

A Logical Framework with Dependently Typed Records

Thierry Coquand, Randy Pollack, Makoto Takeyama

April 16, 2003 (809)

A Logical Framework with Dependently Typed Records

**Thierry Coquand,
Robert Pollack,
Makoto Takeyama**

April 16, 2003

Slide 1

Slide 2

Long Term Goal:

Precise explanation of mathematical vernacular.

- Experiment with dependently typed records as a notation for mathematical structure.
 - Manifest fields
 - Structural subtyping
 - ‘with’ notation to add information to signatures
 - Signature strengthening

$$\mathit{ringSig} = \langle \mathbf{G}:\mathit{grpSig}, \mathbf{M}:\mathit{monSig} \text{ with } \mathit{crr}=\mathbf{G.crr}, \dots \rangle.$$

- Typed equality.
 - η and surjective pairing
 - respects subtyping
- Direct and simple implementation.

Slide 3

This talk

Restrict to a simple core similar to Martin-Löf’s framework.

- PER semantics
 - **Categorical judgements** of Type Theory s.t. (up to η -expansion)
 - * Well-typed objects are normalising
 - * Equality is decidable
 - Extend to **hypothetical judgements** of Type Theory
- **Implementable rules** for a logical framework.
- Extend core system to **subtyping**.

This core framework can be extended with definitions, structures and signatures, manifest fields, ... See our paper.

Syntactic Objects and Syntactic Types

- x, y , range over *identifiers*, \mathcal{I} .
- *objects*, \mathcal{O} , are *untyped* λ -terms.

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

- Equality on objects is β -conversion (\simeq).
- An object is *neutral* iff it is normalisable and of the form

$$\nu ::= x \mid \nu M.$$

- The category of syntactic *types*

$$A, B ::= El M \mid \text{fun } A x.B \mid \star$$

- Objects in \star are “names” of types;
- for $M : \star$, $El M$ is the type named by M .

Slide 4

Categorical Judgements: Outline

- Simultaneously define
 1. A PER of *intensional equality* on the set of syntactic types.
Write $A = B$.
Write $A \in \text{Type}$ for $A = A$.
 2. For $A \in \text{Type}$, a PER, \overline{A} , on objects.
Write $M = N : \overline{A}$.
Write $M : \overline{A}$ for $M = M : \overline{A}$.
- Define eta-expansion
- Key normalization theorem.

Slide 5

Slide 6

Categorical Judgements: the definition

- $\star = \star$.
 $M = N : \bar{\alpha}$ iff M and N are neutral and β -convertible.
- $El M = El N$ whenever $M = N : \bar{\alpha}$.
 $N_1 = N_2 : \overline{El M}$ iff N_1 and N_2 are neutral and β -convertible.
- $\text{fun } A_1 x_1. B_1 = \text{fun } A_2 x_2. B_2$ whenever
 - $A_1 = A_2$,
 - $M_1 = M_2 : \overline{A_1} \implies B_1[M_1] = B_2[M_2]$. $M_1 = M_2 : \overline{\text{fun } A x. B}$ iff
 $N_1 = N_2 : A \implies M_1 N_1 = M_2 N_2 : \overline{B[M_1]}$.

If $A = B$ then \overline{A} and \overline{B} are extensionally equal.

The converse is not true.

Slide 7

Eta-Expansion

Define an operation of η -expansion at type A (written $\eta\{A\}$).

$M : \overline{A}$ will imply $\eta\{A\}M$ is normalising.

$$\begin{aligned} \eta\{\star\} &= \lambda x. x \\ \eta\{El M\} &= \lambda x. x \\ \eta\{\text{fun } A x. B\} &= \lambda u. \lambda z. \eta\{B[\hat{z}]\} (u \hat{z}) \quad \text{where } \hat{z} = \eta\{A\}z \end{aligned}$$

where u and z are distinct and not free in $\eta\{A\}$ or B .

Slide 8

Key Theorem: Normalization

Theorem Let $A \in \text{Type}$.

1. $\eta\{A\}\nu : \overline{A}$, where ν is neutral.
2. If $M : \overline{A}$ then $\eta\{A\}M$ is normalisable.
3. If $M : \overline{A}$ then $M = \eta\{A\}M : \overline{A}$.
4. If $M_1 = M_2 : \overline{A}$ then $\eta\{A\}M_1 \simeq \eta\{A\}M_2$.

