Untyped Algorithmic Equality for Martin-Löf's Logical Framework with Surjective Pairs

Andreas Abel

joint work with Thierry Coquand

Slide 1

TLCA'05
Nara, Japan
April 21, 2005

Work supported by: TYPES & APPSEM-II (EU), CoVer (SSF)

Background: $\beta\eta$-equality

Slide 2

- Checking dependent types requires equality test
- One approach: reduce to normal form and compare syntactically
- Works fine for β-equality
- Problem with η-reduction: surjective pairing destroys confluence (Klop 1980)
- Even subject reduction fails:

$$z : \mathsf{Pair}\, A\, (\lambda x.\, F\, x) \;\vdash\; (z\, \mathsf{L},\, z\, \mathsf{R}) \;:\; \mathsf{Pair}\, A\, (\lambda\_.\, F\, (z\, \mathsf{L}))$$

[I write $\mathsf{Pair}\, A\, (\lambda x B)$ for $\Sigma x \!:\! A.\, B$]

Thierry's Equality Algorithm

Slide 3

- Incremental check for $\beta\eta$-equality in dependently-typed λ-calculus (Coquand 1991)
- Alternates weak head normalization and comparison of head symbols
- We extend this algorithm to Σ-types with surjective pairing
- Challenge: termination and completeness
- Two major technical difficulties to overcome

Martin-Löf's Logical Framework (MLF)

Slide 4

- Expressions = Curry-style λ-terms

c	$::=$	$\mathsf{Fun} \mid \mathsf{El} \mid \mathsf{Set}$	constants
r, s, t, A, B, C	$::=$	$c \mid x \mid \lambda x t \mid r\, s$	expressions

- Examples

$\mathsf{Fun}\, A\, (\lambda x B)$	dependent function space $\Pi x\!:\!A.\, B$
$\mathsf{Fun}\ \mathsf{Set}\ (\lambda a.\ \mathsf{Fun}\ (\mathsf{El}\, a)\ (\lambda\_.\ \mathsf{El}\, a))$	type of identity: $\forall a\!:\!*.\, a \to a$

Martin-Löf's logical framework

- Judgements for typing and equality, e.g.,

$$\begin{array}{ll} \Gamma \vdash t : A & t \text{ has type } A \\ \Gamma \vdash t = t' : A & t \text{ and } t' \text{ are equal terms of type } A \end{array}$$

Slide 5

- Example: application rule

$$\frac{\Gamma \vdash r : \mathsf{Fun}\, A\, (\lambda x B) \qquad \Gamma \vdash s : A}{\Gamma \vdash r\, s : B[s/x]}$$

Weak head evaluation

- Weak head values

$$\begin{array}{lcll} n & ::= & c\,\vec{t} \mid x\,\vec{t} & \text{neutral expressions} \\ w & ::= & n \mid \lambda x t & \text{weak head values} \end{array}$$

Slide 6

- Weak head evaluation (call-by-name)

$$\begin{array}{rcll} (r\,s)\!\downarrow & := & r\!\downarrow\!@s & \\ t\!\downarrow & := & t & t \text{ not application} \\ & & & \\ n@s & := & n\,s & \\ (\lambda x t)@s & := & (t[s/x])\!\downarrow & \end{array}$$

Untyped Algorithmic Equality

- $\beta\eta$-conversion test for weak head values $w \sim w'$
- Two neutral expressions

$$\frac{}{c \sim c} \qquad \frac{}{x \sim x} \qquad \frac{n \sim n' \qquad s{\downarrow} \sim s'{\downarrow}}{n\, s \sim n'\, s'}$$

Slide 7

- At least one λ

$$\frac{t{\downarrow} \sim t'{\downarrow}}{\lambda x t \sim \lambda x t'} \qquad \frac{t{\downarrow} \sim n\, x}{\lambda x t \sim n} \qquad \frac{n\, x \sim t'{\downarrow}}{n \sim \lambda x t'}$$

- Relation $\sim$ is transitive
- Completeness to be shown by model construction

Lambda Model

- Entities

u, v, f, V, F	$\in$	D	elements of the model
ρ	$\in$	$\mathsf{Var} \to \mathsf{D}$	environments

