Second order Isomorphic Types

A proof theoretic study on second order λ-calculus with surjective pairing and terminal object*

Roberto Di Cosmo

LIENS (CNRS) - DMI
Ecole Normale Supérieure
45, Rue d'Ulm
75005 Paris - France
E-mail: dicosmo@dmi.ens.fr

Dipartimento di Informatica
Corso Italia, 40
56100 Pisa -Italy
E-mail: dicosmo@dipisa.DI.UNIPI.IT

September 1, 1993

Abstract

We investigate invertible terms and isomorphic types in the second order lambda calculus extended with surjective pairs and terminal (or *Unit*) type. These two topics are closely related: on one side, the study of invertibility is a necessary tool for the characterization of isomorphic types; on the other hand, we need the notion of isomorphic types to study the typed invertible terms. The result of our investigation is twofold: we give a constructive characterization of the invertible terms, extending previous work by Dezani and Bruce-Longo [Dez76, BL85], and a decidable equational theory of the isomorphisms of types which hold in all models of the calculus, which is a conservative extension to the second order case of the results previously achieved for the case of first order typed calculi. Via the Curry-Howard correspondence, this work also provides a decision procedure for *strong equivalence* of formulae in second order intuitionistic positive propositional logic, that is suitable to search equivalent proofs in automated deduction systems.

KEYWORDS: Type Isomorphisms, Invertibility, System F, Polymorphism, Surjective Pairing, Terminal Object, Equational Theories, λ-Calculus, Proof Search.

Contents

*This work has been partially supported by C.N.R. grant AI89.02010.12

1 Introduction

What is more simple, intuitive and well known than the notion of an *isomorphism*? In any textbook in set theory, it appears at the very beginning: almost immediately after we are told what a set is, we find the definition:

> Two sets A and B are said to be isomorphic, noted $A \cong B$, if and only if there exist two functions $f : A \to B$ and $g : B \to A$ such that
>
> $$\forall x \in A, g(f(x)) = x \text{ and } \forall y \in B, f(g(y)) = y$$

or maybe some equivalent one, that tells us about *surjective* and *injective* mappings.

Later on, if we ever open up a textbook on category theory, we find, always at the very beginning, a very similar definition of isomorphic *objects*, and we are immediately told that isomorphisms are so basic in category theory, that all notions introduced in the book will be "up to isomorphisms".

> An arrow $e : A \to B$ is *invertible* in C if there is an arrow $e' : B \to A$ in C with $e'e = 1_A$ and $ee' = 1_B$ *(omissis ...)* Two objects are *isomorphic* in the category C if there is an invertible arrow (an *isomorphism*) $e : A \to B$; we write $A \cong B$. *([ML71], pag. 19).*

In the world of typed λ-calculus, the *types* play the role of sets, or objects, and typed terms that of functions, or arrows, so we can talk about *definable isomorphic types* using the following definition:

Definition 1.1 (Definable isomorphisms) *Two types A and B are definably isomorphic ($A \cong_d B$) iff there exist λ-terms $M : A \to B$ and $N : B \to A$ such that $M \circ N = I_B$ and $N \circ M = I_A$, where I_A and I_B are $\lambda x : A.x$ and $\lambda x : B.x$, the identities of type A and B. We also write $M : A \cong_d B : N$ when we want to make the associated invertible terms explicit.*

Note that, as usual, the inverse of a term M (if it exists) is unique (up to equality). Suppose that types A and B are definably 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 [Sta83], we may say that the notion of definable isomorphism is typically ambiguous.

There is also a semantic alternative to this syntactic notion of isomorphism of types: when we provide models of typed λ-calculi, one usually interprets types as sets with additional properties, and terms $M : A \to B$ as continuous functions from the interpretation of A to that of B. So we can consider the following definition.

Definition 1.2 (Semantic isomorphisms) *Two types A and B are isomorphic in a specific model $\mathcal{M}$ if their interpretations are isomorphic in $\mathcal{M}$ in the traditional sense (i.e. there are in the model invertible functions f and g between them), and then we write $\mathcal{M} \vdash A \cong B$. Two types will be* semantically isomorphic*, noted $A \cong B$, if $\mathcal{M} \vdash A \cong B$ holds for every model $\mathcal{M}$ of the calculus.*

What is then the relation between the semantic notion of isomorphism of types holding in a specific model of the calculus and the syntactic notion of *definable* isomorphism? In principle, the class of the types isomorphic in every model is larger than that of definable isomorphic types, since the functions f and g that we find in a model need not be definable by terms of the calculus: the adjective *definable* in Definition 1.1 is really meant to stress the fact that $A \cong_d B$ is an isomorphism that can be defined *uniformly* in the calculus, which is not necessarily always the case. In all the cases considered up to now, including this paper, it turns out that one can esily show that the *definable* isomorphisms are exactly the ones that hold in every model of the calculus.

Theorem 1.3 *Let A, $B \in \mathbf{Tp}$. Then $A \cong B \iff A \cong_d B$.*

Proof. ($\Leftarrow$) trivial.
($\Rightarrow$) Take the open term model of the calculus. $\square$

We can then speak simply of *isomorphisms* from now on.

The relevance of isomorphisms of types in specific models is very well known in denotational semantics, where one of the major successes have been the ability to build models of typed calculi that validate domain equations like $D = A + [D \to D]$, i.e. where D and $A + [D \to D]$ are isomorphic: this result is a central tool in giving semantics to programming languages.

While a definable isomorphism $M : A \cong_d B : N$ does hold in every possible model $\mathcal{M}$ of the calculus (the isomorphism in $\mathcal{M}$ being provided by the interpretations of M and N), an isomorphism holding in a specific model usually do not hold in *all* the models of the typed λ-calculus and is not *definable*.

There has been quite a deal of investigation carried on in the last decade concerning isomorphisms of types that do hold in every model of a given typed λ-calculus. Specifically, one is interested in deciding if two given types are or not isomorphic in every model, and in finding a theory of equality Th such that $Th \vdash A = B \iff A \cong B$. We will summarize here the connections of this topic to related ones from functional programming, proof theory and category theory, that also provide the basic motivations for this work.

Before carrying on we suggest the reader have a quick look through Appendix A, where he finds a formal definition of the typed calculi we will be dealing with in this paper, as well as some useful notation.

Functional Programming

Various versions of typed λ-calculi are the core of many strongly typed functional languages of today, like Haskell, Miranda, the family of the ML languages and several others. In the community that uses these systems there is a growing need of formal tools to retrieve functions

in large functional libraries in a smarter way than just scanning an alphabetical list. It turns out that types provide, in this framework, the right search key, when isomorphic types are identified [RT91, Rit91, Rit90, Mor91, DC92b]. In many cases, the search systems based on this idea just use the theory of isomorphisms for some version of typed λ-calculus, typically $\lambda^1\beta\eta\pi*$ (but see [DC92a] for further discussion).

Category Theory

Since in category theory most of the concepts are defined up to isomorphisms, it is particularly important to be able to decide if two objects built in different ways are actually the "same" object or not, i.e. if they are isomorphic. A typical example is the case of the categorical product, that is always commutative:

$$A \times B \cong B \times A.$$

For example, there is work dedicated to the characterization of the isomorphisms holding in all the cartesian closed categories, or *CCC*'s [Sol83].

Now, categorical tools are increasingly used to give semantics to programming languages, especially to functional ones. This is also due to a close connection between some classes of categories and typed λ-calculus: it is by now a very well known fact that the models of $\lambda^1\beta\eta\pi*$ are exactly the cartesian closed categories, or, equivalently, that $\lambda^1\beta\eta\pi*$ is the *internal language* of *CCC*'s (see [Min77, LS86, AL91]).

Due to this connection, this categorical problem coincides with the characterization of isomorphisms in typed lambda calculus, for the case of $\lambda^1\beta\eta\pi*$.

Proof Theory

Since the seminal work by Howard (finally published in [How80], but widely circulated long before), the so-called *Curry-Howard isomorphism* has been one of the central themes in the theory of functional programming, as it is a bridge that allows to carry known results back and forth from one field to the other. It essentially says that typed λ-calculus can be seen as a system of notation for proofs in Intuitionistic Logic (obviously, different logical systems correspond to different calculi), in such a way that a given proposition A can be proved in the logical system $\mathcal{L}$ if and only if one can find a typed term $M : A$ in the corresponding calculus.

There is an exact correspondence, for example, between $\lambda^1\beta\eta\pi*$ and the Intuitionistic Positive Calculus over the connectives $\mathbf{True}, \wedge, \Rightarrow$, or $IPC(\mathbf{True}, \wedge, \Rightarrow)$, when one reads $\mathbf{T}$, $\times$, $\rightarrow$ respectively as $\mathbf{True}, \wedge, \Rightarrow$.

Isomorphic types then come to correspond to propositions that are *strongly equivalent*, in the terminology of [AB91, Mar91]:

Definition 1.4 (Strongly equivalent propositions) *Two propositions A and B are* strongly equivalent *if*

- *$A \Longleftrightarrow B$ (i.e. there exist derivations f:$A \Rightarrow B$ and g:$A \Leftarrow B$)*
- *the composition g$\circ$f of f:$A \Rightarrow B$ and g:$B \Rightarrow A$ by (CUT)*

$$\dfrac{\dfrac{f}{A \Rightarrow B} \qquad \dfrac{g}{B \Rightarrow A}}{A \vdash A}\ (CUT)$$

reduces, after proof normalization to the Axiom $\dfrac{}{A \vdash A}$, and viceversa.

The proof normalization obviously includes the elimination of the *cut* rule, but can also involve other structural equivalences of proof, like the equivalent of η and SP of our calculus, that amount to simplification to atomic assumptions.

The problem here is to characterize *strong equivalence*. Two strongly equivalent formulae A and B have the same constructive content, so that it is possible to immediately turn a proof of A into a proof of B and viceversa. This property can be clearly used to look in a library of theorems for the proof of some formula (type) in a proof system like LF, CoC, NuPRL or the like.

The theories of isomorphisms

As we have briefly seen here, the problems of characterizing isomorphic types, isomorphic objects and strongly equivalent propositions is really just one problem, as the solution to one of these is the solution to all the others. It is time for a summary of the known result: we present in Table 1 the known theories of isomorphic types for typed λ-calculi, using a notation that is consistent with the names of the calculi.

$$\left.\begin{array}{ll} (\textbf{swap}) & A \to (B \to C) = B \to (A \to C) \end{array}\right\} Th^1$$

$$\left.\begin{array}{lll} 1. & A \times B = B \times A & \\ 2. & A \times (B \times C) = (A \times B) \times C & \\ 3. & (A \times B) \to C = A \to (B \to C) & \\ 4. & A \to (B \times C) = (A \to B) \times (A \to C) & \\ 5. & A \times \mathbf{T} = A & \\ 6. & A \to \mathbf{T} = \mathbf{T} & \\ 7. & \mathbf{T} \to A = A & \end{array}\right\} Th^1_{\times T}$$

$$\left.\begin{array}{lll} 8. & \forall X.\forall Y.A = \forall Y.\forall X.A & \\ 9. & \forall X.A = \forall Y.A[Y/X] \quad \textit{(X free for Y in A, } Y \notin FTV(A)\textit{)} & \\ 10. & \forall X.(A \to B) = A \to \forall X.B \quad (X \notin FTV(A)) & \\ & & + \textbf{swap} = Th^2 \\ 11. & \forall X.A \times B = \forall X.A \times \forall X.B & \\ 12. & \forall X.\mathbf{T} = \mathbf{T} & \end{array}\right\} Th^2_{\times T}$$

A, B, C can be arbitrary types and **T** is a constant for the unit type.
Notice that the axiom **swap** of Th^1 is provable in $Th^1_{\times T}$ by axioms 1 and 3.

Table 1: The theories of valid isomorphisms for explicitly typed languages

So, for example, $Th^1_{\times T}$ characterizes the isomorphic types for $\lambda^1\beta\eta\pi*$: the number tells us that it is a first order typed calculus, while the subscripts tell us that it has surjective pairing and unit type. Table 2 gives an overview of the connections between calculi, categories, proof systems and the associated theories, providing also bibliographical pointers to the published proofs of completeness of the theories.

λ- Calculus	Category	Logical System	Theory	Authors
$\lambda^1\beta\eta$		$IPC(\Rightarrow)$	Th^1	([Mar72])[1], [BL85]
$\lambda^1\beta\eta\pi*$	CCC	$IPC(\mathbf{True}, \wedge, \Rightarrow)$	$Th^1_{\times T}$	[Sol83], [BDCL92]
$\lambda^2\beta\eta$		$IPC(\forall, \Rightarrow)$	Th^2	[BL85]
$\lambda^2\beta\eta\pi*$		$IPC(\forall, \mathbf{True}, \wedge, \Rightarrow)$	$Th^2_{\times T}$	this paper

Table 2: Isomorphisms of types, objects and formulae

In this paper, we contribute to this work by providing the characterization of isomorphic types in second order λ-calculus extended with products and a unit type (or $\lambda^2\beta\eta\pi*$, see A). The theory of isomorphic types for this calculus, $Th^2_{\times T}$ in Table 1, subsumes all the previously known ones (the proofs are also fairly more complex than the previous ones).

2 Survey

In this section we will survey the proof techniques that have been used in the literature to show the soundness and completeness of the various theories of isomorphisms that we presented in the introduction. Let's start by making formally clear what we mean by soundness and completeness here.

Definition 2.1 (Soundness, completeness) *We say that an equational theory Th is a* sound *theory of isomorphisms for a calculus (resp. class of categories, logical system) if*

$$\forall A, B\ Th \vdash A = B \Rightarrow A \cong B.$$

Respectively, an equational theory Th is a complete *theory of isomorphisms for a calculus (resp. class of categories, logical system) if*

$$A \cong B \Rightarrow \forall A, B\ Th \vdash A = B.$$

As can be expected, the soundness property is quite easy to show, and does not present any real technical interest, while the completeness is a much harder property, and there are interestingly different techniques that can be used to establish it.

2.1 Soundness

For each of the different theories, it is possible to prove soundness either in a category theoretic way (the axioms of $Th^1_{\times T}$ are valid isomorphisms in every CCC: it is an easy exercise in

[1] This number theoretic study did not address explicitly the problem of isomorphic types, but it is nevertheless related to it, as we explain in Section 2.2.

elementary category theory), or by proof theoretic techniques, but surely the easiest and more uniform way to soundness is by providing invertible terms of the corresponding typed calculi. The following proof actually provides us with soundness not only for $Th^2_{\times T}$, but also for all the other theories seen up to now: it suffices to see the corresponding invertible terms as embedded in the appropriate calculus. For example, $\lambda x : A \to (B \to C).\lambda y : B.\lambda z : A.xzy$ proves $A \to (B \to C) = B \to (A \to C)$ in $\lambda^1\beta\eta$ if seen as a term of $\lambda^1\beta\eta$, or in $\lambda^1\beta\eta\pi*$ if seen as a term of $\lambda^1\beta\eta\pi*$ and so on.

Theorem 2.2 *[Main Theorem, soundness]* $Th^2_{\times T} \vdash A = B \Rightarrow A \cong B$.

Proof. By Theorem 1.3, it is enough to show that $Th^2_{\times T} \vdash$A = B $\Rightarrow$ A $\cong_d$ B.

For this purpose, we give the terms associated to each axiom and rule. As $Th^2_{\times T}$ is a theory of equality, one has first to observe that the usual axioms and inference rules yield and preserve provable isomorphisms:

- λx: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 an invertible N proves B = C, then the term N$\circ$M = λx:A.N (M x), that is clearly invertible, proves A = C ;
- if an invertible term M proves A = B and an invertible term N proves C = D, then the invertible term λx:$A \times C.\langle M(\mathrm{p}_1 x), N(\mathrm{p}_2 x)\rangle$ proves $A \times C = B \times D$;
- if an invertible M proves A = B and an invertible N proves C = D, then λy:$A \to C.\lambda$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 λy:$B \to D.\lambda$x:A.N^{-1} (y (M x))).
- if an invertible M proves A = B, then λx:$\forall X.A.\lambda X.M(x[X])$ proves $\forall X.A = \forall X.B$ and it is invertible (take λy:$\forall X.B.\lambda X.M^{-1}(y[X])$).

