Reasoning with Embedded Formulas and Modalities in SUMO

Christoph Benzmüller and Adam Pease

Articulate Software, Angwin, CA, USA

ARCOE-10, August 16-17, 2010 Lisbon, Portugal

Research funded by
DFG grant BE 2501/6-1

Ontology Reasoning
— SUMO and Sigma —

SUMO and Sigma

- SUMO — Suggested Upper Merged Ontology
  (NilesPease, FOIS, 2001)
  - open source, formal ontology: `www.ontologyportal.org`
  - has been extended for a number of domain specific ontologies
  - altogether approx. 20,000 terms and 70,000 axioms
  - employs the SUO-KIF representation language, a simplification of Genesereth's original Knowledge Interchange Format (KIF)
- Sigma (Pease, CEUR-71, 2003)
  - browsing and inference system for ontology development
  - integrates KIF-Vampire and SystemOnTPTP

SUMO (and similarly Cyc) contains higher-order representations, but there is only very limited automation support so far

⇒ better automation support is goal of DFG project

SUMO and Sigma

- SUMO — Suggested Upper Merged Ontology (NilesPease, FOIS, 2001)
  - open source, formal ontology: `www.ontologyportal.org`
  - has been extended for a number of domain specific ontologies
  - altogether approx. 20,000 terms and 70,000 axioms
  - employs the SUO-KIF representation language, a simplification of Genesereth's original Knowledge Interchange Format (KIF)
- Sigma (Pease, CEUR-71, 2003)
  - browsing and inference system for ontology development
  - integrates KIF-Vampire and SystemOnTPTP

SUMO (and similarly Cyc) contains higher-order representations, but there is only very limited automation support so far

$\Rightarrow$ better automation support is goal of DFG project

Higher-Order Aspects in SUO-KIF and SUMO: Examples

- Embedded formulas

$$term ::= variable | word | string | funterm | number | sentence$$

`(holdsDuring (YearFn 2009) (likes Mary Bill))`

- ...often in combination with modal operators such as `holdsDuring`, `knows`, `believes`, etc.
- Predicate variables, function variables, propositional variables
- Lambda-Abstraction with KappaFN

Higher-Order Aspects in SUO-KIF and SUMO: Examples

- Embedded formulas
- ...often in combination with modal operators such as `holdsDuring`, `knows`, `believes`, etc.
- Predicate variables, function variables, propositional variables
- Lambda-Abstraction with KappaFN

Higher-Order Aspects in SUO-KIF and SUMO: Examples

- Embedded formulas
- . . . often in combination with modal operators such as `holdsDuring`, `knows`, `believes`, etc.
- Predicate variables, function variables, propositional variables

funterm ::= (funword arg+)   relsent ::= (relword arg+)

funword, relword ::= initialchar wordchar* | variable

```
(<=>
   (instance ?REL TransitiveRelation)
   (forall (?INST1 ?INST2 ?INST3)
      (=>
         (and
            (?REL ?INST1 ?INST2)
            (?REL ?INST2 ?INST3))
         (?REL ?INST1 ?INST3))))
```

- Lambda Abstraction with KappaFN

Higher-Order Aspects in SUO-KIF and SUMO: Examples

- Embedded formulas
- ...often in combination with modal operators such as `holdsDuring`, `knows`, `believes`, etc.
- Predicate variables, function variables, propositional variables
- Lambda-Abstraction with KappaFN

```
(=>
  (attribute ?X Celebrity)
  (greaterThan
    (CardinalityFn
      (KappaFn ?A
        (knows ?A (exists (?P) (equal ?P ?X)))))
    1000))
```

Current FO translation 'tricks'

First-order reasoning on a large ontology

(PeaseSutcliffe, CEUR 257, 2007)

- Quoting of embedded formulas

  A: `(holdsDuring (YearFn 2009) (likes Mary Bill))`

  Q: `(holdsDuring (YearFn ?Y) (likes ?X Bill))`

Current project focus:

embedded formulas and modal operators

Current FO translation 'tricks'

First-order reasoning on a large ontology

(PeaseSutcliffe, CEUR 257, 2007)

- Quoting of embedded formulas

A: `(holdsDuring (YearFn 2009) '(likes Mary Bill))`
Q: `(holdsDuring (YearFn ?Y) '(likes ?X Bill))`

