# 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 $\beta$-equality
- Problem with $\eta$-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 $\lambda$-calculus (Coquand 1991)
- Alternates weak head normalization and comparison of head symbols
- We extend this algorithm to $\Sigma$-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 $\lambda$-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

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

**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

**Slide 7**

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

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

- At least one $\lambda$

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

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

## Lambda Model

**Slide 8**

- Entities

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

- Operations

| | | | |
|---|---|---|---|
| $f \cdot v$ | $\in$ | $\mathsf{D}$ | application in the model |
| $t\rho$ | $\in$ | $\mathsf{D}$ | denotation of expression $t$ in environment $\rho$ |

## Lambda Model Axiomatization

**Slide 9**

**Computation ($\beta$)**

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

**Congruences**

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

**Injectivity**

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

## PER Model

**Slide 10**

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

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

- Soundness of typing and equality rules

$$\begin{aligned} &\text{If } \Gamma \vdash t \quad\;\; : A \text{ then } (t\,\rho,\ t\,\rho) \in [A\rho] \text{ for all } \rho \in [\Gamma]. \\ &\text{If } \Gamma \vdash t = t' : A \text{ then } (t\,\rho,\ t'\rho) \in [A\rho] \text{ for all } \rho \in [\Gamma]. \end{aligned}$$

- 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

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

## 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 $\beta$-equivalence classes

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

## Standardization

**Slide 13**

- Using standardization, $\overline{t} \sim \overline{t'}$ implies $t{\downarrow} \sim t'{\downarrow}$.
- Summary ($\rho_0$ is identity valuation):

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

## Extension to Σ-types

**Slide 14**

- Expressions

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

- 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

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

**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

- 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 $\Sigma$-types
- Adams (2001): Luo's LF with $\Sigma$-kinds and type-directed equality

**Slide 19**

## Future Work

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

**Slide 20**

*Thanks to Frank Pfenning, Carsten Schürmann, and Lionel Vaux*