We next check the proper axioms:

1. $A \to (B \to C) = B \to (A \to C)$ is proved by λx:$A \to (B \to C).\lambda$y:B.λz:A.xzy;
2. $A \times B = B \times A$ is proved by λx:$A \times B.\langle \mathrm{p}_2 x, \mathrm{p}_1 x\rangle$;
3. $A \times (B \times C) = (A \times B) \times C$ is proved by λx: $A \times (B \times C).\langle\langle \mathrm{p}_1 x, \mathrm{p}_1(\mathrm{p}_2 x)\rangle, \mathrm{p}_2(\mathrm{p}_2 x)\rangle$, that is invertible;
4. $(A \times B) \to C = A \to (B \to C)$
   is proved by λz:$(A \times B) \to C.\lambda$x:A.$\lambda$y:B.z$\langle x, y\rangle$
   with inverse λz:$A \to (B \to C).\lambda$x:$A \times B$.z (p_1x) (p_2x);
5. $A \to (B \times C) = (A \to B) \times (A \to C)$
   is proved by λz:$A \to (B \times C).\langle \lambda x : A.(\mathrm{p}_1(zx)), \lambda x : A.(\mathrm{p}_2(zx))\rangle$
   with inverse λz:$(A \to B) \times (A \to C).\lambda$x:A.$\langle(\mathrm{p}_1 z)x, (\mathrm{p}_2 z)x\rangle$;
6. $\forall X.\forall Y.A = \forall Y.\forall X.A$
   is proved by λx:$(\forall X.\forall Y.A).\lambda Y.\lambda X.((x[X])[Y])$
   with inverse λy:$(\forall Y.\forall X.A).\lambda X.\lambda Y.((y[Y])[X])$;
7. $\forall X.A = \forall Y.A$[Y/X]
   is proved by λx:$(\forall X.A).\lambda Y.(x[Y])$,
   with inverse λy:$(\forall Y.A$[Y/X]$).\lambda X.(y[X])$,
   provided that X is free for Y in A and Y is not free in A;

8. $\forall X.(A \to B) = A \to \forall X.B$

   is proved by λx:$(\forall X.(A \to B)).\lambda$y:A.$\lambda X.(x[X])y$,

   with inverse λz:$(A \to \forall X.B).\lambda X.\lambda w : A.(zw)[X]$, provided that X is not free in A;

9. $\forall X.A \times B = \forall X.A \times \forall X.B$

   is proved by λx:$(\forall X.A \times B).\langle \lambda X.(\mathrm{p}_1(x[X])), \lambda X.(\mathrm{p}_2(x[X]))\rangle$,

   with inverse λy:$(\forall X.A \times \forall X.B).\lambda X.\langle (\mathrm{p}_1 y)[X], (\mathrm{p}_2 y)[X]\rangle$;

10. $A \times \mathbf{T} = \mathrm{A}$ is proved by λx.p_1x with inverse λx:A.$\langle x, * \rangle$ (to check invertibility, notice that in λw:$A \times \mathbf{T}.\langle \mathrm{p}_1 w, * \rangle$ we have $* = \mathrm{p}_2$w by equality (*top*));

11. $A \to \mathbf{T} = \mathbf{T}$

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

    with inverse $rep(\mathbf{T} \to (A \to \mathbf{T})) = \lambda$y:$\mathbf{T}.\lambda$x:A.$*$;

12. $\mathbf{T} \to A = \mathrm{A}$ is proved by λx:$(\mathbf{T} \to A)$.x$*$ with inverse λy:A.λw:$\mathbf{T}$.y;

13. $\forall X.\mathbf{T} = \mathbf{T}$ is proved by λx:$\forall X.\mathbf{T}.*$ with inverse λy:$\mathbf{T}.\lambda X.*$.

    Here for the difficult side of the equality, notice that if M:$\forall X.\mathbf{T}$ then $M[X] = *$, by equality (*top*), so that, for X a fresh type variable, $\lambda X.M[X] = \lambda X.*$ by rule ξ and finally by η we get $\mathrm{M} = \lambda X.M[X] = \lambda X.*$. Hence

$$\begin{aligned}
(\lambda y : \mathbf{T}.\lambda X.*) \circ (\lambda x : \forall X.\mathbf{T}.*) &= \lambda w : \forall X.\mathbf{T}.(\lambda y : \mathbf{T}.\lambda X.*)((\lambda x : \forall X.\mathbf{T}.*)w) \\
&= \lambda w : \forall X.\mathbf{T}.(\lambda y : \mathbf{T}.\lambda X.*)* \\
&= \lambda w : \forall X.\mathbf{T}.\lambda X.* \\
&= \lambda w : \forall X.\mathbf{T}.w \\
&= \mathrm{I}_{\forall X.\mathbf{T}}
\end{aligned}$$

□

2.2 Completeness

There are essentially two classes of methods that can be used to provide a proof of completeness: semantic proofs or syntactic proofs.

Semantic proofs

Historically, the semantic method was the first to be used, in the case of $Th^1_{\times T}$, in [Sol83]. In this work, the focus was on the Cartesian Closed Categories, and the proof is given by providing a specific CCC where only the isomorphisms of $Th^1_{\times T}$ hold. Clearly then, since the isomorphisms of $Th^1_{\times T}$ hold in every CCC, and there is one CCC where no other isomorphism holds, completeness follows.

The specific CCC used in [Sol83] is the category of *finite sets*, that corresponds to the natural numbers equipped with product and exponentiation. Isomorphisms of two types in this category corresponds to equality of the numeric expressions obtained by reading $a \times b$ as the multiplication ab, $a \to b$ as exponentiation b^a and $\mathbf{T}$ as the number 1. For example, the axiom $a \times b \to c = a \to b \to b$ of $Th^1_{\times T}$ is the valid numeric equation $c^{ab} = c^{b^a}$.

For such structure, there had been some interest before, as it was involved in a number theoretic problem known as *Tarski's High School Algebra Problem*: the axioms of $Th^1_{\times T}$ are all valid arithmetic equations, and correspond to the usual equalities we are taught in high school about product and exponentiation; Tarski asked if these equations (and their equational

consequences) are also the only valid ones [DT69, Hen77]. Martin, one of Tarski's students, provided a positive answer for the exponential and the multiplicative exponential fragments without the constant 1 in [Mar72], by means of number theoretic arguments that are essentially the same used later in [Sol83].

The story of this number theoretic problem and of some of its variations is very interesting and active still today (see for example [Gur90, Gur85, Mac81, HR84]), but the connection with the problem of isomorphisms seems to be useful only in the specific case of $Th^1_{\times T}$: up to now there is no way (to the author's best knowledge) to extend this connection to second order calculi, as quantification over types does not seem to find a useful correspondent on the arithmetic side.

Even dropping the analogy with number theory, it seems hard to find for the second order case a model with the same properties as the category of finite sets. We need a non degenerate model of the calculus, and we already know that it cannot be a simple set theoretic model [Rey84]. Worse, the familiar models (like PER) validate isomorphisms like $\forall X.A = A$, that does not hold in the term model[2].

Syntactic proofs

In [BL85] we find both the first syntactic approach to the proof of completeness of theories of isomorphisms, and the first result concerning second order calculi, as the paper deals with Th^1 and Th^2. The proof techniques used here are all borrowed from the λ-calculus tradition, as the motivation comes from λ-calculus, too. Instead of looking for a specific model satisfying just Th^1, Bruce and Longo use a syntactic characterization of the invertible terms of $\lambda^1\beta\eta$, which is an easy consequence of a classical result by Dezani for untyped λ-calculus.

Theorem 2.3 (Invertible terms of λ-calculus)
Let M be an untyped term possessing normal form. Then M is invertible iff M is finite hereditary permutation (f.h.p.).

Finite hereditary permutations are defined inductively as follows.

Definition 2.4 (Finite Hereditary Permutations, f.h.p.)
An untyped λ term M is a finite hereditary permutation iff

- $M = \lambda x.x$, *or*
- $M = \lambda z.\lambda v_1 \dots v_n.zP_1 \dots P_n$ *and there exists a permutation $\sigma{:}n \to n$, such that $\lambda x_i.P_{\sigma(i)}$ is a finite hereditary permutation.*

These terms are all typable and hence are exactly the invertible terms of $\lambda^1\beta\eta$. Given this simple syntactic characterization, it is possible to proceed inductively on the structure of the terms to prove completeness of the axiom **swap** for $\cong_d$ in the calculus $\lambda^1\beta\eta$(hence for $\cong$, due to Theorem 1.3), as is done in [BL85].

This technique, unlike the semantic one, extends smoothly to the second order case, the only real difficulty being the characterization of invertible terms. In the case of $\lambda^2\beta\eta$, this is done easily in [BL85], always using Dezani's result, and one gets that the invertible terms are the 2-f.h.p.'s defined as follows.

Definition 2.5 (Second order Finite Hereditary Permutations, 2-f.h.p.)
A second order term M of $\lambda^2\beta\eta$ is a second order finite hereditary permutation (2-f.h.p.) iff

- $M = \lambda x.x$, *or*
- $M = \lambda z.\lambda v_1 \dots v_n.zP_1 \dots P_n$ *and there exists a permutation $\sigma{:}n \to n$, such that*

[2]The natural candidates to prove this isomorphism, namely λz:$\forall X.A.$z[X] and λz:A.$\lambda X.z$, when composed reduce to λx:$\forall X.A.\lambda X.x[Y]$, and not to the identity.

if $\lambda v_i = \lambda x_i : C$ *then* $\lambda x_i : C.P_{\sigma(i)}$ *is a 2-f.h.p.*
if $\lambda v_i = \lambda X_i$ *then* $P_{\sigma(i)}$ *is* X_i.

Theorem 2.6 . *2-f.h.p.'s are all and the only invertible terms of* $\lambda^2\beta\eta$.

Proof. By interpretation in the untyped calculus. See [BL85], Lemma 2.4 and Theorem 2.5. □

This can be used to show completeness of Th^2 for isomorphisms of $\lambda^2\beta\eta$.

In the case of the calculi with constants, like $\lambda^1\beta\eta\pi*$ or $\lambda^2\beta\eta\pi*$, though, such a simple syntactic characterization is not already available: actually in [BDCL92] only a subset of invertible terms is characterized, that is sufficient for the purposes of the completeness proof for $Th^1_{\times T}$. The guideline for the proof is to try to deal with the complexities which arise from the different term constructors one at a time, in order to achieve a sort of factorization of the invertibility problem for the full calculus into the invertibility problem for a more manageable subclass of terms. The theory suggests that the type **T** is redundant and that the products in a type can be always pulled out of the other type constructors, while still remaining in the class of isomorphic types. So the completeness proof is not as direct as it was in [BL85]: it needs some intermediate steps, very similar to those we will see here, but still stays rather simple, as Dezani's Theorem can handle a relevant part of the complexity of the proof.

The second order case with constants of $Th^2_{\times T}$ treated in this paper is significantly more complex both than the pure second order case and than the first order case with constants: here too we can give a characterization[3] for a subclass of the invertible terms that is suitable for the purpose of proving the completeness of $Th^2_{\times T}$, but the combination of the two extensions practically forces us to rebuild almost all the complex combinatorial proof of the original Theorem by Dezani, which can no longer be used simply as a tool out of the box.

The rest of this paper is dedicated to the proof of the other implication of the Main Theorem, that is to say the *completeness* of $Th^2_{\times T}$, and to the decidability of $Th^2_{\times T}$.

3 Towards Completeness

As we have seen in the Survey, the only proof technique that we can hope to apply to the case of $\lambda^2\beta\eta\pi*$ is the one based on the syntactic characterization of invertible terms. Unfortunately, no such characterization is known for $\lambda^2\beta\eta\pi*$, so we need to reduce our original problem to a simpler one. In this section we will show that we can actually restrict our attention to isomorphisms of a special form, and to a particular class of invertible terms, for which we will later be able to provide a syntactic characterization.

3.1 Outline of the Section

- **Reduction to a subclass of types.** We identify two relevant classes of types: types not containing products or **T**, that we call *simple types* and products of simple types, that we call *stratified types*. We show that $Th^2_{\times T}$ is complete for isomorphisms of types if and only if it is complete for isomorphisms between *stratified types*.
- **Reduction to a subclass of terms.** We show that any isomorphism between *stratified* types can be proved by invertible terms whose free variables have *simple* types (we call them *canonical* invertible terms).
- **Overall achievement of this section:** we reduce the problem of completeness of $Th^2_{\times T}$ to the problem of completeness for isomorphisms between *stratified* types proved by *canonical* invertible terms, and for these we will be able to provide a syntactic characterization.

[3]For a generic invertible term we can give a procedure to verify if it is invertible and in that case we also know how to build its inverse, but we miss an explicit syntactic characterization.

3.2 Reduction to a subclass of types

We introduce here a type rewriting system, suggested by the form of the axioms of $Th^2_{\times T}$, and the corresponding type normal form. We will then show that two types are isomorphic if and only if their normal forms are. It is to be noticed that the normal form is essentially the usual conjunctive normal form for (second order) propositional calculus. The axioms of $Th^2_{\times T}$ suggest the following rewrite system $\mathcal{R}$ for types (essentially $Th^2_{\times T}$ without commutativity):

Definition 3.1 *(Type rewriting $\mathcal{R}$) Let ">" be the transitive and substitutive type-reduction relation generated by:*

1.	$A \times (B \times C)$	$>$	$(A \times B) \times C$	5.	$A \times \mathbf{T}$	$>$	A
2.	$(A \times B) \to C$	$>$	$A \to (B \to C)$	6.	$\mathbf{T} \times A$	$>$	A
3.	$A \to (B \times C)$	$>$	$(A \to B) \times (A \to C)$	7.	$A \to \mathbf{T}$	$>$	$\mathbf{T}$
4.	$\forall X.A \times B$	$>$	$\forall X.A \times \forall X.B$	8.	$\mathbf{T} \to A$	$>$	A
				9.	$\forall X.\mathbf{T}$	$>$	$\mathbf{T}$.

The system $\mathcal{R}$ yields an obvious notion of normal form for types (type-n.f.), i.e. when no type reduction is applicable. Note that 5, 6, 8, *eliminate the* $\mathbf{T}$*'s*, while 3 and 4 *bring the* $\times$ *outside*. It is then easy to observe that each type-n.f. is $\mathbf{T}$ or has the structure $S_1 \times \ldots \times S_n$ 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 3.2), and say that a normal form is non-trivial if it is not $\mathbf{T}$.

Proposition 3.2 *Each type has a unique type normal form in $\mathcal{R}$.*

Proof. Using the REVE system [Les83, Les86], this is straightforward, but in Appendix D we provide also a direct proof. □

Types in normal form have a very simple shape, that can be described as follows:

Definition 3.3 (simple types, stratified types) *A type A is* **simple** *when there is no occurrence of products or* $\mathbf{T}$*'s in it. A type is* **stratified** *when it is either* $\mathbf{T}$ *or a finite product $A_1 \times \ldots \times A_n$, where the A_i are simple types.*

Remark 3.4 *A type normal form is* stratified. *Furthermore, when $A \cong_d B$, either nf(A) and nf(B) are both* $\mathbf{T}$*, or they are both not* $\mathbf{T}$.

Indeed, a non trivial type-n.f. cannot be isomorphic to $\mathbf{T}$, as is easily seen by taking a non-trivial model, so the case $(A_1 \times \ldots \times A_n) \cong_d \mathbf{T}$ is not possible. Anyway, since all the work done in this section is purely syntactic, we give also an easy syntactic proof of this fact in Proposition D.1 in Appendix D. Now, $\mathcal{R} \vdash$A $>$ B implies $Th^2_{\times T} \vdash$A = B, and using the soundness of $Th^2_{\times T}$ proved in 2.2, we get that any reduction $\mathcal{R} \vdash$ A $>$ B is witnessed (or *proved*, in the *types-as-propositions analogy*) by an invertible term M:$A \to B$. Moreover, one clearly has:

Corollary 3.5 *$Th^2_{\times T} \vdash A = nf(A)$ and, thus, $Th^2_{\times T} \vdash A = B \iff Th^2_{\times T} \vdash nf(A) = nf(B)$*

The same holds for $\cong_d$:

Proposition 3.6 *$A \cong_d B \iff nf(A) \cong_d nf(B)$*

Proof. Recall that A $\cong_d$ B iff there is an invertible term M :$A \to B$, and nf(A) $\cong_d$ nf(B) iff there is an invertible term M' : $nf(A) \to nf(B)$. So it suffices to show that it is possible to turn invertible terms of type $A \to B$ into invertible terms of type $nf(A) \to nf(B)$, and vice-versa. 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 M:$A \to B$ is invertible $\iff$ there exists an invertible term M':$nf(A) \to nf(B)$, such that M = G^{-1}∘M'∘F.

