ON THE DECIDABILITY OF THE EQUALITY THEORY OF SIMPLY TYPED λ-CALCULI

Silvio Valentini

Dipartimento di Matematica Pura ed Applicata
Università di Padova

Sunto. Il lavoro presenta una nuova prova del ben noto risultato di decidibilità della teoria dell'ugualianza per un λ-calcolo tipato semplice basata sull'estensione alla teoria dell'ugualianza dell'approccio utilizzato in [Valentini 92] per dimostrare il teorema di forma normale. La novità principale consiste nel fatto che per ottenere tale risultato non si fa ricorso al teorema di Church-Rosser come avviene nella prova usuale. Il metodo di prova usato è molto semplice, ma al tempo stesso si presenta come molto generale e facilmente applicabile ad una varietà di λ-calcoli tipati semplici la cui teoria dell'ugualianza soddisfi la β e la η ugualianza.

Key words and phrases. Typed λ-calculus, Normal-form theorem, Church-Rosser theorem.

Summary. In this paper we are going to illustrate a new method to prove the theorem of decidability of the equality for simply typed λ-calculi (see [Barendregt 81]) which is both straightforward and general. The main novelty is that we are not going to use the Church-Rosser theorem but we will give a direct proof based on a new definition of the equality rules which is directly inspired by the rules we have introduced in [Valentini 92] in order to prove the normal form theorem for a simply typed λ-calculus. The result is then obtained by showing that the new equality theory is equivalent to the standard one.

1. Introduction

The method we want to illustrate to prove the decidability of the equality theory of a typed λ-calculus seems to be of general applicability, at least if the considered equality theory is closed under the β and η equality. Anyhow, in order to make the exposition easier to understand, we will illustrate here the most elementary example in which only function types can be constructed over some basic types[1].

Definition: (Types).

(Basic types) $$\frac{C \text{ basic type}}{C \text{ type}}$$

(Arrow types) $$\frac{\alpha \text{ type} \qquad \beta \text{ type}}{\alpha \to \beta \text{ type}}$$

In this case one can assume that, for any type α, a numerable quantity of variables of type α exist (notation $x : \alpha$) and that it is possible to form the following lambda expressions, in a context which comprises all the variables used in the lambda expression[2], and, at the same time, collect the variables which appear free within them[3].

Definition: (λ-expression).

(var-rule) $$x : \alpha \ \lambda\text{-exp } [\Gamma, x : \alpha] \qquad FV(x) = \{x\}$$

(app-rule)
$$\frac{b : \alpha \to \beta \ \lambda\text{-exp } [\Gamma] \quad a : \alpha \ \lambda\text{-exp } [\Gamma]}{b(a) : \beta \ \lambda\text{-exp } [\Gamma]} \qquad FV(b(a)) = FV(b) \cup FV(a)$$

(abst-rule)
$$\frac{b : \beta \ \lambda\text{-exp } [\Gamma, x : \alpha]}{(\lambda x.b) : \alpha \to \beta \ \lambda\text{-exp } [\Gamma]} \qquad FV((\lambda x.b)) = FV(b) \backslash \{x\}$$

The substitution of a variable with a term within a lambda expression is introduced in the usual way.

[1] A more general approach can be found in [Bossi-Valentini].

[2] We assume that the reader is aware of the standard operations of thinning, contraction and exchange among assumptions in a context.

[3] We use here the same syntax we already used in [Valentini 92], i.e. we write $a : \alpha$ λ-exp $[\Gamma]$ to mean that a is a λ-expression of type α in the context Γ instead of the more usual $\Gamma \vdash a : \alpha$

Definition: (Substitution). Let $x : \alpha$, $e : \alpha$ λ-exp and $b : \beta$ λ-exp, then the substitution of the variable x with the expression e in the expression b (notation $b[x := e]$) is defined by induction on the structural complexity of b:

(variable step) $$z[x := e] \equiv \begin{cases} e & \text{if } z \equiv x \\ z & \text{otherwise} \end{cases}$$

(application step) $$b(a)[x := e] \equiv b[x := e](a[x := e])$$

(abstraction step)

$$(\lambda z.b)[x := e] \equiv \begin{cases} (\lambda x.b) & \text{if } z \equiv x \\ (\lambda z.b[x := e]) & \text{if } z \not\equiv x \text{ and } z \notin FV(e) \\ (\lambda t.b[z := t][x := e]) & \text{otherwise} \\ \text{where } t \text{ is a new variable of the same type of } z \end{cases}$$

Finally we recall that the standard equality between two λ-expressions induced by $\beta\eta$-conversion[4] is the minimal equivalence relation for which the following conditions hold.

