# Theorem provers within theorem provers

## Experiments with modal logic in HOL Light ⋆

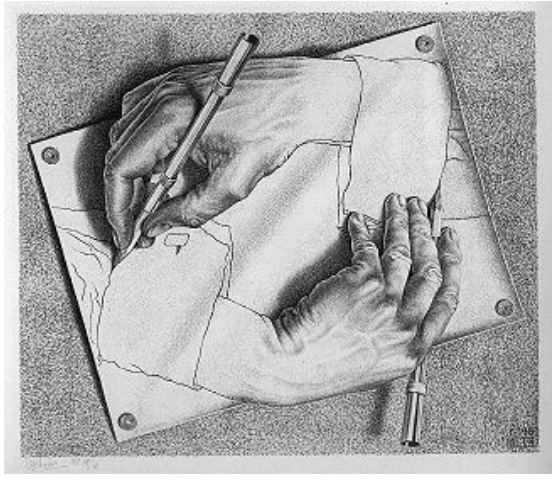
Cosimo Perini Brogi
IMT School for Advanced Studies Lucca

⋆ Based on joint work with Marco Maggesi and Rocco De Nicola
within the project "IT Matters: Methods and Tools for Trustworthy Smart Systems" (PRIN 2017FTXR7S)

Interactions of Proof Assistants and Mathematics
Regensburg, 18–29. Sept. 2023

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

# Drawing Hands



Picture credit: "The Magic of M.C. Escher", WikiMedia

# Quis demonstrat ipsos demonstratores?

- Mathematical proofs are the most reliable form of verification.
- By computerising mathematics through deductive systems, we obtain a highest certificate of correctness.
- But deductive systems are themselves study object of mathematics.

$\therefore$ Deductive systems could be formalised in proof assistants too.

# Our goal

## Current

- Use HOL Light
  - as main environment to formalise results/theorems about syntax and semantics of (normal) modal logics.
  - to make precise the internalisation of semantics in the syntax of (many) contemporary proof systems for modal logics.
  - to mechanise proof search in (the formalised version of) those proof systems by defining new specific tactics reflecting the rules of the proof system under investigation.
- As by-product, obtain a decision procedure implemented in HOL Light for the modal logic under investigation, which works as a countermodel constructor to the input formula when needed.

# Our goal

## Current

- Use HOL Light
  - as main environment to formalise results/theorems about syntax and semantics of (normal) modal logics.
  - to make precise the internalisation of semantics in the syntax of (many) contemporary proof systems for modal logics.
  - to mechanise proof search in (the formalised version of) those proof systems by defining new specific tactics reflecting the rules of the proof system under investigation.
- As by-product, obtain a decision procedure implemented in HOL Light for the modal logic under investigation, which works as a countermodel constructor to the input formula when needed.

