *each* inference rule in the labelled sequent calculus becomes a HOL Light tactic, including the rules for the modal operator(s). For instance, the rule  $R\Box$  is realised by a tactic `BOX_RIGHT_THEN`  $\varnothing$  providing flexibility to handle different variants of the box-right rule depending on the specific position of the “nested antecedents” of the formalised forcing condition for the necessity operator.

A key advantage of our hybrid approach lies in the combination of deep and shallow embeddings. Deep embedding of both syntax and semantics enables the formulation and proof of meta-theorems by structural induction on the involved datatypes. At the same time, proof-search benefits from shallow embedding, leveraging HOL Light’s built-in automation for reasoning about logical connectives such as conjunction, disjunction, and implication. Notably, the structural rules – weakening, contraction, and cut – are not introduced axiomatically, but emerge naturally from HOL Light’s logical engine.

**Design of the `HOLMS_TAC`.** Our tactic `HOLMS_TAC`  $\varnothing$  implements the root-first proof-search in embedded sequent calculi by means of the following loop:

**Step 1 (Normalisation):** Put the goal’s conclusion in normal form;

This step expands the `holds` predicate for each classical operator (e.g., rewriting `holds WR V (p && q) w` as `holds WR V p w /\ holds WR V q w`), converts the result into conjunctive normal form, and splits conjunctions into subgoals. The literals in the conclusion are then reordered to facilitate subsequent steps: negative `holds` literals (used in Step 2), followed by boxed positives (used in Step 3), and then unboxed positives.

**Step 2 (Saturation):** Apply a left rule of the labelled sequent calculus and saturate relational or forcing atoms when assumed;

This step enlarges the assumption list by exhaustively applying propositional left rules, the  $L\Box$  rule, and the rules governing the accessibility relation. For example, in a modal logic with transitive frames, if  $R\ x\ y$  is already assumed and  $R\ y\ z$  is added, then  $R\ x\ z$  is inferred as additional assumption.

**Step 3-4 (Tactics composition and dispatch)**

3. Apply classical right propositional rules;
4. Apply the right rule for the box,  $R\Box$  or  $R\Box_{L\delta b}$ .

These steps assemble, from low-level building blocks, a specialised tactic for each logic, based on its semantic properties from Figure 6 – and the modified modal rule on the right for GL. As  $R\Box_{L\delta b}$  increases the number of worlds in the context, is used only when no other rule is applicable. A dispatching mechanism automatically selects the appropriate tactic based on the goal’s structure, enabling logic-specific reasoning without manual intervention.

<sup>12</sup>Less formally, commas on the left of the sequent symbol translate to HOL Light’s conjunction  $\wedge$ , while commas on the right become the HOL Light’s disjunction  $\vee$ .

The end result is that `HOLMS_TAC` – together with `HOLMS_BUILD_COUNTERMODEL` – either confirms the validity of  $A$  through a(n implicit) formal derivation certified by HOL Light, or provides a countermodel showing its invalidity. It generalises the core mechanism for GL in [77] to the additional systems considered in the current version of HOLMS. Despite the internal differences among the labelled calculi for the seven logics under analysis, the same overarching structure governs the basic proof-search mechanism and the generation of a potential countermodel. Thus, `HOLMS_TAC` reflects a modular design making it straightforward to maintain and extend the framework to other normal modal systems.

**Extraction of Certified Countermodels.** `HOLMS_BUILD_COUNTERMODEL`  $\curvearrowright$  normalises and simplifies the countermodel representation in the goal-stack – potentially expanding definitions or applying basic logical simplifications – to extract the effective relational structure.

From this countermodel representation, it is conceptually straightforward to derive a formal proof that the constructed frame satisfies the required properties of the given modal logic and falsifies the input formula. This process is fully implemented in our framework, with the core handled by the `CERTIFY_COUNTERMODEL_TAC`  $\curvearrowright$  tactic. Although the overall approach is simple in principle, some care is needed in practice, as the necessary checks can be computationally expensive. For instance, when working with transitive frames, one must verify that  $R\ x\ y \wedge R\ y\ z \rightarrow R\ x\ z$  holds for every triple of worlds  $x, y, z$  in the countermodel.

This way, the semantic evaluation is reduced to purely computational verification using the automated reasoning tools of HOL Light, ultimately confirming that the original formula fails to hold at the root of the model generated from the failed automated proof-search.

**Usability and Future Improvements.** Overall, this integration of sequent-based proof-search with modularised formalisation of modal systems underscores the versatility of the framework proposed by HOLMS. Further extensions of the library and refinements of `HOLMS_TAC` will show them as the core engine of a unified tool for reasoning about a broad range of normal modal logics within HOL Light, and general-purpose proof assistant widely conceived.

► **Remark 12 (An aside on decidability).** The modal systems discussed here are decidable, and the formalised completeness proofs can be used to implement model-checking procedures by limiting the size of potential countermodels based on the formula’s structure. The proof-search mechanised by `HOLMS_TAC` offers a proof-theoretic decision method for most of them, halting either by confirming the validity of the formula or providing a finite countermodel, based on [86]. For K4, the algorithm may terminate because of the imposition of a bound on the size of the generated countermodel, to prevent interactions between transitivity closure and logical rules leading to loops. The generated countermodel, however, is not certified by the current version of the tactic. Issues dealing with infinite loops in labelled calculi are well-documented, and solutions ensuring termination [49] can be integrated into future HOLMS versions.

### 5.3 Automated Proof-Search at Work

We now showcase some examples of interaction with our proof-search and countermodel generation algorithms. In Listing 3, we see (in order) that:

- `HOLMS_RULE` verifies that the formula  $\Box\Diamond A \rightarrow \Box B$  is a theorem in the logic GL; and
  - `HOLMS_BUILD_COUNTERMODEL` produces a countermodel to the  $\Box(\Box A \rightarrow A) \rightarrow \Box A$  for the logic T;
- and `CERTIFY_COUNTERMODEL_TAC` certify this countermodel.

Observe that the formula  $\Box\Diamond A \rightarrow \Box B$  is a somewhat unusual theorem for a modal logic, but it yields significant insight when interpreted arithmetically. By mapping  $\Box$  to the arithmetical provability predicate  $\text{Prov}$  and  $\Diamond$  to the arithmetical consistency predicate  $\text{Cons}$ , the formula asserts a strong result: if a theory  $T$  proves the relative consistency of  $A$ , it can effectively prove that *any* sentence  $B$  is provable ( $\text{Prov}(\text{Cons}\ulcorner A^*\urcorner) \rightarrow \text{Prov}\ulcorner B^*\urcorner$ ). This leads to the explosion of the very  $T$  – essentially an internalised version of a well-known consequence of the incompleteness phenomenon [108].