Answer with FO-ATPs ($?Y \leftarrow 2009, ?X \leftarrow Mary$)

Current project focus:

embedded formulas and modal operators

Current FO translation 'tricks'

First-order reasoning on a large ontology

(PeaseSutcliffe, CEUR 257, 2007)

- Quoting of embedded formulas

A: (holdsDuring (YearFn 2009)
    '(and (likes Mary Bill) (likes Sue Bill)))
Q: (holdsDuring (YearFn ?Y) '(likes ?X Bill))

Failure with FO-ATP

Current project focus:

embedded formulas and modal operators

Current FO translation 'tricks'

First-order reasoning on a large ontology

(PeaseSutcliffe, CEUR 257, 2007)

- Quoting of embedded formulas
- Expansion of predicate variables

Current project focus:

embedded formulas and modal operators

Current FO translation 'tricks'

First-order reasoning on a large ontology

(PeaseSutcliffe, CEUR 257, 2007)

- Quoting of embedded formulas
- Expansion of predicate variables

Why not trying higher-order automated theorem proving directly?

Current project focus:

embedded formulas and modal operators

The SUO-KIF to TPTP THF0 Translation

The SUO-KIF to TPTP THF0 Translation

- THF0: new TPTP format for simple type theory
  (SutcliffeBenzmüller, J.Formalized Reasoning, 2010)
- THF0 ATPs: LEO-II, TPS, IsabelleP, Satallax
  THF0 (counter-)model finders: IsabelleM, IsabelleN, Satallax
- achieved:

  SUO-KIF $\longrightarrow$ TPTP THF0

  translation mechanism for SUMO as part of Sigma
- so far only exploits base type ι and o in THF0 ($\longrightarrow$ improvable)
- generally applicable to SUO-KIF representations
- translation example (for SUMO) available at:

  `http://www.ags.uni-sb.de/~chris/papers/SUMO.thf`

The SUO-KIF to TPTP THF0 Translation

- THF0: new TPTP format for simple type theory
  (SutcliffeBenzmüller, J.Formalized Reasoning, 2010)
- THF0 ATPs: LEO-II, TPS, IsabelleP, Satallax
  THF0 (counter-)model finders: IsabelleM, IsabelleN, Satallax
- achieved:

  SUO-KIF $\longrightarrow$ TPTP THF0

  translation mechanism for SUMO as part of Sigma
- so far only exploits base type ι and o in THF0 ($\rightarrow$ improvable)
- generally applicable to SUO-KIF representations
- translation example (for SUMO) available at:

  `http://www.ags.uni-sb.de/~chris/papers/SUMO.thf`

The SUO-KIF to TPTP THF Translation

Main challenge: find consistent typing for untyped SUO-KIF

```
(instance instance BinaryPredicate)
```

The SUO-KIF to TPTP THF Translation

Main challenge: find consistent typing for untyped SUO-KIF

```
(p_instance t_instance BinaryPredicate)
```

Higher-Order Automated Theorem Proving in Ontology Reasoning

Embedded Formulas — An Easy Task for HO-ATP

Example (A: Embedded Formulas)

During 2009 Mary liked Bill and Sue liked Bill. Who liked Bill in 2009?

A: `(holdsDuring (YearFn 2009)`
`    (and (likes Mary Bill) (likes Sue Bill)))`

Q: `(holdsDuring (YearFn 2009) (likes ?X Bill))`

Proof by LEO-II(+E) in 0.19s

Embedded Formulas — An Easy Task for HO-ATP

Example (B: Embedded Formulas)

During 2009 Mary liked Bill and Sue liked Bill. Who liked Bill in 2009?

A:
```
(holdsDuring (YearFn 2009)
   (not (or (not (likes Mary Bill))
            (not (likes Sue Bill)))))
```
Q:
```
(holdsDuring (YearFn 2009) (likes ?X Bill))
```

Proof by LEO-II(+E) in 0.19s

Embedded Formulas — An Easy Task for HO-ATP

Example (C: Embedded Formulas)

At all times Mary likes Bill. During 2009 Sue liked whomever Mary liked. Is there a year in which Sue has liked somebody?

A: `(holdsDuring ?Y (likes Mary Bill))`
B: `(holdsDuring (YearFn 2009)`
`(forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`
Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Proof by LEO-II(+E) in 0.13s

Embedded Formulas — An Easy Task for HO-ATP

Example (D: Embedded Formulas)

What holds that holds at all times. Mary likes Bill. During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?