# Our goal

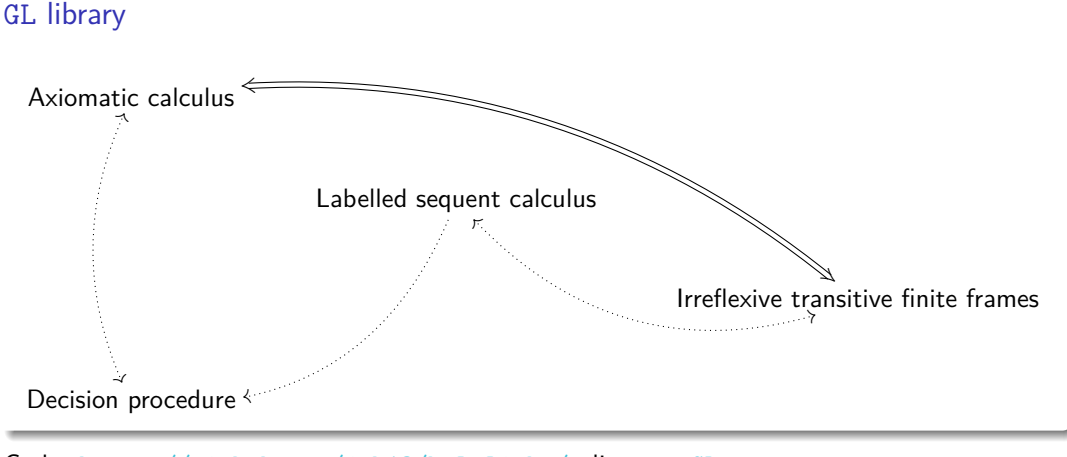
## Long term

- ◇ Our methodology is not strictly related to a specific modal system, and we might apply it to logics relevant to e.g. system security and verification
  ⇒ "Proof-theoretic" alternatives to model checking, SATs, etc.
- ◇ Since it is based on a formal counterpart to internalisation in proof systems, theoretical advances in the latter can be rephrased in our setting, e.g. endowing possible worlds with "structure"
  ⇒ Mechanised deductive systems for compositional/parametric process analysis

# This talk

## The prototype

### GL library




Code: https://github.com/jrh13/hol-light/, directory GL

Paper: *Mechanising Gödel-Löb provability logic in HOL Light*, J. Autom. Reasoning 67, 29 (Open Access)

# Outline

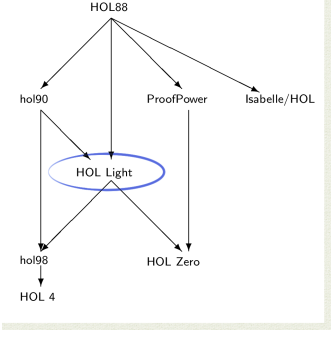
Introduction

Formalization
- Syntax and semantics
- Modal adequacy
- Internal theorem prover

Demo

# Brief glance at HOL Light

(Harrison 2017)




- Clean logical foundations ≈ *Principia Mathematica*
- LCF-style proof checker based on polymorphic simple type theory ≈ small class of *primitive inference rules* for creating theorems + *derived inference rules* to be programmed on top
  - ⇒ 10 primitive rules
  - ⇒ 2 conservative extension principles
  - ⇒ Axioms of choice, extensionality, and infinity
- Written as an OCaml program ≈ three datatypes for the logic: `hol_type`, `term`, and `thm`
- Goal-directed proof development ≈ tactic(al)s + automated methods (in the appropriate domains)

Despite its simple foundations, HOL Light includes a large library of mathematical results in topology, analysis, Euclidean geometry, QBF, floating point algorithms, FOL, limitative results, . . .

# Gödel-Löb Logic

## Syntax

We consider a propositional modal language $\mathcal{L}_{\Box}$ whose formulas have one of the following forms:

$$p \mid \top \mid \bot \mid \neg A \mid A \wedge B \mid A \vee B \mid A \to B \mid A \leftrightarrow B \mid \Box A.$$

### GLaxiom, GLproves

$\mathbb{GL}$ denotes the axiomatic calculus made of:

- Axioms of $\mathbb{CPC}$
- Axiom K : $\Box(A \to B) \to \Box A \to \Box B$
- Axiom GL : $\Box(\Box A \to A) \to \Box A$
- MP Rule $\dfrac{A \to B \qquad A}{B}$
- Nec Rule $\dfrac{A}{\Box A}$

# Gödel-Löb Logic

## Relational semantics

A relational frame is made of a set of "possible worlds" $W$, together with an accessibility relation $R \subseteq W \times W$.

For GL, we consider irreflexive transitive finite frames (`ITF`):

- for no $x \in W$, $xRx$;
- if $xRy$ and $yRz$, then $xRz$;
- $W$ is a finite set.

By introducing a forcing relation $x \Vdash A$ between worlds and formulas, we get a relational model and a general notion of validity in a class of models (`holds`, `holds_in`, and `valid`).

# Adequacy theorem

## Soundness

**Theorem (GL_ITF_VALID)**

*For any formula $A$, if $\mathbb{GL} \vdash A$, then $ITF \vDash A$.*

**Corollary (GL_consistent)**

Gödel-Löb logic is consistent.

