Provable isomorphisms of types

Kim B. Bruce[1] *Roberto Di Cosmo*[2] *Giuseppe Longo*[3]

Abstract

A constructive characterization is given of the isomorphisms which must hold in all models of the typed lambda calculus with surjective pairing. Work of Rittri (1989) on using types as search keys in program libraries provides an interesting example of the use of these characterizations. By the close relation between closed Cartesian categories and models of these calculi, we also produce a characterization of those isomorphisms which hold in all CCC's. By a similar correspondence between these calculi and proofs in intuitionistic positive propositional logic, we also provide a characterization of equivalent formulae of this logic, where the definition of equivalence of terms depends on having "invertible" proofs between the two terms.

[1]Dept. Of Computer Science, Williams College, Williamstown, MA 01267
[2]Dip. di Informatica, Università di Pisa, Italy, and LIENS, 45, Rue d'Ulm, Paris, France

1. Introduction

There has been a great deal of interest over the years in constructing models of the various lambda calculi which satisfy various equations (isomorphisms) between types. One reason is that recursive definitions of data types are usually interpreted as equations to be solved over specific mathematical structures.

In previous work, Bruce and Longo (1985), the first and third authors showed that no such non-trivial isomorphisms existed in either the simple or second-order typed lambda calculus. However, no extension of these results was known at that time for the lambda calculus with more rich type disciplines. Products and higher order types, in particular, proved to be of special interest not only because of their common or increasing use in programming languages, but also because of the close connection between these calculi and relevant categorical structures and proof systems.

In this paper we characterize the isomorphisms holding in all models of the simply typed lambda calculus with surjective pairing (and "terminal object"). Moreover we show that it is decidable whether two types (built from type variables) are isomorphic in all models of this calculus. It is well known that these models are exactly the Cartesian closed categories (CCC).

While we are looking for types which are isomorphic in every model of our lambda calculus (or equivalently, isomorphic in every CCC), there is, in principle, no reason to believe that there is a uniform way to witness these isomorphisms. Nevertheless it turns out that our proof of these results is based on a simple axiomatization of type equations and the notion of provable isomorphisms (those representable by closed terms of the lambda calculus). Moreover any proof of the equality of two types can be used to generate an isomorphism between the types (which holds in every model).

The axioms for our theory are given below:

0.1 Definition. Th is a theory of equality plus the following axiom schemas, where $\mathbf{T}$ is a constant symbol:

1) $A \times \mathbf{T} = A$
2) $A \times B = B \times A$
3) $A \times (B \times C) = (A \times B) \times C$
4) $(A \times B) \rightarrow C = A \rightarrow (B \rightarrow C)$
5) $A \rightarrow (B \times C) = (A \rightarrow B) \times (A \rightarrow C)$
6) $A \rightarrow \mathbf{T} = \mathbf{T}$
7) $\mathbf{T} \rightarrow A = A$

The Main Theorem of this paper shows that two types A and B can be constructively proved to be isomorphic, by programs which act one as the inverse of the other, iff Th $\vdash A = B$.

In order to discuss the soundness of Th, and explain where it comes from, we hint here its categorical meaning. Note, though, that no notion nor result from Category Theory is used in most of the paper.

Since models of the typed lambda calculus with surjective pairing are exactly the Cartesian closed categories, our results translate directly into theorems on when two generic objects are isomorphic due simply to the Cartesian closed structure of the category in which they are interpreted, no matter which particular CCC we choose.

Observe first that Th is realized in every Cartesian Closed Category, when "=" is interpreted as isomorphism. The first three axioms describe properties of the Cartesian product (associativity, commutativity, identity for $\times$), and the second three axioms can be seen as the properties of the three adjunctions of a CCC that relate product, exponent and the terminal object. The last equation ($T \rightarrow A = A$) tells us that the arrows from the terminal object to A in a CCC are the points of A. Thus, the theory Th is sound. A consequence of our main result is the completeness of Th with respect to CCC's. That is, no other isomorphism is valid in all CCC's. (This is not obvious because there are categorical models of Th which are not CCC's: take a Cartesian Category with a bifunctor "$\rightarrow$" such that $A \rightarrow B = B$, say).

Another result of the work below in λ-calculus will be an insight into the composition of derivations in Proof Theory. The typed lambda calculus with surjective pairing is the language for proofs of IPC(**True**, $\wedge$, $\rightarrow$), the intuitionistic positive propositional calculus. In the proof theoretic framework we then characterize *equivalent formulae*, where two formulae A and B are considered equivalent if, given a proof f of the sequent $A \vdash B$, and a proof g of the sequent $B \vdash A$, $g \circ f$ yields, after cut-elimination, the identity proof of the sequent $A \vdash A$ and vice-versa. The details of both the categorical and proof-theoretic applications are discussed in DiCosmo and Longo (1990).

As an example of the use of such results in computer science we note the two papers by Rittri (1989, 1989a) in which the author discusses the problem of finding applicable functions in a program library. For example, one might be interested in looking up various search functions. As a result it might be useful to inspect all functions which take an element and a table and return an index to the table. Because trivial differences in argument order or Currying may lead one to ignore useful functions, it is important to be able to find all those functions whose type is isomorphic to that for which one is searching.

Rittri's application of the result presented here settles on the same notion of provable isomorphism. He cites the paper by Solv'ev (1983), in which the author presents the same result as in our main theorem (Theorem 3.6), although by an entirely different proof which is based on taking the natural numbers as objects in a CCC (with $\times$ interpreted as multiplication and $\rightarrow$ as exponentiation) and then showing the equational completeness of the theory of $(\mathbb{N}, 1, \times, \uparrow)$. (Meyer and Statman, *personal communication*, suggested a similar proof for the exponential fragment only; also the abstract in Martin (1972) states the same fact). Solv'ev also provides a decision procedure similar to that given here.

We note that in a forthcoming paper, the second author extends these results to the second-order typed lambda calculus, with surjective pairings. We know no way of extending the proof given by Solv'ev to this more complex case.

The paper is organized as follows. Section 1 sets out the basic definitions leading up to the notion of a type normal form. Section 2 presents some rather technical lemmas which will be used in section 3 in order to characterize the set of provable isomorphisms. The third section also contains a discussion of the decidability of the theory and the connections with category and proof theory. The proofs of the important (but technically complex) lemmas in section 2 are put off into an appendix at the end of the paper.

1. Basic notions and facts

1.1. DEFINITION. The collection **Tp** of type labels, over a ground set At of atomic type symbols, is inductively defined by:

i. $\mathrm{At} \subseteq \mathrm{Tp}$, where $\mathbf{T} \in \mathrm{At}$ is a fixed constant type symbol

ii. if $A, B \in \mathrm{Tp}$, then $A \rightarrow B \in \mathrm{Tp}$;

iii. if $A, B \in \mathrm{Tp}$, then $A \times B \in \mathrm{Tp}$.

The intended meaning of $\mathbf{T}$ is the terminal object in the categorical sense; thus $*A$ below will stand for the unique map in $A \rightarrow \mathbf{T}$ (as required of a terminal object).

Pure λ-terms are defined as usual. In particular, for every type A there exists a denumerable number of **variables**, ranged over by lower case letters near the end of the alphabet. We use upper case letters M, N, P, . . ., as meta-variables for terms. The fact that a term M has type A will be denoted with the expression "M : A".