A: `(=> ?P (holdsDuring ?Y ?P))`

B: `(likes Mary Bill)`

C:
```
(holdsDuring (YearFn 2009)
    (forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))
```

Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Proof by LEO-II(+E) in 0.16s

Embedded Formulas — An Easy Task for HO-ATP

Example (E: Embedded Formulas)

What holds that holds at all times. Mary likes Bill. During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?

A': `(holdsDuring ?Y True)`

B: `(likes Mary Bill)`

C: `(holdsDuring (YearFn 2009)`
`(forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`

Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Embedded Formulas — An Easy Task for HO-ATP

Example (E: Embedded Formulas)

What holds that holds at all times. Mary likes Bill. During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?

A': `(holdsDuring ?Y` $(1 + 1 = 2)$`)`

B: `(likes Mary Bill)`

C: `(holdsDuring (YearFn 2009)`
`     (forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`

Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Embedded Formulas — An Easy Task for HO-ATP

Example (E: Embedded Formulas)

What holds that holds at all times. Mary likes Bill. During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?

A': `(holdsDuring ?Y (forall (?P) (=> ?P ?P)))`

B: `(likes Mary Bill)`

C: `(holdsDuring (YearFn 2009)`
`(forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`

Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Embedded Formulas — An Easy Task for HO-ATP

Example (E: Embedded Formulas)

What holds that holds at all times. Mary likes Bill. During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?

A': `(holdsDuring ?Y True)`

B: `(likes Mary Bill)`

C: `(holdsDuring (YearFn 2009)`
`(forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`

Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Boolean extensionality: $(P \Leftrightarrow Q) \Leftrightarrow (P = Q)$

Embedded Formulas — An Easy Task for HO-ATP

Example (E: Embedded Formulas)

What holds that holds at all times. Mary likes Bill. During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?

A': `(holdsDuring ?Y True)`
B: `(likes Mary Bill)`
C: `(holdsDuring (YearFn 2009)`
`    (forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`
Q: `(holdsDuring (YearFn ?Y) (likes Sue ?X))`

Proof by LEO-II(+E) in 0.08s

Problem for SUO-KIF Semantics:
Boolean Extensionality versus Modal Operators

Problem: Boolean Extensionality versus Modal Operators

Example (E: Embedded Formulas – Temporal Contexts)

A': `(holdsDuring ?Y True)`
B: `(likes Mary Bill)`
C: `(holdsDuring (YearFn 2009)`
`(forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`
Q: `(holdsDuring (YearFn 2009) (likes Sue Bill))`

Proof by LEO-II(+E) in < 0.08s

Boolean extensionality is in conflict with (epistemic) modalities!
(Has Boolean extensionality ever been questioned for KIF?)

Problem relevant not only for HO-ATPs!

Problem: Boolean Extensionality versus Modal Operators

Example (F: Embedded Formulas – Epistemic Contexts)

A": `(knows ?Y True)`

B: `(likes Mary Bill)`

C': `(knows Chris`
`(forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))`

Q': `(knows Chris (likes Sue Bill))`

Proof by LEO-II(+E) in 0.04s

Boolean extensionality is in conflict with (epistemic) modalities!
(Has Boolean extensionality ever been questioned for KIF?)

Problem relevant not only for HO-ATPs!

Example (F: Embedded Formulas – Epistemic Contexts)

A”: `(knows ?Y (forall (?P) (=> ?P ?P)))`

B: `(likes Mary Bill)`

C’: `(knows Chris`
```
    (forall (?X) (=> (likes Mary ?X) (likes Sue ?X)))
```
Q’: `(knows Chris (likes Sue Bill))`

Proof by LEO-II(+E) in 0.04s

Boolean extensionality is in conflict with (epistemic) modalities!
(Has Boolean extensionality ever been questioned for KIF?)

Problem relevant not only for HO-ATPs!

Problem: Boolean Extensionality versus Modal Operators

Example (F: Embedded Formulas – Epistemic Contexts)

A'': `(knows ?Y True)`

B: `(likes Mary Bill)`

C':
```
(knows Chris
    (forall (?X) (=> (likes Mary ?X) (likes Sue ?X))))
```

Q': `(knows Chris (likes Sue Bill))`

Proof by LEO-II(+E) in 0.04s

Boolean extensionality is in conflict with (epistemic) modalities!

