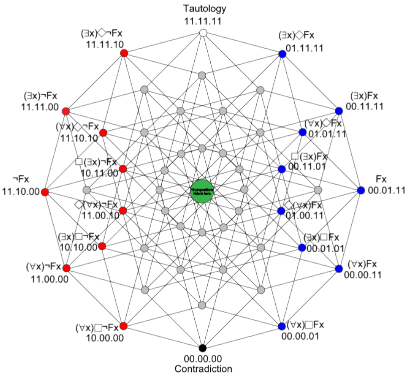


Theorem Provers for Every Normal Modal Logic[1]

Tobias Gleißner   Alexander Steen   Christoph Benzmüller
Freie Universität Berlin

21st Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning

[1]This work has been supported by the DFG under grant BE 2501/11-1 (Leo-III).

Talk outline

1. Higher-Order Modal Logic (HOML)
2. Automating HOML
3. Evaluation
4. Example / Demo

Freie Universität Berlin

Reasoning in Non-Classical Logics

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

Our focus: 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

Reasoning in Non-Classical Logics

- Increasing interest various fields
  - Artificial Intelligence (e.g. **Agents, Knowledge)**
  - Computer Linguistics (e.g. **Semantics)**
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philsophy (e.g. **Metaphysics)**
- Most powerful ATP/ITP: Classical logic only

Our focus: 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

Reasoning in Non-Classical Logics

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

Our focus: 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

Motivation

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

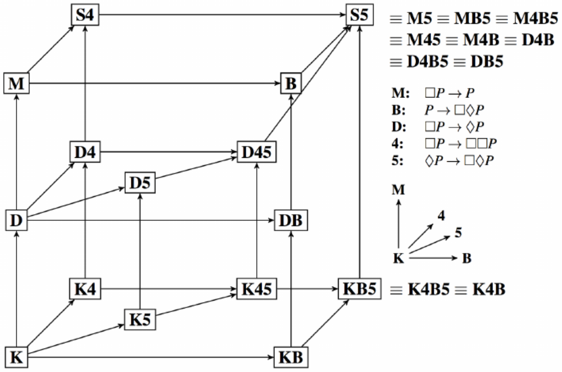
Studies in Metaphysics (e.g. Ontological Argument),

Studies in Computer Ethics

Freie Universität Berlin

Motivation

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

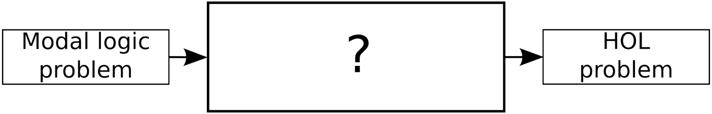


Motivation

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



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

Advantages

- Sophisticated existing systems
  - ATPs: TPS, agsyHOL, Satallax, LEO-II, Leo-III
  - Further: Isabelle, Nitpick
- Not fixed to a proving system
- Semantic variations with minor adjustments
  - Axiomatization
  - Quantification semantics
  - ...

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 a proving system
- Semantic variations with minor adjustments
  - Axiomatization
  - Quantification semantics
  - ...

Higher Order Modal Logic – Syntax

Freie Universität Berlin

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

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and ι

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 $\rightarrow$
- Typically, base types are o and ι

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 $\rightarrow$
- Typically, base types are o and ι

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 $\to$
- Typically, base types are o and ι

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

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

- 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 $\rightarrow$
- Typically, base types are o and ι

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

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

- 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 $\rightarrow$
- Typically, base types are o and ι

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

$$
\begin{aligned}
s,t ::= {} & c_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha . s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta \\
& \mid (\Box^i_{o\rightarrow o}\, s_o)_o
\end{aligned}
$$

- 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 $\to$
- Typically, base types are o and ι

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

$$
\begin{aligned}
s, t ::= & \ c_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha . s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\ t_\alpha)_\beta \\
& \mid (\Box^i_{o\to o} s_o)_o
\end{aligned}
$$

- 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 $\rightarrow$
- Typically, base types are o and ι

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

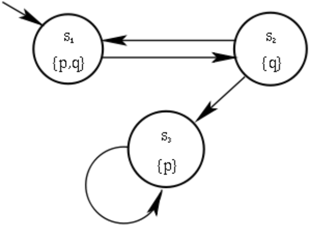
$$
\begin{aligned}
s, t ::= {} & c_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha . s_\beta)_{\alpha \to \beta} \mid (s_{\alpha \to \beta}\, t_\alpha)_\beta \\
& \mid (\Box^i_{o \to o} s_o)_o
\end{aligned}
$$

- 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

$$\mathcal{M} = \left(W, \{R^i\}_{i \in I}, \{\mathcal{D}_w\}_{w \in W}, \{\mathcal{I}_w\}_{w \in W} \right)$$

