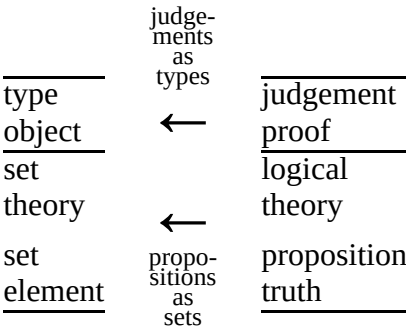


22.2.1987

The Logic of Judgements

Workshop on General Logic, Laboratory for Foundations of Computer Science, University of Edinburgh, 23-27 February 1987



P.M.-L, On the meanings of the logical constants and the justification of the logical laws

Peter Schroeder-Heister, Judgements of higher levels and standardized rules for logical constants in Martin-Löf’s theory of logic, June 1985

The logical theory cannot be entirely logical since it needs at least one set for the quantifiers to ranger over.

Type formation

$$set : type$$

$$\frac{A : set}{elem(A) : type}$$

$$\frac{\alpha : type \quad \beta : type \quad (x : \alpha)}{(fun\ x : \alpha)\beta : type}$$

AUTOMATH

CONSTRUCTIONS

LF

$$(x : \alpha)\beta \quad = \quad \downarrow \quad [x : \alpha]\beta \quad = \quad \downarrow \quad \Pi x : \alpha.\beta$$

$(\alpha)\beta = (x : \alpha)\beta \quad \wedge J \quad \beta \text{ does not depend on } x$

Object formation

$$\frac{\alpha : type}{x : \alpha} \quad (\text{assumption})$$

$$\frac{\begin{array}{c} (x : \alpha) \\ b : \beta \end{array}}{(x)b : (x : \alpha)\beta} \quad (\text{abstraction})$$
$$\lambda x : \alpha. b \quad \longleftarrow \text{LF}$$

$$\frac{c : (x : \alpha)\beta \quad a : \alpha}{c(a) : \beta(a/x)} \quad (\text{application})$$

Equality

$$\frac{\begin{array}{cc} & (x : \alpha) \\ a : \alpha & b : \beta \end{array}}{((x)b)(a) = b(a/x) : \beta(a/x)} \quad (\beta)$$

$$\frac{c : (x : \alpha)\beta}{c = (x)c(x) : (x : \alpha)\beta} \quad (\eta)$$

refl., symm., trans.

equals for equals, spec.

$$\frac{a : \alpha \quad \alpha = \beta : type}{a : \beta}$$

$$\frac{a : elem(A) \quad A = B : set}{a : elem(B)}$$

$$\begin{array}{ll} \alpha : type & \alpha = \beta : type \\ a : \alpha & a = b : \alpha \end{array}$$

Since LF has no equality judgements, $\alpha = \beta : type$ has to be expressed by

$$\alpha, \beta : type, \quad \alpha =_{\beta\eta} \beta,$$

and $a = b : \alpha$ by

$$a, b : \alpha, \quad a =_{\beta\eta} b.$$

The equality judgements are badly needed for formalizing intuitionistic set theory in the logical framework.

A theory, like first order predicate logic or intuitionistic set theory, is specified by typing the constants which make up its signature and writing down the finitely many definitional equations that relate certain combinations of those constants.

In a sensible theory, it is decidable whether or not an expression is wellformed (meaningful) as well as whether or not two wellformed (meaningful) expressions are definitionally equal (have the same meaning).

type checking = checking the wellformedness (meaningfulness) of an expression

$prop : type$

$$\frac{A : prop}{proof(A) : type}$$

A

In the propositions as sets interpretation, we put

$$prop = set : type$$

$$proof(A) = elem(A) : type$$

but it is not necessary for what follows that we have made that identification.

Judgement formation

$$\frac{A : \textit{prop}}{A \textit{ true} : \textit{judg}} \\ \textit{true}(A) \\ A$$

$$\frac{I : \textit{judg} \quad J : \textit{judg}}{I | J : \textit{judg}} \\ \rightarrow \quad (\text{Gentzen}) \\ \Rightarrow \quad (\text{Schroeder-Heister}) \\ \vdash \quad (\text{LF})$$

$$\frac{\alpha : \textit{type} \quad (x : \alpha) \quad J : \textit{Judg}}{|_{x:\alpha} J : \textit{judg}} \\ \rightarrow_{x:\alpha} \\ \Rightarrow_{x:\alpha} \\ \vdash_{x:\alpha}$$

Proof rules

$$\frac{J : \textit{judg}}{J} \quad (\text{assumption})$$

$$\frac{\begin{array}{c} (I) \\ J \end{array}}{I|J} \quad \frac{I|J \quad I}{J}$$

$$\frac{\begin{array}{c} (x : \alpha) \\ J \end{array}}{|_{x:\alpha} J} \quad \frac{|_{x:\alpha} J \quad a : \alpha}{J(a/x)}$$

A context (sequence of assumptions) in this system has the form

$$x_1 : \alpha_1, \dots, x_m : \alpha_m, \underbrace{J_1, \dots, J_n}_{\text{permutable}}$$

Judgements as types

$$judg = type$$

$$true(A) = proof(A)$$

$$I|J = (I)J$$

$$|_{x:\alpha} J = (x : \alpha)J$$

With a proof of J by means of the proof rules above, we can associate an object

$$\text{proof object} \longrightarrow^{\mathcal{C}} : \underbrace{\underbrace{J}_{\text{synthetic judgement}}}_{\text{analytic judgement}}$$

Generalized logical operations

$$\frac{\begin{array}{c} A \text{ prop} \quad (A \text{ true}) \\ B \text{ prop} \end{array}}{A \supset B \text{ prop}} \quad \&$$

The ordinary implication and conjunction are easy enough to type

$$\begin{array}{lcl} \supset & : & (prop)(prop)prop \\ \& & : \quad \quad - \text{ " } - \end{array}$$

but how do we type the generalized implication and conjunction?

Type formation

$$\frac{\begin{array}{c} (I) \\ I : \text{judg} \quad \beta : \text{type} \end{array}}{I|\beta : \text{type}}$$

$$\begin{array}{lcl}
\supset & : & (X : prop)(X|prop)prop \\
& & \\
\& & : & \quad X \text{ true} \\
& & & \quad true(X)
\end{array}$$

Object formation

$$\begin{array}{c}
(I) \\
\frac{b : \beta}{b : I|\beta} \quad \frac{b : I|\beta \quad I}{b : \beta} \\
\\
\frac{A : prop \quad \frac{(A \text{ true})}{B : prop} \quad \overline{B : A \text{ true}|prop}}{A \supset B : prop} \\
\&
\end{array}$$