Definition: (λ-equality).

(var-equality) $$x = x : \alpha\ [\Gamma, x : \alpha]$$

(app-equality) $$\frac{b_1 = b_2 : \alpha \to \beta\ [\Gamma] \quad a_1 = a_2 : \alpha\ [\Gamma]}{b_1(a_1) = b_2(a_2) : \beta\ [\Gamma]}$$

(ξ-equality) $$\frac{b = d : \beta\ [\Gamma, x : \alpha]}{(\lambda x.b) = (\lambda x.d) : \alpha \to \beta\ [\Gamma]}$$

(β-equality) $$\frac{b : \beta\ \lambda\text{-exp}\ [\Gamma, x : \alpha] \quad a : \alpha\ \lambda\text{-exp}\ [\Gamma]}{(\lambda x.b)(a) = b[x := a] : \beta\ [\Gamma]}$$

(η-equality) $$\frac{b : \alpha \to \beta\ \lambda\text{-exp}\ [\Gamma]}{(\lambda x.b(x)) = b : \alpha \to \beta\ [\Gamma]} \quad \text{provided } x \notin FV(b)$$

2. Decidability of the equality theory.

It is well known that the equality theory for typed lambda terms we have just recalled is decidable. The usual proof goes on by using the normal form theorem and the Church-Rosser one in order to show that two typed lambda terms are equal if and only if their normal forms, which always exist and are uniquely determined, differ only for the name of the abstracted variables. However while the normal form theorem is central in obtaining the result, the Church-Rosser one, which holds also for the type-free λ-calculus for which the decidability of the equality theory does not hold, does not seem to be so important. In our proof it will be very clear that the result rests mainly on the normal form theorem.

The notion of normal form is central in the development of a simply typed λ-calculus: it establishes a canonical form for the expressions, i.e. their *simplest* form. In the case we are considering only one form of simplification is introduced.

[4]As usual, we assume that the α-equality holds, i.e. that two expressions which differ only for the name of the abstracted variables are identical; formally the expressions $(\lambda x.b)$ and $(\lambda y.b[x := y])$, where $y \notin FV(b)$, coincide.

Definition: (β-reduction). $(\lambda x.b)(a) \Rightarrow b[x := a]$

The simplest form of an expression is obviously the one in which no reduction can be applied.

Definition: (Normal form). An expression is in normal form if no β-reduction can be applied to any of its sub-expressions.

The following algorithm shows a method to obtain, for any given expression, an equivalent expression in normal form.

Algorithm: (Conversion into normal form). Let e be an expression of type α and consider the following recursive definition.

$$\mathrm{nf}(e) = \begin{cases} (\lambda x.\mathrm{nf}(e(x))) & \text{if } \alpha \equiv \beta \to \gamma \text{ where } x \text{ is a new variable of type } \beta \\ x(\mathrm{nf}(b_1))\ldots(\mathrm{nf}(b_n)) & \text{if } \alpha \equiv C \text{ and } e \equiv x(b_1)\ldots(b_n) \\ \mathrm{nf}(c[x := a](b_1)\ldots(b_n)) & \text{if } \alpha \equiv C \text{ and } e \equiv (\lambda x.c)(a)(b_1)\ldots(b_n) \end{cases}$$

The proof of its total correctness can be found in [Valentini 92] and is based on the fact that the following rules to construct λ-expressions are equivalent to the more standard ones we have shown in the previous section.