# Adequacy theorem

## Completeness

### Theorem (COMPLETENESS_THEOREM_GEN)

*If* $\mathbb{GL} \nvdash A$*, then there exists an irreflexive transitive finite model* $\mathcal{M}$ *such that* $\mathcal{M} \not\Vdash A$.

### Proof.

We formalise (and adapt) the proof in (Boolos 1995), by working with finite *lists* of formulas.

This allows us to prove the theorem for models built on the type `form list` (COMPLETENESS_THEOREM), which is unpleasant from a computational view-point.

Therefore, we generalise the result by

- defining the notion of bisimulation between models (BISIMULATION),
- proving that bisimilar structures satisfy the same formulas (BISIMILAR_VALID),
- defining a bisimulation between the list-based countermodel and its analogous based on sets of formulas (GL_COUNTERMODEL_FINITE_SETS).

# Adequacy theorem

## Completeness

### Theorem (COMPLETENESS_THEOREM_GEN)

*If* $\mathbb{GL} \nvdash A$*, then there exists an irreflexive transitive finite model* $\mathcal{M}$ *such that* $\mathcal{M} \not\Vdash A$.

### Proof.

We formalise (and adapt) the proof in (Boolos 1995), by working with finite *lists* of formulas.

This allows us to prove the theorem for models built on the type `form list` (COMPLETENESS_THEOREM), which is unpleasant from a computational view-point.

Therefore, we generalise the result by

- defining the notion of bisimulation between models (BISIMULATION),
- proving that bisimilar structures satisfy the same formulas (BISIMILAR_VALID),
- defining a bisimulation between the list-based countermodel and its analogous based on sets of formulas (GL_COUNTERMODEL_FINITE_SETS).

# Completeness, directly

The formalism of labelled sequent calculi allows the development of a direct proof of completeness of logical systems w.r.t. relational frames (Negri 2014).

In particular, proof-construction algorithms provide all the information needed for extracting a countermodel from a failed proof-search in a (labelled) sequent calculus (Troelstra and Schwichtenberg 2000).

# G3KGL

$$\frac{}{x : p, \Gamma \Rightarrow \Delta, x : p}\ \mathit{Init} \qquad \frac{}{x : \bot, \Gamma \Rightarrow \Delta}\ \mathcal{L}\bot$$

$$\frac{x : A, x : B, \Gamma \Rightarrow \Delta}{x : A \wedge B, \Gamma \Rightarrow \Delta}\ \mathcal{L}\wedge \qquad \frac{\Gamma \Rightarrow \Delta, x : A \qquad \Gamma \Rightarrow \Delta, x : B}{\Gamma \Rightarrow \Delta, x : A \wedge B}\ \mathcal{R}\wedge$$

$$\frac{x : A, \Gamma \Rightarrow \Delta \qquad x : B, \Gamma \Rightarrow \Delta}{x : A \vee B, \Gamma \Rightarrow \Delta}\ \mathcal{L}\vee \qquad \frac{\Gamma \Rightarrow \Delta, x : A, x : B}{\Gamma \Rightarrow \Delta, x : A \vee B}\ \mathcal{R}\vee$$

$$\frac{\Gamma \Rightarrow \Delta, x : A}{x : \neg A, \Gamma \Rightarrow \Delta}\ \mathcal{L}\neg \qquad \frac{x : A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, x : \neg A}\ \mathcal{R}\neg$$

$$\frac{\Gamma \Rightarrow \Delta, x : A \qquad x : B, \Gamma \Rightarrow \Delta}{x : A \to B, \Gamma \Rightarrow \Delta}\ \mathcal{L}\to \qquad \frac{x : A, \Gamma \Rightarrow \Delta, x : B}{\Gamma \Rightarrow \Delta, x : A \to B}\ \mathcal{R}\to$$

