predicate logic

logical verification

week 6

2004 10 13

workshop in Nijmegen

Types for Mathematics / Libraries of Formal Mathematics

November 1–2, 2004

invited speakers

Bruno Buchberger (of the Theorema system)

Bob Constable (of the NuPRL system)

`http://www.cs.ru.nl/fnds/typesworkshop/`

`typesworkshop@cs.ru.nl`

overview

from propositional to predicate logic

first order propositional logic	$\longleftrightarrow$	simply typed lambda calculus type theory called $\lambda{\to}$
first order **predicate logic**	$\longleftrightarrow$	type theory called λP
second order propositional logic	$\longleftrightarrow$	type theory called $\lambda 2$

inductive types

program extraction

applications of logic

- **propositional logic**

  logical circuits

  correctness of train track switching

- **predicate logic**

  software correctness ‘Hoare logic’

  correctness of driverless metro in Paris

‘a logic’

- syntax of
  - terms
  - formulas
  - judgments
- derivation rules

terms

- x
- $f(M_1, \ldots, M_n)$

symbols f taken from a fixed finite set of function symbols

formulas

- $P(M_1, \ldots, M_n)$
- $\top$
- $\bot$
- $\neg A$
- $A \to B$
- $A \wedge B$
- $A \vee B$
- $\forall x.\, A$
- $\exists x.\, A$

symbols P taken from a fixed finite set of predicate symbols

random example

$$(\forall x.\, \exists y.\, P(f(c,y)) \wedge Q(g(g(x)),y)) \longrightarrow (\exists z.\, \forall w.\, \neg R(z,w))$$

here the signature is

function symbols $\{f, c, g, \ldots\}$

predicate symbols $\{P, Q, R, \ldots\}$

each symbol has an arity

the rules of predicate logic

introduction rules	**elimination rules**
$I\top$	
	$E\bot$
$I[x]\neg$	$E\neg$
$I[x]\rightarrow$	$E\rightarrow$
$I\wedge$	$El\wedge \quad Er\wedge$
$Il\vee \quad Ir\vee$	$E\vee$
$I\forall$	$E\forall$
$I\exists$	$E\exists$

rules for $\top$ and $\bot$

$\top$ introduction

$$\frac{}{\top}\ I\top$$

$\bot$ elimination

$$\frac{\begin{array}{c}\vdots\\ \bot\end{array}}{A}\ E\bot$$

rules for $\neg$

$\neg$ introduction

$$\dfrac{\begin{matrix}[A^x] \\ \vdots \\ \bot\end{matrix}}{\neg A}\; I[x]\neg$$

$\neg$ elimination

$$\dfrac{\begin{matrix}\vdots \\ \neg A\end{matrix} \qquad \begin{matrix}\vdots \\ A\end{matrix}}{\bot}\; E\neg$$

rules for $\longrightarrow$

$\longrightarrow$ introduction

$$\frac{\begin{matrix}[A^x] \\ \vdots \\ B\end{matrix}}{A \to B}\ I[x]\!\!\to$$

$\longrightarrow$ elimination

$$\frac{\begin{matrix}\vdots & & \vdots \\ A \to B & & A\end{matrix}}{B}\ E\!\!\to$$

rules for $\wedge$

$\wedge$ introduction

$$\frac{\begin{array}{cc} \vdots & \vdots \\ A & B \end{array}}{A \wedge B}\ I\wedge$$

$\wedge$ elimination

$$\frac{\begin{array}{c} \vdots \\ A \wedge B \end{array}}{A}\ El\wedge \qquad \frac{\begin{array}{c} \vdots \\ A \wedge B \end{array}}{B}\ Er\wedge$$

rules for $\vee$

$\vee$ introduction

$$\frac{\begin{array}{c}\vdots\\ A\end{array}}{A \vee B}\; I l\vee \qquad \frac{\begin{array}{c}\vdots\\ B\end{array}}{A \vee B}\; I l\vee$$

$\vee$ elimination

$$\frac{\begin{array}{c}\vdots\\ A \vee B\end{array} \qquad \begin{array}{c}\vdots\\ A \to C\end{array} \qquad \begin{array}{c}\vdots\\ B \to C\end{array}}{C}\; E\vee$$

rules for $\forall$

$\forall$ introduction

$$\frac{\begin{matrix}\vdots \\ A\end{matrix}}{\forall x.\, A}\; I\forall$$

variable condition: x not a free variable in any open assumption

$\forall$ elimination

$$\frac{\begin{matrix}\vdots \\ \forall x.\, A\end{matrix}}{A[x := M]}\; E\forall$$

rules for $\exists$

$\exists$ introduction

$$\dfrac{\begin{array}{c}\vdots\\ A[x := M]\end{array}}{\exists x.\, A}\; I\exists$$