■ **Listing 3** Interaction with the HOLMS commands for proof-search and countermodel generation.

```
# HOLMS_RULE '[GL_AX . {} |~ Box Diam a --> Box b]';;
val it : thm = |- [GL_AX . {} |~ Box Diam a --> Box b]

# let tm = '[T_AX . {} |~ Box (Box a --> a) --> Box a]';;
val tm : term = '[T_AX . {} |~ Box (Box a --> a) --> Box a]'

# let ctm = HOLMS_BUILD_COUNTERMODEL tm;;
val ctm : term =
  'y IN W /\ y' IN W /\ w IN W /\
   R y y /\ R y' y' /\ R w w /\ R y y' /\   R w y /\
   ~holds (W,R) V a y /\ ~holds (W,R) V a y' /\
   holds (W,R) V (Box (Box a --> a)) w'

# T_HOLMS_CERTIFY_COUNTERMODEL ctm tm;;
val it : thm = |- ~(! a. RF |= Box (Box a --> a) --> Box a)
```

Notice that in each interaction with the system, we obtain a new HOL Light theorem, certifying provability of the formula, or refutability of it in the stored countermodel. Further examples of the application of these procedures are collected in the file `tests.ml` within the library.

## 6 Conclusions

We introduced an extension of the HOL Light library HOLMS to support principled modal reasoning. We formalised additional normal modal logics within HOLMS along three levels: axiomatic calculi; relational semantics; and labelled sequent calculi. We proved a general completeness theorem for normal modal logics, which we instantiated for K, T, B, K4, S4, S5, and GL. We then implemented root-first proof-search in the corresponding labelled sequent calculi using HOL Light’s goal-stack, leveraging our semantic formalisation and proof-theoretic insights. These procedures illustrate a broader paradigm: developing specialised provers within general-purpose theorem provers. Thanks to the simplicity of HOL Light’s logic engine, our methods are portable to other proof assistants. To measure the modularity of our codebase, we systematically adopted a distinction between parametric/ad hoc polymorphic results, which may serve as a base for a more transparent analysis of the formalisation and of the corresponding reasoning in ordinary mathematical practice.

**Related Work.** The problem of mechanising modal systems – i.e. developing formal and computational tools to represent, analyse, and manipulate them – has given rise to a rich and diverse body of scientific work, including both theoretical contributions and more practice-oriented implementations.

Notable examples include the automation within Prolog of the decision procedures for classical and intuitionistic modal system [57, 54], for Lewis’ conditional logics [52, 51], and for further non-normal logics [35, 34], which define, for each system under investigation,

a program implementing proof-search by a predicate built according to the rules of the corresponding sequent calculus, following the *lean deduction* methodology of [10]. Work using the Rocq proof assistant has mainly focused on formalisation of intuitionistic and classical modal systems in constructive type theory both concerning completeness results (and refinements of classical proofs), and proof systems and analysis [44, 43, 42], [60, 45, 104, 40], [61, 63, 62], [71, 103].

Another line of research, known as the *translational approach*, has embedded various modal and hybrid logics into simple type theory [65, 11, 17, 13], which can then be mechanised within automated theorem provers in the HOL family [14, 15, 110] extending the approach already suggested by Harrison for LTL in HOL Light [66, §20.4]. We refer the reader to the methodological paper [12] and its extensive bibliography for further details, and to [16, 120, 18, 109, 73] for examples of application to philosophy and normative contexts.

The research conducted in [59] introduces a highly effective hybrid methodology that integrates SAT solvers, modal clause-learning, and tableaux techniques for modal systems. This prover focuses on the minimal modal logic K along with its extensions T and S4. The current version of the tool does not provide a proof or countermodel for the given input formula but, with a few minor modifications, it could be adapted to gain these functionalities. A specialised tableaux-based theorem prover for GL is detailed in [58], which also addresses various efficiency-related aspects of the proof system.

An implementation of deductive systems in HOL Light is presented in [92] by a deep embedding of sequent calculi and tailoring specific HOL Light tactics to perform an interactive proof-search in those proof systems. Tableaux for modal logics K, T, and S4 are formalised within the Lean theorem prover in [119], allowing users to get both a closed tableau or a countermodel to the input formula, depending on the correct case. The recent work presented in [26, 27] formalises in the Agda theorem prover extended sequent calculi for several normal modal logics based on the proof-theoretic results discussed in [79].

Additional independent works individually present relevant aspects – such as the natural extensibility of decision procedures [83, 84, 85, 93, 69, 82] and the analogous generality of the formal proofs of completeness via canonical model constructions [47, 48]. By merging those aspects in a uniform framework, we have managed to mechanise in a modular, uniform, and general way six central systems of the modal cube – already captured independently by those works – and, moreover, we substantially improved the performance of proof-search from [77] for GL – one of the most important systems beyond the modal cube.

**Future Work.** A basic task is to extend HOLMS to other logics, such as the entire modal cube [102] and further modal systems for provability and interpretability [80, 95]. Another goal is to support multimodal languages by indexing modal operators – treating current monomodal systems as a special case – thereby significantly enhancing the expressiveness of the framework and broadening its potential applications in system verification [107, 106], [94, 33], [41], [2]. We also aim to integrate deep embeddings of enriched sequent calculi and their metatheory. This would enable the use of HOLMS for interesting proof-theoretic investigations and facilitate its integration with external derivation-search tools. Another direction is generalising our codebase to non-normal modal logics (with neighbourhood semantics) [36, 56, 37] and intuitionistic/constructive modalities [39, 38, 55], which are increasingly relevant in computer science. This would require adapting our methods and relaxing the current parametric approach to semantic adequacy. In terms of performance, an extension to cover concurrent proof-search – as in [29, 30] – is another interesting line for future research.

## References

- 1 Oskar Abrahamsson, Magnus O. Myreen, Ramana Kumar, and Thomas Sewell. Candle: A Verified Implementation of HOL Light. In June Andronick and Leonardo de Moura, editors, *13th International Conference on Interactive Theorem Proving (ITP 2022)*, volume 237 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:17, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITP.2022.3.
- 2 Matteo Acclavio, Fabrizio Montesi, and Marco Peressotti. On propositional dynamic logic and concurrency. *arXiv preprint arXiv:2403.18508*, 2024. doi:10.48550/arXiv.2403.18508.
- 3 Luca Aceto, Anna Ingólfssdóttir, Kim Guldstrand Larsen, and Jiri Srba. *Reactive systems: modelling, specification and verification*. Cambridge University Press, 2007.
- 4 Bahareh Afshari, Sebastian Enqvist, Graham E Leigh, Johannes Marti, and Yde Venema. Proof systems for two-way modal mu-calculus. *The Journal of Symbolic Logic*, pages 1–50, 2023.
- 5 Bahareh Afshari, Graham E Leigh, and Guillermo Menéndez Turata. A cyclic proof system for full computation tree logic. In *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*, volume 252 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2023.5.
- 6 Andrew W Appel, Paul-André Mellies, Christopher D Richards, and Jérôme Vouillon. A very modal model of a modern, major, general type system. In *Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 109–122, 2007. doi:10.1145/1190216.1190235.
- 7 David Basin, Cas Cremers, and Catherine Meadows. *Model Checking Security Protocols*, pages 727–762. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-319-10575-8\_22.
- 8 David A. Basin, Seán Matthews, and Luca Viganò. Labelled Propositional Modal Logics: Theory and Practice. *J. Log. Comput.*, 7(6):685–717, 1997. doi:10.1093/LOGCOM/7.6.685.
- 9 Andrej Bauer and Matija Pretnar. Programming with algebraic effects and handlers. *Journal of logical and algebraic methods in programming*, 84(1):108–123, 2015. doi:10.1016/J.JLAMP.2014.02.001.
- 10 Bernhard Beckert and Joachim Posegga. leanTAP: Lean Tableau-based Deduction. *J. Autom. Reason.*, 15(3):339–358, 1995. doi:10.1007/BF00881804.
- 11 Christoph Benzmueller and Lawrence C. Paulson. Multimodal and intuitionistic logics in simple type theory. *Logic Journal of the IGPL*, 18(6):881–892, January 2010. doi:10.1093/jigpal/jzp080.
- 12 Christoph Benzmüller. Universal (meta-)logical reasoning: Recent successes. *Sci. Comput. Program.*, 172:48–62, 2019. doi:10.1016/J.SCICO.2018.10.008.
- 13 Christoph Benzmüller. Faithful logic embeddings in HOL - A recipe to have it all: deep and shallow, automated and interactive, heavy and light, proofs and counterexamples, meta and object level. *CoRR*, abs/2502.19311, 2025. doi:10.48550/arXiv.2502.19311.
- 14 Christoph Benzmüller, Maximilian Claus, and Nik Sultana. Systematic Verification of the Modal Logic Cube in Isabelle/HOL. In Cezary Kaliszyk and Andrei Paskevich, editors, *Proceedings Fourth Workshop on Proof eXchange for Theorem Proving, PxTP 2015, Berlin, Germany, August 2-3, 2015*, volume 186 of *EPTCS*, pages 27–41, 2015. doi:10.4204/EPTCS.186.5.
- 15 Christoph Benzmüller and Bruno Woltzenlogel Paleo. Higher-Order Modal Logics: Automation and Applications. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 32–74. Springer, 2015. doi:10.1007/978-3-319-21768-0\_2.