Definition: (Expression).

$$(1) \qquad \frac{a_1 : \alpha_1 \text{ exp } [\Gamma] \ldots a_n : \alpha_n \text{ exp } [\Gamma]}{x(a_1)\ldots(a_n) : C \text{ exp } [\Gamma, x : \alpha_1 \to (\ldots(\alpha_n \to C)\ldots)]} \qquad n \geq 0$$

$$(2) \qquad \frac{c\ [x := a](b_1)\ldots(b_n) : C\ [\Gamma] \quad a : \alpha \text{ exp } [\Gamma]}{(\lambda x.c)(a)(b_1)\ldots(b_n) : C \text{ exp } [\Gamma]} \qquad n \geq 0$$

$$(3) \qquad \frac{b(x) : \beta \text{ exp } [\Gamma, x : \alpha]}{b : \alpha \to \beta \text{ exp } [\Gamma]}$$

where in 3. the only occurrence of x in $b(x)$ is the manifested one.

The only non-trivial step in the proof of the equivalence is to prove the closure of the new expression system under substitution. This result is obtained by a double induction on the complexity of the type of the substituted expression and on the length of the derivation of the expression on which the substitution is performed.

So the key to prove the decidability of the equality theory lies in the possibility to obtain a new set of rules, directly suggested by the one we have used to define the new expressions, such that the decidability is almost obvious but which are also equivalent to the standard ones.

Definition: (Equal expressions).

(1) $$\frac{a_1 \simeq b_1 : \alpha_1\ [\Gamma] \ldots a_n \simeq b_n : \alpha_n\ [\Gamma]}{x(a_1)\ldots(a_n) \simeq x(b_1)\ldots(b_n) : C\ [\Gamma, x : \alpha_1 \to (\ldots(\alpha_n \to C)\ldots)]} \quad n \geq 0$$

(2a)
$$\frac{c\ [x := a](b_1)\ldots(b_n) \simeq d : C\ [\Gamma]}{(\lambda x.c)(a)(b_1)\ldots(b_n) \simeq d : C\ [\Gamma]} \quad n \geq 0$$

(2b)
$$\frac{d \simeq c\ [x := a](b_1)\ldots(b_n) : C\ [\Gamma]}{d \simeq (\lambda x.c)(a)(b_1)\ldots(b_n) : C\ [\Gamma]} \quad n \geq 0$$

(3) $$\frac{b(x) \simeq d(x) : \beta\ [\Gamma, x : \alpha]}{b \simeq d : \alpha \to \beta\ [\Gamma]}$$

where in 3. the only occurrence of x in $b(x)$ and $d(x)$ is the manifested one.

It is now almost immediate to show the following theorem.

Theorem. *Let $b : \beta$ exp $[\Gamma]$ and $d : \beta$ exp $[\Gamma]$ be two expressions of the same type; then $b \simeq d : \beta\ [\Gamma]$ is decidable.*

Proof. By induction on the sum of the lengths of the derivations of $b : \beta$ exp $[\Gamma]$ and $d : \beta$ exp $[\Gamma]$.

To conclude the proof of the decidability of the equality between λ-expressions we must show that, for any two expressions $b : \beta$ exp $[\Gamma]$ and $d : \beta$ exp $[\Gamma]$, $b = d : \beta\ [\Gamma]$ is derivable if and only if $b \simeq d : \beta\ [\Gamma]$ is derivable. The proof of the right to left implication is almost immediate while the proof in the other direction is not so obvious.

Let us begin by showing that $\simeq$ is an equivalence relation.

Lemma: Reflexivity. *Let $b : \beta$ exp $[\Gamma]$; then $b \simeq b : \beta\ [\Gamma]$.*

Proof. By induction on the length of the derivation of $b : \beta$ exp $[\Gamma]$.

Lemma: Symmetry. *Let $a \simeq b : \beta\ [\Gamma]$; then $b \simeq a : \beta\ [\Gamma]$.*

Proof. By induction on the length of the derivation of $a \simeq b : \beta\ [\Gamma]$.

The proof of the closure under transitivity is more complex and it recalls that of a standard cut-elimination theorem. First let us observe that in the use ‘upwards’ of the rule 3. the name of the new variable is indifferent, i.e. if $b(x) \simeq d(x) : \beta\ [\Gamma, x : \alpha]$, where x is a variable which appears free neither in b nor in d, is derivable then, for any variable y of type α which appears neither in b nor in d, also $b(y) \simeq d(y) : \beta\ [\Gamma, y : \alpha]$ is derivable by a derivation of the same depth.