($\Leftarrow$) 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∘M∘F^{-1}. Then $M'^{-1} \equiv$ F∘M^{-1}∘G^{-1} and M' is invertible. □

The diagram in Figure 1 shows what's going on in the Proposition.

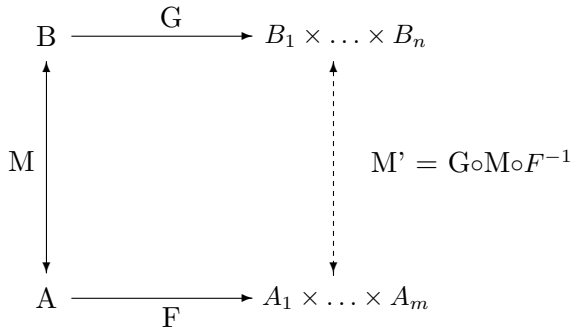

Figure 1: Reduction to a subclass of isomorphic types.

Hence we get the main result in this subsection, that allows us to restrict the analysis to isomorphisms between non-trivial stratified types.

Proposition 3.7 *$Th^2_{\times T}$ is complete for $\cong_d$ iff $Th^2_{\times T}$ is complete for $\cong_d$ restricted to stratified types different from (and hence not containing)* **T**.

Proof. ($\Leftarrow$) If A $\cong_d$ B, then by 3.4 either nf(A) $\equiv$ **T** $\equiv$ nf(B), or nf(A) $\equiv (A_1 \times \ldots \times A_n)$ $\cong_d (B_1 \times \ldots \times B_m) \equiv$ nf(B), where no occurrence of **T** can appear in either type. In both cases the result follows by

$$Th^2_{\times T} \vdash A = B \Longleftrightarrow Th^2_{\times T} \vdash nf(A) = nf(B) \Longleftrightarrow nf(A) \cong_d nf(B) \Longleftrightarrow A \cong_d B$$

The first and third equivalence are just Corollary 3.5 and Proposition 3.6.

The second equivalence is trivially satisfied in the case nf(A) $\equiv$ **T** $\equiv$ nf(B), while is the hypothesis of completeness for stratified types different from **T** otherwise.

($\Rightarrow$) By definition, as stratified types are types. □

Thus we have shown that the characterization of definable isomorphisms between arbitrary types can be reduced to the characterization of definable isomorphisms in the class of **stratified** types different from **T**.

3.3 Reduction to a subclass of terms

There is another simplification we can perform, though. Since equality on lambda terms is substitutive, it is easy to show that we can consider only invertible terms with free variables of **simple** types.

Proposition 3.8 *Let M:$A \to B$ be an invertible term with inverse N:$B \to A$. Then there exist invertible terms M':$A \to B$ and N':$B \to A$ whose free variables have simple types.*

Proof. For every free variable x:A it is easy to build a term t_x:A whose free variables have simple types. If A is simple, then t_x=x. Otherwise, consider the type normal form nf(A) of A, and let G_A:$nf(A) \to A$ be a (closed) invertible term proving the isomorphism A $\cong_d$ nf(A) (as in 3.6). If nf(A) $= A_1 \times \ldots \times A_n$, with $n > 1$, then we can choose fresh variables z_1:A_1, … z_n:A_n and put $t_x = G_A\langle z_1, , \ldots\ z_n\rangle$:A, since the A_i are simple and G_A is closed. If nf(A) $= A_1$ $\neq$ **T** we can choose a fresh variable z_1:A_1 and put $t_x = (G_A\ z_1)$, since A_1 is simple and G_A is closed. Otherwise, nf(A) = **T** and we can put $t_x = (G_A *)$, that is closed. Now we can show that M$[\overrightarrow{t_x}/\overrightarrow{x}]$ and N$[\overrightarrow{t_x}/\overrightarrow{x}]$, where $\overrightarrow{x}$ are all the free variables in M and N, are the required invertible terms M' and N'. Indeed, the free variables in M$[\overrightarrow{t_x}/\overrightarrow{x}]$ and N$[\overrightarrow{t_x}/\overrightarrow{x}]$ have simple

types as they are included in the free variables of the t_x's. Furthermore, by substitutivity of the equality, $N \circ M = I_A$ implies $(N \circ M)[\overrightarrow{t_x}/\overrightarrow{x}] = I_A[\overrightarrow{t_x}/\overrightarrow{x}]$, that is $(N[\overrightarrow{t_x}/\overrightarrow{x}] \circ M[\overrightarrow{t_x}/\overrightarrow{x}]) = I_A$. Similarly, $(M[\overrightarrow{t_x}/\overrightarrow{x}] \circ N[\overrightarrow{t_x}/\overrightarrow{x}]) = I_B$. $\square$

Proposition 3.7 and 3.8 allow, without loss of generality, the restriction of the analysis to invertible terms between stratified types (different from $\mathbf{T}$). By proposition 3.8, we can assume that the free variables in these invertible terms have simple types only. We will call these terms **canonical bijections**, or simply **canonical** terms.

Definition 3.9 (canonical bijections) *A term is a* **canonical bijection** *if it is an invertible term mapping stratified types (different from* $\mathbf{T}$*) into stratified types (different from* $\mathbf{T}$*) and if all its free variables have simple types.*

The next step will be to find a syntactic characterization for them.

4 Characterizing canonical terms: from $\lambda^2\beta\eta\pi*$ to $\lambda^2\beta\eta$

In this section we follow a very natural intuition: if two stratified types $A_1 \times \ldots \times A_m$ and $B_1 \times \ldots \times B_n$ are isomorphic, we expect them to have the same number of components (i.e. $n = m$). Furthermore, we expect that such an isomorphism can be decomposed into an n-tuple of independent, simpler isomorphisms between the different components of the stratified types. These componentwise isomorphisms should not involve products or $\mathbf{T}$, and be invertible terms of $\lambda^2\beta\eta$, rather then of $\lambda^2\beta\eta\pi*$. Essentially, this allows us to express canonical bijections of $\lambda^2\beta\eta\pi*$ in terms of invertible terms of $\lambda^2\beta\eta$, for which a syntactic characterization is known from [BL85].

4.1 Outline of the Section

We define a notion of **coordinates** for a canonical bijection. Such coordinates will be the body of the seeked componentwise isomorphisms, and will allow us to give a syntactic characterization of canonical bijections.

- **Projection of invertibility over coordinates.** We show that the invertibility property of a canonical bijection determines a similar property, that we call **distributed invertibility**, for its set of coordinates.
- **Reduction of coordinates to $\lambda^2\beta\eta$.** Using in an essential way this property of cordinates of a canonical bijection, we can show with a difficult syntactic proof that such coordinates are already terms of $\lambda^2\beta\eta$. This proof is intimately related to the method employed in the original characterization of invertible terms in pure lambda calculus [Dez76].
- **Syntactic Characterization of canonical bijections.** Once we know that coordinates live in $\lambda^2\beta\eta$, we can use the syntactic characterization of invertible terms in $\lambda^2\beta\eta$ provided in [BL85] to obtain a simple syntactic characterization of canonical bijections.
- **Overall achievement of this section:** We explicitly describe the syntactic shape of a canonical bijection: this result will allow to show completeness of $Th^2_{\times T}$ by structural induction on the canonical bijexctions, and also provides us with a decidable test of invertibility for all terms in $\lambda^2\beta\eta\pi*$.

4.2 Projection of invertibility over coordinates

A canonical bijection maps a finite product $A_1 \times \ldots \times A_m$ of non-product types into another finite product $B_1 \times \ldots \times B_n$ of non-product types. It is natural to study the behaviour of the bijection on each one of the components B_i of the target type separately. To do so, we introduce a notion of coordinate as follows:

Definition 4.1 (coordinates) *For a canonical bijection M:*$A_1 \times \ldots \times A_m \to B_1 \times \ldots \times B_n$*, where* $A_1 \times \ldots \times A_m$ *and* $B_1 \times \ldots \times B_n$ *are stratified types, define the collection of its* coordinates *as the sequence* $\overrightarrow{M} = [M_1, \ldots, M_n]$*, where* M_i *is n.f.*$(p_i(M\langle x_1, \ldots, x_m\rangle))$ *and* $x_1, \ldots, x_m$ *are fresh variables.*

Remark 4.2 *The type of a coordinate and of the free variables of a coordinate do not contain products or* **T** *types.*

A canonical bijection can be represented by its coordinates.

Proposition 4.3 *Every canonical bijection M:*$A_1 \times \ldots \times A_m \to B_1 \times \ldots \times B_n$ *can be written in terms of its coordinates* M_i*'s as* $\lambda z.(\lambda x_1 \ldots x_m.(\langle M_1, \ldots, M_n\rangle))(p_1 z) \ldots (p_m z)$.

Proof. Due to surjective pairing, we have

$$\begin{array}{lll} M\langle x_1, \ldots, x_m\rangle & =_{\beta^2\eta^2\pi *} & \langle \mathrm{p}_1(M\langle x_1, \ldots, x_m\rangle), \ldots, \mathrm{p}_n(M\langle x_1, \ldots, x_m\rangle)\rangle \\ & =_{\beta^2\eta^2\pi *} & \langle n.f.(\mathrm{p}_1(M\langle x_1, \ldots, x_m\rangle)), \ldots, n.f.(\mathrm{p}_n(M\langle x_1, \ldots, x_m\rangle))\rangle \\ & =_{\beta^2\eta^2\pi *} & \langle M_1, \ldots, M_n\rangle \end{array}$$

Notice that, by standard currying, $M =_{\beta^2\eta^2\pi *} \lambda z.(\lambda x_1 \ldots x_m.(M\langle x_1, \ldots, x_m\rangle))(\mathrm{p}_1 z) \ldots (\mathrm{p}_m z)$, so we finally get $M = \lambda z.(\lambda x_1 \ldots x_m.(\langle M_1, \ldots, M_n\rangle))(\mathrm{p}_1 z) \ldots (\mathrm{p}_m z)$. □

The property of invertibility of the original bijection is reflected in a similar property for the collection of its coordinates (point 1 of the following Proposition).

Proposition 4.4 (Properties of Coordinates)

Let M:$A_1 \times \ldots \times A_m \to B_1 \times \ldots \times B_n$ and N:$B_1 \times \ldots \times B_n \to A_1 \times \ldots \times A_m$ be canonical bijections in $\lambda^2\beta\eta\pi *$. Then there exist coordinates $\overrightarrow{N} = [N_1, \ldots, N_m]$ and $\overrightarrow{M} = [M_1, \ldots, M_n]$ with $\overrightarrow{x} = \{x_1 : A_1, \ldots, x_m : A_m\} \subseteq FV(\overrightarrow{M})$ and $\overrightarrow{y} = \{y_1 : B_1, \ldots, y_n : B_n\} \subseteq FV(\overrightarrow{N})$ s.t. no x_i is free in any N_j, no y_i is free in any M_j and

1. $M_i[\overrightarrow{N}/\overrightarrow{x}] =_{\beta^2\eta^2\pi *} y_i$ and $N_i[\overrightarrow{M}/\overrightarrow{y}] =_{\beta^2\eta^2\pi *} x_j$

2. $FV(\overrightarrow{M}) \vdash M_i : B_i$ and $FV(\overrightarrow{N}) \vdash N_j : A_j$

3. no type expression occurring in $FV(\overrightarrow{M}) \cup FV(\overrightarrow{N})$ contains occurrences of $\times$ or **T**.

Proof. The condition about the variables ($\overrightarrow{x}$ and $\overrightarrow{y}$) is trivially satisfied by a suitable choice of the new x_i's and y_i's in the coordinates.

(1) By the previous Proposition 4.3 we have

$$M =_{\beta^2\eta^2\pi *} \lambda z.(\lambda x_1 \ldots x_m.M')(\mathrm{p}_1 z) \ldots (\mathrm{p}_m z), \qquad N =_{\beta^2\eta^2\pi *} \lambda z.(\lambda y_1 \ldots y_n.N')(\mathrm{p}_1 z) \ldots (\mathrm{p}_n z)$$

where $M' : B_1 \times \ldots \times B_n =_{\beta^2\eta^2\pi*} \langle M_1, \ldots, M_n\rangle$ and $N' : A_1 \times \ldots \times A_m =_{\beta^2\eta^2\pi*} \langle N_1, \ldots, N_m\rangle$, so we have the following equalities:

$$\begin{array}{rll}
M\circ N & =_{\beta^2\eta^2\pi*} & \lambda w.(M(Nw)) \\
& =_{\beta^2\eta^2\pi*} & \lambda w.(\lambda z.(\lambda x_1 \ldots x_m.M')(\mathrm{p}_1 z)\ldots(\mathrm{p}_m z))(Nw) \\
& =_{\beta^2\eta^2\pi*} & \lambda w.(\lambda x_1 \ldots x_m.M')(\mathrm{p}_1(Nw))\ldots(\mathrm{p}_m(Nw)) \\
& =_{\beta^2\eta^2\pi*} & \lambda w.M'[\mathrm{p}_1(Nw)\ldots\mathrm{p}_m(Nw)/x_1\ldots x_m] \equiv \lambda w.M'[\overrightarrow{\mathrm{p}_i(Nw)}/\overrightarrow{x}] \\
& =_{\beta^2\eta^2\pi*} & \lambda w.\langle \mathrm{p}_1 M', \ldots, \mathrm{p}_n M'\rangle[\overrightarrow{\mathrm{p}_i(Nw)}/\overrightarrow{x}] \\
& =_{\beta^2\eta^2\pi*} & \lambda w.\langle M_1[\overrightarrow{\mathrm{p}_i(Nw)}/\overrightarrow{x}], \ldots, M_n[\overrightarrow{\mathrm{p}_i(Nw)}/\overrightarrow{x}]\rangle \text{ as } M_i \equiv n.f.(\mathrm{p}_i M') \\
& =_{\beta^2\eta^2\pi*} & \lambda w.\langle M_1[\overrightarrow{(N_i[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}])}/\overrightarrow{x}], \ldots, M_n[\overrightarrow{(N_i[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}])}/\overrightarrow{x}]\rangle
\end{array}$$

$$\begin{array}{rll}
\text{as } Nw & =_{\beta^2\eta^2\pi*} & (\lambda z.(\lambda y_1 \ldots y_n.N')(\mathrm{p}_1 z)\ldots(\mathrm{p}_n z))w \\
& =_{\beta^2\eta^2\pi*} & (\lambda y_1 \ldots y_n.N')(\mathrm{p}_1 w)\ldots(\mathrm{p}_n w) \\
& =_{\beta^2\eta^2\pi*} & N'[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}] \\
& =_{\beta^2\eta^2\pi*} & \langle \mathrm{p}_1 N', \ldots, \mathrm{p}_m N'\rangle[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}] \\
& =_{\beta^2\eta^2\pi*} & \langle \mathrm{p}_1 N'[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}], \ldots, \mathrm{p}_m N'[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}]\rangle \\
& =_{\beta^2\eta^2\pi*} & \langle N_1[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}], \ldots, N_m[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}]\rangle
\end{array}$$

Where for substitutions like $[\mathrm{p}_1(Nw), \ldots, \mathrm{p}_m(Nw)/x_1, \ldots, x_m]$ we have used the compact notation $[\overrightarrow{\mathrm{p}_i(Nw)}/\overrightarrow{x}]$.