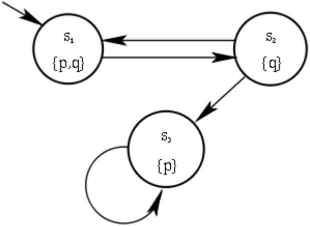


Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

$$\mathcal{M} = \left(W, \{R^i\}_{i \in I}, \{\mathcal{D}_w\}_{w \in W}, \{\mathcal{I}_w\}_{w \in W} \right)$$

Set of possible worlds

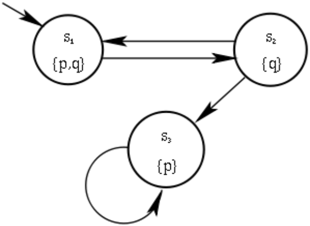


Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

$$\mathcal{M} = \left(W, \{R^i\}_{i \in I}, \{\mathcal{D}_w\}_{w \in W}, \{\mathcal{I}_w\}_{w \in W} \right)$$

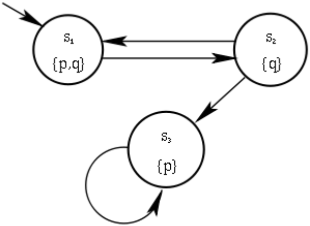
Family of accessibility relations $R^i \subseteq W \times W$



Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

$$\mathcal{M} = \left(W,\ \{R^i\}_{i \in I}\ ,\ \{\mathcal{D}_w\}_{w \in W}\ ,\ \{\mathcal{I}_w\}_{w \in W}\ \right)$$



Family of frames, one for every world

Notion of frames $\mathcal{D} = (D_\tau)_{\tau \in \mathcal{T}}$ as in HOL:

$$D_\iota \neq \emptyset$$

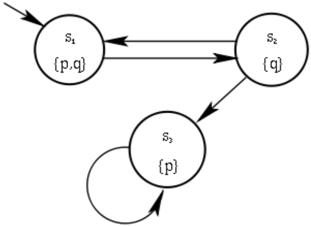
$$D_o = \{T, F\}$$

$$D_{\tau \to \nu} = D_\nu^{D_\tau}$$

Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

$$\mathcal{M} = \left(W,\ \{R^i\}_{i \in I}\ ,\ \{\mathcal{D}_w\}_{w \in W}\ ,\ \{\mathcal{I}_w\}_{w \in W}\ \right)$$



Family of interpretation functions $\mathcal{I}_w$

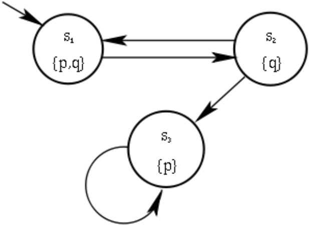
$c_\tau \overset{\mathcal{I}_w}{\mapsto} d \in D_\tau \in \mathcal{D}_w$

Assume $\mathcal{I}_w(\neg), \mathcal{I}_w(\vee) \ldots$ is standard.

Higher Order Modal Logic – Semantics

Extend HOL models with Kripke structures

$$\mathcal{M} = \left(W,\ \{R^i\}_{i \in I}\ ,\ \{\mathcal{D}_w\}_{w \in W}\ ,\ \{\mathcal{I}_w\}_{w \in W}\ \right)$$



Value of a term given by

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

$$\|c_\tau\|^{\mathcal{M},g,w} = \mathcal{I}_w(X)$$

$$\|\,(\lambda X_\tau .\, s_\nu)_{\tau\to\nu}\,\|^{\mathcal{M},g,w} = y \in D_\tau \mapsto \|s_\nu\|^{\mathcal{M},g[X_\tau/y]_w,w}$$

$$\|\,(s_{\tau\to\nu}\, t_\tau)_\nu\,\|^{\mathcal{M},g,w} = \|s_{\tau\to\nu}\|^{\mathcal{M},g,w}\left(\|t_\tau\|^{\mathcal{M},g,w}\right)$$

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

Semantic variants of HOML

1. **Axiomatization of $\Box^i$**
2. **Quantification**
3. **Rigidity**
4. **Consequence**

Semantic variants of HOML

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

   Some popular axiom schemes:

Name	**Axiom scheme**	**Condition on r^i**	**Corr. formula**
K	$\Box^i(s \supset t) \supset (\Box^i s \supset \Box^i t)$	—	—
B	$s \supset \Box^i \Diamond^i s$	symmetric	$wR^i v \supset vR^i w$
D	$\Box^i s \supset \Diamond^i s$	serial	$\exists v. wR^i v$
T/M	$\Box^i s \supset s$	reflexive	$wR^i w$
4	$\Box^i s \supset \Box^i \Box^i s$	transitive	$(wR^i v \wedge vR^i u) \supset wR^i u$
5	$\Diamond^i s \supset \Box^i \Diamond^i s$	euclidean	$(wR^i v \wedge wR^i u) \supset vR^i u$
...	...	...	...

