The World’s Most Widely Applicable Modal Logic Theorem Prover and its Associated Infrastructure

Alexander Steen
Freie Universität Berlin

RuleML Webinar September 29th
Talk outline

1. Motivation
2. Flavours of modal logics
3. How it works (roughly)
4. Evaluation
Introduction

Reasoning in Non-Classical Logics

- Increasing interest in various fields
  - Artificial Intelligence (e.g. Agents, Knowledge)
  - Computer Linguistics (e.g. Semantics)
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philosophy (e.g. Metaphysics)
  - Legal Informatics (e.g. Computable/Smart contracts)

- Most powerful ATP/ITP: Classical logic only

Focus here: Modal logics

- Prover for (propositional) modal logics exist
  - ModLeanTAP, Molle, Bliksem, FaCT++,
  - MOLTAP, KtSeqC, STeP, TRP
  - ...

- Only few for quantified variants
  - MleanTAP, MleanCoP, MleanSeP (J. Otten)
  - f2p+MSPASS
Introduction

Reasoning in Non-Classical Logics

- Increasing interest in various fields
  - Artificial Intelligence (e.g. Agents, Knowledge)
  - Computer Linguistics (e.g. Semantics)
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philosophy (e.g. Metaphysics)
  - Legal Informatics (e.g. Computable/Smart contracts)
- Most powerful ATP/ITP: Classical logic only

Focus here: Modal logics

- Prover for (propositional) modal logics exist
  - ModLeanTAP, Molle, Bliksem, FaCT++,
  - MOLTAP, KtSeqC, STeP, TRP
  - ...
- Only few for quantified variants
  - MleanTAP, MleanCoP, MleanSeP (J. Otten)
  - f2p+MSPASS
Introduction

Reasoning in Non-Classical Logics

- Increasing interest in various fields
  - Artificial Intelligence (e.g. Agents, Knowledge)
  - Computer Linguistics (e.g. Semantics)
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philosophy (e.g. Metaphysics)
  - Legal Informatics (e.g. Computable/Smart contracts)

- Most powerful ATP/ITP: Classical logic only

Focus here: Modal logics

- Prover for (propositional) modal logics exist
  - ModLeanTAP, Molle, Bliksem, FaCT++,
  - MOLTAP, KtSeqC, STeP, TRP
  - ...

- Only few for quantified variants
  - MleanTAP, MleanCoP, MleanSeP (J. Otten)
  - f2p+MSPASS
Motivation

1. First-order quantification is (sometimes) not enough
2. Semantic diversity/flexibility needed

See studies in Metaphysics, e.g.
- Gödel’s Ontological Argument [BenzmüllerW.-Paleo,2017]
  and several variants of it
Motivation

1. First-order quantification is (sometimes) not enough
2. **Semantic diversity/flexibility needed:**

Properties of modal operators *necessary* ($\square$) and *possibly* ($\Diamond$)

... but that’s not all of it!
Automation of Quantified Modal Logic

**Motivation**

1. First-order quantification is (sometimes) not enough
2. Semantic diversity/flexibility needed

**Automation approach**

- Indirect: Via encoding into (classical) HOL
- Use existing general purpose HOL reasoners
Motivation

1. First-order quantification is (sometimes) not enough
2. Semantic diversity/flexibility needed

Automation approach

- Indirect: Via encoding into (classical) HOL
- Use existing general purpose HOL reasoners

Advantages

- Sophisticated existing systems
  - ATPs: TPS, agsyHOL, Satallax, LEO-II, Leo-III
  - Further: Isabelle, Nitpick
- Not fixed to any one proving system
- Semantic variations with minor adjustments
  - Axiomatization
  - Quantification semantics
  - ...
Higher Order Modal Logic – Syntax

Based on Simple type theory  [Church, J.Symb.L., 1940]
augmented with modalities

- **Simple types** $\mathcal{T}$ generated by **base types** and mappings ($\rightarrow$)
- Usually, base types are $\mathcal{O}$ and $\mathcal{I}$
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940] augmented with modalities