- 16 Christoph Benzmüller, Xavier Parent, and Leendert W. N. van der Torre. Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. *Artif. Intell.*, 287:103348, 2020. doi:10.1016/J.ARTINT.2020.103348.
- 17 Christoph Benzmüller and Lawrence C. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis*, 7(1):7–20, 2013. doi:10.1007/S11787-012-0052-Y.
- 18 Christoph Benzmüller and Dana Scott. Notes on Gödel’s and Scott’s variants of the ontological argument. *Monatshefte für Mathematik*, pages 1–43, 2025.
- 19 Antonella Bilotta, Marco Maggesi, and Cosimo Perini Brogi. HOLMS: A HOL Light Library for Modal Systems. Software, swbId: `swb:1:dir:36edca29f53f3fdd0a7fc675702990743eda4c8d` (visited on 2026-02-02). URL: <https://github.com/HOLMS-lib/HOLMS>, doi:10.4230/artifacts.25484.
- 20 Antonella Bilotta, Marco Maggesi, and Cosimo Perini Brogi. A Modular Proof of Semantic Completeness for Normal Systems beyond the Modal Cube, Formalised in HOLMS. In Luca Moscardelli and Francesca Scozzari, editors, *Proceedings of the 26th Italian Conference on Theoretical Computer Science, Pescara, Italy, September 10-12, 2025*, volume 4039 of *CEUR Workshop Proceedings*, pages 154–162. CEUR-WS.org, 2025. URL: <https://ceur-ws.org/Vol-4039/paper10.pdf>.
- 21 Antonella Bilotta, Marco Maggesi, and Cosimo Perini Brogi. Webpage of HOLMS: A HOL Light Library for Modal Systems, 2025. Software (visited on 2026-02-02). URL: <https://holms-lib.github.io/>, doi:10.4230/artifacts.25485.
- 22 Antonella Bilotta, Marco Maggesi, Cosimo Perini Brogi, and Leonardo Quartini. Growing HOLMS, a HOL Light Library for Modal Systems. In Daniele Porello, Cosimo Vinci, and Matteo Zavatteri, editors, *Short Paper Proceedings of the 6th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, OVERLAY 2024, Bolzano, Italy, November 28-29, 2024*, volume 3904 of *CEUR Workshop Proceedings*, pages 41–48. CEUR-WS.org, 2024. URL: <https://ceur-ws.org/Vol-3904/paper5.pdf>.
- 23 Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001. doi:10.1017/CB09781107050884.
- 24 Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter, editors. *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*. North-Holland, 2007. URL: <https://www.sciencedirect.com/bookseries/studies-in-logic-and-practical-reasoning/vol/3/suppl/C>.
- 25 George Boolos. *The logic of provability*. Cambridge University Press, 1995.
- 26 Riccardo Borsetto and Margherita Zorzi. An Agda Implementation of the Modal Logic S4.2: First Investigations. In Luca Moscardelli and Francesca Scozzari, editors, *Proceedings of the 26th Italian Conference on Theoretical Computer Science, Pescara, Italy, September 10-12, 2025*, volume 4039 of *CEUR Workshop Proceedings*, pages 163–168. CEUR-WS.org, 2025. URL: <https://ceur-ws.org/Vol-4039/paper17.pdf>.
- 27 Riccardo Borsetto and Margherita Zorzi. NAMOR: a New Agda Library for Modal Extended Sequents. In *Short Paper Proceedings of the 7th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, OVERLAY 2025, Bologna, Italy, October 26, 2025*, 2025. URL: <https://overlay.uniud.it/workshop/2025/papers/borsetto-zorzi.pdf>.
- 28 Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009. doi:10.1007/S00153-009-0137-3.
- 29 Julie Cailler. *Designing an Automated Concurrent Tableau-Based Theorem Prover for First-Order Logic. (Conception d’un prouveur automatique de théorèmes concurrent basé sur la méthode des tableaux pour la logique du premier ordre)*. PhD thesis, University of Montpellier, France, 2023. URL: <https://tel.archives-ouvertes.fr/tel-04526940>.
- 30 Julie Cailler, Johann Rosain, David Delahaye, Simon Robillard, and Hinde-Lilia Bouziane. Goéland: A Concurrent Tableau-Based Theorem Prover (System Description). In

