Automating Quantified Conditional Logics is (relatively) Easy

Christoph Benzmüller

Freie Universität Berlin

Deduktionstreffen, Koblenz, 16 Sep 2013

for more details:

see IJCAI-2013 paper
come to poster

Quantified Conditional Logics (QCL)

conditional operator vs. material implication

$\varphi \Rightarrow \psi$ $\qquad$ $\varphi \rightarrow \psi \quad (:= \neg\varphi \vee \psi)$

Many applications

- counterfactual reasoning
- default reasoning
- metaphyisical reasoning
- ...
- subsumes normal modal logics ($\Box\varphi := \neg\varphi \Rightarrow \varphi$)
- But: there are no provers yet for QCL!

Quantified Conditional Logics (QCL)

Syntax

$$\varphi, \psi ::= \quad P \mid k(X^1, \ldots, X^n) \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \Rightarrow \psi \mid$$
$$\forall^{co} X\varphi \mid \forall^{va} X\varphi \mid \forall P\varphi$$

Kripke style semantics

$$\ldots$$
$$M, g, s \models \varphi \vee \psi \text{ iff } M, g, s \models \varphi \text{ or } M, g, s \models \psi$$
$$\ldots$$
$$M, g, s \models \varphi \Rightarrow \psi \text{ iff } M, g, t \models \psi \text{ for all } t \in S \text{ such that}$$
$$t \in f(s, [\varphi]) \text{ where } [\varphi] = \{u \mid M, g, u \models \varphi\}$$
$$\ldots$$

Classical Higher-Order Logic (HOL)

Syntax

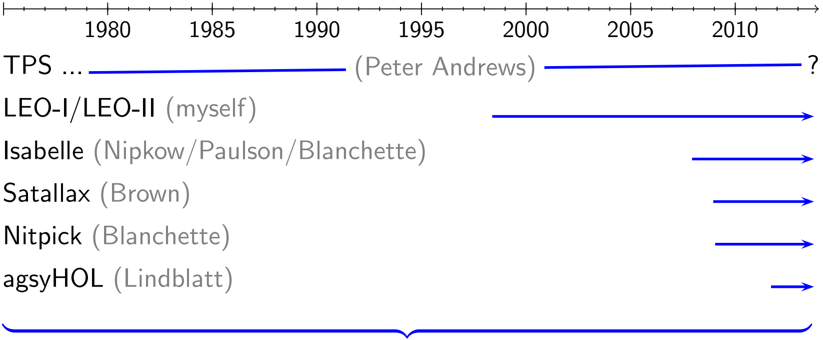
$$s, t ::= c_\alpha \mid X_\alpha \mid (\lambda X_\alpha s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \mid$$
$$(\neg_{o\to o}\, s_o)_o \mid (s_o \vee_{o\to o\to o} t_o)_o \mid (\Pi_{(\alpha\to o)\to o}\, s_{\alpha\to o})_o$$

(Note: Binder notation $\forall X_\alpha t_o$ as syntactic sugar for $\Pi_{(\alpha\to o)\to o}\lambda X_\alpha t_o$)

Henkin semantics well understood

Sound and complete provers do exists

Automated Reasoners for HOL



- all accept TPTP THF Syntax [SutcliffeBenzmüller, J.Form.Reas, 2009]
- can be called remotely via SystemOnTPTP at Miami
- they significantly gained in strength over the last years
- they can be bundled into a combined prover HOL-P

Exploit HOL with Henkin semantics as metalogic
Automate other logics (& combinations) via semantic embeddings
— HOL-P becomes a Universal Reasoner —

QCLs are fragments of HOL

QCL formulas φ are identified with (lifted) HOL terms φ_τ where $\tau := \iota \rightarrow o$

Semantic embedding exploits Kripke style semantics

$$
\begin{aligned}
\neg \quad &= \lambda A_\tau \lambda X_\iota \neg (A\, X) \\
\vee \quad &= \lambda A_\tau \lambda B_\tau \lambda X_\iota (A\, X \vee B\, X) \\
\Rightarrow \quad &= \lambda A_\tau \lambda B_\tau \lambda X_\iota \forall V_\iota (f\, X\, A\, V \rightarrow B\, V) \\
\Pi^{co} \quad &= \lambda Q_{u \rightarrow \tau} \lambda V_\iota \forall X_u (Q\, X\, V) \\
\Pi^{va} \quad &= \lambda Q_{u \rightarrow \tau} \lambda V_\iota \forall X_u (eiw\, V\, X \rightarrow Q\, X\, V) \\
\Pi \quad &= \lambda R_{\tau \rightarrow \tau} \lambda V_\iota \forall P_\tau (R\, P\, V)
\end{aligned}
$$