- Simple types $\mathcal{T}$ generated by base types and mappings ($\rightarrow$)
- Usually, base types are $o$ and $i$

Type of truth-values
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940] augmented with modalities

- **Simple types** $\mathcal{T}$ generated by **base types** and mappings ($\rightarrow$)
- Usually, base types are $\sigma$ and $\iota$

Type of individuals
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940] augmented with modalities

- **Simple types** $\mathcal{T}$ generated by base types and mappings ($\rightarrow$)
- Usually, base types are $\mathtt{o}$ and $\mathtt{l}$

- Terms defined by

$$s, t ::= c_\alpha \mid X_\alpha \quad (\alpha, \beta \in \mathcal{T}, \ c_\alpha \in \Sigma, \ X_\alpha \in \mathcal{V}, \ i \in \mathcal{I})$$

- Allow infix notation for binary logical connectives
- Remaining logical connectives can be defined as usual
- Formulae of HOML are those terms with type $\mathtt{o}$
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940] augmented with modalities

- **Simple types** $\mathcal{T}$ generated by **base types** and mappings ($\rightarrow$)
- Usually, base types are $\mathcal{O}$ and $\mathcal{I}$

**Terms** defined by

$\begin{align*}
s, t & ::= c_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha.s_\beta)_\alpha \rightarrow \beta \mid (s_\alpha \rightarrow \beta t_\alpha)_\beta
\end{align*}$

- Allow infix notation for binary logical connectives
- Remaining logical connectives can be defined as usual
- Formulae of HOML are those terms with type $\mathcal{O}$
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940]
augmented with modalities

- **Simple types** $\mathcal{T}$ generated by base types and mappings ($\rightarrow$)
- Usually, base types are $\mathcal{O}$ and $\mathcal{I}$

- **Terms** defined by $\text{Terms} \; (\alpha, \beta \in \mathcal{T}, \; c_\alpha \in \Sigma, \; X_\alpha \in \mathcal{V}, \; i \in \mathcal{I})$

$$s, t ::= c_\alpha \mid X_\alpha$$
$$\mid (\lambda X_\alpha.s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}t_\alpha)_\beta$$
$$\mid (\Box_i^j o\rightarrow o s_o)_o$$

- Allow infix notation for binary logical connectives
- Remaining logical connectives can be defined as usual
- Formulae of HOML are those terms with type $\mathcal{O}$
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940]
augmented with modalities

- **Simple types** $\mathcal{T}$ generated by base types and mappings ($\rightarrow$)
- Usually, base types are $o$ and $i$

- **Terms** defined by $(\alpha, \beta \in \mathcal{T}, c_\alpha \in \Sigma, X_\alpha \in \nu, i \in I)$

\[
\begin{align*}
s, t &::= c_\alpha | X_\alpha \\
&\quad | (\lambda X_\alpha.s_\beta)_{\alpha\rightarrow\beta} | (s_{\alpha\rightarrow\beta} t_\alpha)_\beta \\
&\quad | (\Box^i_o o \rightarrow o s_0)_o
\end{align*}
\]

- Allow infix notation for binary logical connectives
- Remaining logical connectives can be defined as usual
- Formulae of HOML are those terms with type $o$
Higher Order Modal Logic – Syntax

Based on Simple type theory [Church, J.Symb.L., 1940] augmented with modalities

- **Simple types** $\mathcal{T}$ generated by base types and mappings ($\rightarrow$)
- Usually, base types are $o$ and $i$

- **Terms** defined by

$$ s, t ::= c_\alpha \mid X_\alpha $$

$$ (\lambda X_\alpha.s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}t_\alpha)_{\beta} $$

$$ (\Box^i_{o\rightarrow o}s_o)_{o} $$

- Allow infix notation for binary logical connectives
- Remaining logical connectives can be defined as usual
- Formulae of HOML are those terms with type $o$
Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

\[ \mathcal{M} = (W, \{R^i\}_{i \in I}, \{D_w\}_{w \in W}, \{I_w\}_{w \in W}) \]
Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

\[ M = (W, \{R^i\}_{i \in I}, \{D_w\}_{w \in W}, \{I_w\}_{w \in W}) \]

