Growing a Modular Framework for Modal Systems: HOLMS

Antonella Bilotta[a] Marco Maggesi[b] Cosimo Perini Brogi[c]

[a] *Scuola Normale Superiore di Pisa*
[b] *Università di Firenze*
[c] *Scuola IMT Alti Studi Lucca*

Abstract

We present **HOLMS** (**HOL** Light Library for **M**odal **S**ystems), an evolving modular framework for mechanising modal reasoning within the HOL Light proof assistant. Building on earlier work on Gödel-Löb logic (GL), HOLMS introduces a compositional architecture to formalise modal adequacy proofs and implement automated decision procedures for various normal modal systems, currently including K, T, K4, and GL. To clarify the compositional nature of our framework and illustrate how it bridges general-purpose proof assistants, enriched sequent calculi, and formalised mathematics, we highlight some design choices and structural features of HOLMS, such as its use of the metalanguage, embedding strategies, and modularity metrics.

1 Introduction

Modal Logic. Modal logic extends classical logic by going beyond the true/false dichotomy and introducing operators that reflect non-truth-functional linguistic constructs, such as necessity, possibility, knowledge, belief, or obligation. Its versatility spans domains as diverse as computer science, linguistics, and normative reasoning. Various modal operators have proven capable of applications ranging from knowledge representation and verification of consistency of normative corpora to multi-agent systems, as well as modelling of decision-making. Notable examples include temporal logics, widely used in model checking and system verification [7], and provability logic ($\mathbb{GL}$), which formalises the notion of formal provability within mathematical theories [5].

Proof Assistants and HOL Light. An interactive theorem prover, or a proof assistant, is a software system designed to assist users in developing formal mathematical proofs.

HOL Light [19,20,21] is a proof assistant based on **h**igher-**o**rder **l**ogic, distinguished by its minimalist and elegant deductive system. It employs a simply typed lambda calculus and features a small logical core consisting of three classical axioms, ten inference rules, and a powerful principle for definitional extensions. This foundation ensures relative-reliability, [1] support for a broad range of formalisable mathematical theories, and extensibility, allowing users to construct proofs using built-in functions and to program and exploit new ones.

Mechanisation of Modal Logics. To mechanise a modal system means to develop formal and computational tools to represent, analyse, and manipulate it. This problem has given rise to a rich body of

[1] That is, reliability is guaranteed relative to the soundness of the logical core.

literature, including theoretical contributions [25,29,1] and implementations within proof assistants. Notable examples include Prolog formalisation of $\mathbb{S}5$ and $\mathbb{IS}5$ cubes [17,16], Lewis's conditionals [15,14], and non-normal logics [9,8]. Coq, too, supports intuitionistic epistemic logic [10] and classical modal systems in constructive type theory [18]. HOL-based proof assistants have been used for formalisation in higher-order logic of modal and temporal systems [20], and completeness proofs for some epistemic logics [11,12]. A decision procedure for $\mathbb{GL}$, developed by two co-authors of this paper [23,24], has been integrated into the official HOL Light distribution, covering axiomatisation, relational semantics, and an adequacy theorem.

Despite a high degree of modularity, existing approaches lack tools for assessing portability across modal logics and proof assistants, as well as metrics for evaluating compositional design. Extensible mechanisations have proven valuable in several contexts, such as formal analysis of complex normative texts, e.g. European AI Act [22], hybrid AI [28], and formalisation of classical philosophical argument, e.g. Gödel's ontological proof [2]. By leveraging our additional formalisation of adequacy results for the logics B, S4 and S5, we expect that HOLMS will also contribute to these applicative aspects of research in mechanised modal reasoning in the future.