Lemma: Transitivity. *Let $a \simeq b : \beta\ [\Gamma]$ and $b \simeq c : \beta\ [\Gamma]$; then $a \simeq c : \beta\ [\Gamma]$.*

Proof. The proof is by induction on the sum of the depth of the derivations of $a \simeq b : \beta[\Gamma]$ and $b \simeq c : \beta\ [\Gamma]$. Three main cases have to be considered.

(1) Suppose $\beta \equiv \alpha \to \gamma$ then we can suppose, without losing generality, that $a \simeq b : \alpha \to \gamma\ [\Gamma]$ be derived from $a(x) \simeq b(x) : \gamma\ [\Gamma, x : \alpha]$ and $b \simeq c : \alpha \to \gamma\ [\Gamma]$ be

derived from $b(x) \simeq c(x) : \gamma\ [\Gamma, x : \alpha]$ for some variable x which occurs neither in a nor in b nor in c; then, by inductive hypothesis $a(x) \simeq c(x) : \gamma\ [\Gamma, x : \alpha]$ is derivable and hence also $a \simeq c : \alpha \to \gamma\ [\Gamma]$.

Let us now consider the cases in which the type β is a basic type.

(2) Suppose $a \simeq b : C\ [\Gamma]$ (or $b \simeq c : C\ [\Gamma]$) is derived from $a' \simeq b : C\ [\Gamma]$ (respectively $b \simeq c' : C\ [\Gamma]$) then, by inductive hypothesis, $a' \simeq c : C\ [\Gamma]$ (respectively $a \simeq c' : C\ [\Gamma]$) is derivable and hence also $a \simeq c : C\ [\Gamma]$.

(3a) Suppose $x(a_1) \ldots (a_n) \simeq x(b_1) \ldots (b_n) : C\ [\Gamma, x : \alpha_1 \to (\ldots (\alpha_n \to C) \ldots)]$ and $x(b_1) \ldots (b_n) \simeq x(c_1) \ldots (c_n) : C\ [\Gamma, x : \alpha_1 \to (\ldots (\alpha_n \to C) \ldots)]$ are derived from $a_i \simeq b_i : \alpha_i\ [\Gamma]$ and $b_i \simeq c_i : \alpha_i\ [\Gamma]$ for each $1 \leq i \leq n$, then, by inductive hypothesis, $a_i \simeq c_i : \alpha_i\ [\Gamma]$ and so $x(a_1) \ldots (a_n) \simeq x(c_1) \ldots (c_n) : C\ [\Gamma, x : \alpha_1 \to (\ldots (\alpha_n \to C) \ldots)]$ is derivable.

(3b) Suppose $a \simeq (\lambda x.b)(d)(d_1) \ldots (d_n) : C\ [\Gamma]$ and $(\lambda x.b)(d)(d_1) \ldots (d_n) \simeq c : C\ [\Gamma]$ are derived from $a \simeq b[x := d](d_1) \ldots (d_n) : C\ [\Gamma]$ and $b[x := d](d_1) \ldots (d_n) \simeq c : C\ [\Gamma]$ then, by inductive hypothesis, $a \simeq c : C\ [\Gamma]$ is derivable.

The next fundamental step in the proof of the equivalence between the two equality theories is to show that $\simeq$ is closed under substitution.

Theorem. *Let* $b \simeq d : \beta\ [\Gamma, x : \alpha]$ *and* $a \simeq c : \alpha\ [\Gamma]$*; then* $b[x := a] \simeq d[x := c] : \beta\ [\Gamma]$.

Proof. By double induction: the principal induction is on the complexity of the type α of the substitution and the secondary is on the length of the derivation of $b \simeq d : \beta\ [\Gamma, x : \alpha]$. The proof is completely similar to that of closure under substitution of the expression system we have already recalled. As usual three main cases have to be considered.