Set of possible worlds
Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

\[ \mathcal{M} = (W, \{R^i\}_{i \in I}, \{D_w\}_{w \in W}, \{I_w\}_{w \in W}) \]

Family of accessibility relations \( R^i \subseteq W \times W \)
Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

\[ \mathcal{M} = (W, \{R^i\}_{i \in I}, \{D_w\}_{w \in W}, \{I_w\}_{w \in W}) \]

Family of frames, one for every world
Notion of frames \( D = (D_\tau)_{\tau \in T} \) as in HOL:

\[
\begin{align*}
D_l & \neq \emptyset \\
D_0 & = \{T, F\} \\
D_\tau \rightarrow \nu & = D^{D_\tau}_\nu
\end{align*}
\]
Extend HOL models with Kripke structures

\[ \mathcal{M} = (W, \{R^i\}_{i \in I}, \{D_w\}_{w \in W}, \{I_w\}_{w \in W}) \]

Family of interpretation functions \( I_w \)

\[ c_\tau \xrightarrow{I_w} d \in D_\tau \in D_w \]

Assume \( I_w(\neg), I_w(\lor) \ldots \) is standard.
Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

\[ \mathcal{M} = (W, \{R^i\}_{i \in I}, \{D_w\}_{w \in W}, \{I_w\}_{w \in W}) \]

Value of a term (wrt. var. assignment \( g \)):

\[ \|X_\tau\|^{\mathcal{M},g,w} = g_w(X) \]

\[ \|\Box_i o \to o s_o\|^{\mathcal{M},g,w} = \begin{cases} T & \text{if } \|s_o\|^{\mathcal{M},g,v} = T \text{ for all } v \in W \text{ s.t. } (w, v) \in R^i \\ F & \text{otherwise} \end{cases} \]

Assume Henkin semantics
Semantic variants of HOML

1. Axiomatization of $\square^i$
2. Quantification
3. Rigidity
4. Consequence
1. **Axiomatization of □\(^i\)**

   ▶ What properties does the box operators have?
   ▶ Depending on the application domain

Some popular axiom schemes:

<table>
<thead>
<tr>
<th>Name</th>
<th>Axiom scheme</th>
<th>Condition on ( r^i )</th>
<th>Corr. formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>□(^i)(s ⊃ t) ⊃ (∇(^i)s ⊃ □(^i)t)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>B</td>
<td>s ⊃ □(^i)◊(^i)s</td>
<td>symmetric</td>
<td>( wR^iν ⊃ vR^iw )</td>
</tr>
<tr>
<td>D</td>
<td>□(^i)s ⊃ ◊(^i)s</td>
<td>serial</td>
<td>( ∃v.wR^iv )</td>
</tr>
<tr>
<td>T/M</td>
<td>□(^i)s ⊃ s</td>
<td>reflexive</td>
<td>( wR^iw )</td>
</tr>
<tr>
<td>4</td>
<td>□(^i)s ⊃ □□(^i)s</td>
<td>transitive</td>
<td>( (wR^iv ∧ vR^iu) ⊃ wR^iu )</td>
</tr>
<tr>
<td>5</td>
<td>◊(^i)s ⊃ □◊(^i)s</td>
<td>euclidean</td>
<td>( (wR^iv ∧ wR^iu) ⊃ vR^iu )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

2. **Quantification**

3. **Rigidity**

4. **Consequence**
Semantic variants of HOML

1. **Axiomatization of** $\Box^i$
   - What properties does the box operators have?