- Jasmin Blanchette, Laura Kovács, and Dirk Pattinson, editors, *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, volume 13385 of *Lecture Notes in Computer Science*, pages 359–368. Springer, 2022. doi:10.1007/978-3-031-10769-6\_22.
- 31 Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Logics of Programs*, pages 52–71, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.
  - 32 E.M. Clarke, O. Grumberg, D. Peled, and D.A. Peled. *Model Checking*. The Cyber-Physical Systems Series. MIT Press, 1999.
  - 33 Gabriele Costa and Cosimo Perini Brogi. Toward dynamic epistemic verification of zero-knowledge protocols. In Gianni D’Angelo, Flaminia L. Luccio, and Francesco Palmieri, editors, *Proceedings of the 8th Italian Conference on Cyber Security (ITASEC 2024), Salerno, Italy, April 8-12, 2024*, volume 3731 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2024. URL: <https://ceur-ws.org/Vol-3731/paper25.pdf>.
  - 34 Tiziano Dalmonte, Sara Negri, Nicola Olivetti, and Gian Luca Pozzato. Theorem Proving for Non-normal Modal Logics. In *OVERLAY 2020*, Udine, Italy, September 2021. URL: <https://hal.archives-ouvertes.fr/hal-03159954>.
  - 35 Tiziano Dalmonte, Sara Negri, Nicola Olivetti, Gian Luca Pozzato, and Cyril Terrioux. PRONOM: A theorem prover for nonnormal modal logics. Available at <http://193.51.60.97:8000/pronom/>, 2025.
  - 36 Tiziano Dalmonte, Nicola Olivetti, and Sara Negri. Non-normal modal logics: Bi-neighbourhood semantics and its labelled calculi. In *Advances in Modal Logic 2018*, 2018.
  - 37 Tiziano Dalmonte, Nicola Olivetti, Gian Luca Pozzato, and Cyril Terrioux. HYPNO theorem proving with hypersequent calculi for non-normal modal logics. Available at <http://193.51.60.97:8000/HYPNO/>, 2025.
  - 38 Anupam Das and Sonia Marin. On intuitionistic diamonds (and lack thereof). In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 283–301. Springer, 2023. doi:10.1007/978-3-031-43513-3\_16.
  - 39 Anupam Das, Iris van der Giessen, and Sonia Marin. Intuitionistic Gödel-Löb logic, à la Simpson: Labelled systems and birelational semantics. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy*, volume 288 of *LIPICs*, pages 22:1–22:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.CSL.2024.22.
  - 40 Jim de Groot, Ian Shillito, and Ranald Clouston. Semantical analysis of intuitionistic modal logics between CK and IK. In *40th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2025, Singapore, June 23-26, 2025*, pages 169–182. IEEE, 2025. doi:10.1109/LICS65433.2025.00020.
  - 41 Simon Docherty and Reuben NS Rowe. A non-wellfounded, labelled proof system for propositional dynamic logic. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 335–352. Springer, 2019.
  - 42 Christian Doczkal and Joachim Bard. Completeness and decidability of converse PDL in the constructive type theory of Coq. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2018*, pages 42–52, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3167088.
  - 43 Christian Doczkal and Gert Smolka. Constructive formalization of hybrid logic with eventualities. In Jean-Pierre Jouannaud and Zhong Shao, editors, *Certified Programs and Proofs*, pages 5–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi:10.1007/978-3-642-25379-9\_3.
  - 44 Christian Doczkal and Gert Smolka. Completeness and decidability results for CTL in constructive type theory. *Journal of Automated Reasoning*, 56(3):343–365, 2016. doi:10.1007/S10817-016-9361-9.
  - 45 Hugo Férée, Iris van der Giessen, Sam van Gool, and Ian Shillito. Mechanised Uniform Interpolation for Modal Logics K, GL, and iSL. In Christoph Benzmüller, Marijn J. H. Heule,

- and Renate A. Schmidt, editors, *Automated Reasoning - 12th International Joint Conference, IJCAR 2024, Nancy, France, July 3-6, 2024, Proceedings, Part II*, volume 14740 of *Lecture Notes in Computer Science*, pages 43–60. Springer, 2024. doi:10.1007/978-3-031-63501-4\_3.
- 46 Melvin Fitting. *Proof methods for modal and intuitionistic logics*, volume 169. Springer Science & Business Media, 2013.
- 47 Asta Halkjær From. Formalized Soundness and Completeness of Epistemic Logic. In Alexandra Silva, Renata Wassermann, and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 27th International Workshop, WoLLIC 2021, Virtual Event, October 5-8, 2021, Proceedings*, volume 13038 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2021. doi:10.1007/978-3-030-88853-4\_1.
- 48 Asta Halkjær From. An Isabelle/HOL Framework for Synthetic Completeness Proofs. In *Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP '25*, pages 171–186, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3703595.3705882.
- 49 Deepak Garg, Valerio Genovese, and Sara Negri. Countermodels from sequent calculi in multi-modal logics. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 315–324. IEEE Computer Society, 2012. doi:10.1109/LICS.2012.42.
- 50 James Garson. Modal Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2024 edition, 2024.
- 51 Marianna Girlando, Bjoern Lellmann, Nicola Olivetti, Stefano Pesce, and Gian Luca Pozzato. Theorem proving for Lewis Logics of Counterfactual Reasoning. In *CILC 2020 - 35th Edition of the Italian Conference on Computational Logic, Rende / Virtual, Italy, October 2020*. URL: <https://hal.science/hal-03080670>.
- 52 Marianna Girlando, Bjoern Lellmann, Nicola Olivetti, Gian Luca Pozzato, and Quentin Vitalis. VINTE: an implementation of internal calculi for Lewis' logics of counterfactual reasoning. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 149–159. Springer, 2017.
- 53 Marianna Girlando, Björn Lellmann, Nicola Olivetti, Stefano Pesce, and Gian Luca Pozzato. Calculi, countermodel generation and theorem prover for strong logics of counterfactual reasoning. *Journal of Logic and Computation*, January 2022. exab084. doi:10.1093/logcom/exab084.
- 54 Marianna Girlando and Marianela Morales. MOILab: towards a labelled theorem prover for intuitionistic modal logics. working paper or preprint, December 2020. URL: <https://hal.science/hal-03048966>.
- 55 Marianna Girlando and Marianela Morales. MOILab: a prototype theorem prover for intuitionistic modal logic IK based on labelled sequents. Available at <https://www.mariannagirlando.com/MOILab.html>, 2025.
- 56 Marianna Girlando, Sara Negri, Nicola Olivetti, and Vincent Risch. Conditional beliefs: from neighbourhood semantics to sequent calculus. *The review of symbolic logic*, 11(4):736–779, 2018. doi:10.1017/S1755020318000023.
- 57 Marianna Girlando and Lutz Straßburger. Moin: A nested sequent theorem prover for intuitionistic modal logics (system description). In *International Joint Conference on Automated Reasoning*, pages 398–407. Springer, 2020. doi:10.1007/978-3-030-51054-1\_25.
- 58 Rajeev Goré and Jack Kelly. Automated proof search in Gödel-Löb provability logic. In *abstract, British Logic Colloquium*, 2007.
- 59 Rajeev Goré and Cormac Kikkert. CEGAR-Tableaux: improved modal satisfiability via modal clause-learning and SAT. In *Automated Reasoning with Analytic Tableaux and Related Methods: 30th International Conference, TABLEUX 2021, Birmingham, UK, September 6–9, 2021, Proceedings 30*, pages 74–91. Springer, 2021. doi:10.1007/978-3-030-86059-2\_5.