(1) $b \simeq d : \gamma \to \delta\ [\Gamma, x : \alpha]$ is derived from $b(y) \simeq d(y) : \delta\ [\Gamma, x : \alpha, y : \gamma]$ for some variable y which appears neither in b nor in d; then, by inductive hypothesis, $b(y)[x := a] \simeq d(y)[x := c] : \delta\ [\Gamma, y : \gamma]$, i.e. $b[x := a](y) \simeq d[x := a](y) : \delta\ [\Gamma, y : \gamma]$, is derivable and hence also $b[x := a] \simeq d[x := a] : \gamma \to \delta\ [\Gamma]$ is derivable since we can suppose that y does not appear in a without losing generality.

(2a) $(\lambda y.b)(e)(f_1) \ldots (f_n) \simeq d : C\ [\Gamma, x : \alpha]$ is derived from $b[y := e](f_1) \ldots (f_n) \simeq d : C\ [\Gamma, x : \alpha]$; then, by inductive hypothesis,

$b[y := e](f_1) \ldots (f_n)[x := a] \simeq d[x := c] : C\ [\Gamma]$,

i.e $b[x := a][y := e[x := a]](f_1[x := a]) \ldots (f_n[x := a]) \simeq d[x := c] : C\ [\Gamma]$ since we can suppose that y does not appear in a, is derivable and hence also

$(\lambda y.b[x := a])(e[x := a])(f_1[x := a]) \ldots (f_n[x := a]) \simeq d[x := c] : C\ [\Gamma]$,

i.e. $(\lambda y.b)(e)(f_1) \ldots (f_n)[x := a] \simeq d[x := c] : C\ [\Gamma]$, is derivable.

(2b) $b \simeq (\lambda y.d)(e)(f_1) \ldots (f_n) : C\ [\Gamma, x : \alpha]$ is derived from

$b \simeq d[y := e](f_1) \ldots (f_n) : C\ [\Gamma, x : \alpha]$

is completely analogous to the previous case.

(3a) $z(b_1) \ldots (b_n) \simeq z(d_1) \ldots (d_n) : C\ [\Gamma, x : \alpha, z : \alpha_1 \to (\ldots (\alpha_n \to C) \ldots)]$ is derived from $b_i \simeq d_i : \alpha_i\ [\Gamma, x : \alpha]$, for each $1 \leq i \leq n$, for some variable $z \not\equiv x$; then, by inductive hypothesis, $b_i[x := a] \simeq d_i[x := c] : \alpha_i\ [\Gamma]$ and hence

$z(b_1[x := a]) \ldots (b_n[x := a]) \simeq$
$z(d_1[x := c]) \ldots (d_n[x := c]) : C\ [\Gamma, z : \alpha_1 \to (\ldots (\alpha_n \to C) \ldots)]$,

i.e. $z(b_1) \ldots (b_n)[x := a] \simeq z(d_1) \ldots (d_n)[x := c] : C\ [\Gamma, z : \alpha_1 \to (\ldots (\alpha_n \to C) \ldots)]$, is derivable.

(3b) Suppose $x \simeq x : C\ [\Gamma, x : C]$ then $x[x := a] \simeq x[x := c] : C\ [\Gamma]$, i.e. $a \simeq c : C\ [\Gamma]$ is derivable by hypothesis.

(3c) Suppose $x(b_1)\ldots(b_n) \simeq x(d_1)\ldots(d_n) : C\ [\Gamma, x : \alpha_1 \to (\ldots(\alpha_n \to C)\ldots)]$ is derived from $b_i \simeq d_i : \alpha_i\ [\Gamma, x : \alpha_1 \to (\ldots(\alpha_n \to C)\ldots)]$ for each $1 \leq i \leq n$; then, by inductive hypothesis, $b_i[x := a] \simeq d_i[x := c] : \alpha_i\ [\Gamma]$; let us now consider $a \simeq c : \alpha_1 \to (\ldots(\alpha_n \to C))\ [\Gamma]$: it must be derived from
$a(y_1) \simeq c(y_1) : \alpha_2 \to (\ldots(\alpha_n \to C))\ [\Gamma, y : \alpha_1]$,
where y_1 is a new variable whose type α_1 is simpler then the type $\alpha_1 \to (\ldots(\alpha_n \to C))$ of x, and hence, by principal inductive hypothesis,
$a(y_1)[y_1 := b_1[x := a]] \simeq c(y_1)[y_1 := d_1[x := c]] : \alpha_2 \to (\ldots(\alpha_n \to C))\ [\Gamma]$,
i.e. $a(b_1[x := a]) \simeq c(d_1[x := c]) : \alpha_2 \to (\ldots(\alpha_n \to C))\ [\Gamma]$, is derivable. Now we can repeat this procedure n times in order to show that
$a(b_1[x := a])\ldots(b_n[x := a]) \simeq c(d_1[x := c])\ldots(d_n[x := c]) : C\ [\Gamma]$,
i.e. $x(b_1)\ldots(b_n)[x := a] \simeq x(d_1)\ldots(d_n)[x := c] : C$, is derivable.