2. **Quantification**
   - What is the meaning of $\forall$?
   - Several popular choices exist
     1. Varying domains: As introduced (unrestricted frames)
     2. Constant domains: $D_w = D_v$ for all worlds $w, v \in W$
     3. Cumulative domains: $D_w \subseteq D_v$ whenever $(w, v) \in R^i$
     4. Decreasing domains: $D_w \supseteq D_v$ whenever $(w, v) \in R^i$

3. **Rigidity**

4. **Consequence**
Semantic variants of HOML

1. **Axiomatization of $\Box^i$**
   - What properties does the box operators have?

2. **Quantification**
   - What is the meaning of $\forall$?

3. **Rigidity**
   - Do all constants $c \in \Sigma$ denote the same object at every world?
   - Several popular choices exist
     1. Flexible constants: As introduced (unrestricted $I_w$)
     2. Rigid constants: $I_w(c) = I_v(c)$
        for all worlds $w, v \in W$ and all $c \in \Sigma$

4. **Consequence**
1. **Axiomatization of** $\Box^i$
   ▶ What properties does the box operators have?

2. **Quantification**
   ▶ What is the meaning of $\forall$?

3. **Rigidity**
   ▶ Do all constants $c \in \Sigma$ denote the same object at every world?

4. **Consequence**
   ▶ What is an appropriate notion of logical consequence $|=_{HOML}$?
   ▶ Many choices exist, two of them are
     1. Local consequence: ... *not displayed here* ...
     2. Global consequence: ... *not displayed here* ...
Semantic variants of HOML

1. **Axiomatization of □^i**
   ▶ What properties does the box operators have?

2. **Quantification**
   ▶ What is the meaning of ∀?

3. **Rigidity**
   ▶ Do all constants c ∈ Σ denote the same object at every world?

4. **Consequence**
   ▶ What is an appropriate notion of logical consequence ⊨^{HOML}? 

→ at least 10 × 4 × 2 × 2 = 160 distinct logics
Semantic variants of HOML

1. **Axiomatization of** $\Box^i$
   - What properties does the box operators have?

2. **Quantification**
   - What is the meaning of $\forall$?

3. **Rigidity**
   - Do all constants $c \in \Sigma$ denote the same object at every world?

4. **Consequence**
   - What is an appropriate notion of logical consequence $|=^{\text{HOML}}$?

**Example**: Modal logic S5, constant domains, rigid symbols

Generated by 5: $\Diamond^i s \supset \Box^i \Diamond^i s$

Theorems: *all classical tautologies*,
- $s \supset \Box^i \Diamond^i s$,
- $\Box^i s \supset s$,
- $\forall X. \Box^i f X \supset \Box^i \forall X. f X$
  (Barcan formula)
- (symmetric $r^i$)
- (reflexive $r^i$)

...
Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

HOL (meta-logic):

\[ s, t ::= \]

HOML (target logic):

\[ s, t ::= \]

**Embedding of HOML in HOL**

1. Introduce new type \( \mu \) for worlds
   
   HOML formulas \( s_0 \) are mapped to HOL predicates \( s_{\mu \rightarrow o} \)

2. Introduce new constants \( r^i_{\mu \rightarrow \mu \rightarrow o} \) for each \( i \in I \)

3. Connectives:
   
   \[ \neg_{o \rightarrow o} = \]
   
   \[ \lor_{o \rightarrow o \rightarrow o} = \]
   
   \[ \Pi^\tau_{(\tau \rightarrow o) \rightarrow o} = \]
   
   \[ \Box_{o \rightarrow o} = \]

4. Meta-logical notions:
   
   \[ \text{valid} = \]
Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

**HOL (meta-logic):**

\[ s, t ::= \]

**HOML (target logic):**

\[ s, t ::= \]

**Embedding of HOML in HOL**

1. Introduce new type \( \mu \) for worlds

   HOML formulas \( s_0 \) are mapped to HOL predicates \( s_{\mu \rightarrow 0} \)