But we also have that

$$\begin{array}{rll}
M\circ N & =_{\beta^2\eta^2\pi*} & \lambda w.w \text{ (by hypothesis)} \\
& =_{\beta^2\eta^2\pi*} & \lambda w.\langle \mathrm{p}_1 w, \ldots, \mathrm{p}_n w\rangle \text{ (by Surjective Pairing)}
\end{array}$$

By transitivity, then

$$\lambda w.\langle \mathrm{p}_1 w, \ldots, \mathrm{p}_n w\rangle =_{\beta^2\eta^2\pi*} \lambda w.\langle N_1[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}], \ldots, N_m[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}]\rangle$$

From this equality it is easy to derive (by applications and projections) that $M_k[\overrightarrow{(N_i[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}])}/\overrightarrow{x}] =_{\beta^2\eta^2\pi*} \mathrm{p}_k w$ and then the Church-Rosser property of the rewrite system for our calculus gives us

$$M_k[\overrightarrow{(N_i[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}])}/\overrightarrow{x}] \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{p}_k w$$

By substitution properties

$$M_k[\overrightarrow{(N_i[\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}])}/\overrightarrow{x}] \equiv M_k[\overrightarrow{N}/\overrightarrow{x}][\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}],$$

as no y_j is free in M, so

$$M_k[\overrightarrow{N}/\overrightarrow{x}][\overrightarrow{\mathrm{p}_j w}/\overrightarrow{y}] \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{p}_k w$$

Now, by Theorem B.4 applied to M_k and w, we can deduce that $M_k[\overrightarrow{N}/\overrightarrow{x}] \overset{\beta^2\eta^2\pi*}{\longrightarrow} y_k$, hence $M_k[\overrightarrow{N}/\overrightarrow{x}] = y_k$, as required.

Analogously for $N\circ M$.

(2) Follows from the construction of the M_i and N_j.

(3) Follows from the definition of stratified types and the hypothesis on the free variables of M and N.

□

So, to every bijection we can associate a set of coordinates that not only represents them in the sense of Proposition 4.3, but also enjoys a kind of **distributed invertibility** property (the one expressed by 2 in Proposition 4.4 above). This property, together with the other side conditions in the statement of Proposition 4.4 above, will be used in a crucial way to prove that coordinates are terms of $\lambda^2\beta\eta$. For this reason, we extend it to sequences containing not only terms, but also types, and we give a name to it.

Definition 4.5 (Distributed Invertibility)

Given $\overrightarrow{N} = [N_1, \ldots, N_m]$ and $\overrightarrow{M} = [M_1, \ldots, M_n]$ sequences of second order λ-terms in normal form *and/or second order type variables*, choose $\overrightarrow{x} = V_1, \ldots, V_m \subseteq FV(\overrightarrow{M})$ and $\overrightarrow{y} = W_1, \ldots, W_n \subseteq FV(\overrightarrow{N})$ sequences of variables, that can be either type variables X_i which are part of the list $\overrightarrow{M} = [M_1, \ldots, M_n]$ (resp. type variables Y_j from $\overrightarrow{N} = [N_1, \ldots, N_m]$), or term variables $x_i : A_i$ (resp. $y_i : B_i$). We require that $\overrightarrow{x} \cap FV(\overrightarrow{N}) = \emptyset$ and $\overrightarrow{y} \cap FV(\overrightarrow{M}) = \emptyset$.

We will write $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y})$ if the following conditions hold:

1. no type expression occurring in $FV(\overrightarrow{M}) \cup FV(\overrightarrow{N})$ contains occurrences of $\times$ or $\mathbf{T}$,
2. $FV(\overrightarrow{M}) \vdash M_i : B_i$ if M_i is a term and $FV(\overrightarrow{N}) \vdash N_j : A_j$ if N_j is a term ,
3. $M_i[\overrightarrow{N}/\overrightarrow{x}] = W_i$ and $N_j[\overrightarrow{M}/\overrightarrow{y}] = V_j$.

Remark 4.6 *Notice that this last equality is on terms or types, depending on the nature of the objects that are equated. In particular, this implies that M_i and W_i (resp. N_j and V_j) are either both types or both terms. Furthermore, the M_i (resp. N_j) that are types must be simple type variables, as otherwise they could not be equal to the type variables W_i (resp. V_j), no matter the substitution we apply to them.*

Intermezzo

The coordinates of a canonical bijection M and those of a canonical bijection N that is its inverse are in the same number (i.e. $m = n$) and they are actually terms of the pure calculus $\lambda^2\beta\eta$, as we will show in the next Subsection. The proof of such a fact, though, is far from being a simple one. To understand better the reason for this complexity, and the technique we will use to overcome it, it is convenient here to recall the proof techniques used in previous works. In [BDCL92], exactly the same proof strategy is used up to this point, but for the limited case of the calculus $\lambda^1\beta\eta\pi*$, that has no second order features. There Proposition 3.7 has the same flavour as Proposition 4.4 here, but there it is also possible to show, by induction, that the coordinates are terms of the simple typed lambda calculus. The proof uses in an essential way the fact that the types of a term of $\lambda^1\beta\eta\pi*$ and its free variables carry enough information to exclude the presence of products or terminal constants in the coordinates. This is due to the following relevant facts.

- *Every term in n.f.of $\lambda^1\beta\eta\pi*$, whose type contains no occurrence of $\mathbf{T}$, has no occurrence of *A constants*

  Proof. this is lemma 3.2 in [BDCL92] □

- *Terms in n.f. of $\lambda^1\beta\eta\pi$, whose type is arrow-only, belong to $\lambda^1\beta\eta$*

  Proof. this is lemma 3.6 in [BDCL92] □

Once this reduction from $\lambda^1\beta\eta\pi*$ to $\lambda^1\beta\eta$ is done, it is then possible to prove easily that the coordinates of a canonical bijection M and those of a canonical bijection N that is its inverse are in the same number (i.e. $m = n$), as it is done there in Lemma 3.8.

Unfortunately, these statements do not hold any longer when we consider second order terms, even in normal form, as the following example shows.

Example 4.7 Let A and B be *simple* types and consider the term

$$x : \forall X.X \to B, y : A \vdash x[A \times \mathbf{T}]\langle y, * \rangle : B$$

The term $x[A \times \mathbf{T}]\langle y, * \rangle$ is in normal form and no product or $\mathbf{T}$ appear in its type or the type of any of its variables, but it contains an occurrence of $\mathbf{T}$, a product type, $*$ and a pair as subterms. □

Actually, lemmas 3.2 and 3.6 in [BDCL92] do hold for those second order terms that are coordinates of invertible terms, but we can no longer provide two separate proofs, one for reducing to the pure calculus $\lambda^2\beta\eta$, and one to show that $m = n$. We must show these properties at the same time, in a complex inductive proof that needs also several additional invariants, including the property *DistInv*.

4.3 Reduction of coordinates to $\lambda^2\beta\eta$

To show that the coordinates of an invertible term of $\lambda^2\beta\eta\pi*$ are indeed terms of the simpler calculus $\lambda^2\beta\eta$, we will need to prove a Main Lemma that is comparable for its complexity to the original characterization of invertible terms of the pure λ-calculus provided by Dezani [Bar84, Dez76]. Actually, unless the reader is familiar with the proof technique by Dezani, even the statement of the Lemma can be unreadable without a proper explanation. For this reason, in this section we will focus on showing how the statement of such a complex lemma arises, rather than on its proof, rather technical, that is deferred to Appendix C.

4.3.1 Towards a better Lemma

As we have seen in the Intermezzo, for the calculus $\lambda^2\beta\eta\pi*$ we cannot prove independently that the number of coordinates of an invertible term M is the same as the number of coordinates of any inverse N, *and* that such coordinates are terms of the simpler calculus $\lambda^2\beta\eta$. Actually, it turns out that this last property needs a strengthened version of the first one.

To start, let's recall how in Lemma 3.8 of [BDCL92] it is proved that the length of the coordinates of two invertible terms M and N of $\lambda^1\beta\eta\pi*$ that are each other's inverses is the same (we rephrase the statement using the terminology of this paper).

Lemma 3.8 of [BDCL92] *Let $A_1 \times \ldots \times A_n$ and $B_1 \times \ldots \times B_m$ be type normal forms and $M_1 \ldots M_n$, $N_1 \ldots N_n$ be coordinates of invertible terms M and N of $\lambda^1\beta\eta\pi*$. Then*

- *$n = m$*
- *there exist permutations σ, π over n (and terms P_i, Q_j) such that*

$$M_i = \lambda\overrightarrow{u_i}.x_{\sigma_i}\overrightarrow{P}_i \qquad \text{and} \qquad N_j = \lambda\overrightarrow{v_j}.y_{\pi_j}\overrightarrow{Q}_j$$

(where the x_{σ_i} and y_{π_i} are free in M_i and N_j respectively).

Proof. First, one already knows that the coordinates M_i and N_j, which are terms in normal form, are terms of the simpler calculus $\lambda^1\beta\eta$, so that

$$M_i = \lambda\overrightarrow{u}_i.s_i\overrightarrow{P}_i \qquad \text{and} \qquad N_j = \lambda\overrightarrow{v}_j.t_j\overrightarrow{Q}_j$$

Then it is easy to notice that s_i is a free variable (namely some x_j), since $M_i[\overrightarrow{N}/\overrightarrow{x}] =_{\beta\eta} y_i$, a property of coordinates which cannot hold if s_i is bound. Similarly t_j is some y_i.

This provides us with two functions $\sigma : n \to m$, π: $m \to n$ such that

$$M_i = \lambda \overrightarrow{u}_i.x_{\sigma(i)} \overrightarrow{P}_i \text{ for } 1 \leq i \leq n, \qquad N_j = \lambda \overrightarrow{v}_j.y_{\pi(j)} \overrightarrow{Q}_j \text{ for } 1 \leq i \leq m$$

In conclusion, for $1 \leq i \leq n$ we obtain[4]:

$$\begin{aligned}
y_i =_{\beta\eta} M_i[\overrightarrow{N}/\overrightarrow{x}] &=_{\beta\eta} (\lambda \overrightarrow{u_i}.x_{\sigma(i)} \overrightarrow{P}_i)[\overrightarrow{N}/\overrightarrow{x}] \\
&=_{\beta\eta} \lambda \overrightarrow{u_i}.N_{\sigma(i)}\{\overrightarrow{P}_i[\overrightarrow{N}/\overrightarrow{x}]\} \\
&=_{\beta\eta} \lambda \overrightarrow{u_i}.(\lambda \overrightarrow{v}_{\sigma(i)}.y_{\pi(\sigma(i))} \overrightarrow{Q}_{\sigma(i)})\{\overrightarrow{P}_i[\overrightarrow{N}/\overrightarrow{x}]\} \\
&=_{\beta\eta} \text{if } \overrightarrow{v}_{\sigma(i)} \text{ is longer than } \overrightarrow{P}_i \\
&\quad \text{then } \lambda \overrightarrow{u_i}.\overrightarrow{v'_{\sigma(i)}}.y_{\pi(\sigma(i))} \overrightarrow{Q}_{\sigma(i)}[(\overrightarrow{P}_i[\overrightarrow{N}/\overrightarrow{x}])/(\overrightarrow{v_{\sigma(i)}} - \overrightarrow{v'_{\sigma(i)}})] \\
&\quad \text{else } \lambda \overrightarrow{u_i}.y_{\pi(\sigma(i))}\{\overrightarrow{Q}_{\sigma(i)}[(\overrightarrow{P}_i[\overrightarrow{N}/\overrightarrow{x}])/\overrightarrow{v_{\sigma(i)}}]\}\{\overrightarrow{P'}_i[\overrightarrow{N}/\overrightarrow{x}]\}
\end{aligned}$$

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 reduces in the order to one of the bound variables, so that one can apply η, several times, at the end. The same holds for $N_j[\overrightarrow{M}/\overrightarrow{y}]$ for $1 \leq j \leq m$.

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

Now, back to the second order case, we can no longer assume that the coordinates are terms of the calculus $\lambda^2\beta\eta$. This means first of all that we must find a separate proof of the fact that the head-normal form of the M_i is $\lambda \overrightarrow{u_i}.x_{\sigma_i} \overrightarrow{P}_i$ (respectively, N_j is $\lambda \overrightarrow{v_j}.y_{\pi_j} \overrightarrow{Q}_j$), but this is not a serious problem: we can repair the proof and adapt it to the second order case.

What really complicates matters is the need to prove now that the coordinates are terms of $\lambda^2\beta\eta$. Such a proof will need to examine inductively the full structure of coordinates, not only their head variable, so we will have to turn this Lemma into a stronger one that provide us with an invariant to be used in this inductive proof. Let's see why.

Suppose we try to prove that the coordinates are terms of $\lambda^2\beta\eta$, by induction on their structure. The Lemma tells us that any coordinate M_i is $\lambda \overrightarrow{u_i}.x_{\sigma_i} \overrightarrow{P}_i$, so we know that, say, the "prefix" $\lambda \overrightarrow{u_i}.x_{\sigma_i}$ of M_i can already be seen as a term of $\lambda^2\beta\eta$: seems nice! Unfortunately, here we immediately face a difficulty in applying the inductive hypothesis: we would like to say that, by induction hypothesis, the $\overrightarrow{P}_i$ are already terms of $\lambda^2\beta\eta$, and then conclude, but we could do this only in the case that the $\overrightarrow{P}$ and $\overrightarrow{Q}$ enjoy the properties of coordinates, while the Lemmas stated above seems to tell us nothing at all about these terms. Apparently, we are stuck.

Looking more closely at the proof of the Lemma, though, we see that after all our original Lemma says something interesting, even if it says it not of the P_i's or Q_j's, but of the $P_i[\overrightarrow{N}/\overrightarrow{x}]$'s and the $Q_i[\overrightarrow{M}/\overrightarrow{y}]$'s: they seem to enjoy the properties of coordinates, with respect to the variables v_i's and u_i's. In fact, if

$$Q_i[(\overrightarrow{P}_i[\overrightarrow{N}/\overrightarrow{x}])/\overrightarrow{u_i}] = u_i,$$

[4]We maintain here the original notation from [BDCL92] for delimiting the scope of a substitution: for example, $(P\{M[N/x]\})$ will stand for the term (PM) where the substitution $[N/x]$ is applied only to the subterm M

as shown in the proof of the Lemma, then it is easy to see that also

$$Q_i[\overrightarrow{M}/\overrightarrow{y}][(\overrightarrow{P}_i[\overrightarrow{N}/\overrightarrow{x}])/\overrightarrow{u_i}] = u_i$$

So the right candidates to be considered for an inductive argument will not be just the subterms P_i of the coordinates M_i, but these subterms up to some substitution. We are then in the typical situation that needs an induction loading: to prove that the coordinates of an invertible term are terms of $\lambda^2\beta\eta$, we will prove something more, namely, that terms that enjoy the properties of coordinates after some substitution is applied to them are all in $\lambda^2\beta\eta$. As a special case, we will have our theorem considering empty substitutions.

This brings up two more problems that need to be solved in order to make our new proof work:

1. we must now modify the statement of the Lemma to deal not just with coordinates, but with coordinates to which some substitution is applied, in order for the inductive argument to apply to the case of $Q_i[\overrightarrow{M}/\overrightarrow{y}]$; furthermore, we must be able to deal with n-tuples containing also types, and not only terms, as the Q_i's are not all terms, but can be types also.

2. we must ensure that the number of the Q_i's and P_i's is the same, or, if it is not the case, find a way to extend the shorter one to the length of the longer, in a way as to turn them into coordinates enjoying the **distributed invertibility** property.

The first problem is easily solved by modifying the statement of the Lemma to handle not just coordinates (that enjoy the **distributed invertibility**) but coordinates to which some kind of substitution is applied. As for some of the Q_i's being types, our notion of **distributed invertibility** already takes care of them.

Definition 4.8 (head free terms and substitutions)
A term M is called (second order) head free *when it has a head normal form with free head variable, i.e. its head normal form is $\lambda\overrightarrow{v}.x\overrightarrow{P}$ with x a free term variable (the abstractions can be both term and type abstractions).*
A head free substitution *is a substitution that replaces variables with head free terms and possibly type variables with types. We will use $\bullet$, $\circ$, $\lhd$, $\rhd$... to range over head free substitutions.*
Moreover, if $\bullet$ and $\circ$ are substitutions, $\bullet\circ$ will stand for the usual composition of substitutions, that is done right to left, i.e. first apply $\bullet$ and then $\circ$.

