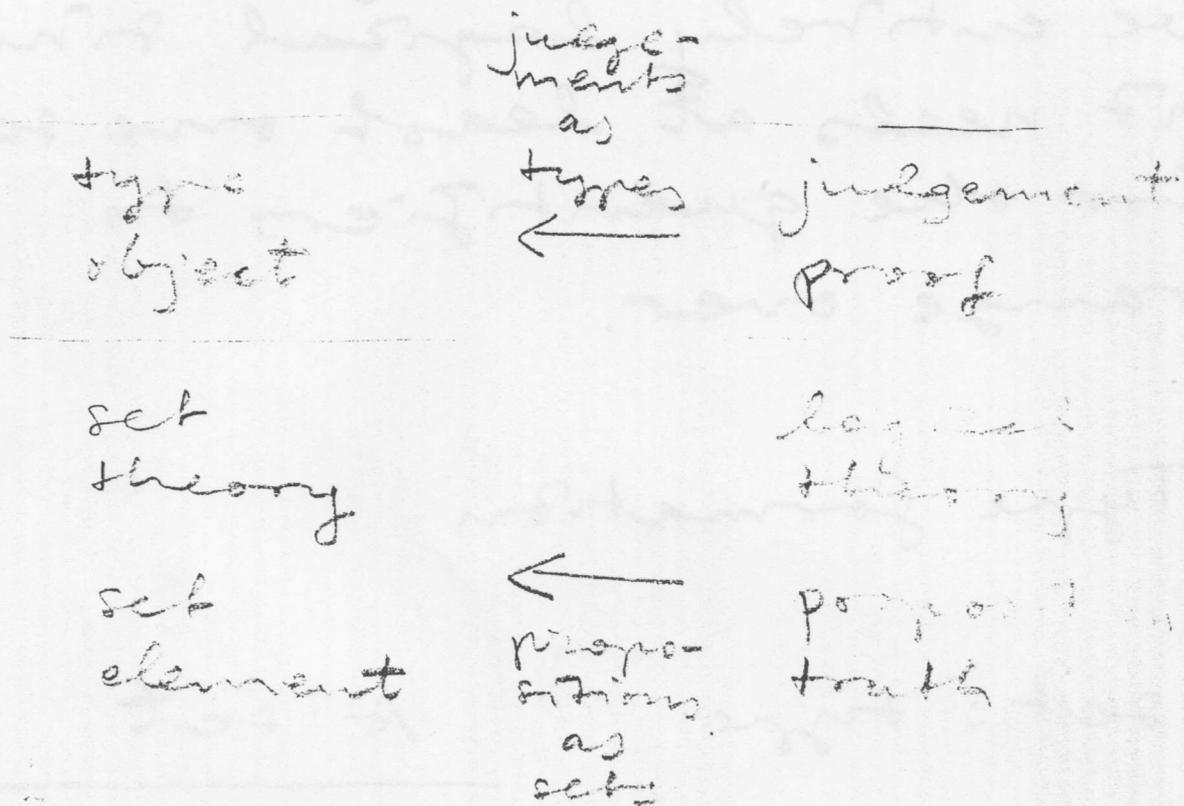


22.2.1987

The Logic of Judgements

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



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

Peter Schroeder-Heister, Judge-
 ments of higher levels and
 standardized rules for lo-
 gical constants in Martin-
 Löf's theory of logiz, June
 1985

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

Type formation

set : type

A : set

elem(A) : type

A

α : type $(x : \alpha)$
 β : type

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

$(x : \alpha) \beta = [x : \alpha] \beta = \prod x : \alpha. \beta$

AUTOMATH
 CONSTRUCTIONS

LF

↓

$(\alpha)\beta = (x:\alpha)\beta$ $\neg \exists \beta$ does not depend on x

Object formation

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

$$(x:\alpha)$$

$$\frac{b = \beta}{(x)b = (x:\alpha)\beta} \quad (\text{abstraction})$$

LF



$$(x)b = (x:\alpha)\beta$$

$$\lambda x:\alpha. b$$

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

Equality

$$(x:\alpha)$$

$$a = \alpha$$

$$b = \beta$$

$$\frac{a = \alpha \quad b = \beta}{((x)b)(a) = b(a/x) : \beta(a/x)} \quad (\beta)$$

$$c = (x:\alpha)\beta$$

$$c = (x)c(x) : (x:\alpha)\beta$$

(η)

refl., symm., trans.
equals for equals, spec.