2. Introduce new constants \( r^i_{\mu \rightarrow \mu \rightarrow 0} \) for each \( i \in I \)

3. Connectives:

   \[ \neg_{ o \rightarrow o } = \]

   \[ \lor_{ o \rightarrow o \rightarrow o } = \]

   \[ \Pi^T_{ (\tau \rightarrow o ) \rightarrow o } = \]

   \[ \Box_{ o \rightarrow o } = \]

4. Meta-logical notions:

   \[ \text{valid} = \]
Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

<table>
<thead>
<tr>
<th>HOL (meta-logic):</th>
<th>s, t ::=</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOML (target logic):</td>
<td>s, t ::=</td>
</tr>
</tbody>
</table>

**Embedding of HOML in HOL**

1. Introduce new type $\mu$ for worlds

   HOML formulas $s_o$ are mapped to HOL predicates $s_{\mu \rightarrow o}$

2. Introduce new constants $r^i_{\mu \rightarrow \mu \rightarrow o}$ for each $i \in I$

3. Connectives:

   $\neg_{o \rightarrow o} =$
   $\lor_{o \rightarrow o \rightarrow o} =$
   $\Pi_{(\tau \rightarrow o) \rightarrow o} =$
   $\Box_{o \rightarrow o} =$

4. Meta-logical notions:

   valid =
Stand-alone tool

Embedding procedure implemented as stand-alone tool

- Semantic specification is analyzed first
- (Meta-)logical notions are included as axioms/definitions
- Output format: "Plain THF" (TH0)
- Integrated as pre-processor into Leo-III
Evaluation

Evaluation setting:
- Translated all 580 mono-modal QMLTP problems to modal THF
- Semantic setting:
  1. Modal operator axiom system ∈ \{K, D, T, S4, S5\}
  2. Quantification semantics ∈ \{constant, varying, cumul., decreasing\}
  3. Rigid constants
  4. Consequence ∈ \{local, global\}
- Native modal logic prover: MleanCoP (J. Otten)
- HOL reasoners: Satallax, LEO-II, Nitpick
- Timeout 60s (2x AMD Opteron 2376 Quad Core/16 GB RAM)

Comments on evaluation result:
- MleanCoP not applicable to modal logic K
- MleanCoP not applicable to decreasing domains semantics
- MleanCoP not applicable to problems with equality symbol
- MleanCoP not applicable for global consequence
- Only first-order modal logic problems
- Embedding approach not restricted to benchmark settings
Evaluation

Evaluation setting:
- Translated all 580 mono-modal QMLTP problems to modal THF
- Semantic setting:
  1. Modal operator axiom system $\in \{K, D, T, S4, S5\}$
  2. Quantification semantics $\in \{\text{constant, varying, cumul., decreasing}\}$
  3. Rigid constants
  4. Consequence $\in \{\text{local, global}\}$
- Native modal logic prover: MleanCoP (J. Otten)
- HOL reasoners: Satallax, LEO-II, Nitpick
- Timeout 60s (2x AMD Opteron 2376 Quad Core/16 GB RAM)

Comments on evaluation result:
- MleanCoP not applicable to modal logic K
- MleanCoP not applicable to decreasing domains semantics
- MleanCoP not applicable to problems with equality symbol
- MleanCoP not applicable for global consequence
- Only first-order modal logic problems
- Embedding approach not restricted to benchmark settings
Evaluation #2

Result excerpt: Theorems

Theorems found

<table>
<thead>
<tr>
<th></th>
<th>LEO-II</th>
<th>Satallax</th>
<th>MleanCoP</th>
</tr>
</thead>
<tbody>
<tr>
<td>D vary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D const</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T vary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T const</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4 vary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4 const</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5 vary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5 const</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation #3

Result excerpt: Counter satisfiable (CSA)

The World’s Most Widely Applicable Modal Logic Theorem Prover, RuleML Webinar
Related work

- Generic theorem proving systems: The Logics Workbench, MetTeL2, LoTREC
- Embedding of further logics: Conditional logics, hybrid logics, many-valued logics, free logic, ...