Remark 4.9 *As suggested by the notation, the composition of two head free substitutions is still a head free substitution (see [Bar84], Lemma 21.2.3, pag. 535).*

Actually, by looking at how the Q_i's relate to $\overrightarrow{M}$, we see that the M's are *terms*, with a free variable in head position, so the substitutions we are interested in are actually head-free ones. Furthermore, the u_i's are distinct from the $\overrightarrow{x}$ and the M's do not contain free any variables from the N's or $\overrightarrow{y}$, so in our general definition we can require that the substitution behaves in the same way, i.e. it affects only those free variables of $\overrightarrow{N}$ that are not also in $\overrightarrow{x}$. We can put all this together to get the definition of the distributed invertibility up to head-free substitutions.

Definition 4.10 (Distributed Invertibility up to suitable head free substitutions)
Given $\overrightarrow{N}$, $\overrightarrow{M}$, $\overrightarrow{x}$, $\overrightarrow{y}$ as above, we will write $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y}))^{\bullet\circ}$ if $DistInv(\overrightarrow{M}^{\circ}, \overrightarrow{x}, \overrightarrow{N}^{\bullet}, \overrightarrow{y})$ holds, where

1. *the head-free substitution $\circ$ affects only variables that are in $(FV(\overrightarrow{N})\setminus\overrightarrow{x})$ and maps them into terms with no occurrences of $FV(\overrightarrow{N})$ or $\overrightarrow{y}$ or type expressions*

2. *the head-free substitution* $\bullet$ *affects only variables that are in* $(FV(\overrightarrow{M}) \setminus \overrightarrow{y})$ *and maps them into terms with no occurrences of* $FV(\overrightarrow{M})$ *or* $\overrightarrow{x}$ *or type expressions*

3. *these head free substitutions* $\circ$ *and* $\bullet$ *substitute type expressions with no occurrences of* $\times$ *or* $\mathbf{T}$ *for type variables.*

Using this definition of distributed invertibility in the modified lemma, we will be able to overcome the first problem described above.

As for the second problem, namely that the number of the Q_i's and P_i's must be the same in order to apply induction, we can actually extend as needed the shorter sequence to match the longer one by means of fresh variables (this will be shown in the proof of the modified Lemma).

Now, we are almost done. Let's put it all together, and see how the modified Lemma looks.

Lemma 4.11 (Main Lemma)
Let $\overrightarrow{N}$, $\overrightarrow{M}$ *(with* $m = \mid \overrightarrow{N} \mid$ *and* $n = \mid \overrightarrow{M} \mid$*) be sequences of terms and/or types and* $\circ$ *and* $\bullet$ *be head-free substitutions such that* $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y}))^{\circ\bullet}$*, then the following hold:*

1. *the substitutions* $\bullet$ *and* $\circ$ *are idempotent, i.e.* $\bullet\bullet = \bullet$ *and* $\circ\circ = \circ$

2. *when* M_i *and* N_j *are terms,*

$$M_i = \lambda \overrightarrow{v_i}.x_{\sigma(i)}\overrightarrow{P_i}, \quad N_j = \lambda \overrightarrow{u_j}.y_{\pi(j)}\overrightarrow{Q_j}$$

where $\sigma{:}n \to m$, $\pi{:}m \to n$ *are integer functions s.t.* $\sigma(\pi(i))=i$ *if* M_i *is a term and* $\pi(\sigma(j))=j$ *if* N_j *is a term*

3. $n = m$

4. *every* P_{i_k} (Q_{j_h}) *that is a type expression is just a type variable*

5. *every variable free in* P_{i_k} (Q_{j_h}) *has no occurrence of* $\times$ *or* $\mathbf{T}$ *in the type expressions occurring in it*

6. P_{i_k} (Q_{j_h}) *has no occurrence of* $\times$ *or* $\mathbf{T}$ *in its type if* P_{i_k} (Q_{j_h}) *is a term*

7. *Define* $s_1 = \mid \overrightarrow{P}_i \mid, r_2 = \mid \overrightarrow{u}_{\sigma(i)} \mid, r_1 = \mid \overrightarrow{v}_i \mid, s_2 = \mid \overrightarrow{Q}_{\sigma(i)} \mid$*. Without loss of generality, suppose* $s_1 \geq r_2$*. Then* $r_1 \geq s_2$*. Furthermore,* $\overrightarrow{Q}_{\sigma(i)}$ *can be extended with a sequence of type and/or term variables* $[u''_1, \ldots, u''_{r_1-s_2}]$ *to a sequence* $\overrightarrow{Q'}_{\sigma(i)}$ *such that* $DistInv(\overrightarrow{P}_i, \overrightarrow{v}_i, \overrightarrow{Q'}_{\sigma(i)}, \overrightarrow{u'}_{\sigma(i)})^{\triangleright\triangleleft}$ *holds, where* $\overrightarrow{u'}_{\sigma(i)} = \overrightarrow{u}_{\sigma(i)} \cup [u''_1, \ldots, u''_{r_1-s_2}]$ *and* $\triangleleft, \triangleright$ *are suitable head-free substitutions.*

Proof. This is done by a tedious case analysis (see Appendix C for full details). □

Remark 4.12 *The original Lemma had just conditions 2 and 3. Conditions 5, 6 and 7 are needed to allow the argument to inductively apply to the subterms, as described before, while the remaining conditions 1 and 4 are needed for technical reasons to make the final proof work.*

The reader familiar with the original characterization from [Dez76] will notice how the proof of this Lemma has a very similar flavour, but with the further complications arising from typing, that do not allow arbitrary η expansions and force us to introduce condition 7.

4.3.2 Relating coordinates to invertible terms in $\lambda^2\beta\eta$

Now, condition 7 is exactly what is needed for our induction argument: we start with a set of coordinates up to substitution, and we end with a set of smaller coordinates, still up to substitution, so that the following Proposition can be very easily proved by induction.

Proposition 4.13 *If* $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y})^{\bullet\circ}$*, then*

1. *The terms in* $\overrightarrow{N}$ *and* $\overrightarrow{M}$ *are terms of* $\lambda^2\beta\eta$
2. *Every occurrence of a type expression is just a type variable.*

Proof. By an easy induction on the complexity n of the longest term in $\overrightarrow{N}$ and $\overrightarrow{M}$.

- Base:
  - (1) n = 1 so every term in $\overrightarrow{N}$ and $\overrightarrow{M}$ is just a variable (cannot be $* : \mathbf{T}$: remember that the type of the elements of the sequences are simple, hence do not contain $\mathbf{T}$)
  - (2) the definition of $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y})^{\bullet\circ}$, point 3, (i.e. $M_i[\overrightarrow{N}/\overrightarrow{x}] = y_i$ and $N_i[\overrightarrow{M}/\overrightarrow{y}] = x_j$), forces every type expression to be a simple type variable. There are no other type expressions as the terms are of length 1.
- Ind. Step: n+1
  - (1) We get from the main lemma that $M_i = \lambda\overrightarrow{v_i}.x_{\sigma(i)}\overrightarrow{P_i}$ and $N_j = \lambda\overrightarrow{u_j}.y_{\pi(j)}\overrightarrow{Q_j}$ with $DistInv(\overrightarrow{P_i}, \overrightarrow{v_i}, {Q'}_{\sigma(i)}, \overrightarrow{u'}_{\sigma(i)})^{\bowtie}$, and where the complexity of the terms in $\overrightarrow{P_i}$ and $\overrightarrow{Q'}_{\sigma(i)}$ is strictly lower than n+1 for every i and j. We can apply the induction hypothesis and get that every term in $\overrightarrow{P_i}$ and $\overrightarrow{Q'}_{\sigma(i)}$ belongs to $\lambda^2\beta\eta$. Furthermore, we know from the main lemma that every type expression in $\overrightarrow{P_i}$ and $\overrightarrow{Q'}_{\sigma(i)}$ is just a type variable. Since $\overrightarrow{Q}_{\sigma(i)} \subseteq \overrightarrow{Q'}_{\sigma(i)}$, this suffices to show that every $M_i = \lambda\overrightarrow{v_i}.x_{\sigma(i)}\overrightarrow{P_i}$ and $N_j = \lambda\overrightarrow{u_j}.y_{\pi(j)}\overrightarrow{Q_j}$ have no occurrence of constants or complex type expressions in them and thus belongs to $\lambda^2\beta\eta$.
  - (2) as in the base case for elements in $\overrightarrow{N}$ and $\overrightarrow{M}$ that are type expressions. Direct consequence of 1 for type expressions occurring inside terms.

□

And we get, as a special case,

Corollary 4.14 *The coordinates* $\overrightarrow{N}$ *= [*$N_1, \ldots, N_m$*] and* $\overrightarrow{M}$ *= [*$M_1, \ldots, M_n$*] in Proposition 4.4 are terms of* $\lambda^2\beta\eta$*. Furthermore,* $n = m$*.*

Proof. It follows from the definitions that $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y})^{\bullet\circ}$ holds with $\bullet$ and $\circ$ empty substitutions. Hence, the result follows from the previous Proposition, point (1), as well as $n = m$. □

So we have factored out pairing and the constant $*$, and we have restricted ourselves to $\lambda^2\beta\eta$. We can do the same for equalities.

Lemma 4.15 *For the coordinates in Proposition 4.4 the equalities* $M_i[\overrightarrow{N}/\overrightarrow{x}] =_{\beta^2\eta^2\pi*} y_i$ *and* $N_i[\overrightarrow{M}/\overrightarrow{y}] =_{\beta^2\eta^2\pi*} x_j$ *hold in* $\lambda^2\beta\eta$ *(i.e.* $M_i[\overrightarrow{N}/\overrightarrow{x}] =_{\beta^2\eta^2} y_i$ *and* $N_i[\overrightarrow{M}/\overrightarrow{y}] =_{\beta^2\eta^2} x_j$ *).*

Proof. Since the reduction $\stackrel{\beta^2\eta^2\pi^*}{\longrightarrow}$ is Church-Rosser and a variable is in normal form (notice that the variables we are considering are not redexes for gentop as their types do not contain $\mathbf{T}$), there must be a reduction path from $M_i[\overrightarrow{N}/\overrightarrow{x}]$ to y_i. We know from the previous Corollary 4.14 that there is no constant in the terms we consider, so there is no π, SP, *gentop*, η_{top}or SP_{top} redex nor any can be created by any reduction. So this reduction path must be made of β and η reductions only, which implies that equality holds for the system consisting of β plus η alone too.
This is a special case of a more general phenomenon, as seen in Proposition A.8. □

4.4 Syntactic Characterization of canonical bijections

We have shown that the coordinates of a canonical bijection are terms of $\lambda^2\beta\eta$, and we also know now that these coordinates, even when seen as terms of $\lambda^2\beta\eta$, still enjoy this peculiar property we called **distributed invertibility**. We are now in a position to relate these coordinates to invertible terms of the pure $\lambda^2\beta\eta$. As we have already seen in the Survey, a characterization of the invertible terms of $\lambda^2\beta\eta$ is provided in [BL85] as the 2-f.h.p. (see 2.5 and 2.6). Now, it is easy to check that the coordinates M_i's of a canonical bijection are the *body* of a 2-f.h.p. (in the sense that they contain only a free variable $x_{\sigma(i)}$, such that $\lambda x_{\sigma(i)}.M_i$ is a 2-f.h.p.).

Theorem 4.16 *For the coordinates in Proposition 4.4 there exist a permutation $\sigma{:}n \to n$ s.t. $\lambda x_{\sigma(i)}.M_i$ and $\lambda y_{\pi(j)}.N_j$ are second order f.h.p.'s, where π is σ^{-1}.*

Proof. Just build the $\lambda^2\beta\eta$ terms, for a suitably typed new variable z,

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

They are terms of $\lambda^2\beta\eta$ (Corollary 4.14) and it is easy to check that they are invertible w.r.t. $\beta\eta$ equality, as

$$\begin{aligned}
M\circ N &=_{\beta^2\eta^2} \lambda w.(M(Nw)) \\
&=_{\beta^2\eta^2} \lambda w.(\lambda z x_1 \dots x_n.z M_1 \dots M_n)((\lambda z y_1 \dots y_n.z N_1 \dots N_n)w) \\
&=_{\beta^2\eta^2} \lambda w.(\lambda z x_1 \dots x_n.z M_1 \dots M_n)(\lambda y_1 \dots y_n.w N_1 \dots N_n) \\
&=_{\beta^2\eta^2} \lambda w.(\lambda x_1 \dots x_n.(\lambda y_1 \dots y_n.w N_1 \dots N_n) M_1 \dots M_n) \\
&=_{\beta^2\eta^2} \lambda w.\lambda x_1 \dots x_n.w N_1 \dots N_n[\overrightarrow{M_i}/\overrightarrow{y_i}] \\
&=_{\beta^2\eta^2} \lambda w.\lambda x_1 \dots x_n.w N_1[\overrightarrow{M_i}/\overrightarrow{y_i}] \dots N_n[\overrightarrow{M_i}/\overrightarrow{y_i}] \\
&=_{\beta^2\eta^2} \lambda w.\lambda x_1 \dots x_n.w x_1 \dots x_n \text{ (due to the previous Lemma 4.15)} \\
&=_{\beta^2\eta^2} \lambda w.w
\end{aligned}$$

Similarly for N∘M.
So N and M are second order f.h.p. and this implies that every M_i has only one occurrence of the x_i's (namely $x_{\sigma(i)}$) and the same for the N_i's. Hence

$$\begin{aligned}
M_i[\overrightarrow{N_i}/\overrightarrow{x_i}] &\equiv M_i[N_{\sigma(i)}/x_{\sigma(i)}] =_{\beta^2\eta^2} y_i, 1 \le i \le n \\
N_i[\overrightarrow{M_i}/\overrightarrow{y_i}] &\equiv N_i[M_{\pi(i)}/y_{\pi(i)}] =_{\beta^2\eta^2} x_i, 1 \le i \le n
\end{aligned}$$

and

$$\lambda x_{\sigma(i)}.M_i : A_{\sigma(i)} \to B_i, \quad \lambda y_i.N_{\sigma(i)} : B_i \to A_{\sigma(i)}$$

are second order f.h.p. □

Now, this exact knowledge of the shape of a coordinate finally gives us the syntactic characterization of canonical bijections.

Theorem 4.17 (shape of a canonical bijection)
Let M:$A_1 \times \ldots \times A_n \to B_1 \times \ldots \times B_n$ be a canonical bijection of $\lambda^2\beta\eta\pi*$. Then either M is the identity or there is a permutation π:$n \to n$ s.t.

1. *$n.f.(M) = \lambda z.(\langle M_1[(p_{\pi(1)}z)/x_{\pi(1)}], \ldots, M_n[(p_{\pi(n)}z)/x_{\pi(n)}]\rangle)$,*
2. *$\lambda x_{\pi(i)}.M_i : A_{\pi(i)} \to B_i$ are 2-f.h.p.'s*
3. *in particular, $M = \lambda z.(\langle M'_1(p_{\pi(1)}z), \ldots, M'_n(p_{\pi(n)}z)\rangle)$, with $M'_i = \lambda x_{\pi(i)}.M_i$'s 2-f.h.p.'s*

Proof. We have shown in Proposition 4.4 that every canonical invertible term M can be written in terms of its coordinates M_i's as $\lambda z.(\lambda x_1 \ldots x_n.(\langle M_1, \ldots, M_n\rangle))(\mathrm{p}_1 z) \ldots (\mathrm{p}_n z)$. This latter term reduces to $\lambda z.(\langle M_1[(\mathrm{p}_{\pi(1)}z)/x_{\pi(1)}], \ldots, M_n[(\mathrm{p}_{\pi(n)}z)/x_{\pi(n)}]\rangle)$, that we claim is in normal form if it does not reduce via SP to $\lambda z.z$. This follows immediately from the fact that the coordinates are (by construction) in normal form and that the substitutions $[(\mathrm{p}_{\pi(i)}z)/x_{\pi(i)}]$ do not create any new redex. So we get 1.

Theorem 4.16 guarantees 2.

