$\lambda 2$

logical verification

week 12

2004 12 01

overview

the course

1st order **propositional** logic	$\leftrightarrow$	simple type theory
		$\lambda{\to}$
1st order **predicate** logic	$\leftrightarrow$	type theory with **dependent types**
		λP
2nd order propositional logic	$\leftrightarrow$	**polymorphic** type theory
		$\lambda 2$

activities

	logic natural deduction		**type theory** type derivations
on paper	■		■
	↓		↑
in Coq	■	→	■

minimal second order propositional logic

formulas

$a \; b \; c \; \ldots$

$A \to B$

$\bot$

$\top$

$\neg A$

$A \land B$

$A \lor B$

$\forall a.\, A$

$\exists a.\, A$

rules for $\longrightarrow$

$\longrightarrow$ introduction

$$\dfrac{\begin{matrix}[A^x]\\ \vdots \\ B\end{matrix}}{A \to B}\; I[x]{\longrightarrow}$$

$\longrightarrow$ elimination

$$\dfrac{\begin{matrix}\vdots \\ A \to B\end{matrix} \qquad \begin{matrix}\vdots \\ A\end{matrix}}{B}\; E{\longrightarrow}$$

rules for $\forall$

$\forall$ introduction

$$\cfrac{\begin{matrix}\vdots \\ B\end{matrix}}{\forall a.\, B}\; I\forall$$

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

$\forall$ elimination

$$\cfrac{\begin{matrix}\vdots \\ \forall a.\, B\end{matrix}}{B[a := A]}\; E\forall$$

$\lambda 2$

terms

$$*, \; \square$$

$$x, \; y, \; z, \; \ldots$$

$$MN$$

$$\lambda x : M . \, N$$

$$\Pi x : M . \, N$$

rules

$$\frac{}{\vdash * : \Box} \text{ axiom}$$

$$\frac{\Gamma \vdash M : \Pi x : A.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]} \text{ application}$$

$$\frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\, B : s}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B} \text{ abstraction}$$

$$\frac{\Gamma \vdash A : s \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x : A.\, B : *} \text{ product}$$

rules (continued)

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B} \text{ weakening}$$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \text{ variable}$$

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'} \text{ conversion}$$

with $B =_\beta B'$

the three product rules

all systems

$$\frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x : A.\, B : *}$$

only in λP

$$\frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : \Box}{\Gamma \vdash \Pi x : A.\, B : \Box}$$

$$\texttt{nat} \rightarrow *$$

only in $\lambda 2$

$$\frac{\Gamma \vdash A : \Box \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x : A.\, B : *}$$

$$\Pi a : *.\, a \rightarrow a$$

Curry-Howard-de Bruijn

$\longrightarrow$ introduction versus abstraction rule

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

$$\dfrac{\Gamma,\, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\, B : *}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B}$$

$\longrightarrow$ elimination versus application rule

$$\dfrac{\begin{array}{c}\vdots \\ A \to B\end{array} \qquad \begin{array}{c}\vdots \\ A\end{array}}{B}\ E\!\to$$

$$\dfrac{\Gamma \vdash M : \Pi x : A.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

$\forall$ introduction versus abstraction rule

$$\frac{\begin{array}{c}\vdots \\ B\end{array}}{\forall a.\, B}\; I\forall$$

$$\frac{\Gamma, a : * \vdash M : B \qquad \Gamma \vdash \Pi a : *.\, B : *}{\Gamma \vdash \lambda a : *.\, M : \Pi a : *.\, B}$$

∀ elimination versus application rule

$$\dfrac{\begin{matrix}\vdots \\ \forall a.\, B\end{matrix}}{B[a := A]}\ E\forall$$

$$\dfrac{\Gamma \vdash M : \Pi a : *.\, B \qquad \Gamma \vdash A : *}{\Gamma \vdash M\,A : B[a := A]}$$

example 1

$$(\forall b.\, b) \rightarrow a$$

example 2

$$a \to \forall b.\, (b \to a)$$

$$a : * \vdash \Pi b : *.\, (b \to a) : *$$

corresponds to

$$\prod_{b \in \mathsf{Set}} \mathcal{P}(b) \in \mathsf{Set}$$

week 10 $\longrightarrow$ paradox

example 3

$$a \longrightarrow \forall b.\, ((a \longrightarrow b) \longrightarrow b)$$

detour elimination for $\longrightarrow$

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

detour elimination for $\forall$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ B\end{array}}{\forall a.\, B}\, I\forall}{B[a := A]}\, E\forall \qquad \longrightarrow \qquad \begin{array}{c}\vdots\, *\\ B[a := A]\end{array}$$