Conclusion

- Provided a quite general semantics for HOML
- Presented a procedure that automatically converts HOML into HOL
- Implemented a stand-alone tool (e.g. as preprocessor)
  - standard HOL provers can be used to reason about problems encoded in the modal THF syntax
- Approach feasible (no evaluation for higher-order problems yet)
- Many new problems contributed in the modal THF format
The penultimate slide

Related work

- Generic theorem proving systems:
  The Logics Workbench, MetTeL2, LoTREC
- Embedding of further logics:
  Conditional logics, hybrid logics, many-valued logics, free logic, ...

Conclusion

- Provided a quite general semantics for HOML
- Presented a procedure that automatically converts HOML into HOL
- Implemented a stand-alone tool (e.g. as preprocessor)
  - standard HOL provers can be used to reason about problems encoded in the modal THF syntax
- Approach feasible (no evaluation for higher-order problems yet)
- Many new problems contributed in the modal THF format
Thank you for your attention!

Penguins are black and white. Some old TV shows are black and white. Therefore, some penguins are old TV shows.

Logic: another thing that penguins aren't very good at.
### Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

<table>
<thead>
<tr>
<th>HOL (meta-logic):</th>
<th>s, t ::=</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOML (target logic):</td>
<td>s, t ::=</td>
</tr>
</tbody>
</table>

**Embedding of HOML in HOL**

1. **Introduce new type $\mu$ for worlds**
   - HOML formulas $s_0$ are mapped to HOL predicates $s_{\mu \to o}$

2. **Introduce new constants $r^i_{\mu \to o}$ for each $i \in I$**

3. **Connectives:**

   - $=$
   - $=$
   - $=$
   - $=$

4. **Meta-logical notions:**

   - $=$
Embedding of HOML within HOL

Automation approach: Encode HOML semantics within (classical) HOL

HOL (meta-logic):
\[ s, t ::= \]
HOML (target logic):
\[ s, t ::= \]

Embedding of HOML in HOL

(1) Introduce new type \( \mu \) for worlds

HOML formulas \( s_\mu \) are mapped to HOL predicates \( s_{\mu \rightarrow o} \)

(2) Introduce new constants \( r^i_{\mu \rightarrow \mu \rightarrow o} \) for each \( i \in I \)

(3) Connectives:

(4) Meta-logical notions:
Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

HOL (meta-logic): \[ s, t ::= \]

HOML (target logic): \[ s, t ::= \]

**Embedding of HOML in HOL**

1. Introduce new type \( \mu \) for worlds

   HOML formulas \( s_o \) are mapped to HOL predicates \( s_{\mu \rightarrow o} \)

2. Introduce new constants \( r^i_{\mu \rightarrow \mu \rightarrow o} \) for each \( i \in I \)

3. Connectives:

4. Meta-logical notions:
Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

**HOL (meta-logic):**

\[ s, t ::= \]

**HOML (target logic):**

\[ s, t ::= \]

**Embedding of in (1) Introduce new type \( \mu \) for worlds**

HOML formulas \( s_o \) are mapped to HOL predicates \( s_{\mu \rightarrow o} \)

**(2) Introduce new constants \( r^i_{\mu \rightarrow o} \) for each \( i \in I \)**

**(3) Connectives:**

\[
\neg_{o \rightarrow o} = \lambda S_{\mu \rightarrow o}. \lambda W_\mu. \neg(S W) \\
\vee_{o \rightarrow o \rightarrow o} = \lambda S_{\mu \rightarrow o}. \lambda T_{\mu \rightarrow o}. \lambda W_\mu. (S W) \vee (T W) \\
\Pi^\tau_{(\tau \rightarrow o) \rightarrow o} = \lambda P_{\tau \rightarrow o \rightarrow \mu}. \lambda W_\mu. \forall X_\tau. P X W \\
\Box_{o \rightarrow o} = \lambda S_{\mu \rightarrow o}. \lambda W_\mu. \forall V_\mu. \neg(r^i W V) \vee S V
\]

**(4) Meta-logical notions:**

\[ = \]