$\exists$ elimination

$$\dfrac{\begin{array}{c}\vdots\\ \exists x.\, A\end{array} \qquad \begin{array}{c}\vdots\\ \forall x.\, (A \to B)\end{array}}{B}\; E\exists$$

variable condition: x not a free variable in B

alternative versions of $E\vee$ and $E\exists$

$\vee$ elimination

$$\dfrac{\begin{matrix} \\ \vdots \\ A \vee B \end{matrix} \qquad \begin{matrix} [A] \\ \vdots \\ C \end{matrix} \qquad \begin{matrix} [B] \\ \vdots \\ C \end{matrix}}{C}$$

$\exists$ elimination

$$\dfrac{\begin{matrix} \\ \vdots \\ \exists x.\, A \end{matrix} \qquad \begin{matrix} [A] \\ \vdots \\ B \end{matrix}}{B}$$

variable condition: x not a free variable in B or any open assumption

minimal versus intuitionistic versus classical

- **minimal predicate logic**

  just the connectives $\rightarrow$ and $\forall$

- **intuitionistic predicate logic**

  the system just presented

- **classical predicate logic**

  add any of

  $$A \vee \neg A$$

  $$\neg\neg A \rightarrow A$$

  $$((A \rightarrow B) \rightarrow A) \rightarrow A \quad \text{(Peirce's law)}$$

empty domains

$$\dfrac{\dfrac{}{\top}\ I\top}{\exists x.\,\top}\ I\exists$$

$\exists x.\,\top$

means

'there exists an object x'

the $I\exists$ rule is not valid when the domain is empty!

terms

- `x`
- `f M1 M2 ... Mn`

curried function application: not a first order system!

formulas

- `P M1 M2 ... Mn`
- `True`
- `False`
- `~A`
- `A -> B`
- `A /\ B`
- `A \/ B`
- `forall x:D, A`
- `exists x:D, A`

tactics

			$I[x]{\longrightarrow}$	$I\forall$	`intro`
			$E{\longrightarrow}$	$E\forall$	`apply`
$E\bot$	$El\wedge$	$Er\wedge$	$E\vee$	$E\exists$	`elim`
				$I\wedge$	`split`
				$Il\vee$	`left`
				$Ir\vee$	`right`
				$I\exists$	`exists`
				$I\top$	`exact I`

example 1

$$(\forall x.\, P(x) \to Q(x)) \to (\forall x.\, P(x)) \to \forall y.\, Q(y)$$

example 2

$$\forall x.\,(P(x) \longrightarrow \neg(\forall y.\,\neg P(y)))$$

example 3

$$(\exists x.\, P(x) \vee Q(x)) \rightarrow (\exists x.\, P(x)) \vee (\exists x.\, Q(x))$$

variable conditions

$\forall$ introduction

$$\dfrac{\begin{array}{c}\vdots\\ A\end{array}}{\forall x.\, A}\; I\forall$$

variable condition: x not a free variable in any open assumption

$\exists$ elimination

$$\dfrac{\begin{array}{c}\vdots\\ \exists x.\, A\end{array} \qquad \begin{array}{c}\vdots\\ \forall x.\, (A \to B)\end{array}}{B}\; E\exists$$

variable condition: x not a free variable in B

example 4: violates the variable condition of $I\forall$

$$\forall x.\,(P(x) \to \forall x.\,P(x))$$

example 5: violates the variable condition of $E\exists$

$$\forall x.\,((\exists x.\,P(x)) \rightarrow P(x))$$

detour elimination

detours

often called ‘cuts’

introduction rule of a connective

directly followed by the

elimination rule of the **same** connective

detour elimination for $\to$

$$\dfrac{\dfrac{\begin{array}{c}[A^x]\\ \vdots\\ B\end{array}}{A \to B}\; I[x]{\to} \qquad \begin{array}{c}\vdots\\ A\end{array}}{B}\; E{\to} \qquad \longrightarrow \qquad \begin{array}{c}\vdots\\ A\\ \vdots\\ B\end{array}$$

'proof of B using a **lemma** A'

detour elimination for $\wedge$

$$\dfrac{\dfrac{\begin{matrix}\vdots \\ A\end{matrix} \qquad \begin{matrix}\vdots \\ B\end{matrix}}{A \wedge B}\, I\wedge}{A}\, El\wedge \quad \longrightarrow \quad \begin{matrix}\vdots \\ A\end{matrix}$$

detour elimination for $\forall$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ A\end{array}}{\forall x.\, A}\; I\forall}{A[x := M]}\; E\forall \quad \longrightarrow \quad \begin{array}{c}\vdots\, * \\ A[x := M]\end{array}$$

$*$ replace x everywhere by M

'proof of $A[x := M]$ from the **generalization** A'

a theorem by Gödel

- **propositional logic**

  provability is decidable

- **predicate logic**

  provability is undecidable

first order provers

- **programs that search for proofs in predicate logic**

  Otter

  Bliksem

  Vampire

  E-SETHEO

  ...

- **tactics that search for proofs in predicate logic**

  coq: jprover

the CASC competition

CASC	=	CADE ATP System Competition
CADE	=	Conference on Automated Deduction
ATP	=	Automated Theorem Proving

yearly competition of first order provers

this year the winner was: Vampire
(solved 180 out of 200 problems)