$$\frac{y : A, xRy, x : \Box A, \Gamma \Rightarrow \Delta}{xRy, x : \Box A, \Gamma \Rightarrow \Delta}\ \mathcal{L}\Box \qquad \frac{xRy, y : \Box A, \Gamma \Rightarrow \Delta, y : A}{\Gamma \Rightarrow \Delta, x : \Box A}\ \mathcal{R}\Box\ \substack{\mathit{Löb}\\(y!)}$$

$$\frac{}{xRx, \Gamma \Rightarrow \Delta}\ Irref \qquad \frac{xRz, xRy, yRz, \Gamma \Rightarrow \Delta}{xRy, yRz, \Gamma \Rightarrow \Delta}\ Trans$$

# Correspondences

## Forcing

| | |
|---|---|
| Semantic notation | $x \Vdash A$ |
| Labelled sequent calculus notation | $x : A$ |
| HOL Light notation | `holds (W,R) V A x` |

How to employ these correspondences in our setting?

# Shallow embedding

We adapt the goal stack mechanism of HOL Light to automate the proof development in G3KGL via new HOL Light tactics: when the very goal stack is dealing with

- forcing statement as goal
  $\rightsquigarrow$ tactics mimicking $\mathcal{R}$-rules;
- forcing statemant as hypothesis
  $\rightsquigarrow$ tactics mimicking $\mathcal{L}$-rules.

# Decision algorithm

To check whether a formula $A$ is a theorem of GL, we programmed the tactic `GL_TAC` to perform a complete proof-search in G3KGL embedded in HOL Light.
From that tactic, we define the expected `GL_RULE` automating the whole process: given the formula $A$, the rule returns:

- a new theorem *in* HOL Light, stating that $A$ is derivable in $\mathbb{GL}$, whenever the proof-search positively terminates;
- a countermodel to $A$, that is extracted from the top-most sequent built during the automated proof-search.

# Short Demo

# Put in perspective

- Formalising meta-theoretical results about non-classical logics provides a basis for implementing *within* the theorem prover at hand an automated theorem prover/decision procedure for the logic under investigation
- Internalisation methods through labelling are a formalisation in disguise: such a correspondence is made precise by using a proof assistant
- The prototype for GL can be extended and refined; in particular, we would like to return a complete tree of labelled sequents whenever the automated proof-search positively terminates
- We have worked so far with labelled sequent calculi involving world labels, but if the structure of a label is defined by rules (SOS?) we can extend the embedding by extending the very sequent calculus
- Extending the theory would provide an alternative tool for formal verification of processes which differs from both model checking (no black boxes) and main-stream automated theorem provers implemented in Prolog

*Many thanks for listening!*

# Put in perspective

- Formalising meta-theoretical results about non-classical logics provides a basis for implementing *within* the theorem prover at hand an automated theorem prover/decision procedure for the logic under investigation
- Internalisation methods through labelling are a formalisation in disguise: such a correspondence is made precise by using a proof assistant
- The prototype for GL can be extended and refined; in particular, we would like to return a complete tree of labelled sequents whenever the automated proof-search positively terminates
- We have worked so far with labelled sequent calculi involving world labels, but if the structure of a label is defined by rules (SOS?) we can extend the embedding by extending the very sequent calculus
- Extending the theory would provide an alternative tool for formal verification of processes which differs from both model checking (no black boxes) and main-stream automated theorem provers implemented in Prolog

*Many thanks for listening!*

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

# References


- Boolos, G. (1995). *The logic of provability*. Cambridge University Press.
- Harrison, J. (2017). *The HOL Light Tutorial*.
- Maggesi, M., PB, C. (2023). *Mechanising Gödel-Löb provability logic in HOL Light*, J. Autom. Reasoning 67, 29.
- Negri, S. (2014). *Proofs and countermodels in non-classical logics*. Logica Universalis, 8, 25-60.
- Troelstra, A. S., Schwichtenberg, H. (2000). Basic proof theory (No. 43). Cambridge University Press.