The four parts are proved simultaneously by induction on the proof that $A \in \text{Type}$.

Corollary Let $A \in \text{Type}$, $M_1 : \overline{A}$ and $M_2 : \overline{A}$.

1. If $\eta\{A\}M_1 \simeq \eta\{A\}M_2$ then $M_1 = M_2 : \overline{A}$.
2. The relation $M_1 = M_2 : \overline{A}$ is decidable.

Slide 9

Hypothetical Judgements: Outline

- Introduce *environments*, ρ , *contexts*, C .
- Define a judgement $\rho_1 = \rho_2 : C$.
- Simultaneously define hypothetical judgements
 - C **valid**,
 - $A_1 = A_2 [C]$,
 - $M_1 = M_2 : A [C]$.

For C **valid**, $\rho_1 = \rho_2 : C$, $A_1 = A_2 [C]$ and $M_1 = M_2 : A [C]$ are PERs.

- Show that hypothetical judgements satisfy the rules of Type Theory.
- Give conditions for $M_1 = M_2 : A [C]$ to be decidable.
- Define a relation $C \vdash A_1 = A_2$ of *syntactic type equality* that is decidable and sound for the semantic relation $A_1 = A_2 [C]$.

Contexts and Environments

Contexts

$C ::= \nabla \mid C, x:A$ (∇ is the empty context.)

Write $x \in C$ if $x:A$ in C for some A .

In writing $C, x:A$ we assume $x \notin C$.

Environments

- An environment, ρ , is a function $\mathcal{I} \rightarrow \mathcal{O}$.
- ρ_0 is the identity environment.
- Environments are applied as simultaneous substitutions: $M\rho$, $A\rho$.
- Write $(\rho, x=M)$ for the *update* of ρ , defined by

$$(\rho, x=M)(x) = M, \quad (\rho, x=M)(y) = \rho(y) \text{ if } y \neq x.$$

Slide 10

Equal Environments

Inductively define a judgement of form $\rho_1 = \rho_2 : C$:

$$\frac{\overline{\rho_1 = \rho_2 : \nabla}}{\frac{\rho_1 = \rho_2 : C \quad A\rho_1 \in \text{Type} \quad \rho_1 x = \rho_2 x : \overline{A\rho_1}}{\rho_1 = \rho_2 : C, x:A}}$$

Write $\rho : C$ for $\rho = \rho : C$.

Slide 11

Hypothetical Judgements Defined

Simultaneously define three judgement forms:

validity

$$\frac{}{\nabla \text{ valid}} \quad \frac{x \notin C \quad A = A [C]}{C, x:A \text{ valid}}$$

type equality (write $A \text{ type } [C]$ for $A = A [C]$)

$$\frac{C \text{ valid} \quad \forall \rho_1, \rho_2 . \rho_1 = \rho_2 : C \implies A_1 \rho_1 = A_2 \rho_2}{A_1 = A_2 [C]}$$

object equality in a type (write $M : A [C]$ for $M = M : A [C]$)

$$\frac{A \text{ type } [C] \quad \forall \rho_1, \rho_2 . \rho_1 = \rho_2 : C \implies M_1 \rho_1 = M_2 \rho_2 : \overline{A} \rho_1}{M_1 = M_2 : A [C]}$$

Slide 12

Properties of Hypothetical Judgements

type formation and type equality

$$\frac{C \text{ valid} \quad M = N : \star [C] \quad A_1 = A_2 [C] \quad B_1 = B_2 [C, x:A_1]}{\star \text{ type } [C] \quad El M = El N [C] \quad \text{fun } A_1 x.B_1 = \text{fun } A_2 x.B_2 [C]}$$

objects

$$\frac{C, x:A \text{ valid}}{x : A [C, x:A]} \quad \frac{M : B [C, x:A]}{\lambda x.M : \text{fun } A x.B [C]}$$

$$\frac{M : \text{fun } A x.B [C] \quad N : A [C]}{M N : B[N] [C]}$$

type conversion

$$\frac{M = N : A [C] \quad A = B [C]}{M = N : B [C]}$$

weakening

$$\frac{B_1 = B_2 [C] \quad C, x:A \text{ valid}}{B_1 = B_2 [C, x:A]} \quad \frac{M = N : B [C] \quad C, x:A \text{ valid}}{M = N : B [C, x:A]}$$