Setting now $M'_i = \lambda x_{\pi(1)}.M_i$, we get 3.
Notice that in the particular case $n = 1$, $n.f.(M) = \lambda z.M_1$ and it is a 2-f.h.p. □

Corollary 4.18 *It is possible to decide if a generic term $M : A \to B$ is invertible.*

Proof. Proceed as in Proposition 3.6 to build the term $M' : n.f(A) \to n.f.(B)$. Then normalize it and check if its shape is as in Theorem 4.17. □

Now we can use this characterization to prove completeness of $Th^2_{\times T}$ for isomorphisms of type normal forms given by canonical bijections (and hence for all isomorphisms of types, Propositions 3.7 and 3.8) in the following Section.

5 Completeness for isomorphisms

We know now that two types A and B are isomorphic iff their normal forms are; furthermore, $Th^2_{\times T}$ equates a type to its normal form. Thus it is possible to show that $Th^2_{\times T}$ is complete for the definable isomorphisms if we can show that $Th^2_{\times T}$ is complete for isomorphisms between stratified types. But if two stratified types are definably isomorphic, then the isomorphism can be defined by a canonical invertible term, and we can easily get the result by inspecting its structure, as given in

Proposition 5.1 *Let A, B be stratified type expressions that are not* **T**.
Then $A \cong_d B \Rightarrow Th^2_{\times T} \vdash A = B$.

Proof. Suppose A $\cong_d$ B via a canonical invertible term M. We can assume that free and bound type variables are all different in M, as the theory $Th^2_{\times T}$ allows renaming of bound type variables. If A= $A_1 \times \ldots \times A_m$, then B= $B_1 \times \ldots \times B_m$ (as the length of two isomorphic stratified types is the same, see Corollary 4.14).
If m=1 then A $\cong_d$ B via a 2-f.h.p. M and we will show the completeness of $Th^2_{\times T}$ by induction on the depth of the Böhm-tree BT(M) of M.

- Depth 1: M $\equiv \lambda$z : C. z. Thus M : $C \to C$, and $Th^2_{\times T} \vdash$ C = C by reflexivity.
- Depth d+1: M $\equiv \lambda$z : A. $\lambda\overrightarrow{x}$: $\overrightarrow{B}$. z $\overrightarrow{N}$. Recall z $\overrightarrow{N}$ = z$N_1 \ldots N_n$ where, for some permutation σ, if the ith abstraction in λx:D is λx_i:D_i then $\lambda x_{\sigma(i)} : D_i.N_i$ is a 2-f.h.p. We will proceed by induction on the length n of $\overrightarrow{N}$.
  If n=0 then the result follows as in the case of depth 1.

If n=k+1, then

$$M\equiv \lambda z : A.\lambda x_1 : B_1 \dots \lambda x_{\sigma(k+1)} : B_{\sigma(k+1)} \dots \lambda x_{k+1} : B_{k+1}.zN_1 \dots N_{k+1} : A \to B$$

In what follows, A $\square$ B will stand for one among $A \to B$ and $\forall A.B$. First, we notice that, in order to type check, we must have

$$A = (A_1 \square_1 \dots \square_k A_{k+1} \square_{k+1} E) \text{ for some } E,$$

$$B = (B_1 \square_1 \dots \square_k B_{k+1} \square_{k+1} F) \text{ for some } F,$$

with F = E$[N_{i_1}/A_{i_1}] \dots [N_{i_r}/A_{i_r}]$, where the i_j are all the indexes s.t. $\square_{i_r} = \forall$ in A. Then, we proceed by cases on N_{k+1}:

1. N_{k+1} is a term. Then we have

(i) $\lambda x_{\sigma(k+1)} : B_{\sigma(k+1)}.N_{k+1} : B_{\sigma(k+1)} \to Type(N_{k+1})$

(ii) $\square_{\sigma(k+1)} = \ \to$

As $\mid BT(\lambda x_{\sigma(k+1)} : B_{\sigma(k+1)}.N_{k+1}) \mid < d+1$, we know by induction hypothesis on d that.

(iii) $\quad Th^2_{\times T} \vdash B_{\sigma(k+1)} = Type(N_{k+1}).$

Furthermore, it is straightforward to see that, since free and bound type variables in M are different, the term

$$\lambda z : A.\lambda x_1 : B_1 \dots \lambda x_{\sigma(k+1)-1} : B_{\sigma(k+1)-1}.$$
$$\lambda x_{\sigma(k+1)+1} : B_{\sigma(k+1)+1} \dots \lambda x_{k+1} : B_{k+1}.zN_1 \dots N_k : A \to B'$$

is well formed and is still a 2-f.h.p., with only k N's. Then, by induction hypothesis on n,

(iv) $\quad Th^2_{\times T} \vdash A = B'.$

Now notice that

$$B' = (B_1 \square_1 \dots B_{\sigma(k+1)-1} \square_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \square_{\sigma(k+1)+1} \dots \square_k B_{k+1} \square_{k+1} Type(N_{k+1}) \to F)$$

So, since equality is substitutive, (ii), (iii) and (iv) yield

$$\begin{aligned} Th^2_{\times T} \vdash A \ &= B' \\ &= (B_1 \square_1 \dots B_{\sigma(k+1)-1} \square_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \square_{\sigma(k+1)+1} \dots \square_k B_{k+1} \square_{k+1} B_{\sigma(k+1)} \to F) \\ &= (B_1 \square_1 \dots B_{\sigma(k+1)-1} \square_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \square_{\sigma(k+1)+1} \dots \square_k B_{k+1} \square_{k+1} B_{\sigma(k+1)} \square_{\sigma(k+1)} F) \end{aligned}$$

Now, it suffices to show that this last type is equal to B in $Th^2_{\times T}$. Since we assumed free and bound type variables to be different, if $\square_j$ is a $\forall$, with j greater than σ(k+1), then B_j cannot be free in $B_{\sigma(k+1)}$. This allows to use axiom 10, together with 8 and the equality $A \to (B \to C) = B \to (A \to C)$ derived from axioms 1 and 3 (that are always applicable), in order to repeatedly swap $B_{\sigma(k+1)}$ $\square_{\sigma(k+1)}$ with B_j $\square_j$ up to $B_{\sigma(k+1)+1}$ $\square_{\sigma(k+1)+1}$. By this, we obtain the required equality.

2. N_{k+1} is a type. Then we have

$$M\equiv \lambda z : A.\lambda x_1 : B_1 \dots \lambda X_{\sigma(k+1)} \dots \lambda x_{k+1} : B_{k+1}.zN_1 \dots N_k[X_{\sigma(k+1)}] : A \to B$$

As in the previous case, we can reduce to a smaller k via the term

$$\lambda z : A.\lambda x_1 : B_1 \dots \lambda x_{\sigma(k+1)-1} : B_{\sigma(k+1)-1}.$$
$$\lambda x_{\sigma(k+1)+1} : B_{\sigma(k+1)+1} \dots \lambda x_{k+1} : B_{k+1}.zN_1 \dots N_k : A \to B'$$

that is well formed, as free and bound variables are different, so that removing the abstraction does not cause any capture of variables by other type abstractions. It is still a 2-f.h.p., so we get $Th^2_{\times T} \vdash$ A= B'. Now,

$$B' = (B_1 \Box_1 \dots B_{\sigma(k+1)-1} \Box_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \Box_{\sigma(k+1)+1} \dots \Box_k B_{k+1} \Box_{k+1} \forall Z.F')$$

where F $\equiv$ F'$[X_{\sigma(k+1)}/\mathrm{Z}]$, so that $\forall X_{\sigma(k+1)}.F \equiv \forall X_{\sigma(k+1)}.F'[X_{\sigma(k+1)}/\mathrm{Z}]$, that is equal, by axiom 9, to $\forall Z.F'$. Hence,

$$\begin{aligned} Th^2_{\times T} \vdash A &= B' \\ &= (B_1 \Box_1 \dots B_{\sigma(k+1)-1} \Box_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \Box_{\sigma(k+1)+1} \dots \Box_k B_{k+1} \Box_{k+1} \forall Z.F') \\ &= (B_1 \Box_1 \dots B_{\sigma(k+1)-1} \Box_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \Box_{\sigma(k+1)+1} \dots \Box_k B_{k+1} \Box_{k+1} \forall X_{\sigma(k+1)}.F) \\ &= (B_1 \Box_1 \dots B_{\sigma(k+1)-1} \Box_{\sigma(k+1)-1} B_{\sigma(k+1)+1} \Box_{\sigma(k+1)+1} \dots \Box_k B_{k+1} \Box_{k+1} B_{\sigma(k+1)} \Box_{\sigma(k+1)} \mathrm{F}) \end{aligned}$$

To show that this last type is equal to B in $Th^2_{\times T}$, we can use the swap axioms 8 and 10, but this last one is a conditional axiom, that can be used as we want only if we can ensure that $B_{\sigma(k+1)} \equiv X_{\sigma(k+1)}$ is not free in any B_j s.t. $\Box_j$ is $\to$.

Indeed, we can show this fact, proceeding by contradiction. Suppose that $X_{\sigma(k+1)}$ is free in some B_j s.t. $\Box_j$ is $\to$. Notice that the types equated by the theory $Th^2_{\times T}$ have the same free type variables, so that $N_{\sigma^{-1}(j)}$ is then a term, and $X_{\sigma(k+1)}$ is free in its type, which is isomorphic to B_j as already remarked. For M to type-check, then, the type of $zN_1 \dots N_{(\sigma^{-1}(j))-1}$ must be $Type(N_{\sigma^{-1}(j)}) \to G$ for some G.
Now notice that, due to the structure of 2-f.h.p.'s, $X_{\sigma(k+1)}$ occurs (in type applications) only once and exactly as N_{k+1} in M: under these conditions, we can prove by induction on k that $X_{\sigma(k+1)}$ must occur in the type of z. This leads to a contradiction, though, as then the subterm $\lambda X_{\sigma(k+1)}. \dots \lambda x_{k+1} : B_{k+1}.zN_1 \dots N_k[X_{\sigma(k+1)}]$ of M is not well formed: there is a type abstraction over a type variable $(X_{\sigma(k+1)})$ occurring free in the type of a free term variable (z), and this violates the term formation rule for type abstraction of Definition A.1.

If $m > 1$, then by Theorem 4.17 there exist an integer m and a permutation $\pi{:}m \to m$ s.t. the normal form of M is λz.$(\langle M_1[(\mathrm{p}_{\pi(1)}z)/x_p(1)], \dots, M_n[(\mathrm{p}_{\pi(m)}z)/x_p(m)]\rangle)$, where the terms $\lambda x_i.\mathrm{M}\pi(\mathrm{i}){:}A_i \to B_{\pi(i)}$ are second order f.h.p.'s. This means that $A_i \cong_d B_{\pi(i)}$ via 2-f.h.p.'s, that are invertible. We have already shown that $Th^2_{\times T}$ is complete for isomorphisms definable by 2-f.h.p.'s, and to get the result it is enough to notice that $Th^2_{\times T}$ includes commutativity and associativity for the product. □

So we can finally get our Main Theorem.

Theorem 5.2 (Main Theorem, difficult implication) *$A \cong B \Rightarrow Th^2_{\times T} \vdash A = B$.*

Proof. By Theorem 1.3, it is enough to show that A $\cong_d$ B $\Rightarrow Th^2_{\times T} \vdash$A = B. This is now an easy consequence of Propositions 3.7 and 5.1. □

5.1 (Definable) isomorphisms and uniform isomorphisms in every model

Corollary 5.3 *If $A \cong B$, then there is an uniform isomorphism between A and B.*

Proof. If A and B are isomorphic in every model, then they are provably equal in $Th^2_{\times T}$ and since for every axiom of $Th^2_{\times T}$ there is an uniform isomorphism, we can derive a uniform isomorphism for B and A by just composing the ones associated to every step of the proof of B = A. □

6 Decidability of the equational theory

An immediate consequence of the Main Theorem is

Theorem 6.1 *Given types A and B it is decidable whether they are isomorphic in all models of* $\lambda^2\beta\eta\pi*$.

Proof. Let the type normal forms of A and B be $(A_1 \times \ldots \times A_n)$ and $(B_1 \times \ldots \times B_m)$, respectively. We know that none of the A_i and B_j can contain any occurrence of $\mathbf{T}$ or $\times$ (by Remark 3.4). Now we know that A and B are isomorphic if and only if $n = m$ (Lemma 4.14), and there exist a permutation $\sigma : n \to n$ such that $A_i \cong_d B_{\sigma(j)}$. Hence, to decide A $\cong_d$ B it suffices to be able to decide $A_i \cong_d B_j$, as then we can just try $A_i \cong_d B_{\sigma(i)}$ for all $1 \leq i \leq n$ and all permutations $\sigma : n \to n$.
By inspecting the proof of Proposition 5.1, we see that to axiomatise $A_i \cong_d B_{\sigma(i)}$, we only need the equational theory (that we will call S) made up of axioms 8, 9, 10 and the equality $A \to (B \to C) = B \to (A \to C)$ derived from axioms 1 and 3.
These equalities do not change the length of formulae (if we use the notation of Proposition 5.1), hence, if S $\vdash A_i = B_j$, then A_i and B_j have the same length.
Axiom 9, though, changes the alphabet of a formula, so that we need to gain some control on its application in the proofs of equality modulo S if we want to provide an effective decision procedure. Actually, if we start with two types A and B equal in S where free and bound type variables are different (and we can do it without loss of generality), we can find a proof of equality where any use of axiom 9 appears only at the end as follows:

- first, given a proof A $= A_1 = \ldots = A_n =$ B, we transform it into a proof A $= A'_1 = \ldots = A'_n =$ B where at every stage all free and bound variables are distinct[5]. It suffices, whenever a variable X is replaced by Y via axiom 9 in the proof, to replace Y by a fresh variable Z never used in the proof. It is easy to see that the resulting sequence of formulae is still a proof in S, as the only constraint on the name of the variables appears when using axiom 10 from right to left, and if a bound occurrence of Y is not free in some formula C outside the scope of the $\forall$ that binds Y, then *a fortiori* Z is not free in C, by the way it has been chosen. The proof obtained after applying orderly this procedure to every occurrence of axiom 9 is the required proof A $= A'_1 = \ldots = A'_n =$ B.

- second, in a proof where all the free and bound variables are distinct, we can push all the applications of axiom 9 at the end. This can be shown by noticing that a sequence of equalities $A_{i-1} =_{axiom\ 9} A_i = A_{i+1}$, where the second equality is not an application of axiom 9 can be always replaced by a sequence $A_{i-1} = A'_i =_{axiom\ 9} A_{i+1}$, where the first equality is not an application of axiom 9 and free and bound variables are all distinct in A'_i too. The only nontrivial case is axiom 10 after axiom 9, but under the hypothesis of having all distinct variables, everything works fine.

This means that if A $=_S$ B, then B is just an α variant of one of the formulae equal to A in S less axiom 9. These formulae are in finite number (as these last axioms do not change the length nor the alphabet of a formula) and can be effectively generated (for example, by a depth-first search of the equality proofs). So it suffices to check B against each of these for α equality, which is a decidable task. □

Since $Th^2_{\times T}$ is the theory of isomorphic types, an immediate consequence of the theorem above is the decidability of $Th^2_{\times T}$.

Corollary 6.2 *Equality in* $Th^2_{\times T}$ *is decidable.*

[5]Actually, as remarked by one of the referees, what we obtain is not exactly B, but an α variant of B, as we may rename bound variables in our procedure. In that case, we can easily complete the proof with some steps of α equality, so we need not consider this case explicitly in the proof.

Deciding equality in theories containing associative and commutative operators, as well as binding operators, like $\forall$, is in general far from trivial (actually it is at least as hard as deciding Graph Isomorphism, see [Bas90]), so that the complexity of the decision procedure given above (it essentially requires to examine all the search space) is no surprise. Actually, the very fact that $Th^2_{\times T}$ is decidable is not an obvious result and the insight gained by the study of invertible terms (in Proposition 5.1) is essential in order to establish it.

7 Conclusions and Future Work

We have provided a finite, complete and decidable axiomatization of the types isomorphic in every model of $\lambda^2\beta\eta\pi*$, by means of purely syntactical proof theoretic methods. Due to the well-known connection between typed lambda calculus and intuitionistic logic, this work also fully characterizes the *constructively equivalent* formulae of IPC($\forall$, $\Rightarrow$, $\wedge$, True), the second order intuitionistic positive propositional calculus.