Slide 8

- Operations

$f \cdot v$	$\in$	D	application in the model
$t\rho$	$\in$	D	denotation of expression t in environment ρ

Lambda Model Axiomatization

Slide 9

Computation (β)

$$(\lambda x t)\rho \cdot v \quad = \quad t(\rho, x\!=\!v)$$

Congruences

$$\begin{array}{rcl} c\rho & = & c \\ x\rho & = & \rho(x) \\ (r\,s)\rho & = & r\rho \cdot (s\rho) \end{array}$$

Injectivity

$$\begin{array}{rcll} \mathsf{El} \cdot v & = & \mathsf{El} \cdot v' & \text{implies } v = v' \\ \mathsf{Fun} \cdot V \cdot F & = & \mathsf{Fun} \cdot V' \cdot F' & \text{implies } V = V' \text{ and } F = F' \end{array}$$

PER Model

Slide 10

- Assume a basic partial equivalence relation (PER) $\mathcal{S}$ on D
- Interpretation of *types* in D as sub-PERs of $\mathcal{S}$

$$\begin{array}{lcl} [\mathsf{Set}] & = & \mathcal{S} \\ [\mathsf{El} \cdot v] & = & \mathcal{S} \\ [\mathsf{Fun} \cdot V \cdot F] & = & \{(f, f') \mid (f \cdot v, f' \cdot v') \in [F \cdot v] \text{ for all } (v, v') \in [V]\} \end{array}$$

- Soundness of typing and equality rules

  If $\Gamma \vdash t \quad : A$ then $(t\,\rho,\ t\,\rho) \in [A\rho]$ for all $\rho \in [\Gamma]$.
  If $\Gamma \vdash t = t' : A$ then $(t\,\rho,\ t'\rho) \in [A\rho]$ for all $\rho \in [\Gamma]$.

- Implication: $(t\,\rho,\ t'\rho) \in \mathcal{S}$

Substitution and Extensionality

Slide 11

- Difficulty 1: Soundness proof of application rule

$$\frac{\Gamma \vdash r : \mathsf{Fun}\, A\, (\lambda x B) \qquad \Gamma \vdash s : A}{\Gamma \vdash r\, s : B[s/x]}$$

- requires substitution property

$$(B[s/x])\rho = B(\rho, x{=}s\rho).$$

- Hence, model needs additional axiom

$$\begin{array}{ll} (\xi) & (\lambda x t)\rho = (\lambda x t')\rho' \\ & \quad \text{if } t(\rho, x{=}v) = t'(\rho', x{=}v) \text{ for all } v \in \mathsf{D} \end{array}$$

Completeness of Algorithmic Equality

Slide 12

- Recall: $\vdash t = t' : A$ implies $(t, t') \in \mathcal{S}$
- Take model instance

$$\begin{array}{lcl} \mathsf{D} & = & \beta\text{-equivalence classes} \\ f \cdot v & = & \overline{f\, v} \\ t\rho & = & \overline{t[\rho]} \\ \mathcal{S} & = & \text{lifted algorithmic equality } \sim \end{array}$$

- algorithmic equality on β-equivalence classes

$$\bar{t} \sim \overline{t'} \;\; :\Longleftrightarrow \;\; t =_\beta v \text{ and } t' =_\beta v' \text{ for some } v, v' \text{ with } v \sim v'$$

Standardization

- Using standardization, $\bar{t} \sim \overline{t'}$ implies $t\!\downarrow \sim t'\!\downarrow$.
- Summary (ρ_0 is identity valuation):

Slide 13

$$\begin{array}{c} \Gamma \vdash t = t' : A \\ \Big\Downarrow \text{Soundness of judgement} \\ (t\rho_0, t'\rho_0) \in [A\rho_0] \\ \Big\Downarrow [A\rho_0] \subseteq \mathcal{S} \\ \bar{t} \sim \overline{t'} \\ \Big\Downarrow \text{Standardization} \\ t\!\downarrow \sim t'\!\downarrow \end{array}$$