1.2. DEFINITION. (i) The terms of $\lambda\beta\eta\pi^{*\mathbf{t}}$, the typed λ-calculus with surjective pairing and terminal object, and their associated types, are defined according to the following formation rules:

1. every variable $x : A$ is a term;
2. if $x : A$ is a variable, and $M : B$ is a term, then $\lambda x{:}A.M : A \to B$ is a term;
3. if $M : A \to B$ is a term and $N : A$ is a term, then $MN : B$ is a term;
4. if $M : A$ is a term and $N : B$ is a term, then $<M,N> : A \times B$ is a term;
5. $p_1 : A \times B \to A$;
6. $p_2 : A \times B \to B$;
7. $*A : A \to \mathbf{T}$.

(ii) The equational theory of terms is the minimal congruence relation "=" which satisfies the following axioms schemas, rule ξ and the rule for the terminal arrow:

alpha-beta-eta-csi:

(α) $\lambda x{:}A.M = \lambda y{:}A.[y/x]M$, if y is free for x in M

($\to\beta$) $(\lambda x{:}A.M)N = [N/x]M$, if N is free for x in M.

($\to\eta$) $\lambda x{:}A.(Mx) = M$, if $x \notin FV(M)$

(ξ) if $M=N$ then $\lambda x{:}A.M=\lambda x{:}A.N$

surjective pairing:

($\times\beta 1$) $p_1(<M,N>) = M$

($\times\beta 2$) $p_2(<M,N>) = N$

($\times\eta$) $<p_1(M),p_2(M)> = M$

terminal object:

($*$) If $M : A \to \mathbf{T}$ then $M = *A$.

Given a sequence $\underline{M} = M_1,..., M_n$ of terms, and sequence $\underline{x} = x_1, . . . , x_n$ of variables, $N[\underline{x}:=\underline{M}]$ denotes the simultaneous substitution of every term M_i for the variable x_i in the term N (for simplicity, we always assume bound variables are renamed as necessary to avoid capture of free variables). We also use the notation $N[\underline{x}:=M]$ to express the simultaneous substitution of the term M for all the variables in $\underline{x}$. Let $I_A = \lambda x{:}A.x$ be the identity of type A.

NOTATION. We write $<M_1, ... M_n>$ for $<...<<M_1, M_2>,M_3> ....>$. $\lambda\beta\eta\pi^{\mathbf{t}}$ is the calculus without terminal object and related rules, $\lambda\beta\eta^{\mathbf{t}}$ is the classical typed calculus, and $\lambda\beta\eta$ the type-free calculus.

REMARK. (*Notion of reduction for $\lambda\beta\eta\pi^{*t}$*) The notion of reduction associated with the equational theory of $\lambda\beta\eta\pi^{*t}$ obtained by just orienting the equalities in the axioms to the right is not Church-Rosser. It is possible, though, to derive for this equality theory another notion of reduction that has the Church-Rosser property; in the following we will refer to this latter one when talking about reduction, normal forms, and so on for $\lambda\beta\eta\pi^{*t}$ (see for example Pottinger (1981), Curien (1990)).

1.3. DEFINITION. Let $A, B \in Tp$. Then A and B are **provably isomorphic** ($A \cong_p B$) iff there exist closed λ-terms $M : A \to B$ and $N : B \to A$ such that $\lambda\beta\eta\pi^{*t} \vdash M \circ N = I_B$ and $\lambda\beta\eta\pi^{*t} \vdash N \circ M = I_A$, where I_A and I_B are the identities of type A and B. We then say that M and N are **invertible** terms, and that M is an inverse of N, in $\lambda\beta\eta\pi^{*t}$.

Note that, as usual, the inverse of a term M (if it exists) is unique up to "=." Suppose that types A and B are provably isomorphic and consistently substitute the common base types by arbitrary types. Then the isomorphism still holds: just use the corresponding terms with updated types. Borrowing terminology from Statman (1983) we may say that the notion of provable isomorphism is typically ambiguous.

1.4. MAIN THEOREM (*easy implication*) $\quad Th \vdash A = B \;\Rightarrow\; A \cong_p B$.

PROOF. We give the terms associated to each axiom and rule. As T is a theory of equality, one has first to observe that the usual axioms and inference rules yield and preserve provable isomorphisms:

- $\lambda x{:}A.x$ proves $A = A$;
- if M, with inverse N, proves $A = B$, then N proves $B = A$;
- if an invertible M proves $A = B$ and N proves $B = C$, then $N \circ M = \lambda x{:}A.N\,(M\,x)$ proves $A = C$ and it is clearly invertible;
- if an invertible M proves $A = B$ and N proves $C = D$, then $\lambda x{:}A{\times}C.<M\,(p_1\,x),\ N\,(p_2\,x)>$ proves $A \times C = B \times D$ and it is invertible;
- if an invertible M proves $A = B$ and N proves $C = D$, then $\lambda y{:}A{\to}C.\lambda x{:}B.N\,(y\,(M^{-1}\,x))$, where M^{-1} is the inverse of M, proves $A \to C = B \to D$ and it is invertible (take $\lambda y{:}B{\to}D.\lambda x{:}A.N^{-1}\,(y\,(M\,x))$).

We next check the proper axioms:

1) $\quad A \times \mathbf{T} = A$ is proved by p_1 with inverse $\lambda x{:}A.< x, {*}A\, x >$ (to check invertibility, note that ${*}A \circ p_1 = {*}(A \times \mathbf{T}) = p_2$).

2) $\quad A \times B = B \times A$ is proved by $\lambda x{:}A{\times}B.< p_2\,x, p_1\,x >$;

3) $\quad A \times (B \times C) = (A \times B) \times C$ is proved by $\lambda x{:}A \times (B \times C).<< p_1\,x, p_1\,(p_2\,x) >, p_2\,(p_2\,x)>$, which is invertible;

4) $\quad (A \times B)\to C = A \to (B \to C)$ is proved by $\lambda z{:}(A{\times}B){\to}C.\lambda x{:}A.\lambda y{:}B.z\,<x,y>$ with inverse $\lambda z{:}A{\to}(B{\to}C).\lambda x{:}A{\times}B.z\,(p_1\,x)\,(p_2\,x)$;

5) $A \to (B \times C) = (A \to B) \times (A \to C)$ is proved by
$\lambda z{:}A\to(B\times C).\langle\lambda x{:}A.(p_1\ (z\ x)),\lambda x{:}A.(p_2\ (z\ x))\rangle$ with inverse
$\lambda z{:}(A\to B)\times(A\to C).\lambda x{:}A.\langle(p_1\ z)\ x,(p_2\ z)\ x\rangle$;

6) $A \to \mathbf{T} = \mathbf{T}$ is proved by $*(A\to\mathbf{T})$, with inverse $\lambda x{:}\mathbf{T}.*A$;

7) $\mathbf{T} \to A = A$ is proved by $\lambda z{:}\mathbf{T}\to A.z\ (*(\mathbf{T}\to A)\ z)$ with inverse $\lambda x{:}A.\lambda y{:}\mathbf{T}.x$. ♦

The rest of this section, as well sections 2 and 3, are dedicated to the proof of the other implication of the Main Theorem. The first steps are done by reducing types to a "type normal form".

The axioms of Th suggest the following rewrite system R for types (essentially Th without commutativity):

1.5 DEFINITION (*Type rewriting R*) Let ">" be the transitive and substitutive **type-reduction** relation generated by:

1) $A \times \mathbf{T} > A$

1') $\mathbf{T} \times A > A$

3) $A \times (B \times C) > (A \times B) \times C$

4) $(A \times B) \to C > A \to (B \to C)$

5) $A \to (B \times C) > (A \to B) \times (A \to C)$

6) $A \to \mathbf{T} > \mathbf{T}$

7) $\mathbf{T} \to A > A$.