2 HOLMS Framework

HOLMS, which stands for **HOL** Light Library for **M**odal **S**ystems, is the ongoing modular framework we are introducing to enhance the capabilities of the HOL Light proof assistant in (automated) modal reasoning. More concretely, we have extended the HOL Light deductive system with a new inference rule (`HOLMS_RULE`) which automatically decides whether a given modal formula is a theorem of a particular modal logic or, alternatively, constructs a countermodel. In its current state, HOLMS supports decision procedures for $\mathbb{K}$, $\mathbb{T}$, $\mathbb{K}4$ [2] and $\mathbb{GL}$, by implementing root-first proof search in the associated labelled calculi. The modular architecture of HOLMS — described in the following sections — allows the user to extend the library to cover the entire modal cube and, potentially, all normal modal logics. With additional effort, the framework may also be adapted to support non-normal logics.

In this talk, we present the library from a general perspective, highlighting key design choices and structural features of HOLMS. A more detailed and technically focused account is currently being refined. An earlier, embryonic version of the framework was also presented at the international workshop *OVERLAY 2024* [4], shortly before the defence of the first author's MA thesis [3]. An additional extension covering modal adequacy of further normal systems within the so-called S_5-cube is available from our repository and discussed in a different communication paper, currently under review.

2.1 Implementation Choices

Metalanguage and Object Languages. In our formalisation of modal systems, HOL Light serves as the *metatheory*, while modal logics are treated as *object logics*. This distinction requires the embedding of an object language within HOL Light—one capable of explicitly differentiating between formal statements of the modal language (e.g. $\Box A \to \Box\Box A$) and statements about modal systems in the metatheory (e.g. `|- !`A. $\vdash_{\mathbb{K}4} \Box A \to \Box\Box A$).

HOL Light is implemented in OCaml, a functional programming language that also serves as its *metalanguage*. Through this meta-level, we can define and manipulate the deductive apparatus of HOL Light. For instance, new inference rules can be encoded and executed via OCaml code such as `HOLMS_RULE '!`A. $\vdash_{\mathbb{K}4} \Box A \to \Box\Box A$'.

Embeddings. When representing formal systems within HOL-based proof assistants, two primary techniques are commonly used: *deep* and *shallow embedding* [6]. These two approaches differ in how they encode syntax and semantics of the object theory within the host HOL-based metatheory; the former requires the user to define a new type and an interpretation function to represent the embedded theory, while the second inherits them from the metatheory.

HOLMS adopts a *deep embedding* of a standard syntax for modal logic and Kripke semantics, from which

[2] The general mechanism provides a semi-decision procedure for $\mathbb{K}4$. However, the literature on labelled sequent calculi offers uniform solutions to this issue [13], which may be incorporated in future versions of HOLMS.

it defines a general provability predicate and modularly proves adequacy results. To develop decision procedures, it additionally implements a *shallow embedding* of Sara Negri's labelled sequent calculi [26,27] via HOL Light's goal stack mechanism and embedded logics. Henceforth, in HOLMS 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 shallowly embedded in HOL Light.

Modularity. HOLMS generalises the previously developed $\mathbb{GL}$ library [23,24], with the ultimate goal of equipping HOL Light with a uniform mechanism for automated theorem proving and countermodel generation for modal logics. To do so, we developed the previously mentioned compositional implementation methodology— which Figure 1 summarises—centred on a scalable and uniform proof of modal adequacy, inspired by George Boolos' proofs in [5, §5].

To precisely measure and enhance code modularity, we adopted a precise coding discipline, inspired by [30], and distinguished between: *parametric polymorphic* code, fully independent of specific parameter instantiations; and *ad-hoc polymorphic* code, whose components are tailored to the modal logic under consideration. Figure 2 reports the measured modularity of the different components of the HOLMS implementation. Although HOL Light lacks explicit mechanisms to support parametric and/or ad hoc polymorphism (unlike, e.g., Isabelle/HOL), this distinction remains helpful in presenting the abstract structure of our formalisation and discussing the potential portability of our results to proof assistants that implement this distinction through specific mechanisms. We remark *en passant* that the approach we have used in HOL Light can be applied in other theorem provers. This possibility is mainly due to the simplicity of the logical framework we relied on: HOL Light features a comparatively "weaker" logic, extended only through basic extra/meta-theoretical mechanisms. The simpler the language, the broader the scope of possible generalisation(s).