The World’s Most Widely Applicable Modal Logic Theorem Prover, RuleML Webinar
Embedding of HOML within HOL

**Automation approach:** Encode HOML semantics within (classical) HOL

HOL (meta-logic):
\[ s, t ::= \]

HOML (target logic):
\[ s, t ::= \]

**Embedding of HOML in HOL**

1. Introduce new type \( \mu \) for worlds
   - HOML formulas \( s_\mu \) are mapped to HOL predicates \( s_{\mu \to o} \)
2. Introduce new constants \( r^i_{\mu \to o} \) for each \( i \in I \)
3. Connectives:
   \[ \neg_{o \to o} = \lambda S_{\mu \to o}. \lambda W_{\mu}. \neg (S W) \]
   \[ \vee_{o \to o} = \lambda S_{\mu \to o}. \lambda T_{\mu \to o}. \lambda W_{\mu}. (S W) \vee (T W) \]
   \[ \Pi_{(\tau \to o) \to o} = \lambda P_{\tau \to \mu \to o}. \lambda W_{\mu}. \forall X_{\tau}. P X W \]
   \[ \Box_{o \to o} = \lambda S_{\mu \to o}. \lambda W_{\mu}. \forall V_{\mu}. \neg (r^i W V) \vee S V \]
4. Meta-logical notions:
   \[ \text{valid} = \lambda S_{\mu \to o}. \forall W_{\mu}. S W \]
Embedding semantic variants

1. Axiomatization of □ᵢ
2. Quantification
3. Rigidity
4. Consequence
Embedding semantic variants

1. **Axiomatization of □\(^i\)**
   
   Recall correspondences:

<table>
<thead>
<tr>
<th>Name</th>
<th>Axiom scheme</th>
<th>Condition on (r^i)</th>
<th>Corr. formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>(s \sqcup □^i \Diamond^i s)</td>
<td>symmetric</td>
<td>(wR^i v \sqcup vR^i w)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

   For each desired axiom scheme for □\(^i\):
   
   Postulate frame condition on \(r^i\) as HOL axiom

2. **Quantification**

3. **Rigidity**

4. **Consequence**
Embedding of HOML within HOL #2

**Embedding semantic variants**

1. **Axiomatization of □^i**
   Postulate frame condition on r^i as HOL axiom

2. **Quantification**
   Choose appropriate definition/axiomatization of quantifier:
   - **Constant domains quantifier:**
     \[ \Pi_{\tau \rightarrow o \rightarrow o} = \lambda P_{\tau \rightarrow o}. \lambda W_{\mu}. \forall X_{\tau}. P X W \]
   - **Varying domains quantifier:**
     \[ \Pi_{\tau(\rightarrow o) \rightarrow o, va} = \lambda P_{\tau \rightarrow o}. \lambda W_{\mu}. \forall X_{\tau}. \neg(eiw X W) \vee (P X W) \]
   - **Cumulative/decreasing domains quantifier:**
     Add axioms on eiw

3. **Rigidity**

4. **Consequence**
Embedding semantic variants

1. Axiomatization of $\Box^i$
   Postulate frame condition on $r^i$ as HOL axiom

2. Quantification
   Choose appropriate definition/axiomatization of quantifier

3. Rigidity
   Rigid constants:
   Only translate Boolean types to predicates: $o = \mu \rightarrow o$

   Rigid constants:
   Also translate individuals types to predicates: $i = \mu \rightarrow i$

4. Consequence
Embedding of HOML within HOL #2

Embedding semantic variants

1. **Axiomatization of** $\square^i$
   Postulate frame condition on $r^i$ as HOL axiom

2. **Quantification**
   Choose appropriate definition/axiomatization of quantifier