Slide 13

Deciding Hypothetical Equality

- We saw that if $M_1, M_2 : \overline{A}$ then $M_1 = M_2 : \overline{A}$ is decidable.
- Lift this property to hypothetical judgements:
 - need the “most general environment”,
 - intuitively, the identity environment, ρ_0 .
- By the Key Theorem, we need to η -expand ρ_0 at C :

Slide 14

$$\eta\{\nabla\}\rho = \rho$$

$$\eta\{C, x:A\}\rho = (\rho', x=\eta\{A\rho'\}(\rho x)) \quad \text{where } \rho' = \eta\{C\}\rho$$

Lemma Write ρ_C for $\eta\{C\}\rho_0$.

- If C valid then $\rho_C : C$. • $\rho_C \circ \rho = \eta\{C\}\rho$.
- If $M_1 : A [C]$ and $M_2 : A [C]$ then

$$M_1 = M_2 : A [C] \quad \text{iff} \quad \eta\{A\rho_C\}M_1\rho_C \simeq \eta\{A\rho_C\}M_2\rho_C$$

which is decidable.

Approximating Type Equality

Define a syntactic relation of shape $C \vdash A_1 = A_2$ which is decidable and sound for the semantic relation $A_1 = A_2 [C]$.

Slide 15

$$\frac{}{C \vdash \star = \star} \quad \frac{M_1 = M_2 : \star [C]}{C \vdash \mathit{El} M_1 = \mathit{El} M_2}$$

$$\frac{C \vdash A_1 = A_2 \quad C, x:A_1 \vdash B_1 = B_2}{C \vdash \text{fun } A_1 x.B_1 = \text{fun } A_2 x.B_2}$$

Lemma

1. If A_1 **type** $[C]$ and A_2 **type** $[C]$ then $C \vdash A_1 = A_2$ is decidable.
2. If C **valid** and $C \vdash A_1 = A_2$ then $A_1 = A_2 [C]$.

A Logical Framework in Syntax: Outline

We give a concrete implementation.

- Syntax of expressions and expression contexts
- Judgement forms
- Typechecking rules
 - syntax directed
 - terminating
 - sound

Slide 16

Expressions

The syntax of expressions and expression contexts:

$$\begin{aligned}
 e & ::= z \mid e e \mid [z:e]e \mid * \mid E! e \mid \{z:e\}e \mid e \rightarrow e \\
 \Gamma & ::= \blacktriangledown \mid \Gamma, x:e \quad (\blacktriangledown \text{ is the empty context})
 \end{aligned}$$

Judgement Forms

Two judgement forms are defined simultaneously,

- $C \vdash e \Rightarrow A$, (expression e is interpreted in C as type A)
- $C \vdash e \Rightarrow M : A$,
(expression e is interpreted in C as object M having type A)

while a third can be defined afterwards.

- $\Gamma \Rightarrow C$, (Γ is interpreted as the valid context C)

Slide 17

Typechecking

type formation

$$\frac{}{C \vdash * \Rightarrow *} \quad \frac{C \vdash e \Rightarrow M : *}{C \vdash \text{El } e \Rightarrow \text{El } M} \quad \frac{C \vdash e_1 \Rightarrow A \quad C \vdash e_2 \Rightarrow B}{C \vdash e_1 \rightarrow e_2 \Rightarrow A \rightarrow B}$$

$$\frac{C \vdash e_1 \Rightarrow A \quad C, x:A \vdash e_2 \Rightarrow B}{C \vdash \{x:e_1\}e_2 \Rightarrow \text{fun } A \ x.B}$$

Slide 18

objects

$$\frac{x:A \text{ in } C}{C \vdash x \Rightarrow x : A} \quad \frac{C \vdash e_1 \Rightarrow A \quad C, x:A \vdash e_2 \Rightarrow M : B}{C \vdash [x:e_1]e_2 \Rightarrow \lambda x.M : \text{fun } A \ x.B}$$

$$\frac{C \vdash e_1 \Rightarrow M_1 : \text{fun } A_1 \ x.B \quad C \vdash e_2 \Rightarrow M_2 : A_2 \quad C \vdash A_1 = A_2}{C \vdash e_1 e_2 \Rightarrow M_1 M_2 : B[M_2]}$$

validity

$$\frac{}{\blacktriangledown \Rightarrow \nabla} \quad \frac{\Gamma \Rightarrow C \quad C \vdash e \Rightarrow A}{\Gamma, x:e \Rightarrow C, x:A}$$