- 60 Rajeev Goré, Revantha Ramanayake, and Ian Shillito. Cut-elimination for provability logic by terminating proof-search: Formalised and deconstructed using coq. In Anupam Das and Sara Negri, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 299–313, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-86059-2\_18.
- 61 Christian Hagemeyer. Formalizing intuitionistic epistemic logic in Coq. BSc thesis, 2021.
- 62 Christian Hagemeyer and Dominik Kirst. Constructive and mechanised meta-theory of IEL and similar modal logics. *J. Log. Comput.*, 32(8):1585–1610, 2022. doi:10.1093/LOGCOM/EXAC068.
- 63 Christian Hagemeyer and Dominik Kirst. Constructive and mechanised meta-theory of intuitionistic epistemic logic. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2022, Deerfield Beach, FL, USA, January 10-13, 2022, Proceedings*, volume 13137 of *Lecture Notes in Computer Science*, pages 90–111. Springer, 2022. doi:10.1007/978-3-030-93100-1\_7.
- 64 Joseph Y. Halpern, Yoram Moses, and Mark R. Tuttle. A knowledge-based analysis of zero knowledge (preliminary report). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 132–147. ACM, 1988. doi:10.1145/62212.62224.
- 65 Moritz Hardt and Gert Smolka. Higher-order syntax and saturation algorithms for hybrid logic. In Patrick Blackburn, Thomas Bolander, Torben Braüner, Valeria de Paiva, and Jørgen Villadsen, editors, *Proceedings of the International Workshop on Hybrid Logic, HyLo@FLoC 2006, Seattle, WA, USA, August 11, 2006*, volume 174(6) of *Electronic Notes in Theoretical Computer Science*, pages 15–27. Elsevier, 2006. doi:10.1016/J.ENTCS.2006.11.023.
- 66 John Harrison. HOL Light tutorial. <http://www.cl.cam.ac.uk/~jrh13/hol-light/tutorial.pdf>, 2017.
- 67 John Harrison. The HOL Light system reference. <https://www.cl.cam.ac.uk/~jrh13/hol-light/reference.pdf>, 2023.
- 68 John Harrison. The HOL Light Theorem Prover. Official webpage: <https://hol-light.github.io/>, 2025.
- 69 Ullrich Hustadt, Fabio Papacchini, Cláudia Nalon, and Clare Dixon. Model construction for modal clauses. In *International Joint Conference on Automated Reasoning*, pages 3–23. Springer, 2024. doi:10.1007/978-3-031-63501-4\_1.
- 70 Andrzej Indrzejczak. Sequents and trees. *Studies in Universal Logic*, Birkhäuser, Cham, 2021.
- 71 Dominik Kirst and Ian Shillito. Completeness of first-order bi-intuitionistic logic. In Jörg Endrullis and Sylvain Schmitz, editors, *33rd EACSL Annual Conference on Computer Science Logic, CSL 2025, February 10-14, 2025, Amsterdam, Netherlands*, volume 326 of *LIPICs*, pages 40:1–40:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.CSL.2025.40.
- 72 Roman Kuznets. Communication modalities. In Ludovic Levy Patey, Elaine Pimentel, Lorenzo Galeotti, and Florin Manea, editors, *Twenty Years of Theoretical and Practical Synergies*, pages 60–71, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-64309-5\_6.
- 73 Lara Lawniczak and Christoph Benzmüller. Logical Modalities within the European AI Act: An Analysis, 2025.
- 74 Daan Leijen and Erik Meijer. Domain-Specific embedded compilers. In *2nd Conference on Domain-Specific Languages (DSL 99)*, Austin, TX, October 1999. USENIX Association. URL: <https://www.usenix.org/conference/dsl-99/domain-specific-embedded-compilers>.
- 75 Ioana Leustean and Bogdan Macovei. DELP: Dynamic epistemic logic for security protocols. *2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 275–282, 2021. URL: <https://api.semanticscholar.org/CorpusID:237491844>.
- 76 Marco Maggesi and Cosimo Perini Brogi. A Formal Proof of Modal Completeness for Provability Logic. In Liron Cohen and Cezary Kaliszzyk, editors, *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *Leibniz International Proceedings*

- in Informatics (LIPICs)*, pages 26:1–26:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.ITP.2021.26.
- 77 Marco Maggesi and Cosimo Perini Brogi. Mechanising Gödel-Löb Provability Logic in HOL Light. *J. Autom. Reason.*, 67(3):29, 2023. doi:10.1007/S10817-023-09677-Z.
- 78 Sonia Marin, Marianela Morales, and Lutz Straßburger. A fully labelled proof system for intuitionistic modal logics. *J. Log. Comput.*, 31(3):998–1022, 2021. doi:10.1093/LOGCOM/EXAB020.
- 79 Simone Martini, Andrea Masini, and Margherita Zorzi. Cut elimination for extended sequent calculi. *Bulletin of the Section of Logic*, 52(4):459–495, 2023. doi:10.18778/0138-0680.2023.22.
- 80 Luka Mikec. Satisfiability verifiers for certain modal logics concerned with provability. Available at <https://luka.doublebuffer.net/o/il/>; Accessed: December 2025, 2025. URL: [https://github.com/luka-mikec/provability\\_sat](https://github.com/luka-mikec/provability_sat).
- 81 Hiroshi Nakano. A modality for recursion. In *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332)*, pages 255–266. IEEE, 2000. doi:10.1109/LICS.2000.855774.
- 82 Cláudia Nalon. Efficient Theorem-Proving for Modal Logics. In Agata Ciabattoni, David Gabelaia, and Igor Sedlár, editors, *Advances in Modal Logic, AiML 2024, Prague, Czech Republic, August 19-23, 2024*, pages 13–16. College Publications, 2024.
- 83 Cláudia Nalon, Clare Dixon, and Ullrich Hustadt. Modal resolution: proofs, layers, and refinements. *ACM Transactions on Computational Logic (TOCL)*, 20(4):1–38, 2019. doi:10.1145/3331448.
- 84 Cláudia Nalon, Ullrich Hustadt, Fabio Papacchini, and Clare Dixon. Local reductions for the modal cube. In *International Joint Conference on Automated Reasoning*, pages 486–505. Springer International Publishing Cham, 2022. doi:10.1007/978-3-031-10769-6\_29.
- 85 Cláudia Nalon, Ullrich Hustadt, Fabio Papacchini, and Clare Dixon. Buy one get 14 free: Evaluating local reductions for modal logic. In Brigitte Pientka and Cesare Tinelli, editors, *Automated Deduction – CADE 29*, pages 382–400, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-38499-8\_22.
- 86 Sara Negri. Proof analysis in modal logic. *Journal of Philosophical Logic*, 34(5):507–544, 2005. doi:10.1007/S10992-005-2267-3.
- 87 Sara Negri. Kripke Completeness Revisited. In Giuseppe Primiero, editor, *Acts of Knowledge: History, Philosophy and Logic*, pages 233–266. College Publications, 2009.
- 88 Sara Negri. Proofs and countermodels in non-classical logics. *Logica Universalis*, 8(1):25–60, 2014. doi:10.1007/S11787-014-0097-1.
- 89 Sara Negri and Jan von Plato. *Structural proof theory*. Cambridge university press, 2008.
- 90 Sara Negri and Jan von Plato. *Proof analysis: a contribution to Hilbert’s last problem*. Cambridge University Press, 2011.
- 91 Nicola Olivetti and Gian Luca Pozzato. NESCOND: an implementation of nested sequent calculi for conditional logics. In *International Joint Conference on Automated Reasoning*, pages 511–518. Springer, 2014. doi:10.1007/978-3-319-08587-6\_39.
- 92 Petros Papapanagiotou and Jacques Fleuriot. Object-Level Reasoning with Logics Encoded in HOL Light. In *Fifteenth Workshop on Logical Frameworks and Meta-Languages: Theory and Practice*, pages 18–34. Open Publishing Association, 2021.
- 93 Dirk Pattinson, Nicola Olivetti, and Cláudia Nalon. Resolution calculi for non-normal modal logics. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 322–341. Springer, 2023. doi:10.1007/978-3-031-43513-3\_18.
- 94 Cosimo Perini Brogi, Rocco De Nicola, and Omar Inverso. Simpson’s proof systems for process verification: A fine-tuning (short paper). In Ugo de’Liguoro, Matteo Palazzo, and Luca Roversi, editors, *Proceedings of the 25th Italian Conference on Theoretical Computer Science, Torino, Italy, September 11-13, 2024*, volume 3811 of *CEUR Workshop Proceedings*, pages 292–299. CEUR-WS.org, 2024. URL: <https://ceur-ws.org/Vol-3811/paper050.pdf>.