$*$ replace a everywhere by A

typing the proof term of a detour

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma, x : A \vdash M : B} \qquad \dfrac{\vdots}{\Gamma \vdash \Pi x : A.\, B : s}}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B} \qquad \dfrac{\vdots}{\Gamma \vdash N : A}}{\Gamma \vdash (\lambda x : A.\, M)N : B[x := N]}$$

type checking problem

input

context Γ and terms M and N

$$\Gamma \overset{?}{\vdash} M : N$$

output

true typing judgment is derivable

false typing judgment is not derivable

generally decidable

type synthesis problem

input

context Γ and term M

$$\Gamma \vdash M : ?$$

output

true + term N

typing judgment is derivable with N substituted for $?$

false for no term substituted for $?$ is the judgment derivable

generally decidable

type inhabitation problem

input

context Γ and term N

$$\Gamma \vdash \,?: N$$

output

true + term M

typing judgment is derivable with M substituted for $?$

false for no term substituted for $?$ is the judgment derivable

generally undecidable

proof checking problem

input

formula A + possibly incorrect 'proof'

correct?

output

true the 'proof' is a correct proof of A

false the 'proof' is not a correct proof of A

generally decidable

corresponds to type checking problem

provability problem

input

formula A

A?

output

true + proof of A

A is proved by the proof in the output

false A is not provable

generally undecidable

corresponds to type inhabitation problem

uniqueness of types

$$\left.\begin{array}{l} \Gamma \vdash A : B \\ \\ \Gamma \vdash A : B' \end{array}\right\} \Rightarrow \quad B =_\beta B'$$

subject reduction

$$\left.\begin{array}{l} \Gamma \vdash A : B \\ \\ B \twoheadrightarrow_\beta B' \end{array}\right\} \Rightarrow \quad \Gamma \vdash A : B'$$