(Has Boolean extensionality ever been questioned for KIF?)

Problem relevant not only for HO-ATPs!

Problem: Boolean Extensionality versus Modal Operators

Example (F: Embedded Formulas – Epistemic Contexts)

A'': `(knows ?Y True)`

B: `(likes Mary Bill)`

C': `(knows Chris`
`      (forall (?X) (=> (likes Mary ?X) (likes Sue ?X)))`

Q': `(knows Chris (likes Sue Bill))`

Proof by LEO-II(+E) in 0.04s

Boolean extensionality is in conflict with (epistemic) modalities!
(Has Boolean extensionality ever been questioned for KIF?)

Problem relevant not only for HO-ATPs!

Proposed Solution: Possible World Semantics for SUMO

SUMO $\longrightarrow$ Quantified Multimodal Logic (QML) $\longrightarrow$ TPTP THF
(QML is fragment of HOL (BenzmüllerPaulson, SR-2009-02, 2009))

- T-Box like information in SUMO:

  `(instance holdsDuring AsymmetricRelation)` $\longrightarrow$
  $\forall W_\iota.\text{(instance holdsDuring AsymmetricRelation)}_{\iota\to o}\ W$

- A-Box like information as in query problem: current world cw_ι

  (likes Mary Bill) $\longrightarrow$ $\text{(likes Mary Bill)}_{\iota\to o}\ cw$

  (knows Chris (likes Sue Bill)) $\longrightarrow$
  $(\Box_{\text{Chris}}\text{(likes Sue Bill)})_{\iota\to o}\ cw$

Proposed Solution: Possible World Semantics for SUMO

SUMO $\longrightarrow$ Quantified Multimodal Logic (QML) $\longrightarrow$ TPTP THF
(QML is fragment of HOL (BenzmüllerPaulson, SR-2009-02, 2009))

- T-Box like information in SUMO:

  `(instance holdsDuring AsymmetricRelation)` $\longrightarrow$

  $\forall W_\iota \blacksquare (\text{instance holdsDuring AsymmetricRelation})_{\iota \to o}\ W$

- A-Box like information as in query problem: current world cw_ι

  (likes Mary Bill) $\longrightarrow$ $(\text{likes Mary Bill})_{\iota \to o}\ cw$

  (knows Chris (likes Sue Bill)) $\longrightarrow$

  $(\Box_{\text{Chris}}\ (\text{likes Sue Bill}))_{\iota \to o}\ cw$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota \rightarrow o} . \lambda W_{\iota} . \neg (\phi\, W)$$

$$\vee = \lambda \phi_{\iota \rightarrow o} . \lambda \psi_{\iota \rightarrow o} . \lambda W_{\iota} . \phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o} . \lambda \phi_{\iota \rightarrow o} . \lambda W_{\iota} . \forall V_{\iota} . \neg (R\, W\, V) \vee \phi\, V$$

$$(\forall^{i}) \quad \Pi^{\mu} = \lambda \tau . \lambda W . \forall X . (\tau\, X)\, W$$

$$(\forall^{p}) \quad \Pi^{\iota \rightarrow o} = \lambda \tau . \lambda W . \forall P . (\tau\, P)\, W$$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota \rightarrow o} . \lambda W_{\iota} . \neg (\phi\, W)$$

$$\vee = \lambda \phi_{\iota \rightarrow o} . \lambda \psi_{\iota \rightarrow o} . \lambda W_{\iota} . \phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o} . \lambda \phi_{\iota \rightarrow o} . \lambda W_{\iota} . \forall V_{\iota} . \neg (R\, W\, V) \vee \phi\, V$$

$$(\forall^{i}) \quad \Pi^{\mu} = \lambda \tau_{\mu \rightarrow (\iota \rightarrow o)} . \lambda W_{\iota} . \forall X_{\mu} . (\tau\, X)\, W$$

$$(\forall^{p}) \quad \Pi^{\iota \rightarrow o} = \lambda \tau_{(\iota \rightarrow o) \rightarrow (\iota \rightarrow o)} . \lambda W_{\iota} . \forall P_{\iota \rightarrow o} . (\tau\, P)\, W$$