Extension to Σ-types

- Expressions

c	$::=$	$\cdots \mid \mathsf{Pair}$	constants
r, s, t, A, B, C	$::=$	$\cdots \mid (r, s) \mid t\,\mathsf{L} \mid t\,\mathsf{R}$	expressions

Slide 14

- Example: $\mathsf{Pair}\, A\, (\lambda x B)$ dependent type of pairs $(\Sigma x\!:\!A.\, B)$
- Surjective pairing rule

$$\frac{\Gamma \vdash r = r' : \mathsf{Pair}\, A\, (\lambda x B)}{\Gamma \vdash (r\,\mathsf{L},\ r\,\mathsf{R}) = r' : \mathsf{Pair}\, A\, (\lambda x B)}$$

Extended Algorithmic Equality

Slide 15

- Neutral expressions

$$\frac{n \sim n'}{n\,\mathsf{L} \sim n'\,\mathsf{L}} \qquad \frac{n \sim n'}{n\,\mathsf{R} \sim n'\,\mathsf{R}}$$

- At least one pair

$$\frac{r{\downarrow} \sim r'{\downarrow} \qquad s{\downarrow} \sim s'{\downarrow}}{(r, s) \sim (r', s')}$$

$$\frac{r{\downarrow} \sim n\,\mathsf{L} \qquad s{\downarrow} \sim n\,\mathsf{R}}{(r, s) \sim n} \qquad \frac{n\,\mathsf{L} \sim r'{\downarrow} \qquad n\,\mathsf{R} \sim s'{\downarrow}}{n \sim (r', s')}$$

Transitivity

Slide 16

- Problem 2: Alg. Eq. not transitive
- $\lambda x.\, z\, x \sim z$ and $z \sim (z\,\mathsf{L}, z\,\mathsf{R})$, but *not* $\lambda x.\, z\, x \sim (z\,\mathsf{L}, z\,\mathsf{R})$
- Solution: "Transitivization" $\overset{+}{\sim}$ through additional rules

$$\frac{t{\downarrow} \overset{+}{\sim} n\, x \qquad n\,\mathsf{L} \overset{+}{\sim} r \qquad n\,\mathsf{R} \overset{+}{\sim} s}{\lambda x t \overset{+}{\sim} (r, s)}$$

+ symmetrical rule

- If t, t' are of the same type, $t \overset{+}{\sim} t'$ does not use extra rules.
- Equality *is* transitive for expressions of the same type

Slide 17

Summary of Completeness Proof

$$\begin{array}{c}
\Gamma \vdash t = t' : A \\
\Big\Downarrow \text{ Soundness of judgement} \\
(t\rho_0, t'\rho_0) \in [A\rho_0] \\
\Big\Downarrow \; [A\rho_0] \subseteq \mathcal{S} \\
\overline{t} \overset{+}{\sim} \overline{t'} \\
\Big\Downarrow \text{ Standardization} \\
t\!\downarrow \overset{+}{\sim} t'\!\downarrow \\
\Big\Downarrow \text{ Transitivity (with } \Gamma \vdash t, t' : A) \\
t\!\downarrow \sim t'\!\downarrow
\end{array}$$

Slide 18

Proof Economics

Injectivity	required
Inversion of typing	required
Standardization	required
Subject reduction	not required
Confluence (Church-Rosser)	not required
Normalization	not required
Certificate	good economics!

Related Work

Slide 19

- Vaux (2004): PER model for MLF with intersection
- Aspinall/Hofmann (TAPL II), Goguen (2005): completeness of algorithmic equality using standard meta theory
- Coquand, Pollack, and Takeyama (2003): extension of MLF by records with manifest fields
- Harper and Pfenning (2005): algorithmic equality for ELF directed by simple types (obtained by erasure)
- Schürmann and Sarnat (2004): extension to Σ-types
- Adams (2001): Luo's LF with Σ-kinds and type-directed equality

Future Work

Slide 20

- Logical framework with proof-irrelevant propositions
- Type-directed equality *without* erasure
- An open problem?!

Thanks to Frank Pfenning, Carsten Schürmann, and Lionel Vaux