The system R yields an obvious notion of **normal form for types** (type-n.f.), i.e. when no type reduction is applicable. Note that 1, 1', 6 and 7 "eliminate the $\mathbf{T}$'s", while 4 and 5 "bring the $\times$ outside". It is then easy to observe that each type-n.f. is $\mathbf{T}$ or has the structure $S_1\times \dots \times S_n$ where each S_i does not contain $\mathbf{T}$ or "$\times$". We write **nf(S)** for the normal form of S (there is exactly one, see 1.6), and say that a normal form is non-trivial if it is not T.

1.6 PROPOSITION. Each type has a unique type normal form in R .

PROOF. Notice that in any R-reduction, starting with a given type S:
(i) Rules 4 and 5 can be applied only finitely many times, as they strictly decrease the number of $\times$'s in the scope of an arrow of S and this number is finite and is not changed by any other rule.

(ii) Between an application of rule 4 or 5 (yielding type S') and the next one, the remaining rules can be applied only finitely many times (1, 1', 6 and 7 simply throw away some subformula reducing by one the number of products or arrows, which is finite; 3 is just associativity to the left).

So, after a finite reduction path we get a type S" with no redex for rule 4 and 5, and then, again, the remaining rules can be applied only finitely many times (at most the length of S" plus the times required for associating S" to the left). The resulting type nf(S) has then no products in the scope of any arrow (otherwise 4 or 5 could be applied), and is either **T** or a type with no occurrence of **T** (otherwise 1, 1', 6 or 7 could be applied). Thus nf(S) is a product of types, each of which has no occurrence of $\times$.

It is easy to observe that R is Church-Rosser too and, thus, that nf(S) is unique. (Note also that we have actually proved that R strongly normalizes) ♦

From the implication proved above of the Main Theorem, since $R \models S > R$ implies $Th \vdash S = R$, it is clear that any reduction $R \models S > R$ is witnessed (or, proved, in the "types-as-propositions" analogy) by an invertible term of type $S \to R$. Moreover, one clearly has:

1.7 COROLLARY. $Th \vdash S = nf(S)$ and, thus, $Th \vdash S = R \Leftrightarrow Th \vdash nf(S) = nf(R)$.

In conclusion, when $Th \vdash S = R$, either we have $nf(S) \equiv \mathbf{T} \equiv nf(R)$, or $Th \vdash nf(S) \equiv (S_1\times \ldots \times S_n) = (R_1\times \ldots \times R_m) \equiv nf(R)$. A crucial lemma below will prove that, in this case, one also has $n = m$.

The assertion in the corollary can be reformulated for invertible terms in a very convenient way:

1.8. PROPOSITION. (*Commuting diagram*) Given types A and B, assume that $F : A \to nf(A)$ and $G : B \to nf(B)$ prove the reductions to type-n.f.. Then a term $M : A \to B$ is invertible iff there exist an invertible term $M' : nf(A) \to nf(B)$, such that $M = G^{-1} \circ M' \circ F$.

PROOF. ($\Leftarrow$) Set $M^{-1} \equiv (G^{-1} \circ M' \circ F)^{-1} \equiv F^{-1} \circ M'^{-1} \circ G$, then M is invertible.
($\Rightarrow$) Just set $M' = G \circ M \circ F^{-1}$. Then $M'^{-1} \equiv F \circ M^{-1} \circ G^{-1}$ and M' is invertible. ♦

A diagram easily represents the situation in the corollary:

$$
\begin{tikzcd}
A \arrow[r, "M"] \arrow[d, "F"'] & B \arrow[d, "G"] \\
A_1 \times \vdots \times A_n \arrow[r, "M'"] & B_1 \times \vdots \times B_m
\end{tikzcd}
$$

Thus we have reduced isomorphisms between arbitrary types to the same problems with respect to type normal forms. We examine next how this may affect the structure of the terms which prove the isomorphisms.

2. More Lemmas: From $\lambda\beta\eta\pi^{*t}$ to the Classical $\lambda\beta\eta^t$.

This is a technical section, where we display the statements of some crucial lemmas. Their proofs are postponed to the appendix.

Recall first that, when $Th \vdash S = R$, one has $nf(S) \equiv \mathbf{T} \equiv nf(R)$, or $Th \vdash nf(S) \equiv (S_1 \times ... \times S_n) = (R_1 \times ... \times R_m) \equiv nf(R)$. Notice now that, in the latter case, there cannot be any occurrence of $\mathbf{T}$ in either type. Indeed, a non trivial type-n.f. cannot be provably equated to T, as can be easily seen by taking a non-trivial model. Thus we restrict our attention to equations like $(S_1 \times ... \times S_n) = (R_1 \times ... \times R_m)$ with no occurrence of $\mathbf{T}$ and, hence, to invertible terms with no occurrence of the type constant $\mathbf{T}$ in their types. We can show that these terms do not contain any occurrence of $*A$ either, for any type A, via the following lemmas.

2.1 Lemma. (*Terms of a product type*) Given a term M of $\lambda\beta\eta\pi^{*t}$ in normal form such that $M: A \times B$, then either $M \equiv <M_1, M_2>$ or there is an $x : C$ such that $x \in FV(M)$ and $A \times B$ is a type subexpression of C.

Proof. By induction on the length of the structure of M (see appendix).

2.2 Lemma. (*Every term, whose type contains no occurrence of* $\boldsymbol{T}$, *has no occurrence of* $*A$ *constants*) Assume that in a term M of $\lambda\beta\eta\pi^{*t}$ in normal form there is an occurrence of *A, for some type A. Then there is some occurrence of the type constant **T** in the type of M or in the type of some free variable of M.

PROOF. By induction on the structure of M (see appendix).

2.3 PROPOSITION. (*Isomorphisms between type-n.f.'s are given by terms in* $\lambda\beta\eta\pi^t$). Assume that S and R are non trivial type-n.f.'s. If the closed terms M and N prove $S \cong_p R$ in $\lambda\beta\eta\pi^{*t}$, then their normal forms contain no occurrences of the constants *A. (Thus, M and N are actually in $\lambda\beta\eta\pi^t$).

PROOF. By the previous lemmas, as the terms are closed and no **T** occurs in their type.♦

So we have factored out the first class of constants *A, and we have restricted ourselves to $\lambda\beta\eta\pi^t$. In the next step we eliminate pairing as well, in a sense.

There is a problem though. Our aim is to reduce the investigation of invertible terms in $\lambda\beta\eta\pi^{*t}$ to that of terms in $\lambda\beta\eta^t$. This is done on the grounds of Proposition 1.8 by examining each component of the product, where the isomorphism will be given by terms of $\lambda\beta\eta^t$. However, in the notation of Proposition 1.8, consider the term M' : nf(A) $\rightarrow$ nf(B). M' is invertible in (the equational theory of) $\lambda\beta\eta\pi^{*t}$ and, thus, also the subterms yielding the isomorphism of the components (see 2.4 and 2.6 below) are, a priori, invertible in $\lambda\beta\eta\pi^{*t}$. We get rid of the problem by the following remark.