Meta-level notion of validity defined as

$$\text{valid} = \lambda A_\tau \forall S_\iota (A\, S)$$

A „lean" QCL Theorem Prover (18 lines of code)

```
%---- file: Axioms.ax -----------------------------------
%--- type mu for individuals
thf(mu,type,(mu:$tType)).
%--- reserved constant for selection function f
thf(f,type,(f:$i>($i>$o)>$i>$o)).
%--- `exists in world' predicate for varying domains;
%--- for each v we get a non-empty subdomain eiw@v
thf(eiw,type,(eiw:$i>mu>$o)).
thf(nonempty,axiom,(![V:$i]:?[X:mu]:(eiw@V@X))).
%--- negation, disjunction, material implication
thf(not,type,(not:($i>$o)>$i>$o)).
thf(or,type,(or:($i>$o)>($i>$o)>$i>$o)).
thf(not_def,definition,(not = (^[A:$i>$o,X:$i]:~(A@X)))).
thf(or_def,definition,(or = (^[A:$i>$o,B:$i>$o,X:$i]:((A@X)|(B@X))))).
%--- conditionality
thf(cond,type,(cond:($i>$o)>($i>$o)>$i>$o)).
thf(cond_def,definition,(cond = (^[A:$i>$o,B:$i>$o,X:$i]:![W:$i]:((f@X@A@W)=>(B@W))))).
%--- quantification (constant dom., varying dom., prop.)
thf(all_co,type,(all_co: (mu>$i>$o)>$i>$o)).
thf(all_va,type,(all_va:(mu>$i>$o)>$i>$o)).
thf(all,type,(all:(($i>$o)>$i>$o)>$i>$o)).
thf(all_co_def,definition,(all_co = (^[A:mu>$i>$o,W:$i]:![X:mu]:(A@X@W)))).
thf(all_va_def,definition,(all_va = (^[A:mu>$i>$o,W:$i]:![X:mu]:((eiw@W@X)=>(A@X@W))))).
thf(all_def,definition,(all = (^[A:($i>$o)>$i>$o,W:$i]:![P:$i>$o]:(A@P@W)))).
%--- notion of validity of a conditional logic formula
thf(vld,type,(vld:($i>$o)>$o)).
thf(vld_def,definition,(vld = (^[A:$i>$o]:![S:$i]:(A@S)))).
%---- end file: Axioms.ax -------------------------------
```

Automating QCL — Application in Default Reasoning

Freie Universität Berlin

The following examples are taken from [Delgrande, Artif.Intell., 1998]

$\phi \Rightarrow_x \psi$ stands for $(\exists^{va} x \phi) \Rightarrow \forall^{va} x(\phi \rightarrow \psi)$

"Birds (b) normally fly (f), but Opus (o) is a bird that normally does not fly."

$$b(x) \Rightarrow_x f(x), \quad b(o), \quad b(o) \Rightarrow \neg f(o)$$

HOL-P: Satisfiable (constant domain HOL-P: Unsatisfiable)

"Birds normally fly and necessarily Opus the bird does not fly."

$$b(x) \Rightarrow_x f(x), \quad \Box(b(o) \wedge \neg f(o))$$

HOL-P: Satisfiable (constant domain HOL-P: Unsatisfiable)

"Birds normally fly, penguins normally do not fly and all penguins are necessarily birds."

$$b(x) \Rightarrow_x f(x), \quad p(x) \Rightarrow_x \neg f(x), \quad \forall^{va} \Box(p(x) \rightarrow b(x))$$

HOL-P: Satisfiable (constant domain HOL-P: Satisfiable)

for more see [Benzmüller, IJCAI, 2013]

Soundness and Completeness (and Cut-elimination)

Freie Universität Berlin

$$\models^{L} \varphi \quad \text{iff} \quad \models^{HOL}_{\text{Henkin}} \text{valid}\, \varphi_{\iota\to o} \quad (\text{iff} \quad \vdash^{\text{seq}^{\text{HOL}}}_{\text{cut-free}} \text{valid}\, \varphi_{\iota\to o})$$

Logic L:

- Propositional Conditional Logics [BenzmüllerEtAl., AMAI, 2012]
- Quantified Conditional Logics [Benzmüller, IJCAI, 2013]

Further results:

- Propositional Multimodal Logics [BenzmüllerPaulson, Log.J.IGPL, 2010]
- Quantified Multimodal Logics [BenzmüllerPaulson, Log.Univ., 2012]
- Intuitionistic Logics [BenzmüllerPaulson, Log.J.IGPL, 2010]
- Access Control Logics [Benzmüller, IFIP SEC, 2009]
- . . . more is on the way . . .

Related Application — of Public Interest

SPIEGEL ONLINE WISSENSCHAFT

Login | Registrierung

Politik | Wirtschaft | Panorama | Sport | Kultur | Netzwelt | Wissenschaft | Gesundheit | einestages | Karriere | Uni | Schule | Reise | Auto

Nachrichten > Wissenschaft > Mensch > Mathematik > Formel von Kurt Gödel: Mathematiker bestätigen Gottesbeweis

Formel von Kurt Gödel: Mathematiker bestätigen Gottesbeweis

Von Tobias Hürter

picture-alliance/ Imagno/ Wiener Stadt- und Landesbibliothek

Kurt Gödel (um das Jahr 1935): Der Mathematiker hielt seinen Gottesbeweis jahrzehntelang geheim

Ein Wesen existiert, das alle positiven Eigenschaften in sich vereint. Das bewies der legendäre Mathematiker Kurt Gödel mit einem komplizierten Formelgebilde. Zwei Wissenschaftler haben diesen Gottesbeweis nun überprüft - und für gültig befunden.

Montag, 09.09.2013 – 12:03 Uhr

Drucken | Versenden | Merken

Jetzt sind die letzten Zweifel ausgeräumt: Gott existiert tatsächlich. Ein Computer hat es mit kalter Logik bewiesen - das MacBook des Computerwissenschaftlers Christoph Benzmüller von der Freien Universität Berlin.