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota \rightarrow o} . \lambda W_{\iota} . \neg (\phi\, W)$$

$$\vee\ \phi = \lambda \psi_{\iota \rightarrow o} . \lambda W_{\iota} . \phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o} . \lambda \phi_{\iota \rightarrow o} . \lambda W_{\iota} . \forall V_{\iota} . \neg (R\, W\, V) \vee \phi\, V$$

$$(\forall^{i}) \quad \Pi^{\mu} = \lambda \tau_{\mu \rightarrow (\iota \rightarrow o)} . \lambda W_{\iota} . \forall X_{\mu} . (\tau\, X)\, W$$

$$(\forall^{p}) \quad \Pi^{\iota \rightarrow o} = \lambda \tau_{(\iota \rightarrow o) \rightarrow (\iota \rightarrow o)} . \lambda W_{\iota} . \forall P_{\iota \rightarrow o} . (\tau\, P)\, W$$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota\rightarrow o}.\lambda W_\iota.\neg(\phi\, W)$$

$$\vee\ \phi\ \psi = \lambda W_\iota.\phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota\rightarrow\iota\rightarrow o}.\lambda \phi_{\iota\rightarrow o}.\lambda W_\iota.\forall V_\iota.\neg(R\, W\, V) \vee \phi\, V$$

$$(\forall^i) \quad \Pi^\mu = \lambda \tau_{\mu\rightarrow(\iota\rightarrow o)}.\lambda W_\iota.\forall X_\mu.(\tau\, X)\, W$$

$$(\forall^p) \quad \Pi^{\iota\rightarrow o} = \lambda \tau_{(\iota\rightarrow o)\rightarrow(\iota\rightarrow o)}.\lambda W_\iota.\forall P_{\iota\rightarrow o}.(\tau\, P)\, W$$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota \rightarrow o} . \lambda W_\iota . \neg(\phi\, W)$$

$$(\vee\, \phi\, \psi)\, W = \phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o} . \lambda \phi_{\iota \rightarrow o} . \lambda W_\iota . \forall V_\iota . \neg(R\, W\, V) \vee \phi\, V$$

$$(\forall^i) \quad \Pi^\mu = \lambda \tau_{\mu \rightarrow (\iota \rightarrow o)} . \lambda W_\iota . \forall X_\mu . (\tau\, X)\, W$$

$$(\forall^p) \quad \Pi^{\iota \rightarrow o} = \lambda \tau_{(\iota \rightarrow o) \rightarrow (\iota \rightarrow o)} . \lambda W_\iota . \forall P_{\iota \rightarrow o} . (\tau\, P)\, W$$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda\phi_{\iota\rightarrow o}.\lambda W_{\iota}.\neg(\phi\, W)$$

$$\vee = \lambda\phi_{\iota\rightarrow o}.\lambda\psi_{\iota\rightarrow o}.\lambda W_{\iota}.\phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota\rightarrow\iota\rightarrow o}.\lambda\phi_{\iota\rightarrow o}.\lambda W_{\iota}.\forall V_{\iota}.\neg(R\, W\, V) \vee \phi\, V$$

$$(\forall^{i}) \quad \Pi^{\mu} = \lambda\tau_{\mu\rightarrow(\iota\rightarrow o)}.\lambda W_{\iota}.\forall X_{\mu}.(\tau\, X)\, W$$

$$(\forall^{p}) \quad \Pi^{\iota\rightarrow o} = \lambda\tau_{(\iota\rightarrow o)\rightarrow(\iota\rightarrow o)}.\lambda W_{\iota}.\forall P_{\iota\rightarrow o}.(\tau\, P)\, W$$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \to o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota\to o} . \lambda W_{\iota} . \neg(\phi\, W)$$

$$\vee = \lambda \phi_{\iota\to o} . \lambda \psi_{\iota\to o} . \lambda W_{\iota} . \phi\, W \vee \psi\, W$$

$$\Box_R = \lambda \phi_{\iota\to o} . \lambda W_{\iota} . \forall V_{\iota} . \neg(R\, W\, V) \vee \phi\, V$$

$$(\forall^i) \quad \Pi^{\mu} = \lambda \tau_{\mu\to(\iota\to o)} . \lambda W_{\iota} . \forall X_{\mu} . (\tau\, X)\, W$$

$$(\forall^p) \quad \Pi^{\iota\to o} = \lambda \tau_{(\iota\to o)\to(\iota\to o)} . \lambda W_{\iota} . \forall P_{\iota\to o} . (\tau\, P)\, W$$