REMARK. (The equational theory of) $\lambda\beta\eta\pi^{*t}$ is a conservative extension of (the equational theory of) $\lambda\beta\eta\pi^t$. Similarly for $\lambda\beta\eta\pi^t$ with respect to $\lambda\beta\eta^t$. Indeed, both $\lambda\beta\eta\pi^{*t}$ and $\lambda\beta\eta\pi^t$ are Church-Rosser, where "the theory of reduction" for $\lambda\beta\eta\pi^t$ is given by orienting the equalities in the axioms from left to right (for the C-R property see the references in the remark before 1.3) . Consider now M and N in $\lambda\beta\eta\pi^t$ such that $\lambda\beta\eta\pi^{*t} \vdash N = M$ and let P be the common reductum. Then $\lambda\beta\eta\pi^{*t} \vdash N > P$ is actually a reduction $\lambda\beta\eta\pi^t \vdash N > P$, as N contains no *-redex. The same applies to $\lambda\beta\eta\pi^{*t} \vdash M > P$ and, thus, $\lambda\beta\eta\pi^t \vdash N = M$. Similarly for $\lambda\beta\eta\pi^t$ w.r.t $\lambda\beta\eta^t$.

NOTATION. By $\underline{x}$, $\underline{y}$, $\underline{M}$... we denote vectors of variables, terms, etc.

2.4 PROPOSITION. (*Isolate the relevant* $<M_1,...,M_n>$ *in an isomorphism*) Let $S \equiv S_1 \times ... \times S_m$ and $R \equiv R_1 \times ... \times R_n$ be type-n.f.'s where neither the S_i's nor the R_j's contain any occurrences of **T** or $\times$.
Then $S \cong_p R$ iff there exist $M_1,..., M_n$ and $N_1,..., N_m$ such that

$$x_1 : S_1, ..., x_m : S_m \vdash <M_1,...,M_n> : (R_1\times ... \times R_n)$$
$$y_1 : R_1, ..., y_n : R_n \vdash <N_1,...,N_m> : (S_1\times ... \times S_m)$$

and

$$M_i [\underline{x} := \underline{N}] =_{\beta\eta} y_i , \quad \text{for } 1 \le i \le n$$
$$N_j [\underline{y} := \underline{M}] =_{\beta\eta} x_j , \quad \text{for } 1 \le j \le m$$

(where substitution of vectors of equal length is meant componentwise).

PROOF. (See appendix: it is not obvious).

2.5 LEMMA. (*Terms of* $\lambda\beta\eta\pi^t$ *whose type is arrow-only belong to* $\lambda\beta\eta^t$) Let M be a term of $\lambda\beta\eta\pi^t$ in normal form such that $M : A$, where A is a type with no occurrence of $\times$ in it. If no free variable of M has a type with occurrences of $\times$, then M is actually a term in $\lambda\beta\eta^t$.

PROOF. By induction on the structure of M (see appendix).

In conclusion, we have isolated some interesting terms from which every constant has been factored out. Next we prove that provably equal types in normal form have equal length.

2.6 LEMMA. (*Isomorphic type-n.f.'s have equal length*). Assume that $R_1 \times ... \times R_n$ and $S_1 \times ... \times S_m$ are type-n.f.'s and $M \equiv <M_1,...,M_n>$, $N \equiv <N_1,...,N_m>$ are terms in $\lambda\beta\eta\pi^t$ such that

$$x_1: S_1, ..., x_m : S_m \vdash <M_1,...,M_n> : (R_1 \times ... \times R_n)$$
$$y_1: R_1, ..., y_n : R_n \vdash <N_1,...,N_m> : (S_1 \times ... \times S_m)$$
$$M_i [\underline{x} := \underline{N}] =_{\beta\eta} y_i , \quad \text{for } 1 \le i \le n$$
$$N_j [\underline{y} := \underline{M}] =_{\beta\eta} x_j , \quad \text{for } 1 \le j \le m$$

then $n = m$ and there exist permutations σ, π over n such that

$$M_i = \lambda\underline{u}_i.x_{\sigma i}\, \underline{P}_i \quad \text{and} \quad N_j = \lambda\underline{v}_j.y_{\pi j}\, \underline{Q}_j.$$

PROOF. By the previous lemma (recall that we may assume that each M_i and N_j is in normal form) one has that M_i and N_j are in $\lambda\beta\eta^t$. Then,

$M_i \equiv \lambda \underline{u}_i.s_i\, \underline{P}_i$

$N_j \equiv \lambda \underline{v}_j.t_j\, \underline{Q}_j$

Note that s_i is a free variable (namely some x_j), since $M_i[\underline{x} := \underline{N}] =_{\beta\eta} y_i$, which cannot hold if s_i is bound (it is easy to see that there would be no way to reduce it to a term without abstraction), and similarly t_j is some y_i. So there are two functions $\sigma : n \to m$, $\pi: m \to n$ such that $M_i = \lambda \underline{u}_i.x_{\sigma i}\, \underline{P}_i$ for $1 \le i \le n$ and $N_j = \lambda \underline{v}_j.y_{\pi j}\, \underline{Q}_i$ for $1 \le i \le m$.

In conclusion, for $1 \le i \le n$ we obtain:

$$
\begin{aligned}
y_i =_{\beta\eta} M_i\, [\underline{x} := \underline{N}] \quad &=_{\beta\eta} (\lambda \underline{u}_i.x_{\sigma i}\, \underline{P}_i)\, [\underline{x} := \underline{N}] \\
&=_{\beta\eta} \lambda \underline{u}_i.N_{\sigma i}\, (\underline{P}_i\, [\underline{x} := \underline{N}]) \\
&=_{\beta\eta} \lambda \underline{u}_i.(\lambda \underline{v}_i.y_{\pi\sigma i}\, \underline{Q}_i\,)\, (\underline{P}_i[\underline{x} := \underline{N}]) \\
&=_{\beta\eta} \textit{ if } \underline{v} \textit{ is longer than } \underline{P} \\
&\qquad \textit{then } \lambda \underline{u}_i.\lambda \underline{v}'_i.y_{\pi\sigma i}\, \underline{Q}_i\, [\underline{v}_i := \underline{P}_i[\underline{x} := \underline{N}]] \\
&\qquad \textit{else } \lambda \underline{u}_i.y_{\pi\sigma i}\, (\underline{Q}_i\, [\underline{v}_i := \underline{P}_i[\underline{x} := \underline{N}]])\, (\underline{P}_i'[\underline{x} := \underline{N}])
\end{aligned}
$$

In either case of the last equality, each term can reduce to y_i iff $y_i = y_{\pi\sigma i}$ and each of the Q's and P's left orderly reduce to one of the bound variables, so that one can apply $(\to\eta)$, several times, at the end. The same holds for $N_j\, [\underline{y} := \underline{M}]$ for $1 \le j \le m$.

Thus $i = \pi\sigma i$, for $1 \le i \le n$, and $j = \sigma\pi j$, for $1 \le j \le m$ and we can conclude that $m = n$, σ is a permutation and π is its inverse. ♦

We are then reduced to examining componentwise the terms which prove an isomorphism. The next point is to show that each component, indeed a term of $\lambda\beta\eta^t$ by lemma 2.5, yields an isomorphism.

3. FINITE HEREDITARY PERMUTATIONS

In order to prove that the isomorphism between two type-n.f.'s can be expressed componentwise, we use a theorem in Dezani[1976]. The same result will also be applied to obtain, at last, the remaining part of the proof of our Main Theorem.