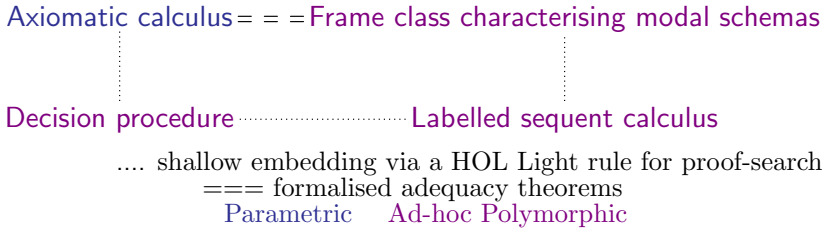

Fig. 1. The implementation methodology behind HOLMS.

Syntax		Parametric Polimorphic
Semantics		Parametric Polimorphic
Correspondence Theory	Concepts	Parametric Polimorphic
	Lemmata	Ad-hoc Polimorphic
Soundness		Parametric Polimorphic
Completeness	Standard Model	Parametric Polimorphic
	Truth Lemma	Parametric Polimorphic
	Countermodel Lemma	Parametric Polimorphic
	Standard Relation	Ad-hoc Polimorphic
	Identification of the Standard Model	Ad-hoc Polimorphic
	Type Generalisation	Ad-hoc Polimorphic
Shallow Embedding		Ad-hoc Polimorphic
Decision Procedure		Ad-hoc Polimorphic

Fig. 2. Measure of the modularity of the implementation.

References

[1] Benzmüller, C.: 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 (2025)

[2] Benzmüller, C., Scott, D.S.: Notes on Gödel's and Scott's variants of the ontological argument (Isabelle/HOL dataset). Archive of Formal Proofs (2025)

[3] Bilotta, A.: Growing a Modular Framework for Modal Systems- HOLMS: a HOL Light Library. Master's thesis, University of Florence (2025), https://arxiv.org/abs/2506.10048

[4] Bilotta, A., Maggesi, M., Perini Brogi, C., Quartini, L.: Growing HOLMS, a HOL Light Library for Modal Systems. In: Porello, D., Vinci, C., Zavatteri, M. (eds.) 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. CEUR Workshop Proceedings, vol. 3904, pp. 41–48. CEUR-WS.org (2024), https://ceur-ws.org/Vol-3904/paper5.pdf

[5] Boolos, G.: The logic of provability. Cambridge University Press (1995)

[6] Boulton, R., Gordon, A., Gordon, M., Harrison, J., Herbert, J., Tassel, J.V.: Experience with embedding hardware description languages in HOL. pp. 129–156

[7] Clarke, E., Grumberg, O., Peled, D., Peled, D.: Model Checking. The Cyber-Physical Systems Series, MIT Press (1999)

[8] Dalmonte, T., Negri, S., Olivetti, N., Pozzato, G.L.: Theorem Proving for Non-normal Modal Logics. In: OVERLAY 2020. Udine, Italy (Sep 2021), https://hal.science/hal-03159954

[9] Dalmonte, T., Negri, S., Olivetti, N., Pozzato, G., Terrioux, C.: PRONOM: A theorem prover for nonnormal modal logics. Available at http://193.51.60.97:8000/pronom/

[10] Doczkal, C., Smolka, G.: Constructive formalization of hybrid logic with eventualities. In: Jouannaud, J.P., Shao, Z. (eds.) Certified Programs and Proofs. pp. 5–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[11] From, A.H.: Formalized soundness and completeness of epistemic logic. In: Silva, A., Wassermann, R., de Queiroz, R.J.G.B. (eds.) Logic, Language, Information, and Computation - 27th International Workshop, WoLLIC 2021, Virtual Event, October 5-8, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13038, pp. 1–15. Springer (2021). https://doi.org/10.1007/978-3-030-88853-4_1, https://doi.org/10.1007/978-3-030-88853-4_1