- 95 Cosimo Perini Brogi, Sara Negri, and Nicola Olivetti. Modular Sequent Calculi for Interpretability Logics. *Rev. Symb. Log.*, 18(3):704–743, 2025. doi:10.1017/S1755020325100701.
- 96 Gordon Plotkin and Matija Pretnar. A logic for algebraic effects. In *2008 23rd Annual IEEE symposium on logic in computer science*, pages 118–129. IEEE, 2008. doi:10.1109/LICS.2008.45.
- 97 Francesca Poggiolesi. The method of tree-hypersequents for modal propositional logic. In *Towards mathematical philosophy*, pages 31–51. Springer, 2009. doi:10.1007/978-1-4020-9084-4\_3.
- 98 Francesca Poggiolesi. *Gentzen calculi for modal propositional logic*, volume 32. Springer Science & Business Media, 2010.
- 99 John Power. Generic models for computational effects. *Theoretical Computer Science*, 364(2):254–269, 2006. doi:10.1016/J.TCS.2006.08.006.
- 100 F. Rajaona, I. Boureau, R. Ramanujam, and S. Wesemeyer. Epistemic model checking for privacy. In *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*, pages 1–16, Los Alamitos, CA, USA, July 2024. IEEE Computer Society. doi:10.1109/CSF61375.2024.00020.
- 101 Jan Rooduijn, Dexter Kozen, and Alexandra Silva. A cyclic proof system for Guarded Kleene Algebra with Tests. In *International Joint Conference on Automated Reasoning*, pages 257–275. Springer, 2024. doi:10.1007/978-3-031-63501-4\_14.
- 102 Renate Schmidt. Advances in Modal Logic Tools. Last modified: 03 Mar 23; Accessed: December 2025, 2025. URL: <https://www.cs.man.ac.uk/~schmidt/tools/>.
- 103 Ian Shillito and Dominik Kirst. A mechanised and constructive reverse analysis of soundness and completeness of bi-intuitionistic logic. In Amin Timany, Dmitriy Traytel, Brigitte Pientka, and Sandrine Blazy, editors, *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2024, London, UK, January 15-16, 2024*, pages 218–229. ACM, 2024. doi:10.1145/3636501.3636957.
- 104 Ian Shillito, Iris van der Giessen, Rajeev Goré, and Rosalie Iemhoff. A New Calculus for Intuitionistic Strong Löb Logic: Strong Termination and Cut-Elimination, Formalised. In Revantha Ramanayake and Josef Urban, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 32nd International Conference, TABLEAUX 2023, Prague, Czech Republic, September 18-21, 2023, Proceedings*, volume 14278 of *Lecture Notes in Computer Science*, pages 73–93. Springer, 2023. doi:10.1007/978-3-031-43513-3\_5.
- 105 Alex K. Simpson. The Proof Theory and Semantics of Intuitionistic Modal Logic. *PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics*, 1994.
- 106 Alex K. Simpson. Compositionality via cut-elimination: Hennessy-milner logic for an arbitrary GSOS. In *Proceedings, 10th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, USA, June 26-29, 1995*, pages 420–430. IEEE Computer Society, 1995. doi:10.1109/LICS.1995.523276.
- 107 Alex K. Simpson. Sequent calculi for process verification: Hennessy-Milner logic for an arbitrary GSOS. *J. Log. Algebraic Methods Program.*, 60-61:287–322, 2004. doi:10.1016/J.JLAP.2004.03.004.
- 108 Craig Smorynski. The incompleteness theorems. In Jon Barwise, editor, *Handbook of mathematical logic*, pages 821–865. North-Holland, 1977.
- 109 Alexander Steen and Christoph Benzmüller. Challenges for Non-Classical Reasoning in Contemporary AI Applications. *Künstliche Intelligenz*, 38(1-2):7–16, 2024. doi:10.1007/S13218-024-00855-8.
- 110 Alexander Steen, Geoff Sutcliffe, and Christoph Benzmüller. Solving quantified modal logic problems by translation to classical logics. *Journal of Logic and Computation*, page exaf006, 2025.
- 111 Colin Stirling. *Modal and Temporal Properties of Processes*. Texts in Computer Science. Springer, 2001. doi:10.1007/978-1-4757-3550-5.
- 112 Colin Stirling. *Bisimulation and logic*, pages 173–196. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2011.

- 113 Christopher Strachey. Fundamental concepts in programming languages. *Higher-order and symbolic computation*, 13:11–49, 2000. doi:10.1023/A:1010000313106.
- 114 Alfred Tarski. *Undecidable Theories*. North-Holland Pub. Co., Amsterdam, 1968.
- 115 Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic proof theory, Second Edition*, volume 43 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 2000.
- 116 Johan Van Benthem. *Correspondence Theory*, pages 325–408. Springer Netherlands, Dordrecht, 2001. doi:10.1007/978-94-017-0454-0\_4.
- 117 Rineke (L.C.) Verbrugge. Provability Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2024 edition, 2024.
- 118 Luca Viganò. *Labelled non-classical logics*. Springer Science & Business Media, 2013.
- 119 Minchao Wu and Rajeev Goré. Verified Decision Procedures for Modal Logics. In John Harrison, John O’Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:19, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITP.2019.31.
- 120 Yiming Xu and Michael Norrish. Mechanised modal model theory. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning*, pages 518–533, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-51074-9\_30.

## A Source Code

This appendix documents the structure and purpose of the source files composing the HOLMS repository. The loaded files are designed in a modular way, and modularity of the code can be analysed – according to the distinction examined in Appendix B – in terms of *ad hoc* and *parametric* polymorphism. The codebase is organised into the following layers:

### Top-level Files.

- `README.md`  
provides a usage guide for the HOLMS library in its current state.
- `make.ml`  
loads all the other files in order.
- `misc.ml`  
collects auxiliary definitions and utility lemmas.

### Syntax, Semantics, Axiomatic Calculus.

- `modal.ml`  
deep-embeds the syntax of propositional modal logic and Kripke semantics.
- `calculus.ml`  
formalises a parametric notion of deducibility from hypotheses in an axiomatic system for normal modal logics and proves the deduction theorem.

### Correspondence Theory.

- `ad_hoc_correspondence.ml`  
defines axiom schemas and characteristic properties, and proves correspondence lemmas.
- `parametric_correspondence.ml`  
introduces the notions of frames, characteristic frames, and appropriate frames, and establishes their relationships. It also proves the general validity lemma and shows that the class of all frames is characteristic to K.

**Adequacy Theorems.**

- [conjlist.ml](#)  
defines and proves lemmas to treat iterated conjunctions of formulas.
- [consistent.ml](#) ↗  
provides a treatment of consistent sets of formulas.
- [gen\\_completeness.ml](#) ↗  
proves lemmas about completeness theorem.
- [\\*\\_completeness.ml](#)  
these files, one for each logic \* under analysis, exploit results from the previous file and correspondence theory to prove completeness theorems.

**Decision Procedures and Countermodels.**

- [gen\\_decid.ml](#)  
common part of all the decision procedures for all the implemented logics.
- [gen\\_countermodel.ml](#)  
common part of all the procedures for the countermodel generation for all the logics.
- [\\*\\_decid.ml](#)  
these files, one for each logic \* under analysis, develop decision procedures, countermodel construction and verification.
- [tests.ml](#) ↗  
contains further application tests and examples of the decision procedure for the seven modal systems under analysis (not loaded by `make.ml`).