3.1 DEFINITION. Let M be an untyped term. Then M is a **finite hereditary permutation** (f.h.p.) iff either

(i) $\lambda\beta\eta \vdash M = \lambda x.\, x$, or

(ii) $\lambda\beta\eta \vdash M = \lambda z.\, \lambda \underline{x}.\, z\, \underline{N}_\sigma$, where if $|\underline{x}| = n$ then σ is a permutation over n and

$z\,\underline{N}_\sigma = z\, N_{\sigma 1}\, N_{\sigma 2} \ldots N_{\sigma n}$, such that, for $1 \le i \le n$, $\lambda x_i.\, N_i$ is a finite hereditary permutation.

Thus $\lambda z.\, \lambda x_1.\, \lambda x_2.\, z\, x_2\, x_1$ and $\lambda z.\, \lambda x_1.\, \lambda x_2.\, z\, x_2\, (\lambda x_3.\, \lambda x_4.\, x_1\, x_4\, x_3)$ are f.h.p.'s. F.h.p.'s can also be tidily described in terms of Böhm-trees. Recall that a **Böhm-tree** of a term M is (informally) given by:

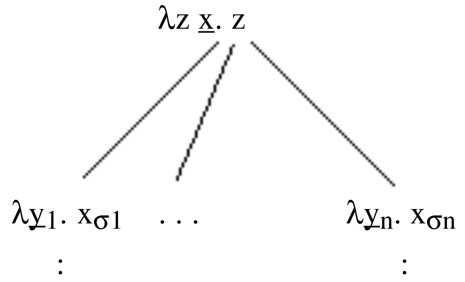
$BT(M) = \Omega$ if M has no head normal form

$BT(M) = \lambda x_1 \ldots x_n.\, y$ if $M =_\beta \lambda x_1 \ldots x_n.\, y\, M_1 \ldots M_p$

/ ... \

$BT(M_1)$ $BT(M_p)$

Recall also that $BT(M)$ is finite and Ω-free iff M has a normal form. Then one may look at f.h.p.'s as Böhm-trees, as follows:



and so on, up to a finite depth (note that $\underline{y}_i$ may be an empty string of variables). Clearly f.h.p.'s are closed terms and they possess normal form. In particular, exactly the abstracted variables at level $n+1$ appear at level $n+2$, modulo some permutation of the order (note the special case of z at level 0). The importance of f.h.p.'s arises from the following theorem. (Clearly, the notion of invertible term given in 1.3 easily translates to the untyped λ-calculus).

3.2 THEOREM. (Dezani[1976]): Let M be an untyped term possessing normal form. Then M is $\lambda\beta\eta$-invertible iff M is a f.h.p..

REMARK**.** One may easily show that the f.h.p.'s are typeable terms (Hint: Just follow the inductive definition and give z, for instance, type $A_1 \to (A_2 \ldots\ldots \to B)$, where the A_i's are the types of the $N_{\sigma i}$.) By the usual abuse of language we may then speak of typed f.h.p.'s.

Recall now that all typed terms possess a (unique) normal form (see Barendregt [1984]). As we now need an interplay between typed and type-free terms, we are going to be more explicit about which sort of terms we are dealing with, when needed. Let M be a typed λ-term. We write e(M) for the **erasure** of M, i.e. for M with all type labels on variables erased.

REMARK: Observe that the erasures of all axioms and rules of the typed lambda calculus are themselves axioms and rules of the untyped lambda calculus. Then, in particular, if M and N are terms of $\lambda\beta\eta^t$ and $\lambda\beta\eta^t \vdash M = N$, one has $\lambda\beta\eta \vdash e(M) = e(N)$.

3.3 THEOREM: If $M : \sigma \to \tau$ and $N : \tau \to \sigma$ are invertible terms in $\lambda\beta\eta^t$, then e(M) and e(N) are f.h.p.'s.

PROOF: $e(N \circ M) = e(N) \circ e(M)$, and hence, by the remark, $\lambda\beta\eta \vdash e(M) \circ e(N) = e(I_\sigma) = I$ and $\lambda\beta\eta \vdash e(N) \circ e(M) = e(I_\sigma) = I$. Thus by Theorem 3.2, e(M) and e(N) are f.h.p.'s.

♦

The first application of 3.2 we need is the following.

3.4 PROPOSITION. Let $M_1, \ldots M_n$, $N_1, \ldots ,N_n$ and permutation σ satisfy all the assumptions in lemma 2.6. Then $\lambda x_{\sigma i}. M_i : S_{\sigma i} \to R_i$ and $\lambda y_i. N_{\sigma i} : R_i \to S_{\sigma i}$ are invertible terms.

PROOF. For a suitable typing of the variables it is possible to build the following terms of $\lambda\beta\eta^t$:

$$M = \lambda z\, x_1 \ldots x_n.\ z\, M_1 \ldots M_n$$
$$N = \lambda z\, y_1 \ldots y_n.\ z\, N_1 \ldots N_n\ .$$

It is an easy computation to check, by the definition of the M_i's and of the N_i's, that M and N are invertible. Moreover, they are (by the construction given in the Appendix) in normal form, thus, by Dezani's theorem, (the erasures of) M and N are f.h.p.'s. This is enough to show that every M_i has only one occurrence of the x_i's (namely $x_{\sigma i}$); similarly for the N_i's.

Thus we obtain

$$M_i\, [\underline{x} := \underline{N}] \equiv M_i\, [\, x_{\sigma i} := N_{\sigma i}] =_{\beta\eta} y_i,\ \text{ for } 1 \le i \le n,$$
$$N_i\, [\underline{y} := \underline{M}] \equiv N_i\, [\, y_{\pi i} := M_{\pi i}] =_{\beta\eta} x_i,\ \text{ for } 1 \le i \le n,$$

and, hence, for each i, $\lambda x_{\sigma i}.M_i : S_{\sigma i} \to R_i$ and $\lambda y_i.N_{\sigma i} : R_i \to S_{\sigma i}$ are invertible. ♦

As a result of all the work done so far, we can then focus on invertible terms whose types contain only "$\to$", i.e. investigate componentwise the isomorphisms of type-n.f.'s. Of course, these isomorphisms will be given just by a fragment of the theory Th.

Let **S** be the subtheory of Th given by just one proper axiom (plus the usual axioms and rules for "="), namely

(swap) $A \to (B \to C) = (B \to (A \to C))$.

S is a subtheory of Th by axioms 2 and 4 of Th.

3.5 PROPOSITION: Let A, B be type expressions with no occurrences of **T** or $\times$. Then

$$A \cong_p B \;\Rightarrow\; S \vdash A = B.$$

PROOF: Suppose $A \cong_p B$ via M and N. As usual, we may assume without loss of generality that M and N are in normal form. By Lemma 2.5, M and N actually live in $\lambda\beta\eta^t$ and, by Theorem 3.3, e(M) and e(N) are f.h.p.'s. We prove $S \vdash A = B$ by induction on the depth of the Böhm-tree of M.

Depth 1: $M \equiv \lambda z : C.\ z$. Thus $M : C \to C$, and $S \vdash C = C$ by reflexivity.