Correctness and Termination of Typechecking

- If $C \vdash a \Rightarrow A$ and C valid then A type $[C]$.
- If $C \vdash e \Rightarrow M : A$ and C valid then $M : A [C]$.
- If $\Gamma \Rightarrow C$ then C valid.
- Typechecking is decidable.

Slide 19

“Eta-conversion” of expressions

- Suppose $v : \{A:*\}\text{El } A$.
- The expressions v and $[A:*\](v \ A)$ are indistinguishable by typechecking.

Subtyping: A Top Type

For a simple example of subtyping we add a top type, $\mathbf{1}$, with $M = N : \bar{\mathbf{1}}$ for any M, N .

- Extend objects and syntactic types

$$\begin{aligned} M & ::= x \mid M M \mid \lambda x.M \mid () \\ A & ::= El M \mid \text{fun } A x.B \mid \star \mid \mathbf{1} \end{aligned}$$

Slide 20

- Categorical judgement: $\mathbf{1} = \mathbf{1}$. $\bar{\mathbf{1}}$ is $\mathcal{O} \times \mathcal{O}$.
- $\eta\{\mathbf{1}\} = \lambda u.()$
- Normalization theorem still holds.
- Derivable properties

$$\frac{C \text{ valid}}{\mathbf{1} \text{ type } [C]} \quad \frac{C \text{ valid}}{M = N : \mathbf{1} [C]}$$

Subtyping

- **Categorical Subtype** Written $A \sqsubseteq B$, is defined

$$A \sqsubseteq B = \bar{A} \subseteq \bar{B} \quad (\text{extensionally subrelation})$$

- **Hypothetical Subtype** Written $A \sqsubseteq B [C]$, is defined

$$A \sqsubseteq B [C] = A \text{ type } [C] \wedge B \text{ type } [C] \wedge \forall \rho : C . A\rho \sqsubseteq B\rho$$

Slide 21

- **Subsumption (Subtype Conversion) Rule** is derivable

$$\frac{M = N : A [C] \quad A \sqsubseteq B [C]}{M = N : B [C]}$$

Syntactic Subtyping

$$\frac{}{C \vdash \star \sqsubseteq \star} \quad \frac{M_1 = M_2 : \star [C]}{C \vdash El M_1 \sqsubseteq El M_2} \quad \frac{}{C \vdash A \sqsubseteq 1}$$

$$\frac{C \vdash A_2 \sqsubseteq A_1 \quad C, x:A_2 \vdash B_1 \sqsubseteq B_2}{C \vdash \text{fun } A_1 x.B_1 \sqsubseteq \text{fun } A_2 x.B_2}$$

Slide 22

Syntactic subtyping is decidable and sound.

Lemma If A_1 **type** $[C]$ and A_2 **type** $[C]$ then $C \vdash A_1 \sqsubseteq A_2$ is decidable and implies $A_1 \sqsubseteq A_2 [C]$.

Typechecking Expressions with Subtyping

Extend expression language with top, <> and ().

$$e ::= z \mid e e \mid [z:e]e \mid * \mid El e \mid \langle \rangle \mid ()$$

Replace the typechecking rule for applications

$$\frac{C \vdash e_1 \Rightarrow M_1 : \text{fun } A_1 x.B \quad C \vdash e_2 \Rightarrow M_2 : A_2 \quad C \vdash A_2 \sqsubseteq A_1}{C \vdash e_1 e_2 \Rightarrow M_1 M_2 : B[M_2]}$$

Slide 23

New typechecking rules for <> and ().

$$\frac{}{C \vdash \langle \rangle \Rightarrow 1} \quad \frac{}{C \vdash () \Rightarrow () : 1}$$

Typechecking is terminating and sound.

Eta-conversion for <>

Slide 24

Other Features

- Definitions using singleton types
- Type coercions
- First class signatures and records with manifest fields
- The *with* notation for manifest fields
- Type families
- Type abbreviations
- Inductive sets.