(Normal) QML as Fragment of HOL

— related, but significantly extending (Ohlbach, 1988/93) —

Straightforward encoding

- base type ι: non-empty set of possible worlds
- base type μ: non-empty set of individuals

QML formulas $\longrightarrow$ HOL terms of type $\iota \to o$

QML operators as abbreviations for specific HOL terms

$$\neg = \lambda \phi_{\iota\to o} \lambda W_{\iota} \neg(\phi\, W)$$

$$\vee = \lambda \phi_{\iota\to o} \lambda \psi_{\iota\to o} \lambda W_{\iota} \phi\, W \vee \psi\, W$$

$$\Box = \lambda R_{\iota\to\iota\to o} \lambda \phi_{\iota\to o} \lambda W_{\iota} \forall V_{\iota} \neg(R\, W\, V) \vee \phi\, V$$

$$(\forall^{i}) \quad \Pi^{\mu} = \lambda \tau_{\mu\to(\iota\to o)} \lambda W_{\iota} \forall X_{\mu} (\tau\, X)\, W$$

$$(\forall^{p}) \quad \Pi^{\iota\to o} = \lambda \tau_{(\iota\to o)\to(\iota\to o)} \lambda W_{\iota} \forall P_{\iota\to o} (\tau\, P)\, W$$

Example (F: Embedded Formulas – Epistemic Contexts)

A": $\forall Y_{\iota\to\iota\to o}.(\Box_Y \top)$ cw

B: $(likes\ Mary\ Bill)$ cw

C': $(\Box_{Chris} (\forall^i X_{\mu}.((likes\ Mary\ X) \supset (likes\ Sue\ X))))$ cw

Q': $(\Box_{Chris} (likes\ Sue\ Bill))$ cw

Axioms for $\Box_{Chris}$ can be added:

M: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\,\phi \supset \phi)\ W$

4: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\,\phi \supset \Box_{Chris}\,\Box_{Chris}\,\phi)\ W$

5: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\,\neg\phi \supset \Box_{Chris}\,\neg\Box_{Chris}\,\phi)\ W$

Challenge: Embedded Formulas — Epistemic Context

Example (F: Embedded Formulas – Epistemic Contexts)

A”: $\forall Y_{\iota\to\iota\to o}.(\Box_Y \top)\ cw$

B: $(likes\ Mary\ Bill)\ cw$

C’: $(\Box_{Chris}(\forall^i X_\mu.((likes\ Mary\ X) \supset (likes\ Sue\ X))))\ cw$

Q’: $(\Box_{Chris}(likes\ Sue\ Bill))\ cw$

Axioms for $\Box_{Chris}$ can be added:

M: $\forall W_\iota.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\phi \supset \phi)\ W$

4: $\forall W_\iota.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\phi \supset \Box_{Chris}\Box_{Chris}\phi)\ W$

5: $\forall W_\iota.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\neg\phi \supset \Box_{Chris}\neg\Box_{Chris}\phi)\ W$

Challenge: Embedded Formulas — Epistemic Context

Example (F: Embedded Formulas – Epistemic Contexts)

A'': $\forall Y_{\iota\to\iota\to o}\centerdot(\Box_Y \top)\ cw$

B: $(likes\ Mary\ Bill)\ cw$

C': $(\Box_{Chris}\,(\forall^i X_{\mu}\centerdot((likes\ Mary\ X) \supset (likes\ Sue\ X))))\ cw$

Q': $(\Box_{Chris}\,(likes\ Sue\ Bill))\ cw$

Axioms for $\Box_{Chris}$ can be added:

M: $\forall W_{\iota}\centerdot(\forall^p \phi_{\iota\to o}\centerdot\Box_{Chris}\,\phi \supset \phi)\ W$

4: $\forall W_{\iota}\centerdot(\forall^p \phi_{\iota\to o}\centerdot\Box_{Chris}\,\phi \supset \Box_{Chris}\,\Box_{Chris}\,\phi)\ W$

5: $\forall W_{\iota}\centerdot(\forall^p \phi_{\iota\to o}\centerdot\Box_{Chris}\,\neg\phi \supset \Box_{Chris}\,\neg\Box_{Chris}\,\phi)\ W$

LEO-II(+E) cannot solve this problem anymore!