**B Ad Hoc and Parametric Polymorphism**

As anticipated in the introduction, an important feature of our work is the precise study of the compositional nature of the theory of modal systems – as presented in Boolos’ textbook [25] – with respect to the various logics under consideration. We achieved that by measuring the modularity of the whole framework via a *polymorphic* treatment of the axiom schemas. More precisely, we mechanised definitions, theorems, and proofs positing those schemas as parameters.

We thus make a conceptual distinction between the two kinds of polymorphic constructions that we incur in our development:

**Parametric Polymorphism:** A construction is called *parametric* when a single, uniform implementation works for all values of a given parameter within the chosen logic or formal language.

**Ad hoc Polymorphism:** By contrast, a construction is *ad hoc* when it requires multiple implementations, each tailored to specific values of the parameter, with the appropriate implementation being selected at the meta-theoretical level – for example through overloading, instance resolution, or other external manual or automatic rule.

Historically and in most common usage, this distinction between parametric and ad hoc polymorphism has been formulated by Christopher Strachey in terms of types [113]. Nonetheless, we claim that this concept can be naturally extended beyond the type-theoretic context.<sup>13</sup> It naturally applies to any parameter on which a construction may depend – such

---

<sup>13</sup>Indeed, the more general coding discipline can be concretely implemented with static-typing techniques such as phantom types [74].

as structures, operations, or other contextual objects. In this broader and more general sense, the same criteria of uniformity vs. case-based definition remain meaningful and practically useful. Therefore, we have systematically adopted it in HOLMS development, as summarised in Figure 8.

<b>Syntax</b>		Parametric
<b>Semantics</b>		Parametric
<b>Correspondence Theory</b>	Concepts	Parametric
	Lemmas	Ad hoc
<b>Soundness</b>		Parametric
<b>Completeness</b>	“Standard” Model	Parametric
	Truth Lemma	Parametric
	Countermodel Lemma	Parametric
	“Standard” Relation	Ad hoc
	Identification of the “Standard” Model	Ad hoc
<b>Shallow Embedding</b>		Ad hoc
<b>Decision Procedure</b>		Ad hoc

■ **Figure 8** Measure of the modularity of our implemented results.

We remark that HOL Light does not provide explicit support for ad hoc polymorphism of the kind requested in this work. The categorisation we propose here, and the corresponding design embodied in our code, reflects a conceptual discipline that is independent of any concrete extra-logical feature of the underlying theorem prover.

## C Details of the Formalised Completeness Proof

This appendix provides a detailed account of the modular proof of completeness formalised in HOLMS, expanding on the proof sketch displayed in Section 4.

In particular, we want to prove the following claim:

▷ **Claim 13 (Completeness)**. Let  $\mathcal{S}$  be a set of axiom schemas. Let  $\mathfrak{S}$  be the characteristic class of *finite* frames for  $\mathcal{S}$ . For every  $A \in \mathbf{Form}_\square$ , if  $\mathfrak{S} \models A$ , then  $\mathcal{S}.\emptyset \vdash A$ .

We proceed by contraposition; we prove that, given a generic modal formula  $A$ :

▷ **Claim 14**. *If  $\mathcal{S}.\emptyset \not\vdash A$  then  $\mathfrak{S} \not\models A$*

This means that for each set of axioms  $\mathcal{S}$  and for each modal formula  $A$  such that  $\mathcal{S}.\emptyset \not\vdash A$ , we must find a countermodel  $\mathcal{M}_A^{\mathcal{S}}$  inhabiting  $\mathfrak{S}$ , and a “counterworld”  $w_A^{\mathcal{S}}$  inhabiting  $\mathcal{M}_A^{\mathcal{S}}$  such that  $w_A^{\mathcal{S}} \not\models_{\mathcal{M}_A^{\mathcal{S}}} A$ . To do so, we formalise the argument in [25, §5] and implement the following key-strategy.

### C.1 Fully Parametric Part of the Proof

Our file `gen_completeness.ml` <sup>14</sup> contains all the parametric results involved in the completeness proof. Some of those lines of code depend on further parametric notions formalised in `consistent.ml` <sup>14</sup>. We recap them by the following definitions and results.<sup>14</sup>

<sup>14</sup>The file `conjlist.ml` <sup>14</sup> contains additional parametric results for conjunctions on lists of formulas, which have a technical nature and can be omitted from here.

► **Definition 15** (Consistent lists of formulas). **CONSISTENT**  $\varnothing$  Given  $l$  a list of formulas and  $\mathcal{S}$  a set of axiom schemas,  $l$  is  $\mathcal{S}$ -consistent iff  $\mathcal{S}.\emptyset \not\vdash \neg \bigwedge l$ .

**MAXIMAL\_CONSISTENT**  $\varnothing$  Given  $A$  a modal formula,  $l$  a list of formulas and  $\mathcal{S}$  a set of axiom schemas,  $l$  is  $\mathcal{S}, A$ -maximal consistent iff  $l$  is  $\mathcal{S}$ -consistent,  $l$  contains no repetitions and for all modal formulas  $B$  that are subformulas of  $A$ ,  $B \in l$  or  $\neg B \in l$ . We denote by  $MAX_{\mathcal{S}}^A$  the set of  $\mathcal{S}, A$ -maximal consistent lists of formulas.

The proof of the following lemma is based on the standard Lindenbaum construction for consistent sets [114, pp. 15-16]:

► **Lemma 16** (Extension lemma). **EXTEND\_MAXIMAL\_CONSISTENT**  $\varnothing$  Let  $A$  be a formula and  $\mathcal{S}$  modal schemas. Then any  $\mathcal{S}$ -consistent list made of subformulas and negations of subformulas of  $A$ <sup>15</sup> can be extended to a  $\mathcal{S}, A$ -maximal consistent list made of subsentences of  $A$ .

Finally, we can prove that, whenever  $A$  is not a theorem of  $\mathcal{S}$ , the class  $MAX_{\mathcal{S}}^A$  is non-empty (**NONEMPTY\_MAXIMAL\_CONSISTENT**  $\varnothing$ ).

Next, we identify the central parametric notions we are interested in:

1. Let  $\mathcal{M}_{\mathcal{S}}^A := \langle W_{\mathcal{S}}^A, R_{\mathcal{S}}^A, V_{\mathcal{S}}^A \rangle \varnothing$ , where
  - a.  $W_{\mathcal{S}}^A$  is the subset of  $MAX_{\mathcal{S}}^A$  made of subformulas of  $A$  and their negations;
  - b.  $R_{\mathcal{S}}^A$  is an accessibility relation such that:
    - i.  $\langle W_{\mathcal{S}}^A, R_{\mathcal{S}}^A \rangle \in \mathfrak{G}$ ;
    - ii. for all  $B \in \mathbf{Form}_{\square}$ , if  $\square B$  is a subformula of  $A$  then, for every  $w \in W_{\mathcal{S}}^A$ ,  $\square B \in w$  if and only if, for all  $x \in W_{\mathcal{S}}^A$ ,  $wR_{\mathcal{S}}^A x$  implies  $B \in x$ .
  - c.  $V_{\mathcal{S}}^A$  is an evaluation over  $W_{\mathcal{S}}^A$  such that
 
$$aV_{\mathcal{S}}^A w \text{ if and only if } a \text{ is a subformula of } A \text{ and } a \in w.$$

► **Remark 17.** Notice that, despite the notion is parametric polymorphic, the definition of  $R_{\mathcal{S}}^A$  for each singular logic  $\star \in \{K, T, K4, S4, B, S5, GL\}$  is what actually makes the remaining part of the completeness proof ad hoc polymorphic. Therefore, in the specific files  $\star\_completeness.ml$  we define a specific accessibility relation  $Rel_{\star}^A$  and then we prove that  $\langle W_{\star}^A, Rel_{\star}^A \rangle$  satisfies requirements 1(b)i and 1(b)ii.