Depth n+1: $M \equiv \lambda z : E.\ \lambda \underline{x} : \underline{D}.\ z\, \underline{N}_\sigma$. Recall $z\, \underline{N}_\sigma = z\, N_{\sigma 1} \ldots N_{\sigma n}$ where if the ith abstraction in $\lambda \underline{x} : \underline{D}$ is $\lambda x_i : D_i$ then the erasure of $\lambda x_i : D_i.\ N_i$ is a f.h.p. Let F_i be the type of N_i. In order to type check, we must have $E = (F_{\sigma 1} \to \ldots \to F_{\sigma n} \to B)$ for some B. Thus the type of M is $(F_{\sigma 1} \to \ldots \to F_{\sigma n} \to B) \to (D_1 \to \ldots \to D_n \to B)$. Since $\lambda x_i : D_i.\ N_i$ is a f.h.p, $\lambda x_i : D_i.\ N_i$ gives (half of) a provable isomorphism from D_i to F_i. By induction, since the height of the Böhm tree of (of the erasure of) each $\lambda x_i : D_i.\ N_i$ is less than the height of the Böhm tree of M, one has $S \vdash D_i = F_i$ for $1 \le i \le n$. By repeated use of the rules for "=", we get

$$S \vdash (F_{\sigma 1} \to \ldots \to F_{\sigma n} \to B) = (D_{\sigma 1} \to \ldots \to D_{\sigma n} \to B).$$

Hence it suffices to show

$$S \vdash (D_{\sigma 1} \to \ldots \to D_{\sigma n} \to B) = (D_1 \to \ldots \to D_n \to B).$$

This is quite simple to show by repeated use of axiom (swap) above in conjunction with the rules. ♦

Clearly, also the converse of Proposition 3.5 holds, since the "$\Leftarrow$" part in 3.5 is provable by a fragment of the proof in Theorem 1.4. Thus one has:

$$S \vdash A = B \;\Leftrightarrow\; A \cong_p B \text{ by terms in } \lambda\beta\eta^t.$$

The result we aim at is just the extension of this fact to Th and $\lambda\beta\eta\pi*^t$.

3.6 MAIN THEOREM. $S \cong_p R \;\Leftrightarrow\; Th \vdash S = R$

PROOF. In view of Theorem 1.4, we only need to prove $S \cong_p R \Rightarrow \mathsf{Th} \vdash S = R$. By Proposition 1.8, this is equivalent to proving $nf(S) \cong nf(R) \Rightarrow \mathsf{Th} \vdash nf(S) = nf(R)$.

Now, in Lemmas 2.4 and 2.6, for $nf(S) \equiv (S_1 \times ... \times S_n)$ and $(R_1 \times ... \times R_m) \equiv nf(R)$, we have shown that $nf(S) \cong nf(R) \Rightarrow n = m$ and there exist $M_1, ..., M_n, N_1, ..., N_n$ and a permutation σ such that $\lambda x_{\sigma i}. M_i : S_{\sigma i} \to R_i$ and $\lambda y_i. N_{\sigma i} : R_i \to S_{\sigma i}$.

By 3.4, these terms are invertible, for each i. Thus, by 3.5, $\mathsf{S} \vdash R_i = S_{\sigma i}$ and, hence, by the rules, $\mathsf{Th} \vdash S = R$. ♦

This concludes the proof of the main theorem. Here is an immediate consequence.

3.7 COROLLARY. Given types A and B, it is decidable whether they are isomorphic in all models of $\lambda\beta\eta\pi{*}^t$. (And thus whether A and B name isomorphic objects in all CCC's.)

PROOF. Let the type-n.f. of A be $R_1 \times ... \times R_n$ and that of B be $S_1 \times ... \times S_m$ where neither the R_i's nor the S_j's contain any occurrences of **T** or $\times$. (If one of A or B is T, the other must be as well if they are to be isomorphic.) By Propositions 2.4 and 2.6, and Theorem 3.6, A and B are isomorphic iff $m = n$ and there is a permutation σ over n such that for $1 \le i \le n$, $R_i \cong_p S_{\sigma i}$. By Proposition 3.5, we must have that $\mathsf{S} \vdash R_i = S_{\sigma i}$. Note that the axioms and rules of S do not change the length of type expressions. Hence if $\mathsf{S} \vdash R_i = S_{\sigma i}$, R_i and $S_{\sigma i}$ have the same length. We provide a decision procedure to determine if $\mathsf{S} \vdash R = S$ (and hence whether they are isomorphic in all models) by induction on the length of R (and hence S). We restrict ourselves to type expressions of the same length since otherwise they are not provably equal. If R and S are both type symbols then they are equal if and only if they are the same symbol. Suppose we have a decision procedure for all types of length less than n, and R and S have length n. Decompose R and S into terms of the form $R_1 \to R_2 \to ... \to R_k$ and $S_1 \to S_2 \to ... \to S_m$ where R_k and S_m are type symbols. If R_k and S_m are different or $k \neq m$ then it is not the case that $\mathsf{S} \vdash R = S$. Otherwise, for each R_i determine if there is a distinct S_j such that $\vdash R_i = S_j$. Each of these tests is decidable by hypothesis. If each R_i can be paired with a distinct S_j, then $\vdash \sigma = \tau$. Otherwise it fails. The proof of the correctness of this decision proceed follows the same lines as the proof of the ($\Rightarrow$) direction of Proposition 3.5. ♦

Indeed, more can be said about the connection to Category Theory; we also hint here a simple application to Proof Theory, but refer to DiCosmo and Longo (1990) for more discussions on both topics.

Take the intuitionistic positive calculus, IPC, i.e. Intuitionistic Logic with only $\rightarrow$, $\times$ (i.e. conjunction), and True, and consider the following notion of constructive equivalence.

3.7 DEFINITION. Two formulas A and B of IPC are constructively equivalent iff there are proofs f of the sequent $A \vdash B$ and g of the sequent $B \vdash A$ such that the proofs $g \circ f$ and $f \circ g$ obtained by composition reduce, by cut-elimination, to the one step deductions $A \vdash A$ and $B \vdash B$.

Notice that this notion of equivalence is much stronger than the classical notion of logical equivalence: all tautologies of IPC are logically equivalent, for example, but only a few are equivalent constructively.

3.8 COROLLARY (*Connection with deductive systems*) Two formulas A and B of IPC are constructively equivalent iff $\mathsf{Th} \vdash A = B$.

APPENDIX

We give here the proofs of the lemmas in section 2. The numbers refer to that section.

2.1 LEMMA. (*Terms of a product type*) Given a term M of $\lambda\beta\eta\pi^{*t}$ in normal form such that $M : A \times B$, then either $M \equiv \langle M_1, M_2\rangle$ or there is an $x : C$ such that $x \in FV(M)$ and $A \times B$ is a type subexpression of C.

PROOF. By induction on the length of structure of M.
Basis of induction: if M is of length 1, then it can be only a free variable of type $A \times B$.
Inductive step: $M \equiv \lambda \underline{x}.\ r\ \underline{P}$, as it is in normal form. Observe first that this case reduces to $M \equiv r\,\underline{P}$, as its type is $\alpha \times \beta$, and we proceed by case analysis on r as follows:
r is a variable: Then r is free and has type $\text{type}(P_1) \rightarrow (\ldots \rightarrow (\text{type}(P_n) \rightarrow A \times B) \ldots)$.
r is $\langle M_1,M_2\rangle$: Then $M \equiv \langle M_1,M_2\rangle$, in order to type check.
r is p_1 or p_2: Then $M \equiv (\ldots(p_i\, M_1)\, M_2 \ldots M_k)$ with $M_1 : S \times U$ and M_1 in normal form with S or $U \equiv \text{type}(M_2) \rightarrow (\ldots \rightarrow (\text{type}(M_k) \rightarrow A \times B) \ldots)$. By induction hypothesis either M_1 is $\langle N_1,N_2\rangle$ or M_1 has a free variable $x : C$ with $S \times U$ (hence $A \times B$ too) a type subexpression of C. The first case is not possible, as $p_i \langle N_1,N_2\rangle$ is a redex, so M has a free variable $x : C$ with $A \times B$ a type subexpression of C.