Example (F: Embedded Formulas – Epistemic Contexts)

A”: $\forall Y_{\iota\to\iota\to o}.(\Box_Y \top)\ cw$
B: $(\Box_{Chris}\ (likes\ Mary\ Bill))\ cw$
C’: $(\Box_{Chris}\ (\forall^i X_{\mu}.((likes\ Mary\ X) \supset (likes\ Sue\ X))))\ cw$
Q’: $(\Box_{Chris}\ (likes\ Sue\ Bill))\ cw$

Axioms for $\Box_{Chris}$ can be added:

M: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\ \phi \supset \phi)\ W$
4: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\ \phi \supset \Box_{Chris}\ \Box_{Chris}\ \phi)\ W$
5: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\ \neg\phi \supset \Box_{Chris}\ \neg\Box_{Chris}\ \phi)\ W$

But LEO-II(+E) can solve this problem in 0.15s!

Challenge: Embedded Formulas — Epistemic Context

Example (F: Embedded Formulas – Epistemic Contexts)

A”: $\forall Y_{\iota\to\iota\to o}.(\Box_Y \top)$ *cw*
B: $(\Box_{fool}\ (likes\ Mary\ Bill))$ *cw*
C’: $(\Box_{Chris}\ (\forall^i X_{\mu}.((likes\ Mary\ X) \supset (likes\ Sue\ X))))$ *cw*
Q’: $(\Box_{Chris}\ (likes\ Sue\ Bill))$ *cw*

Axioms for $\Box_{Chris}$ can be added:

M: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\ \phi \supset \phi)$ W
4: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\ \phi \supset \Box_{Chris}\ \Box_{Chris}\ \phi)$ W
5: $\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{Chris}\ \neg\phi \supset \Box_{Chris}\ \neg\Box_{Chris}\ \phi)$ W

Axioms for $\Box_{fool}$ can be added …
$\forall W_{\iota}.(\forall^p \phi_{\iota\to o}.\Box_{fool}\ \phi \supset \Box_{Chris}\ \phi)$ W

…

Significant Improvements for Large Theories

Significant Improvements for Large Theories (PAAR-2010)

LEO-II(+E) version v1.1

Ex.	A	B	C	D	E			F	
local	.19	.19	.13	.16	.08	.34	.18	.04	2642.55
SInE	–	–	–	–	–	–	–	–	–
global	–	–	–	–	–	–	–	–	–

global: all SUMO axioms given to LEO-II

SInE: filters SUMO axioms for problem — ˜400 axioms given to LEO-II

local: only handselected axioms given to LEO-II

LEO-II(+E) version v1.2.1 (with relevance filtering)

Ex.	A	B	C	D	E			F	
local	.19	.18	.11	.08	.10	.38	.32	.14	.18
SInE	.43	.40	.21	.54	.37	.12	.70	.06	.26
global	2.8	2.7	1.6	4.9	1.4	0.9	4.7	1.3	0.9

Significant Improvements for Large Theories (PAAR-2010)

LEO-II(+E) version v1.1

Ex.	A	B	C	D	E			F	
local	.19	.19	.13	.16	.08	.34	.18	.04	2642.55
SInE	–	–	–	–	–	–	–	–	–
global	–	–	–	–	–	–	–	–	–

global: all SUMO axioms given to LEO-II

SInE: filters SUMO axioms for problem — ˜400 axioms given to LEO-II

local: only handselected axioms given to LEO-II

LEO-II(+E) version v1.2.1 (with relevance filtering)

Ex.	A	B	C	D	E			F	
local	.19	.18	.11	.08	.10	.38	.32	.14	.18
SInE	.43	.40	.21	.54	.37	.12	.70	.06	.26
global	2.8	2.7	1.6	4.9	1.4	0.9	4.7	1.3	0.9

Conclusion

- SUMO (similarly Cyc) employs higher-order representations
- support with first-order ATPs good but not perfect
- additional support with higher-order ATPs seems feasible
  - translation SUO-KIF $\longrightarrow$ THF0
  - example problems solved effectively (in large theory context!) by LEO-II(+E)
  - simple relevance filtering mechanism implemented for LEO-II(+E)
- various problems in SUMO detected, including:

  Boolean extensionality versus modal operators
- solution
  - possible world semantics for SUO-KIF resp. SUMO
  - exploitation of embedding of quantified multimodal logic in THF for automation with higher-order ATPs
  - supports combinations with further logic embeddings in THF0