2. For such a model we can prove the following:

► **Lemma 18** (Truth lemma). **GEN\_TRUTH\_LEMMA**  $\varnothing$  Let  $A \in \mathbf{Form}_{\square}$  such that  $\mathcal{S}.\emptyset \not\vdash A$ . Then for every  $w \in W_{\mathcal{S}}^A$  and for every  $B$  subformula of  $A$  the following equivalence holds:  $w \Vdash_{\mathcal{M}_{\mathcal{S}}^A} B$  iff  $B \in w$ .

3. From Lemmas 16 and 18 we obtain:

► **Lemma 19** (Countermodel lemma). **GEN\_COUNTERMODEL\_LEMMA\_ALT**  $\varnothing$  Let  $A \in \mathbf{Form}_{\square}$  such that  $\mathcal{S}.\emptyset \not\vdash A$ . Let the frame  $\langle W_{\mathcal{S}}^A, R_{\mathcal{S}}^A \rangle$  satisfy requirements 1(b)i and 1(b)ii. Then  $\langle W_{\mathcal{S}}^A, R_{\mathcal{S}}^A \rangle \not\models A$ .

**Proof.** Given these hypotheses, we can use  $\langle W_{\mathcal{S}}^A, R_{\mathcal{S}}^A, V_{\mathcal{S}}^A \rangle$  as countermodel and the maximal extension of  $[\neg A]$  as the desired counterworld. Indeed, given  $\mathcal{S}.\emptyset \not\vdash A$  then  $[\neg A]$  is  $\mathcal{S}$ -consistent; thus by Lemma 16 we build  $X_{\mathcal{S}}^A$  such that  $X_{\mathcal{S}}^A \in W_{\mathcal{S}}^A$  and  $\neg A \in X_{\mathcal{S}}^A$ . Then, by  $\mathcal{S}$ -consistency,  $A \notin X_{\mathcal{S}}^A$ . Finally, by applying Lemma 18, the thesis holds:

$$X_{\mathcal{S}}^A \not\models_{\langle W_{\mathcal{S}}^A, R_{\mathcal{S}}^A, V_{\mathcal{S}}^A \rangle} A. \quad \blacktriangleleft$$

<sup>15</sup> We denote by the predicate **SUBSENTENCE**  $\varnothing$  the set of subformulas of  $A$  together with their negations, that is the set identified in [25, p. 79] by the notion of “formula”.

## C.2 Ad Hoc Polymorphic Part of the Proof

As previously remarked, we need now to define for each normal logic a specific accessibility relation on maximal consistent lists verifying the requirements of  $\mathcal{M}_S^A$ . Still, we can leverage a parametric notion of accessibility relation which subsumes any relation specific to  $\mathcal{S}$  we are going to define. Listing 4 codifies such a class of “standard” relations for the completeness proof.

■ **Listing 4** Parametric definition of a class of standard accessibility relations.

```
GEN_STANDARD_REL S p w x <=>
  MAXIMAL_CONSISTENT S p w /\ (!q. MEM q w ==> q SUBSENTENCE p) /\
  MAXIMAL_CONSISTENT S p x /\ (!q. MEM q x ==> q SUBSENTENCE p) /\
  (!B. MEM (Box B) w ==> MEM B x)
```

In each file `*_completeness.ml` we then proceed as follows:

### Step I

Following Boolos [25, §5], we formalise the standard accessibility relation  $Rel_*^A$  specific to each  $* \in \{K \checkmark, T \checkmark, K4 \checkmark, S4 \checkmark, B \checkmark, S5 \checkmark, GL \checkmark\}$ , which we recall in Listing 5.

■ **Listing 5** Definition of the standard accessibility relation for each system in HOLMS.

```
K_STANDARD_REL p = GEN_STANDARD_REL {} p

T_STANDARD_REL p w x <=> GEN_STANDARD_REL T_AX p w x

K4_STANDARD_REL p w x <=> GEN_STANDARD_REL K4_AX p w x /\
  (!B. MEM (Box B) w ==> MEM (Box B) x)

S4_STANDARD_REL p w x <=> GEN_STANDARD_REL S4_AX p w x /\
  (!B. MEM (Box B) w ==> MEM (Box B) x)

B_STANDARD_REL p w x <=> GEN_STANDARD_REL B_AX p w x /\
  (!B. MEM (Box B) x ==> MEM B w)

S5_STANDARD_REL p w x <=> GEN_STANDARD_REL S5_AX p w x /\
  (!B. MEM (Box B) w <=> MEM (Box B) x)

GL_STANDARD_REL p w x <=> GEN_STANDARD_REL GL_AX p w x /\
  (!B. MEM (Box B) w ==> MEM (Box B) x) /\
  (?E. MEM (Box E) x /\ MEM (Not (Box E)) w)
```

### Step II

We prove that  $\langle W_*^A, Rel_*^A, V_*^A \rangle$  verifies the two constraints for  $\mathcal{M}_*^A$ :

- (1) Our lemma `*_MAXIMAL_CONSISTENT` verifies the requirement 1(b)i, i.e.  $\langle W_*^A, Rel_*^A \rangle \in \mathfrak{G}$ ;
- (2) Our lemma `*_ACCESSIBILITY_LEMMA` verifies the requirement 1(b)ii.

Figure 9 collects these results in a single table. Some parametric lemmas proved in `gen_completeness.ml`, e.g. `GEN_ACCESSIBILITY_LEMMA`  $\checkmark$ , simplify this work. Thanks to these lemmas distributed in the files for each modal system, we are able to apply Lemma 19 to obtain instances of Theorem 10 for each  $\mathcal{S}$  among K, T, K4, S4, B, S5, and GL.

*	Maximal consistent lemma for *	Accessability lemma for *
K	K_MAXIMAL_CONSISTENT ↗	K_ACCESSIBILITY_LEMMA ↗
T	RF_MAXIMAL_CONSISTENT ↗	T_ACCESSIBILITY_LEMMA ↗
K4	TF_MAXIMAL_CONSISTENT ↗	K4_ACCESSIBILITY_LEMMA ↗
S4	RTF_MAXIMAL_CONSISTENT ↗	S4_ACCESSIBILITY_LEMMA ↗
B	RSF_MAXIMAL_CONSISTENT ↗	B_ACCESSIBILITY_LEMMA ↗
S5	REF_MAXIMAL_CONSISTENT ↗	S5_ACCESSIBILITY_LEMMA ↗
GL	ITF_MAXIMAL_CONSISTENT ↗	GL_ACCESSIBILITY_LEMMA ↗

■ **Figure 9** Ad hoc parametric results in HOLMS.