On the practical side, these results give the necessary theoretical basis to the development both of library search tools based on the *type as specification* paradigm, and of extensions of the usual type-checking algorithms for strongly typed functional languages. In this direction, the development of efficient algorithms to decide equality in our theory $Th^2_{\times T}$, as well as the study of matching and unification up to isomorphism, need to be addressed. As pointed out by one of the referees, another promising application is in retrieving proofs of propositions from a theorem library.

We intend also to investigate how far our methods can be extended to yield similar results for other or further extensions of the λ-calculus. It would be interesting to provide similar characterizations in the presence of additional axioms, like the ones used to describe recursive or inductive types. More difficult seems the study of isomorphisms when restricting the definition of type isomorphism to closed terms (as is suggested for example in [CMMS91] and [ACC93]).

Finally, let us just hint at another possible application of type isomorphisms. Terms inhabiting isomorphic types are truly interchangeable, as they *do* compute the same class of functions, but they *need not have the same complexity*: this is the case for example for the distributivity of arrow over product. We may expect to exploit these isomorphisms to perform program transformations in the optimizing phase of a compiler, where one has the freedom to choose the most efficient implementation.

Acknowledgements

Part of this work was carried out at the Department of Computer Science of the Cornell University, Ithaca - N.Y. I wish to thank Robert Constable, for inviting me there. He gave me the opportunity to access the great facilities of Cornell University.

I am greatly indebted to my advisor, Giuseppe Longo, for continuous encouragement in this work, to Eugenio Moggi for insights and to Kim Bruce for valuable discussions. Thanks also to Andy Pitts and Martin Hyland for many helpful discussions, and to Radha Jagadeesan for listening to the first expositions of the fully detailed syntactical proofs during my stay in Cornell University.

A special thanks goes to the anonymous referees, that carefully read this work providing an extremely valuable feedback. This paper benefited greatly of their continuous and detailed comments.

References

[AB91] Franco Alessi and Franco Barbanera. Strong conjunction and intersection types. Dipartimento di Informatica, Universitá di Torino (Italy), manuscript., 1991.

[ACC93] Martín Abadi, Luca Cardelli, and Pierre-Louis Curien. Formal parametric polymorphism. In *Ann. ACM Symp. on Principles of Programming Languages (POPL)*. ACM, 1993.

[AL91] Andrea Asperti and Giuseppe Longo. *Categories, Types, and Structures*. MIT Press, 1991.

[Bar84] Henk Barendregt. *The Lambda Calculus; Its syntax and Semantics (revised edition)*. North Holland, 1984.

[Bas90] David Basin. Equality of Terms Containing Associative-Commutative Functions and Commutative Binding Operators is Isomorphism Complete in 10th Int. Conf. on Automated Deduction. *Lecture Notes in Computer Science*, 449, July 1990.

[BDCL92] Kim Bruce, Roberto Di Cosmo, and Giuseppe Longo. Provable isomorphisms of types. *Mathematical Structures in Computer Science*, 2(2):231–247, 1992. Proc. of Symposium on Symbolic Computation, ETH, Zurich, March 1990.

[BL85] Kim Bruce and Giuseppe Longo. Provable isomorphisms and domain equations in models of typed languages. *ACM Symposium on Theory of Computing (STOC 85)*, May 1985.

[CDC91] Pierre-Louis Curien and Roberto Di Cosmo. A confluent reduction system for the λ-calculus with surjective pairing and terminal object. In Leach, Monien, and Artalejo, editors, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 510 of *Lecture Notes in Computer Science*, pages 291–302. Springer-Verlag, 1991.

[CMMS91] Luca Cardelli, Simone Martini, John C. Mitchell, and Andre Scedrov. An extension of system F with subtyping. In T. Ito and A. R. Meyer, editors, *Theoretical Aspects of Computer Software*, volume 526 of *Lecture Notes in Computer Science*, pages 750–770. Springer-Verlag, September 1991.

[DC92a] Roberto Di Cosmo. Deciding type isomorphisms in a type assignment framework. *Journal of Functional Programming*, 1992. To appear in the Special Issue on ML.

[DC92b] Roberto Di Cosmo. Type isomorphisms in a type assignment framework. In *Ann. ACM Symp. on Principles of Programming Languages (POPL)*, pages 200–210. ACM, 1992.

[DCK93] Roberto Di Cosmo and Delia Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 1993.

[Dez76] Mariangiola Dezani-Ciancaglini. Characterization of normal forms possessing an inverse in the $\lambda\beta\eta$ calculus. *Theoretical Computer Science*, 2:323–337, 1976.

[DT69] J. Doner and Alfred Tarski. An extended arithmetic of ordinal numbers. *Fundamenta Mathematica*, 65:95–127, 1969.

[Gur85] R. Gurevic. Equational theory of positive numbers with exponentiation. *Proceedings of the American Mathematical Society*, 94(1):135–141, May 1985.

[Gur90] R. Gurevic. Equational theory of positive numbers with exponentiation is not finitely axiomatizable. *Annals of Pure and Applied Logic*, 49:1–30, 1990.

[Hen77] Leon Henkin. The logic of equality. *American Mathematical Monthly*, 84:597–612, October 1977.

[How80] W.A. Howard. The formulae-as-types notion of construction. In Hindley and Seldin, editors, *To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus and formalism*, pages 479–490. Academic Press, 1980.

[HR84] C. W. Henson and L. A. Rubel. Some applications of Nevanlinna theory to mathematical logic: Identities of exponential functions. *Trans. Am. Math. Soc.*, 282(1):1–32, March 1984.

[Les83] P. Lescanne. Computer experiments with the REVE term rewriting systems generator. In *Proceedings of 10th ACM Symposium on Principles of Programming Languages*, pages 99–108. Association for Computing Machinery, 1983.

[Les86] P. Lescanne. Reve a rewrite rule laboratory. In J. Siekmann, editor, *Proc. 8th Conf. on Automated Deduction*, Lecture Notes in Computer Science, pages 696–697, Oxford (England), 1986. Springer Verlag.

[LS86] Joachim Lambek and Philip J. Scott. *An introduction to higher order categorical logic*. Cambridge University Press, 1986.

[Mac81] A. Macintyre. The laws of exponentiation. In C. Berline, K. McAloon, and J.-P. Ressayre, editors, *Model Theory and Arithmetic*, volume 890 of *Lecture Notes in Mathematics*, pages 185–197. Springer-Verlag, 1981.

[Mar72] C.F. Martin. Axiomatic bases for equational theories of natural numbers. *Notices of the Am. Math. Soc.*, 19(7):778, 1972.

[Mar91] Simone Martini. Strong equivalence in positive propositional logic: provable realizability and type assignment. Dipartimento di Informatica, Universitá di Pisa (Italy), Internal Note., June 1991.

[Min77] Gregory Mints. Closed categories and the theory of proofs. *Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta im. V.A. Steklova AN SSSR*, 68:83–114, 1977.

[ML71] Saunders Mac Lane. *Categories for the working mathematician*, volume 5 of *GTM*. Springer, 1971.

[Mor91] R. Morgan. *Component Library Retrieval using property models*. PhD thesis, University of Durham - England, rick@easby.dur.ac.uk, 1991.

[Pot81] Garrel Pottinger. The Church Rosser Theorem for the Typed lambda-calculus with Surjective Pairing. *Notre Dame Journal of Formal Logic*, 22(3):264–268, 1981.

[Rey84] J.C. Reynolds. Polymorphism is not set-theoretic. *Lecture Notes in Computer Science*, 173, 1984.

[Rit90] Mikael Rittri. Retrieving library identifiers by equational matching of types in 10th Int. Conf. on Automated Deduction. *Lecture Notes in Computer Science*, 449, July 1990.

[Rit91] Mikael Rittri. Using types as search keys in function libraries. *Journal of Functional Programming*, 1(1):71–89, 1991.

[RT91] Colin Runciman and Ian Toyn. Retrieving re-usable software components by polymorphic type. *Journal of Functional Programming*, 1(2):191–211, 1991.

[Sol83] Serjey V. Soloviev. The category of finite sets and cartesian closed categories. *Journal of Soviet Mathematics*, 22(3):1387–1400, 1983.

[Sta83] Rick Statman. λ-definable functionals and $\beta\eta$ conversion. *Arch. Math. Logik*, 23:21–26, 1983.

A The calculus $\lambda^2\beta\eta\pi*$ and some basic notations

Definition A.1 *$\lambda^2\beta\eta\pi*$ is the extension of the second order lambda calculus defined as follows:*

- Types are defined by the following grammar:

  $Type ::= At \mid Var \mid Type \to Type \mid Type \times Type \mid \forall X.Type$

  where At is a set of countably many atomic types including the distinguished constant $\mathbf{T}$ and Var is a set of countably many type variables

  The intended meaning of $\mathbf{T}$ is the terminal object in the categorical sense, so $*$ below will stand for the unique term of type $\mathbf{T}$ (as required of a terminal object)[6].

- Terms (M:A will stand for *M is a term of type A*)

  - the set of terms contains a countable set $x, y, \ldots$ of term variables for each type and a constant $*$:$\mathbf{T}$
  - terms are constructed from variables and constants via the following term formation rules (notice the perfect analogy with the introduction and elimination rules for second order logic in natural deduction style)

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash (MN) : B}$$

$$\frac{\Gamma \vdash M : A \quad N : B}{\Gamma \vdash \langle M, N\rangle : A \times B} \qquad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \mathrm{p}_1 M : A} \quad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \mathrm{p}_2 M : B}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \lambda X.M : \forall X.A}\ {}^{7} \qquad \frac{\Gamma \vdash M : \forall X.A}{\Gamma \vdash M[B] : A[B/X]} \text{ for any type B.}$$

    Notice that pairing and projections are new *term formation rules* and not constants added to the language.

- Equality

(β)	$(\lambda x.M)N = M[\mathrm{N}/x]$	(η)	$\lambda x.Mx = M$ if $x \notin FV(M)$
(π)	$\mathrm{p}_i\langle M_1, M_2\rangle = M_i$	$(\mathbf{SP})$	$\langle \mathrm{p}_1 M, \mathrm{p}_2 M\rangle = M$
		$(\mathbf{top})$	$M = *$ if $M : \mathbf{T}$
$(\beta^{\mathbf{2}})$	$(\lambda X.M)[A] = M[A/X]$	$(\eta^{\mathbf{2}})$	$\lambda X.M[X] = M$ if X is not free in M

We will note $=_{\beta^2\eta^2\pi*}$ the theory of equality generated by β, η, π, SP, top, β^2 and η^2.

Notation A.2 (sequences, substitutions)

We will often use sequences of variables [x_1,...,x_n] or terms [M_1,...,M_n], and we will note them respectively $\overrightarrow{x}$ and $\overrightarrow{M}$. The *length* of a sequence $\overrightarrow{M}$ of terms (or variables) is the number of elements in the sequence, noted $\mid \overrightarrow{M} \mid$. A sequence can be empty. As is standard notation in the theory of λ-calculus, we will often use $\lambda\overrightarrow{x}.\overrightarrow{N}$ as a short-hand for $\lambda x_1 \ldots x_n.(\ldots(N_1N_2)\ldots N_m)$, where it is intended that N_1 is not an application $N_{11}N_{12}$, as otherwise we could take N_{11} as the starting term of the sequence.

[6] This notation is different from the one originally used in [BDCL92], where the symbol $*_A$ stands for the unique arrow of type $A \to \mathbf{T}$, and, though completely equivalent and interchangeable, is preferred here for ease of reference to [CDC91], where confluence of related systems is studied.

[7] With the proviso that the type variable X is not free in the type of any free variable of the term M.

Given a term N, a sequence $\overrightarrow{M} = [M_1,\ldots,M_n]$ of terms, and a sequence $\overrightarrow{x} = [x_1,\ldots,x_n]$ of variables, of the same length, $N[\overrightarrow{M}/\overrightarrow{x}]$ denotes the simultaneous substitution of every variable x_i with the term M_i in the term N (for simplicity, we always assume that bound variables are renamed as necessary to avoid capture of free variables). We may also use $\overrightarrow{N}[\mathrm{M}/\overrightarrow{x}]$ for the simultaneous substitution of all the variables in $\overrightarrow{x}$ with the same term M (similarly for type variables).

Our calculus has also pairs, so we need to introduce some less standard abbreviations for nested pairs, or n-tuples.

Notation A.3 (n-tuples, identities, free variables, types)

We write $\langle M_1, \ldots M_n\rangle$ for $\langle M_1, \langle M_2, \langle \ldots, \langle M_{n-1}, M_n\rangle\rangle \ldots\rangle\rangle$. Let $I_A = \lambda$x:A.x be the identity of type A and M∘N be the usual composition λx.(M(Nx)) of lambda terms. As usual, by $FV(M)$ we mean the free term variables $x : A$ of a term *declared with their type*, but, being in a second order setting, we also use $FTV(M)$ for the free type variables of a term M. As is usual notation in the theory of λ-calculus, we will say that a variable x it is *free for* a term N in a term M if the substitution of N for the free occurrences of x in M does not provoke capture of the free variables of N. Since the calculus is explicitly typed, we can write $Type(M)$ for the type of the term M, which is uniquely determined once we are given the type of its free variables.

Notation A.4 (calculi)

We will denote by $\lambda^2\beta\eta\pi$ the calculus $\lambda^2\beta\eta\pi*$ without terminal object and related rules, $\lambda^2\beta\eta$ the polymorphic typed calculus, and λ the type-free calculus, while $\lambda^1\beta\eta\pi*$ and $\lambda^1\beta\eta\pi$ are the first order restrictions of $\lambda^2\beta\eta\pi*$ and $\lambda^2\beta\eta\pi$ respectively.

Remark A.5 (Notion of reduction for $\lambda^2\beta\eta\pi*$) *The notion of reduction associated with the equational theory of $\lambda^2\beta\eta\pi*$ 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 and is weakly normalizing, thus allowing us to speak about normal forms for every term (see [CDC91]). In this paper we refer to this latter one when talking about reduction, normal forms, and so on, for $\lambda^2\beta\eta\pi*$, so we briefly recall here the necessary definitions.*

Definition A.6 (Terminal types and Canonical terms)

1. $Iso(\mathbf{T})$ (the collection of types isomorphic to $\mathbf{T}$) is the set defined as follows:
   (a) $\mathbf{T} \in Iso(\mathbf{T})$
   (b) if B $\in Iso(\mathbf{T})$, then $A \to B \in Iso(\mathbf{T})$ for every type A
   (c) if A $\in Iso(\mathbf{T})$ and B $\in Iso(\mathbf{T})$, then $A \times B \in Iso(\mathbf{T})$
   (d) if A $\in Iso(\mathbf{T})$ and X is a type variable, then $\forall X.A \in Iso(\mathbf{T})$.
2. for each type A $\in Iso(\mathbf{T})$, the associated *canonical* representative rep(A) is defined inductively as follows:
   (a) $rep(\mathbf{T})$ is $*$
   (b) $rep(A \to B)$ is λx:A.rep(B)
   (c) $rep(A \times B)$ is $\langle rep(A), rep(B)\rangle$
   (d) $rep(\forall X.A)$ is $\lambda X.rep(A)$.