Finally, r cannot be any constant *A as *A has type $A \rightarrow \mathbf{T}$, so M would definitely not have type $A \times B$. ♦

2.2 LEMMA. (*Every term, whose type contains no occurrence of* ***T****, has no occurrence of *A constants*) Assume that in a term M of $\lambda\beta\eta\pi^{*t}$ in normal form there is an occurrence of *A, for some type A. Then there is some occurrence of the type constant **T** in the type of M or in the type of some free variable of M.

PROOF. By induction on the structure of M.
Basis for induction: *A has type $A \rightarrow \mathbf{T}$.
Inductive step: $M \equiv \lambda\underline{x}.r\,\underline{P}$, as M is in normal form, and we proceed by case analysis on r as follows:
r is a variable: Then r has type $\text{type}(P_1) \rightarrow (\ldots \rightarrow (\text{type}(P_n) \rightarrow C) \ldots)$; by hypothesis, the P_i's are in normal form and in some P_j there are occurrences of a constant *A, so by induction hypothesis there are **T**'s in $\text{type}(P_j)$, hence in the type of r. By this, either r is a free variable or (since r occurs among the $\underline{x}$) there are **T**'s in the type of M.

r is <P,Q>: Then $M \equiv \lambda\underline{x}.\langle P,Q\rangle$ where P and Q are in normal form The type of M is $D_1 \to \ldots \to D_n \to (A \times B)$, with $P : A$ and $Q : B$, and *A occurs in P or Q. By inductive hypothesis, either **T** occurs in $A \times B$ (hence in the type of M, too) or in the type of some free variable y of P or Q. In either case, as above, some **T**'s occur in the type of M or in the type of y, which is free in M.

r is p_1 or p_2: then $M \equiv \lambda\underline{x}.((p_i\, M_1)\, M_2 \ldots M_k)$ where:

- M_j is in normal form, for each j.
- $M_1 : S \times U$ with either S or $U \equiv \text{type}(M_2) \to (\ldots \to (\text{type}(M_k) \to C) \ldots)$.
- *A occurs in M_j for some j; consider than

case j = 1: then **T** occurs in $S \times U$, by induction hypothesis. By Lemma 2.1, as M cannot be a redex, M_1 is not a pair and M_1 has a free variable $y : C$ with $S \times U$ a type subexpression of C. Notice that y is also free in $((p_i\, M_1)\, M_2 \ldots M_k)$. Thus as in the earlier cases either y is free in M or some T's occur in the type of M (because y is one of the variables in $\underline{x}$);

case j > 1: then by induction hypothesis either

(a) there is a **T** occurring in the type of M_j, and, hence, in $S \times U$ or

(b) there is a free variable y of M_j with type **T** occurring in its type. In case (a), we can conclude the proof as in the case for j = 1 above. In case (b), if y is free in M_j then it is also free in $((p_i\, M_1)\, M_2 \ldots M_k)$. We can thus conclude the proof again as for i = 1.

r is $*\alpha$: Then $M \equiv \lambda\underline{x}.{*A}\, M_1$ or $M \equiv \lambda\underline{x}.{*A}$ and the type of M is $D_1 \to \ldots D_n \to \mathbf{T}$, for some $D_1, \ldots, D_n$. ♦

2.4 PROPOSITION. (*Isolate the relevant* $\langle M_1,\ldots,M_n\rangle$ *in an isomorphism*) Let $S \equiv S_1 \times \ldots \times S_m$ and $R \equiv R_1 \times \ldots \times R_n$ be type-n.f.'s where neither the S_i's nor the R_j's contain any occurrences of **T** or $\times$.

Then $S \cong_p R$ iff there exist $M_1,\ldots, M_n$ and $N_1,\ldots, N_m$ such that

$$x_1 : S_1, \ldots, x_m : S_m \vdash \langle M_1,\ldots,M_n\rangle : (R_1\times \ldots \times R_n)$$
$$y_1 : R_1, \ldots, y_n : R_n \vdash \langle N_1,\ldots,N_m\rangle : (S_1\times \ldots \times S_m)$$

and
$$M_i [\, \underline{x} := \underline{N}] =_{\beta\eta} y_i\,, \quad \text{for } 1 \le i \le n$$
$$N_j [\, \underline{y} := \underline{M}] =_{\beta\eta} x_j\,, \quad \text{for } 1 \le j \le m$$

(where substitution of vectors of equal length is meant componentwise).

PROOF. ($\Rightarrow$) Let $M^\circ : S \to R$ and $N^\circ : R \to S$ be closed terms (in normal form) of $\lambda\beta\eta\pi^t$ such that $M^\circ \circ N^\circ = I_R$ and $N^\circ \circ M^\circ = I_S$. Then by standard currying, consider the term $\lambda x_1 \ldots x_m.M^\circ \langle x_1, \ldots x_m\rangle : (S_1 \to \ldots \to (S_m \to (R_1 \times \ldots \times R_n) \ldots)$, and observe that the normal form M' of $M^\circ <x_1, \ldots x_m> : R_1 \times \ldots \times R_n$, by Lemma 2.1, must be of the form $<M_1,\ldots,M_n>$, with $FV(M') = \{x_1: S_1, \ldots x_m: S_m\}$ (by assumption, the S_i's contain no occurrences of $\times$). The same applies for N°.

As for the other properties, let

$M'' \equiv \lambda z.(\lambda x_1\ldots x_m.M^\circ <x_1, \ldots x_m>)(p_1 z) \ldots (p_m z)$, and

$N'' \equiv \lambda z.(\lambda y_1\ldots y_n.N^\circ <y_1, \ldots y_n>)(p_1 z) \ldots (p_n z)$,

where the x_i's, y_j's, and z are chosen to be distinct.

Then

$M'' =_\beta \lambda z.M^\circ <p_1 z, \ldots p_m z> =_\eta \lambda z.M^\circ z =_\eta M^\circ$, and

$N'' =_\beta \lambda z.N^\circ z =_\eta N^\circ$.

Compute then

$M^\circ \circ N^\circ =_{\beta\eta} M''\circ N'' \equiv \lambda x.(M''(N''x))$ for x a variable not occurring in M'' or N''.

$=_{\beta\eta} \lambda x.(\lambda z.(\lambda x_1\ldots x_m.M')(p_1 z) \ldots (p_m z))(N''x)$

$=_{\beta\eta} \lambda x.<M_1[\underline{x}_j := \underline{p}_j (N''x)],\ldots,M_n[\underline{x}_j := \underline{p}_j (N''x)]>$

where the substitution is done simultaneously for all $1 \le j \le m$,

$=_{\beta\eta} \lambda x.<M_1[\underline{x} := \underline{N}[\underline{y}_i := \underline{p}_i x]],\ldots,M_n[\underline{x} := \underline{N}[\underline{y}_i := \underline{p}_i x]]>$ since $N''x =_{\beta\eta}$

$\lambda y_1\ldots y_n.N'(p_1 x) \ldots (p_n x) =_{\beta\eta} <N_1[\underline{y}_i := \underline{p}_i x],\ldots,N_m[\underline{y}_i := \underline{p}_i x]>$

where substitution is done simultaneously for all $1 \le i \le n$

$=_{\beta\eta} \lambda x.<M_1[\underline{x} := \underline{N}][\underline{y}_i := \underline{p}_i x],\ldots,M_n[\underline{x} := \underline{N}][\underline{y}_i := \underline{p}_i x]>$

by substitution properties, as no y_i is free in M'

$=_{\beta\eta} \lambda x.<p_1 x,\ldots, p_n x>$

since $M^\circ\circ N^\circ =_{\beta\eta} \lambda x.x$ and $x =_{\beta\eta} <p_1 x,\ldots, p_n x>$.