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

$$a : \beta$$

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

$$a : \text{elem}(B)$$

$$\alpha : \text{type} \quad \alpha = \beta : \text{type}$$

$$a : \alpha \quad a = b : \alpha$$

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

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

and $a = b : \alpha$ by

$$a, b : \alpha, \quad a = b \beta \eta$$

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) ex-

expressions are definitionally equal (have the same meaning).

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

$\text{prop} : \text{type}$

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

A

In the propositions as sets interpretation, we put

$$\text{prop} = \text{set} : \text{type}$$
$$\text{proof}(A) = \text{elem}(A) : \text{type}$$

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

Judgement formation

$$\frac{A : \text{prop}}{A \text{ true} : \text{judg}}$$

true(A)
A

$$\frac{I : \text{judg} \quad J : \text{judg}}{I | J : \text{judg}}$$

I | J = judg

→ (Gentzen)

⇒ (Schroeder-Heister)

⊢ (LF)

$$\frac{\alpha : \text{type} \quad J : \text{judg}}{|x = \alpha} \quad (x : \alpha)$$

|x = α J : judg

→ x = α

⇒ x = α

⊢ x = α

Proof rules

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

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

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

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

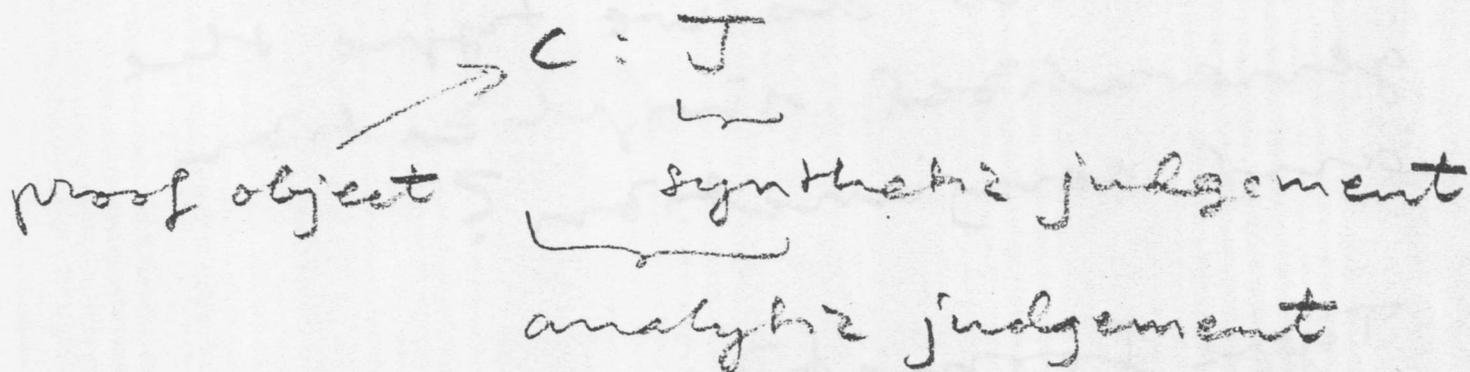
$$\text{judg} = \text{type}$$

$$\text{true}(A) = \text{proof}(A)$$

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

$$|x:\alpha J = (x:\alpha) J$$

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



$$\supset : (X : \text{prop}) (X \mid \text{prop}) \text{prop}$$

$$\& \quad \begin{array}{l} X \text{ true} \\ \text{true}(X) \end{array}$$

Object formation

$$\begin{array}{l} \text{(I)} \\ \frac{\alpha : \beta}{\alpha : \text{I} \mid \beta} \end{array} \quad \frac{\alpha : \text{I} \mid \beta \quad \text{I}}{\alpha : \beta}$$

$$\frac{\begin{array}{l} (A \text{ true}) \\ B : \text{prop} \end{array}}{\frac{A : \text{prop} \quad B : A \text{ true} \mid \text{prop}}{\quad}}$$

$$A \supset B = \text{prop}$$

$$\&$$