Now the proof of the equivalence of the two equality theories is straightforward. First of all we have the following lemma.

Lemma. *Let $x : \alpha$; then $x \simeq x : \alpha\ [\Gamma, x : \alpha]$ is derivable.*

Proof. By induction on the type complexity α of the variable x.

Closure of $\simeq$ under app-equality is an immediate consequence of the theorem of closure under substitution.

Lemma. *Let $b_1 \simeq b_2 : \alpha \to \beta\ [\Gamma]$ and $a_1 \simeq a_2 : \alpha\ [\Gamma]$; then $b_1(a_1) \simeq b_2(a_2) : \beta\ [\Gamma]$.*

Proof. $b_1 \simeq b_2 : \alpha \to \beta\ [\Gamma]$ must be derived from $b_1(x) \simeq b_2(x) : \beta\ [\Gamma, x : \alpha]$ where x appears neither in b_1 nor in b_2; hence from $a_1 \simeq a_2 : \alpha\ [\Gamma]$ we obtain $b_1(a_1) \simeq b_2(a_2) : \beta\ [\Gamma]$.

Also ξ-equality is immediate.

Lemma. *Let $b \simeq d : \beta\ [\Gamma, x : \alpha]$; then $(\lambda x.b) \simeq (\lambda x.d) : \alpha \to \beta\ [\Gamma]$.*

Proof. $b \simeq d : \beta\ [\Gamma, x : \alpha]$ must be derived from
$b(y_1)\ldots(y_m) \simeq d(y_1)\ldots(y_m) : C\ [\Gamma, x : \alpha, y_1 : \alpha_1 \ldots y_m : \alpha_m]$,
then, since obviously $b[x := x] \equiv b$ and $d[x := x] \equiv d$, we obtain
$(\lambda x.b)(x)(y_1)\ldots(y_m) \simeq (\lambda x.d)(x)(y_1)\ldots(y_m) : C\ [\Gamma, x : \alpha, y_1 : \alpha_1 \ldots y_m : \alpha_m]$
and hence $(\lambda x.b) \simeq (\lambda x.d) : \alpha \to \beta\ [\Gamma]$ because x appears neither in $(\lambda x.b)$ nor in $(\lambda x.d)$.

As to β-equality, it is built-in in our equality rules.

Lemma. *Let $b : \beta$ exp $[\Gamma, x : \alpha]$ and $a : \alpha$ exp $[\Gamma]$; then $(\lambda x.b)(a) \simeq b[x := a] : \beta\ [\Gamma]$.*

Proof. If $b : \beta$ exp $[\Gamma, x : \alpha]$ and $a : \alpha$ exp $[\Gamma]$ then $b[x := a] : \beta$ exp $[\Gamma]$, because of closure under substitution, and hence $b[x := a] \simeq b[x := a] : \beta\ [\Gamma]$, because of reflexivity, but the last equality must be derived from
$b[x := a](y_1)\ldots(y_m) \simeq b[x := a](y_1)\ldots(y_m) : C\ [\Gamma, y_1 : \alpha_1 \ldots y_m : \alpha_m]$
from which we obtain
$(\lambda x.b)(a)(y_1)\ldots(y_m) \simeq b[x := a](y_1)\ldots(y_m) : C\ [\Gamma, y_1 : \alpha_1 \ldots y_m : \alpha_m]$
and hence $(\lambda x.b)(a) \simeq b[x := a] : \beta\ [\Gamma]$.