[12] From, A.H.: An Isabelle/HOL Framework for Synthetic Completeness Proofs. In: Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs. p. 171–186. CPP '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3703595.3705882, https://doi.org/10.1145/3703595.3705882

[13] Garg, D., Genovese, V., Negri, S.: 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. pp. 315–324. IEEE Computer Society (2012). https://doi.org/10.1109/LICS.2012.42, https://doi.org/10.1109/LICS.2012.42

[14] Girlando, M., Lellmann, B., Olivetti, N., Pesce, S., Pozzato, G.L.: Theorem proving for Lewis Logics of Counterfactual Reasoning. In: CILC 2020 - 35th Edition of the Italian Conference on Computational Logic. Rende / Virtual, Italy (Oct 2020), https://hal.science/hal-03080670

[15] Girlando, M., Lellmann, B., Olivetti, N., Pozzato, G.L., Vitalis, Q.: Vinte: An implementation of internal calculi for lewis' logics of counterfactual reasoning. In: Schmidt, R.A., Nalon, C. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods. pp. 149–159. Springer International Publishing, Cham (2017)

[16] Girlando, M., Morales, M.: MOILab: towards a labelled theorem prover for intuitionistic modal logics (Dec 2020), https://hal.science/hal-03048966, working paper or preprint

[17] Girlando, M., Straßburger, L.: Moin: A nested sequent theorem prover for intuitionistic modal logics (system description). In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Automated Reasoning. pp. 398–407. Springer International Publishing, Cham (2020)

[18] Hagemeier, C.: Formalizing intuitionistic epistemic logic in Coq. Ph.D. thesis, BSc thesis (2021)

[19] Harrison, J.: The HOL Light manual (1.1). University of Cambridge. https://www.cl.cam.ac.uk/~jrh13/hol-light/manual-1.1.pdf (2000)

[20] Harrison, J.: HOL Light tutorial. Intel Corporation. http://www.cl.cam.ac.uk/~jrh13/hol-light/tutorial.pdf (2017)

[21] Harrison, J.: The HOL Light Theorem Prover. https://hol-light.github.io/ (2025)

[22] Lawniczak, L., Benzmüller, C.: Logical modalities within the european ai act: An analysis (2025)

[23] Maggesi, M., Perini Brogi, C.: A formal proof of modal completeness for provability logic. In: Cohen, L., Kaliszyk, C. (eds.) 12th International Conference on Interactive Theorem Proving (ITP 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 193, pp. 26:1–26:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). https://doi.org/10.4230/LIPIcs.ITP.2021.26, https://drops.dagstuhl.de/opus/volltexte/2021/13921

[24] Maggesi, M., Perini Brogi, C.: Mechanising Gödel-Löb Provability Logic in HOL Light. Journal of Automated Reasoning **67**(3), 29 (2023). https://doi.org/10.1007/S10817-023-09677-Z, https://doi.org/10.1007/s10817-023-09677-z

[25] Nalon, C., Hustadt, U., Papacchini, F., Dixon, C.: Buy one get 14 free: Evaluating local reductions for modal logic. In: Pientka, B., Tinelli, C. (eds.) Automated Deduction – CADE 29. pp. 382–400. Springer Nature Switzerland, Cham (2023)

[26] Negri, S.: Proof analysis in modal logic. Journal of Philosophical Logic **34**(5), 507–544 (2005)

[27] Negri, S., von Plato, J.: Structural proof theory. Cambridge university press (2008)

[28] Steen, A., Benzmüller, C.: Challenges for non-classical reasoning in contemporary AI applications. Künstliche Intelligenz **38**(1-2), 7–16 (2024)

[29] Steen, A., Sutcliffe, G., Benzmüller, C.: Solving quantified modal logic problems by translation to classical logics. Journal of Logic and Computation p. exaf006 (02 2025). https://doi.org/10.1093/logcom/exaf006, https://doi.org/10.1093/logcom/exaf006

[30] Strachey, C.: Fundamental concepts in programming languages. Higher-order and symbolic computation **13**, 11–49 (2000)