Observe now that the equality just proved implies, componentwise, that

$$M_k[\underline{x} := \underline{N}][\underline{y}_i := \underline{p}_i x] =_{\beta\eta} p_k x.$$

For the purpose of the final argument of the proof, we refer now to "$\rightarrow_{\beta\eta\pi'}^{*}$" as a "several steps reduction" in $\lambda\beta\eta\pi^t$. In view of the Church-Rosser property for this calculus, the last equality is equivalent to

$$M_k[\underline{x}_i := N_i][\underline{y}_j := p_j w] \rightarrow_{\beta\eta\pi'}^{*} p_k w$$

where w is a fresh variable (to avoid confusion between $\underline{x}$ and x; in other words, w is not free in M_k nor in any N_i and cannot be free in any reduct of $M_k[\underline{x}_i := N_i]$ either.) Furthermore, the substituition $[\underline{y}_j := p_j w]$ creates no new redexes, so that if we have $M_k[\underline{x}_i := N_i][\underline{y}_j := p_j w] \rightarrow_{\beta\eta\pi'}^{*} Q$ then $M_k[\underline{x}_i := N_i] \rightarrow_{\beta\eta\pi'}^{*} Q'$ with $Q \equiv Q'[\underline{y}_j := p_j w]$. This implies, in the case of $M_k[\underline{x}_i := N_i][\underline{y}_j := p_j w] \rightarrow_{\beta\eta\pi'}^{*} p_k w$, that $M_k[\underline{x}_i := N_i] \rightarrow_{\beta\eta\pi'}^{*} Q'$ with $p_k w \equiv Q'[\underline{y}_j := p_j w]$, that is $M_k[\underline{x}_i := N_i] \rightarrow_{\beta\eta\pi'}^{*} y_k$. In conclusion, $M_k[\underline{x}_i := N_i] = y_k$, as required.

Symmetrically, one obtains $N_j[\,\underline{y} := \underline{M}] =_{\beta\eta} x_j$ from $N^\circ \circ M^\circ = \lambda x.x$.

$\Leftarrow$) Just step through the above proof in reverse order, defining the required closed terms by

$$M^\circ \equiv \lambda z.(\lambda x_1 \ldots x_m.\langle M_1, \ldots M_n\rangle)(p_1\, z) \ldots (p_m\, z),$$
$$N^\circ \equiv \lambda z.(\lambda y_1 \ldots y_n.\ \langle N_1, \ldots N_m\rangle)(p_1\, z) \ldots (p_n\, z). \qquad \blacklozenge$$

2.5 LEMMA. (*Terms of $\lambda\beta\eta\pi^t$ whose type is arrow-only belong to $\lambda\beta\eta^t$*) Let M be a term of $\lambda\beta\eta\pi^t$ in normal form such that $M : A$, where A is a type with no occurrence of $\times$ in it. If no free variable of M has a type with occurrences of $\times$, then M is actually a term in $\lambda\beta\eta^t$.

Proof By induction on the structure of M.
Basis for induction: if M is of length 1, then it can be only a variable of type A, as any constant has a type with occurrences of $\times$.
Inductive step: $M \equiv \lambda\underline{x}.r\,\underline{P}$, as M is in normal form, and we proceed by case analysis on r as follows:
r is a variable: then r has type $\mathrm{type}(P_1)\to (\ldots \to (\mathrm{type}(P_n)\to C) \ldots)$ and no matter if r is free or bound, by hypothesis on the type of M and its free variables, the P_i's (which are in normal form) have a type with no $\times$'s and free variables whose type have no $\times$'s, so by induction hypothesis they contain no constants nor pairs, hence M contains no constants or pairs either.
r is $\langle P,Q\rangle$: this is impossible, otherwise $M \equiv \lambda\underline{x}.\langle P,Q\rangle$ and the type of M would be $S_1 \to \ldots \to S_n \to (A \times B)$, which contains $\times$.
r is p_1 or p_2: this cannot be either, since:
- $M \equiv \lambda\underline{x}.p_i$ must have a type containing $\times$,
- $M \equiv \lambda\underline{x}.((p_i\, M_1)\, M_2 \ldots M_k)$ implies, by Lemma 2.1, that either M_1 is $\langle N_1,N_2\rangle$ or M_1 has a free variable $x : C$ with $S \times U$ a type subexpression of C. The first case is not possible, as $p_i \langle N_1,N_2\rangle$ is a redex while M is in normal form. Thus M_1 has a free variable $x : C$ with $S \times U$ a type subexpression of C, and, hence, either $x \in FV(M)$ or $S \times U$ is a type subexpression of the type of M, since the type of M includes the types of bound variables. Impossible. $\blacklozenge$

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Gregory Mints for pointing out to us the work of Soloviev, and Rittri for informing us of his own work in the topic. We would also like to thank Albert Meyer and John Mitchell for several helpful conversations on these matters.

REFERENCES

Asperti A., Longo G. (1990) **Categories, types and structures: an introduction to category theory for the working computer scientist**, M.I.T. Press, *to appear.*

Babaev A., Soloviev S.V. (1982) "Coherence theorem for canonical maps in Cartesian Closed Categories" **J. Soviet Mathematics** 20.

Barendregt, H. (1984) **The lambda calculus; its syntax and semantics**. Revised edition, North Holland.

Böhm, C., Dezani-Ciancaglini, M.(1974) "Combinatorial problems, combinator equations and normal forms, in : J. Loeckx, ed., **Automata, Languages and Programming**, LNCS 14 (185-199).

Dezani-Ciancaglini, M. (1976) "Characterization of normal forms possessing an inverse in the $\lambda\beta\eta$-calculus," **Theoretical Computer Science**, 2 (323-337).

Di Cosmo, R., Longo G. (1990) "Constructively Equivalent Propositions and Isomorphisms of Objects (or Terms as Natural Transformations)" Workshop on **Logic from Computer Science**, M.S.R.I., Berkeley (Ca.), November 1989; *to appear.*

Martin C.F. (1972) "Axiomatic bases for equational theories of natural numbers," **Notices Am. Math. Soc.** 19,7, p.778.

Narendran P., Pfenning F., Statman R. (1989) "On the unification problem for Cartesian Closed Categories," abstract, CMU.

Reynolds, J. (1984), "Polymorphism is not set-theoretic," Symposium on Semantics of Data Types, (Kahn, MacQueen,Plotkin, eds.) LNCS 173, Springer-Verlag.

Rittri M. (1989) "Using types as Search Keys in Function Libraries," **IV Conf. in Funct. Progr. and Computer Architecture**, London, ACM Press, Addison-Wesley.

Rittri M. (1989) "Retrieving library identifiers whose types are at least as general, modulo CCC-isomorphisms, as a given type". Dept. of C.S., Chalmers University, Sweden.

Soloviev S.V. (1983) "The category of finite sets and Cartesian Closed Categories" **J. Soviet Mathematics** 22, 3, pp. 1387-1400.

Statman, R. (1983) "λ-definable functionals and $\beta\eta$-conversion," **Arch. Math. Logik**, 23 (21-26).