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 ∀?
   - Several popular choices exist
     (1) Varying domains: As introduced (unrestricted frames)
     (2) Constant domains: $\mathcal{D}_w = \mathcal{D}_v$ for all worlds $w, v \in W$
     (3) Cumulative domains: $\mathcal{D}_w \subseteq \mathcal{D}_v$ whenever $(w, v) \in R^i$
     (4) Decreasing domains: $\mathcal{D}_w \supseteq \mathcal{D}_v$ 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 $\mathcal{I}_w$)
     (2) Rigid constants: $\mathcal{I}_w(c) = \mathcal{I}_v(c)$
     for all worlds $w, v \in W$ and all $c \in \Sigma$
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?
4. **Consequence**
   - What is an appropriate notion of logical consequence $\models^{\mathrm{HOML}}$?
   - 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 $\Box^i$**
   - 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 $\models^{\mathrm{HOML}}$?

$\longrightarrow$ at least $10 \times 4 \times 2 \times 2 = 160$ distinct logics

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 μ for worlds

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

(2) Introduce new constants $r^i_{\mu \to \mu \to o}$ 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 in

(1) Introduce new type μ 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 μ for worlds

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

(2) Introduce new constants $r^i_{\mu \to \mu \to o}$ 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 in

(1) Introduce new type μ for worlds

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

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

(3) Connectives:

$$
\begin{aligned}
\neg_{o \to o} &= \lambda S_\sigma . \lambda W_\mu .\ \neg (S\ W) \\
\vee_{o \to o \to o} &= \lambda S_\sigma . \lambda T_\sigma . \lambda W_\mu .\ (S\ W) \vee (T\ W) \\
\Pi^\tau_{(\tau \to o) \to o} &= \lambda P_{\tau \to \sigma} . \lambda W_\mu . \forall X_\tau .\ P\ X\ W \\
\Box_{o \to o} &= \lambda S_\sigma . \lambda W_\mu . \forall V_\mu .\ \neg (r^i\ W\ V) \vee S\ V
\end{aligned}
$$

(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 μ for worlds

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

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

(3) Connectives:

$$\begin{aligned}
\neg_{o \to o} &= \lambda S_\sigma . \lambda W_\mu .\ \neg (S\ W) \\
\vee_{o \to o \to o} &= \lambda S_\sigma . \lambda T_\sigma . \lambda W_\mu .\ (S\ W) \vee (T\ W) \\
\Pi^\tau_{(\tau \to o) \to o} &= \lambda P_{\tau \to \sigma} . \lambda W_\mu . \forall X_\tau .\ P\ X\ W \\
\Box_{o \to o} &= \lambda S_\sigma . \lambda W_\mu . \forall V_\mu .\ \neg (r^i\ W\ V) \vee S\ V
\end{aligned}$$

(4) Meta-logical notions:

$$\text{valid} = \lambda s_\sigma . \forall W_\mu . s\ W$$

Embedding of HOML within HOL #2

Embedding semantic variants

1. **Axiomatization of $\Box^i$**
2. **Quantification**
3. **Rigidity**
4. **Consequence**

Embedding of HOML within HOL #2

Embedding semantic variants

1. **Axiomatization of $\Box^i$**

   Recall correspondences:

Name	**Axiom scheme**	**Condition on r^i**	**Corr. formula**
...	...	...	...
B	$s \supset \Box^i \Diamond^i s$	symmetric	$wR^i v \supset vR^i w$
...	...	...	...

   For each desired axiom scheme for $\Box^i$:
   Postulate frame condition on r^i as HOL axiom
2. **Quantification**
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:
   Constant domains quantifier:

$$\Pi^{\tau}_{(\tau\to o)\to o} = \lambda P_{\tau\to\sigma}.\lambda W_{\mu}.\forall X_{\tau}.\ P\,X\,W$$

   Varying domains quantifier:

$$\Pi^{\tau_{(\tau\to o)\to o},va} = \lambda P_{\tau\to\sigma}.\lambda W_{\mu}.\forall X_{\tau}.\ \neg(\mathrm{eiw}\,X\,W) \vee (P\,X\,W)$$

   Cumulative/decreasing domains quantifier:

   Add axioms on eiw

3. **Rigidity**
4. **Consequence**

Embedding of HOML within HOL #2

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: $\iota = \mu \rightarrow \iota$
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**
   Appropriate type lifting
4. **Consequence**
   Global consequence: Apply $\text{valid}_{(\mu\to o)\to o}$ to all translated $s_{\mu\to o}$

   $$s_o = \text{valid}_{(\mu\to o)\to o}\, s_{\mu\to o}$$

   Local consequence: Apply *actuality* operator $\mathcal{A}$ to all translated $s_{\mu\to o}$

   $$s_o = \mathcal{A}_{(\mu\to o)\to o}\, s_{\mu\to o}$$

   where $\mathcal{A} = \lambda S_\sigma . s\, w_{\text{actual}}$ and w_{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

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**

```
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**

```
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))).
```

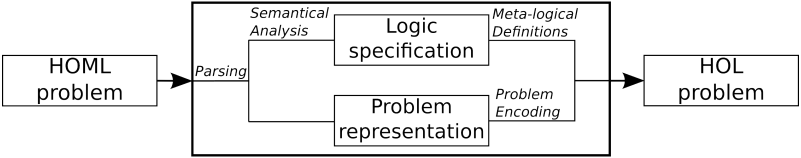
(2) **Semantics configuration**

Add ”logic”-annotated statements to the problem:

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

- Intended semantics is attached to the problem

Embedding procedure implemented as stand-alone tool



- Semantic specification is analyzed first
- Adequate definitions of logical and meta-logical notions are included as axioms and definitions
- The problem is translated as presented
- Output format: Modal THF
- 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

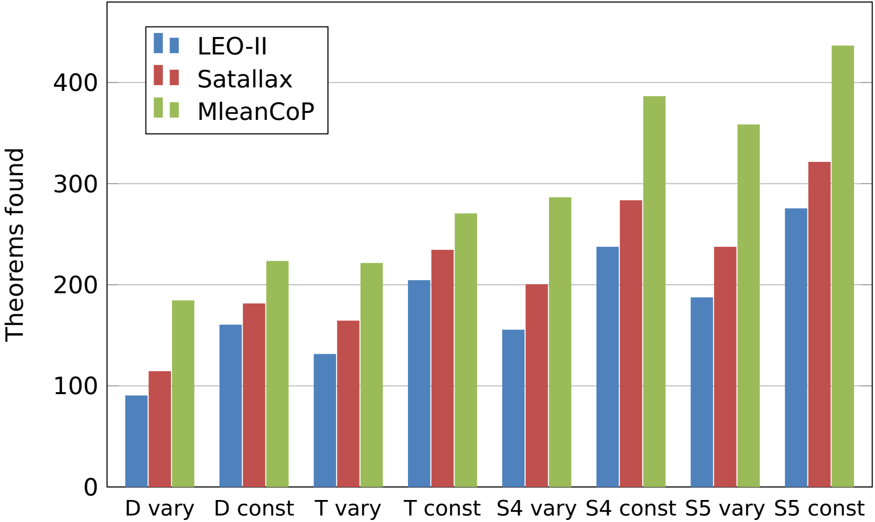
Evaluation setting:

- 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 $\in$ {constant, varying, cumul., decreasing}
  3. Rigid constants
  4. Consequence $\in$ {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)

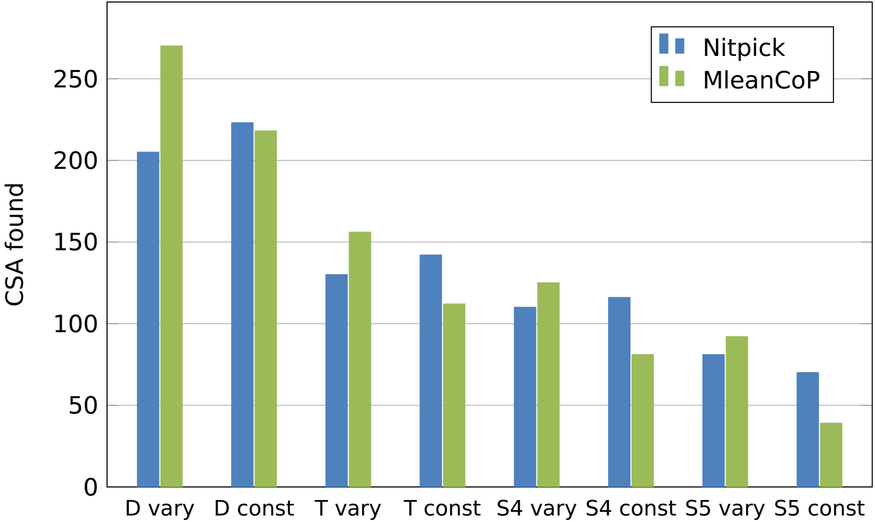
Comments on evaluation result:

- 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

Result excerpt: Theorems



Result excerpt: Counter satisfiable (CSA)



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, ...

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, ...

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!