Now let us consider η-equality.

Lemma. *Let $b : \alpha \to \beta$ exp $[\Gamma]$ and $x : \alpha$ be a variable which does not appear in b; then $(\lambda x.b(x)) \simeq b : \alpha \to \beta$ $[\Gamma]$.*

Proof. If $b : \alpha \to \beta$ exp $[\Gamma]$ and $x : \alpha$ then $b(x) : \beta$ exp $[\Gamma, x : \alpha]$ is derivable and hence, by reflexivity, also $b(x) \simeq b(x) : \beta$ $[\Gamma, x : \alpha]$, i.e. $b(x)[x := x] \simeq b(x) : \beta$ $[\Gamma, x : \alpha]$ is derivable; now, this equality must be derived from

$b(x)[x := x](y_1) \ldots (y_m) \simeq b(x)(y_1) \ldots (y_m) : C$ $[\Gamma, x : \alpha, y_1 : \alpha_1 \ldots y_m : \alpha_m]$

hence $(\lambda x.b(x))(x)(y_1) \ldots (y_m) \simeq b(x)(y_1) \ldots (y_m) : C$ $[\Gamma, x : \alpha, y_1 : \alpha_1 \ldots y_m : \alpha_m]$ is derivable and then also $(\lambda x.b(x))(x) \simeq b(x) : \beta$ $[\Gamma, x : \alpha]$; now, if x does not appear in b, we can conclude that $(\lambda x.b(x)) \simeq b : \alpha \to \beta$ $[\Gamma]$.

Finally we must not forget that as far as standard equality is concerned also α-equality is tacitly assumed and hence here it must be proved.

Lemma. *Let $b : \beta$ exp $[\Gamma, x : \alpha]$ and $y : \alpha$ be a variable which does not appear in b; then $(\lambda x.b) \simeq (\lambda y.b[x := y]) : \alpha \to \beta$ $[\Gamma]$.*

Proof. Suppose y does not occur in b then $b \equiv b[x := y][y := x]$; hence $b : \beta$ exp $[\Gamma, x : \alpha]$ implies by reflexivity that $b[x := x] \simeq b[x := y][y := x] : \beta$ $[\Gamma, x : \alpha]$ is derivable and it must be derived from

$b[x := x](y_1) \ldots (y_m) \simeq b[x := y][y := x](y_1) \ldots (y_m) : C$ $[\Gamma, x : \alpha, y_1 : \alpha_1 \ldots y_m : \alpha_m]$

and hence

$$(\lambda x.b)(x)(y_1) \ldots (y_m) \simeq$$
$$(\lambda y.b[x := y])(x)(y_1) \ldots (y_m) : C \; [\Gamma, x : \alpha, y_1 : \alpha_1 \ldots y_m : \alpha_m]$$

is derivable from which $(\lambda x.b) \simeq (\lambda y.b[x := y]) : \alpha \to \beta$ $[\Gamma]$ follows.

So we have completed the proof of equivalence of the two equality systems and hence also the proof of the theorem of decidability of the equality.

It is necessary to observe that the method here presented fits very well with the β-η-equality and, as also the referee noted, it would be very interesting to know how to modify it in order to treat with a λ-calculus such that only the β-equality is present. But this is an another problem.

References

[Barendregt 81] H.P. Barendregt, *The Lambda Calculus: its syntax and semantics*, North Holland, Amsterdam, 1981.

[Bossi-Valentini 89] A. Bossi, S. Valentini, *The expressions with arity*, rapporto interno **61/89** (1989), Dip. Scienze dell'Informazione, Univ. di Milano, Milano.

[Valentini 92] S. Valentini, *A note on a straightforward proof of normal form theorem for simply typed λ-calculi*, Bollettino dell'Unione Matematica Italiana **??/??** (1993), ??-??.

Dipartimento di Matematica Pura ed Applicata, Università di Padova, via G. Belzoni n.7, 35131 Padova (ITALY)

E-mail address: valentini@pdmat1.unipd.it