3. **Rigidity**
   Appropriate type lifting

4. **Consequence**
   Global consequence: Apply $\text{valid}(_\mu o) \rightarrow o$ to all translated $s_{\mu o}$
   \[ S_o = \text{valid}(_\mu o) \rightarrow o \ s_{\mu o} \]
   Local consequence: Apply *actuality* operator $\mathcal{A}$ to all translated $s_{\mu o}$
   \[ S_o = \mathcal{A}(_\mu o) \rightarrow o \ s_{\mu o} \]
   where $\mathcal{A} = \lambda S_{\mu o}. s \ w_{\text{actual}}$ and $w_{\text{actual}}$ is an uninterpreted symbol
Problem representation

**Ongoing work:** Extension of TPTP THF syntax for modal logic

(1) **Formula syntax**

```
thf( classical, axiom, ! [X:$i]: (p @ X)).
```

↓ Extend syntax with modalities

```
thf( modal, axiom, ! [X:$i]: ($box @ (p @ X))).
thf( multi_modal, axiom, ! [X:$i]: ($box_int @ 1 @ (p @ X))).
```

(2) **Semantics configuration**

Add "logic"-annotated statements to the problem:

```
thf(simple_s5, logic, ($modal := [  
  $constants := $rigid,  
  $quantification := $constant,  
  $consequence := $global,  
  $modalities := $modal_system_S5 ]))).
```

▶ Intended semantics is attached to the problem
**Problem representation**

**Ongoing work:** Extension of TPTP THF syntax for modal logic

(1) **Formula syntax**

```prolog
thf(classical, axiom, ![X:i]: (p @ X)).
```

↓ Extend syntax with modalities

```prolog
thf(modal, axiom, ![X:i]: ($box @ (p @ X))).
thf(multi_modal, axiom, ![X:i]: ($box_int @ 1 @ (p @ X))).
```

(2) **Semantics configuration**

Add "logic"-annotated statements to the problem:

```prolog
thf(simple_s5, logic, ($modal := [
  $constants := $rigid,
  $quantification := $constant,
  $consequence := $global,
  $modalities := $modal_system_S5 ])).
```

▶ Intended semantics is attached to the problem
Problem representation

Ongoing work: Extension of TPTP THF syntax for modal logic

(1) Formula syntax

```thf( classical, axiom, ! [X:$i]: (p @ X)).
```

↓ Extend syntax with modalities

```thf( modal, axiom, ! [X:$i]: ($box @ (p @ X))).
thf( multi_modal, axiom, ! [X:$i]: ($box_int @ 1 @ (p @ X))).
```

(2) Semantics configuration

Add "logic"-annotated statements to the problem:

```thf(simple_s5, logic, ($modal := [
    $constants := $rigid,
    $quantification := $constant,
    $consequence := $global,
    $modalities := $modal_system_S5 ]))).
```

► Intended semantics is attached to the problem
Problem representation

Ongoing work: Extension of TPTP THF syntax for modal logic

(1) Formula syntax

\[
\text{thf( classical, axiom, ! \[X:\$i\]: (p @ X))}.
\]

\[ \downarrow \text{Extend syntax with modalities} \]

\[
\begin{align*}
\text{thf( modal, axiom, ! \[X:\$i\]: (}$\text{\texttt{box}} @ (p @ X)))). \\
\text{thf( multi_modal, axiom, ! \[X:\$i\]: (}$\text{\texttt{box_int}} @ 1 @ (p @ X)))}.
\end{align*}
\]

(2) Semantics configuration

Add "logic"-annotated statements to the problem:

\[
\begin{align*}
\text{thf( mydomain_type, type, ( human : $tType ))).} \\
\text{thf( myconstant_declaration, type, ( myconstant : $i ))).} \\
\text{thf( complicated_s5, logic, ( $modal := [} \\
\text{ $constants := [ $rigid, myconstant := $flexible ],} \\
\text{ $quantification := [ $constant, human := $varying ],} \\
\text{ $consequence := [ $global, myaxiom := $local ],} \\
\text{ $modalities := [ $modal_system_S5, $box_int @ 1 := $modal_system_T ] ] ) ).}
\end{align*}
\]

\[ \text{\textbullet\ Intended semantics is attached to the problem} \]