Curry-Howard-de Bruijn isomorphism

isomorphism

between

the set of proofs in a logic

and

the set of typed lambda terms in a type theory

formulas as types

proofs as terms

Brouwer-Heyting-Kolmogorov interpretation

intuitive semantics of intuitionistic logic

explains

what it means to prove a formula

in terms of

what it means to prove its components

minimal versus intuitionistic versus classical logic

- **minimal logic**

  only $\rightarrow$ and $\forall$

- **intuitionistic logic**

  all connectives, just the intro and elim rules

- **classical logic**

  $\ldots$ $+$ one of the classical principles

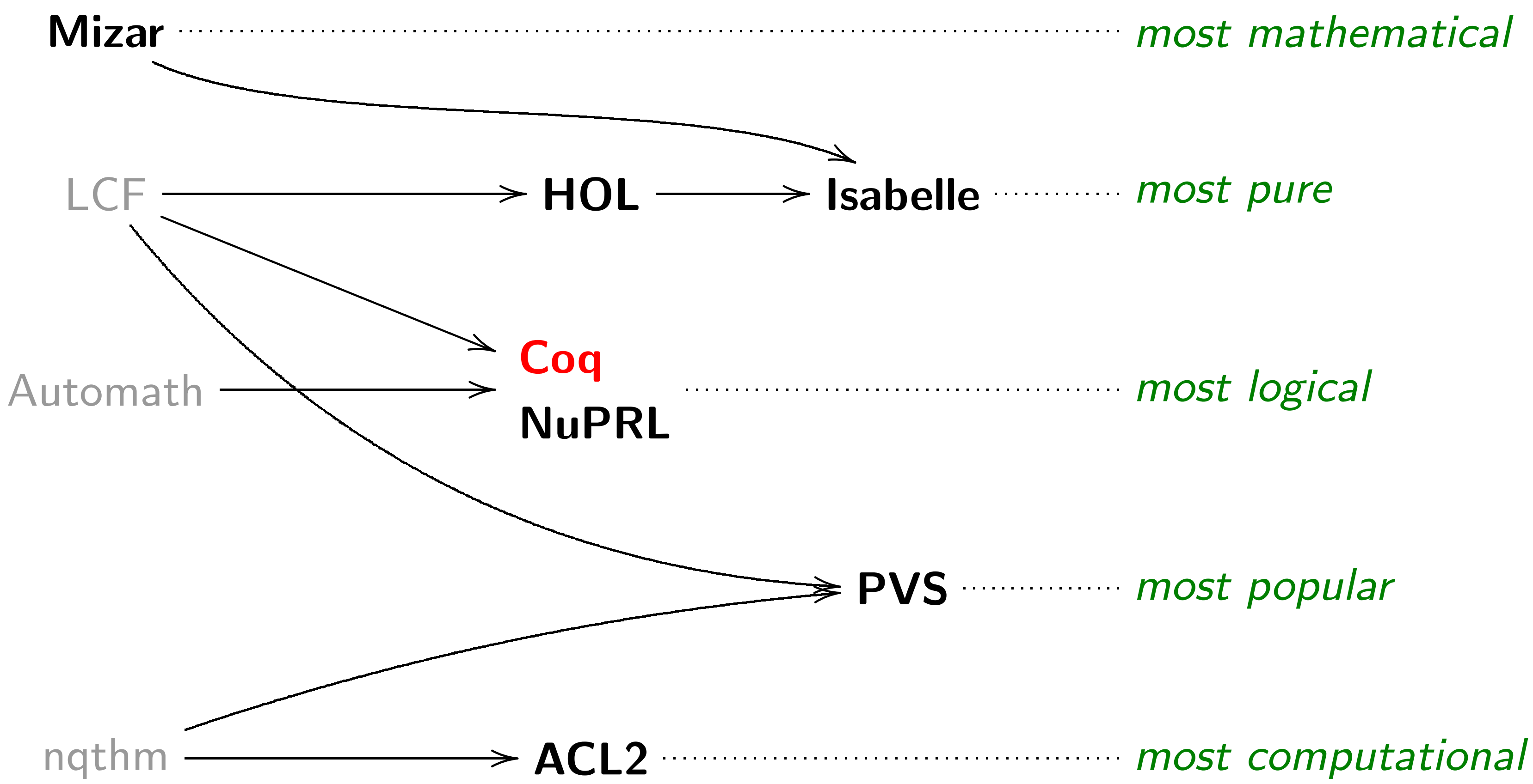
  excluded middle

$$A \lor \neg A$$

$$\forall a.\, a \lor \neg a$$

Coq versus the other proof assistants

seven provers for mathematics



foundations

- **primitive recursive arithmetic**

  ACL2

- **type theory**

  typed lambda calculus + inductive types, constructive

  Coq, NuPRL

- **higher order logic**

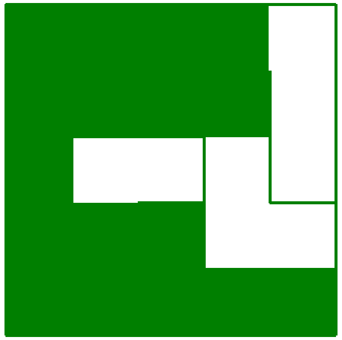
  typed lambda calculus + choice operator, classical

  HOL, Isabelle, PVS

- **set theory**

  Mizar

procedural versus declarative



- **procedural**

  E E S E N E S S S W W W S E E E

  HOL, PVS, Coq, NuPRL

- **declarative**

  (0,0) (1,0) (2,0) (3,0) (3,1) (2,1) (1,1) (0,1) (0,2) (0,3) (0,4) (1,4) (1,3) (2,3) (2,4) (3,4) (4,4)

  Mizar, Isabelle