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

Andreas Abel

*joint work with Thierry Coquand*

**Slide 1**

ProgLog Seminar
Chalmers, Göteborg, Sweden
May 11, 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

$$\begin{array}{lcll} c & ::= & \mathsf{Fun} \mid \mathsf{El} \mid \mathsf{Set} & \text{constants} \\ r, s, t & ::= & c \mid x \mid \lambda x t \mid r\, s & \text{expressions} \\ A, B, C & ::= & \mathsf{Set} \mid \mathsf{El}\, t \mid \mathsf{Fun}\, A\, (\lambda x B) & \text{types} \end{array}$$

- Examples

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

## Martin-Löf's logical framework

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

| | |
|---|---|
| $\Gamma \vdash t : A$ | $t$ has type $A$ |
| $\Gamma \vdash t = t' : A$ | $t$ and $t'$ are equal terms of type $A$ |

- Example: application rule

**Slide 5**

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

- $\beta$- and $\eta$-rules

$$\frac{\Gamma, x\!:\!A \vdash t = t' : B \qquad \Gamma \vdash s = s' : A}{\Gamma \vdash (\lambda x t)\, s = t'[s'/x] : B[s/x]}$$

$$\frac{\Gamma \vdash t = t' : \mathsf{Fun}\, A\, (\lambda x B)}{\Gamma \vdash (\lambda x.\, t\, x) = t' : \mathsf{Fun}\, A\, (\lambda x B)}\; x \notin \mathsf{FV}(t)$$

## Lambda Model

- Entities

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

**Slide 6**

- 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 7**

**Computation** ($\beta$)

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

**Congruences**

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

**Injectivity**

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

## PER Model

**Slide 8**

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

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

- 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

- Difficulty 1: Soundness proof of application rule

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

**Slide 9**

- requires substitution property $(B[s/x])\rho = B(\rho, x\!=\!s\rho)$.
- Hence, model needs additional axiom

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

- Also gives *irrelevance* $t(\rho, x\!=\!v) = t\rho$ if $x \notin \mathsf{FV}(t)$, needed for $\eta$.

## 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 10**

- Weak head evaluation (call-by-name)

$$\begin{array}{rcll} (r\,s){\downarrow} & := & r{\downarrow}@s & \\ t{\downarrow} & := & t & t \text{ not application} \end{array}$$

$$\begin{array}{rcl} n@s & := & n\,s \\ (\lambda x t)@s & := & (t[s/x]){\downarrow} \end{array}$$

## Untyped Algorithmic $\beta\eta$-Equality

**Slide 11**

- $\beta\eta$-conversion test for normalizable 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 a PER

## Transitivity of Algorithmic Equality

**Slide 12**

- Lemma:
  1. If $\mathcal{D}_1 :: w \sim n\,\vec{x}$ and $\mathcal{D}_2 :: n \sim n'$ then $w \sim n'\,\vec{x}$ (plus symmetrical proposition).
  2. If $\mathcal{D}_1 :: w_1 \sim w_2$ and $\mathcal{D}_2 :: w_2 \sim w_3$ then $w_1 \sim w_3$.
- Proof by simultaneous induction on $\mathcal{D}_1$ and $\mathcal{D}_2$.
- 1. is needed for the following case of 2.

$$\mathcal{D}_1 = \frac{\begin{array}{c}\mathcal{D}_1' \\ t{\downarrow} \sim n\,x\end{array}}{\lambda x t \sim n} \qquad \begin{array}{c}\mathcal{D}_2 \\ n \sim n'\end{array}$$

## Completeness of Algorithmic Equality

**Slide 13**

- 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

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

## Standardization

**Slide 14**

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

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

## Extension to $\Sigma$-types

**Slide 15**

- Expressions

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

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

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

## $\eta$-Reduction Destroys Subject Reduction

**Slide 16**

- Pair intro: types of $s$ and $t$ *do not* determine type of $(s, t)$

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

- E.g., if $B[s/x] = \mathsf{Eq}\,A\,s\,s$, then $B \in \{\mathsf{Eq}\,A\,x\,x,\ \mathsf{Eq}\,A\,x\,s,\ \ldots\}$
- Change typing through $\eta$-expansion

$$\frac{\dfrac{z : \mathsf{Pair}\,A\,(\lambda x B)}{z\,\mathsf{L} : A} \qquad \dfrac{z : \mathsf{Pair}\,A\,(\lambda x B)}{z\,\mathsf{R} : B[z\,\mathsf{L}/x]}}{(z\,\mathsf{L},\ z\,\mathsf{R})\ :\ \mathsf{Pair}\,A\,(\lambda\_.\,B[z\,\mathsf{L}/x])}$$

- Subtyping does not solve this problem

**Slide 17**

## Extended Algorithmic Equality

- 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

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

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

**Slide 18**

## Transitivity

- 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 $\sim$ *is* transitive for expressions of the same type

## Proof of Transitivity

**Slide 19**

- Measure # on derivations

$$\textsc{aq}^{+}\textsc{-fun-pair}\;\frac{\begin{array}{ccc}\mathcal{D}_1 & \mathcal{D}_{21} & \mathcal{D}_{22}\\ t{\downarrow} \overset{+}{\sim} n\,x & n\,\mathsf{L} \overset{+}{\sim} r & n\,\mathsf{R} \overset{+}{\sim} s\end{array}}{\lambda xt \overset{+}{\sim} (r, s)}$$

$$\begin{array}{lcl}\#\textsc{aq}^{+}\textsc{-fun-pair}(\mathcal{D}_1, \mathcal{D}_{21}, \mathcal{D}_{22}) & = & 1 + \#\mathcal{D}_1 + \max(\#\mathcal{D}_{21}, \#\mathcal{D}_{22})\\ \#r(\mathcal{D}_1, \ldots, \mathcal{D}_n) & = & 1 + \max\{\#\mathcal{D}_i \mid 1 \leq i \leq n\}\end{array}$$

- Prove simultaneously by induction on $\#\mathcal{D}_1 + \#\mathcal{D}_2$:
  1. If $\mathcal{D}_1 :: w \overset{+}{\sim} n\,\vec{e}$ and $\mathcal{D}_2 :: n \overset{+}{\sim} n'$ then $\mathcal{E} :: w \overset{+}{\sim} n'\,\vec{e}$.
  2. If $\mathcal{D}_1 :: w_1 \overset{+}{\sim} w_2$ and $\mathcal{D}_2 :: w_2 \overset{+}{\sim} w_3$ then $\mathcal{E} :: w_1 \overset{+}{\sim} w_3$.
- In both cases, $\#\mathcal{E} < \#\mathcal{D}_1 + \#\mathcal{D}_2$.

## Summary of Completeness Proof

**Slide 20**

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

**Slide 21**

## 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! |

**Slide 22**

## 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

## Future Work

- Logical framework with proof-irrelevant propositions
- Type-directed equality *without* erasure

**Slide 23**

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