Definition A.7 (Confluent notion of reduction for $\lambda^2\beta\eta\pi*$) $\overset{\beta^2\eta^2\pi*}{\longrightarrow}$ *is the notion of reduction for $\lambda^2\beta\eta\pi*$ generated by the following rewriting rules:*

Rules obtained by orienting the equalities:

(β) $(\lambda \mathrm{x.M})\mathrm{N} \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M[N/x]}$

(η) $\lambda \mathrm{x.Mx} \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M}$ if $x \notin FV(M)$

(π) $\mathrm{p}_i\langle M_1, M_2\rangle \overset{\beta^2\eta^2\pi*}{\longrightarrow} M_i$

(SP) $\langle \mathrm{p}_1 M, \mathrm{p}_2 M\rangle \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M}$

(β^2) $(\lambda X.M)[A] \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M[A/X]}$

(η^2) $\lambda X.M[X] \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M}$ if X is not a free type variable in M

Rules coming from completion:

$(gentop)$ $\mathrm{M{:}A} \overset{\beta^2\eta^2\pi*}{\longrightarrow} rep(\mathrm{A})$ if M:A and $\mathrm{A} \in Iso(\mathbf{T})$ and M is not already $rep(\mathrm{A})$

(SP_{top}) $\langle rep(A), \mathrm{p}_2 M\rangle \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M}$ if $\mathrm{M{:}}A \times B$ (and, of course, $\mathrm{A} \in Iso(\mathbf{T})$)

(SP_{top}) $\langle \mathrm{p}_1 M, rep(B)\rangle \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M}$ if $\mathrm{M{:}}A \times B$ (and, of course, $\mathrm{B} \in Iso(\mathbf{T})$)

(η_{top}) $\lambda \mathrm{x{:}A.M}rep(\mathrm{A}) \overset{\beta^2\eta^2\pi*}{\longrightarrow} \mathrm{M}$ if $\mathrm{A} \in Iso(\mathbf{T})$ and $\mathrm{x} \notin \mathrm{FV(M)}$.

Actually, the form of the rules in this rewriting system allows us to get a generalized conservativity result for the equational theories of $\lambda^2\beta\eta\pi*$ over $\lambda^2\beta\eta\pi$ and $\lambda^2\beta\eta$, and of $\lambda^2\beta\eta\pi$ over $\lambda^2\beta\eta$. This fact is relevant by itself, and essential in the reduction of coordinates to terms of $\lambda^2\beta\eta$ (in Lemma 4.15).

Proposition A.8 *(The equational theory of) $\lambda^2\beta\eta\pi*$ is a conservative extension of (the equational theory of) $\lambda^2\beta\eta\pi$. Similarly for $\lambda^2\beta\eta\pi$ with respect to $\lambda^2\beta\eta$.*

Proof.

Both the equality for $\lambda^2\beta\eta\pi*$ and that for $\lambda^2\beta\eta\pi$ can be derived from a Church-Rosser notion of reduction, (for the C-R property without terminal object, see [Pot81] for early results on $\lambda^1\beta\eta\pi$ and [CDC91] for a more recent detailed discussion and the references therein). Consider now M and N in $\lambda^2\beta\eta\pi$ such that $\lambda^2\beta\eta\pi* \vdash \mathrm{N} = \mathrm{M}$. By the Church-Rosser property, there is common reductum P (i.e. $\mathrm{M} \to \mathrm{P}$ and $\mathrm{N} \to \mathrm{P}$). Then $\lambda^2\beta\eta\pi* \vdash \mathrm{N} \to \mathrm{P}$ is actually a reduction $\lambda^2\beta\eta\pi \vdash \mathrm{N} \to \mathrm{P}$, as N contains no *gentop* redex, and no *gentop* redex can be created by the application of reduction rules. The same applies to $\lambda^2\beta\eta\pi* \vdash \mathrm{M} \to \mathrm{P}$ and, thus, $\lambda^2\beta\eta\pi \vdash \mathrm{N} = \mathrm{M}$. Similarly for $\lambda^2\beta\eta\pi$ w.r.t $\lambda^2\beta\eta$. □

B Properties of n-tuples

Notation B.1 (n-fold projections) *Let $M : A_1 \times \ldots \times A_n$ where the $A_i's$ have no occurrence of product or $\mathbf{T}$. We will then write $p_i^k M$ to note, if $k \leq n, 0 < i < k$, the term*

$$p_1 \underbrace{p_2 \ldots p_2}_{i-1} M,$$

and, if $k \leq n, 1 < i = k$, the term

$$\underbrace{p_2 \ldots p_2}_{i-1} M.$$

The idea behind this notation is that a sequence of binary projections can be considered as a projection over an n-tuple. Then p_i^k is a notation for the sequence of projections that selects the ith component in a k-tuple. Obviously, any n-tuple is also a k-tuple if $k \leq n$, so one could be tempted to drop the suffix k from p_i^k, but this is not correct: the k-th component of an n-tuple $\langle M_1, \ldots, M_n \rangle$ is M_k, that is not at all the same thing as the last component of the same term when considered as a k-tuple, which is $\langle M_k, \ldots, M_n \rangle$. The sequence of projections needed to select the ith component of a term considered as an n-tuple really depends both on i and k. So, we will drop the suffix k only when it is well understood from the context.

Remark B.2 (Simple projection arithmetics) *It is easy to check the following equalities:*

$$p_1 p_k^k M = p_k^{k+1} M \text{ if } k < n$$

$$p_2 p_k^k M = p_{k+1}^{k+1} M \text{ if } k < n$$

Lemma B.3 *Let M be a term in normal form and $w : A_1 \times \ldots \times A_n$, where the $A_i's$ have no occurrence of product or $\mathbf{T}$, be a variable not occurring free in it. Then $M[\overrightarrow{p_i^n w}/\overrightarrow{y_i}]$ can only contain Surjective Pairing redexes. During any reduction to normal form of this term the only created redexes are Surjective Pairing redexes.*

Proof. By inspection of the form of the redexes. □

Theorem B.4 *Let $w : A_1 \times \ldots \times A_n$, where the $A_i's$ have no occurrence of product or $\mathbf{T}$, be a variable not occurring free in a term Q in normal form. Then the following implications hold:*

1. *if $Q[\overrightarrow{p_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} p_k^{n'} w$, then $Q = \begin{cases} y_k & \text{if } k < n' \\ y_n & \text{if } k = n' = n \\ \langle y_{n'}, \ldots, y_n \rangle & \text{otherwise} \end{cases}$*
2. *if $Q[\overrightarrow{p_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} w$, then $Q = \langle y_1, \ldots, y_n \rangle$*

Proof. By induction on the structure of Q.

- **Q is a variable**
  1. if Q is not one of the $y's$, then the claim holds vacuously, else Q is some y_i and $Q[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}]$ is just $\mathrm{p}_i^n \mathrm{w}$, so that Q is as required
  2. holds vacuously
- **Q is an application $Q_1 Q_2$**
  The only new redexes created by the substitution are Surjective Pairing redexes, and their elimination cannot create any new β or η redexes(by Lemma B.3), so that there is no way to get rid of the outermost application and the claims hold vacuously.
- **Q is an abstraction $\lambda x.Q'$** The only way to get rid of the top-level abstraction in a reduction would be by means of an η reduction, but no such reductions are possible, due to Lemma B.3.
- **Q is a projection $\mathrm{p}_1 Q'$**
  1. for $\mathrm{p}_1 Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_k^{n'} w$, we have two possibilities:
     - $\mathrm{p}_k^{n'}$ is actually $\mathrm{p}_1 \mathrm{p}_k^k$ and $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_k^k w$. First notice that this implies that k is strictly smaller than n, otherwise the term $\mathrm{p}_1 \mathrm{p}_k^k w$ would not be well typed. Then, by inductive hypothesis 1, the term Q' is a pair $\langle y_k, \ldots, y_n \rangle$, hence Q is not in normal form, contradicting the hypothesis of the theorem.

– $\mathrm{p}_k^{n'}$ does not start with a first projection, or $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}]$ does not reduce to $\mathrm{p}_k^k w$. Then necessarily $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow}$ to a pair $\langle \mathrm{p}_k^{n'} w, \dots \rangle$, in order for $\mathrm{p}_1 Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}]$ to reduce to $\mathrm{p}_k^{n'} w$. But by Lemma B.3 in the reduction path there are only SP redexes, that can make disappear, but not create pairs. So Q' is already a product and Q is not in normal form, contradicting the hypothesis of the theorem.

2. if $\mathrm{p}_1 Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} w$, then necessarily $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \langle w, \dots \rangle$, so that, similarly as for the previous point, Q' is already a pair and Q in not in normal form contradicting the hypothesis of the theorem.

- **Q is a projection $\mathbf{p}_2 Q'$**

1. for $\mathrm{p}_2 Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_k^{n'} w$, we have two possibilities:
   – $\mathrm{p}_k^{n'}$ is actually $\mathrm{p}_{n'}^{n'}$ and $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_{n'-1}^{n'-1} w$. First notice that $n' - 1 < n' \leq n$. Then, by inductive hypothesis 1, the term Q' is a pair $\langle y_{n'-1}, \dots, y_n \rangle$, hence Q is not in normal form, contradicting the hypothesis of the theorem.
   – $\mathrm{p}_k^{n'}$ does not start with a second projection, or $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}]$ does not reduce to $\mathrm{p}_{n'-1}^{n'-1} w$. Then necessarily $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow}$ to a pair $\langle \dots, \mathrm{p}_{n'}^{n'} w \rangle$, in order for $\mathrm{p}_2 Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}]$ to reduce to $\mathrm{p}_k^{n'} w$. But Lemma B.3 tells us that in the reduction path there are only SP redexes, that can make disappear, but not create, pairs. So Q' is already a product and Q is not in normal form, contradicting the hypothesis of the theorem.

2. if $\mathrm{p}_2 Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} w$, then necessarily $Q'[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \langle \dots, w \rangle$, so that, similarly as for the previous point, Q' is already a pair and Q in not in normal form contradicting the hypothesis of the theorem.

- **Q is a pair $\langle Q_1, Q_2 \rangle$**

1. for $\langle Q_1, Q_2 \rangle[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_k^{n'} w$, we have two possibilities:
   – $k < n'$: then $\mathrm{p}_k^{n'} w = \mathrm{p}_1 \underbrace{\mathrm{p}_2 \dots \mathrm{p}_2}_{k-1} M = \mathrm{p}_k^n w$, so the type of $\mathrm{p}_k^{n'}$w is A_k, that does not contain products, and cannot be equal to a pair, whose type is a product.
   – $k = n'$: then $n' < n$ as $\mathrm{p}_n^n w$ cannot be equal to a pair, because its type A_n contains no product. Then,

$$Q_1[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_1 \mathrm{p}_{n'}^{n'} w = \mathrm{p}_{n'}^{n'+1} w = \mathrm{p}_{n'}^n w,$$

   and

$$Q_2[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_2 \mathrm{p}_{n'}^{n'} w = \mathrm{p}_{n'+1}^{n'+1} w,$$

   so by induction we have that

$$Q_1 = y_{n'}, \quad Q_2 = \langle y_{n'+1}, \dots, y_n \rangle$$

   and $Q = \langle y_{n'}, \dots, y_n \rangle$ as required.

2. if $\langle Q_1, Q_2 \rangle[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} w$ then necessarily $Q_1[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_1 w = \mathrm{p}_1^2 w$ and $Q_2[\overrightarrow{\mathrm{p}_i^n w}/\overrightarrow{y_i}] \overset{\beta^2\eta^2\pi^*}{\longrightarrow} \mathrm{p}_2 w = \mathrm{p}_2^2 w$, so by induction hypothesis

$$Q_1 = y_1, \quad Q_2 = \langle y_2, \dots, y_n \rangle$$

and

$$Q = \langle Q_1, Q_2 \rangle = \langle y_1, \ldots, y_n \rangle$$

as required.

- Q **is a second order application** $Q_1[B]$
  There is no way to reduce Q_1 to a second order abstraction by means of SP reductions (the only ones that are allowed by Lemma B.3), and it is not already one since Q is in normal form. So, there is no way to get rid of the second order application at the top level, and the two claims hold vacuously.

- Q **is a second order abstraction** $\lambda X.Q_1$
  Again, there is no way to get rid of the top-level second order abstraction, and the claims hold vacuously.

□

C Technical Lemmas

This appendix contains some technical lemmas and the proof of the Main Lemma 4.11. Before proceeding to the proof of the Main Lemma, let's study some properties of *DistInv*, in the following Lemma, and then some connections between types and the structure of terms in the lemmas that follow.

Lemma C.1 *Let* $DistInv(\overrightarrow{M}, \overrightarrow{V}, \overrightarrow{N}, \overrightarrow{W})$ *, let*

$$\begin{aligned}
\overrightarrow{M}_{types} &= \{M_i \mid M_i \textit{ is a type variable}\} \\
\overrightarrow{N}_{types} &= \{N_i \mid N_i \textit{ is a type variable}\} \\
\overrightarrow{V}_{types} &= \{V_i \mid V_i \textit{ is a type variable}\} \\
\overrightarrow{W}_{types} &= \{W_i \mid W_i \textit{ is a type variable}\}
\end{aligned}$$

and define

$$\begin{aligned}
m_{type} &= \mid \overrightarrow{M}_{types} \mid \\
n_{type} &= \mid \overrightarrow{N}_{types} \mid \\
v_{type} &= \mid \overrightarrow{V}_{types} \mid \\
w_{type} &= \mid \overrightarrow{W}_{types} \mid
\end{aligned}$$

then $m_{type} = n_{type} = v_{type} = w_{type}$.

Proof. First recall condition 3 of Definition 4.5: we have

$$M_i[\overrightarrow{N}/\overrightarrow{V}] = W_i \quad N_j[\overrightarrow{M}/\overrightarrow{W}] = V_j$$

for $\overrightarrow{V} = V_1, \ldots, V_m \subseteq FV(\overrightarrow{M})$ and $\overrightarrow{W} = W_1, \ldots, W_n \subseteq FV(\overrightarrow{N})$ sequences of variables, that can be either type variables X_i which are part of the list $\overrightarrow{M} = [M_1, \ldots, M_n]$ (resp. type variables Y_j from $\overrightarrow{N} = [N_1, \ldots, N_m]$), or term variables $x_i : A_i$ (resp. $y_i : B_i$).

Since in our calculus terms are not types, the members of the above equalities are either both types or both terms, so we can already establish that

$$m_{type} = w_{type} \quad \text{and} \quad n_{type} = v_{type}$$

Now types do not contain terms either, so when we specialize condition 3 of Definition 4.5 to type expressions we get

$$\overrightarrow{M}_{types}[\overrightarrow{Y}/\overrightarrow{V}_{types}] = \overrightarrow{W}_{types} \quad \text{and} \quad \overrightarrow{N}_{types}[\overrightarrow{X}/\overrightarrow{W}_{types}] = \overrightarrow{V}_{types}$$

Now, recall that the $\overrightarrow{M}_{types}$ and the $\overrightarrow{N}_{types}$ are simple type variables (Remark 4.6), and let's focus on the first equality: we know that no $X_i \in \overrightarrow{M}_{types}$ can be equal to a W_i, as W_i cannot be a free variable of $\overrightarrow{M}$, while X_i clearly is. This means that the substitution $[\overrightarrow{Y}/\overrightarrow{V}_{types}]$ must affect all the $\overrightarrow{V}_{types}$, and this can happen only if $\overrightarrow{V}_{types}$ includes $\overrightarrow{M}_{types}$, i.e. if

$$m_{type} = \mid \overrightarrow{M}_{types} \mid \leq \mid \overrightarrow{V}_{types} \mid = v_{type} = n_{type}$$

Analogously we get

$$n_{type} = \mid \overrightarrow{N}_{types} \mid \leq \mid \overrightarrow{W}_{types} \mid = w_{type} = m_{type}$$

Hence $m_{type} = n_{type} = v_{type} = w_{type}$. □

Lemma C.2 *Given $r\overrightarrow{P}$: C where r is a variable and C a type containing an occurrence of a product type $A \times B$ then either r:E and $D \times H$ occurs in E for some D, H or some $P_i = [E]$ and $D \times H$ occurs in E for some D, H.*

Proof. By induction on the length n of $\overrightarrow{P}$.
Base: for $n = 0$ then $r : C$, that contains the product type $A \times B$ by hypothesis.
Inductive step: the lemma holds for $n \leq k$, we prove it for $n = k + 1$
By cases on P_{k+1}:

- it is a term. Then (r $\overrightarrow{P_{1...k}}$)$P_{k+1}$:C and (r $\overrightarrow{P_{1...k}}$): $Type(P_{k+1}) \to C$ so we can apply the induction hypothesis and get either r:E and $D \times H$ occurs in E for some D, H or some $P_i = [E]$ and $D \times H$ occurs in E for some D, H for $i \leq k$, hence for $i \leq k + 1$ too.
- it is a type $[F]$. Then (r $\overrightarrow{P_{1...k}}$): $\forall X.T$ with T[F/X] = C. Now we have again two cases: either T does not contain any products, but then F must, or T contains products and we can apply induction as in the previous case.

□

Lemma C.3 *Given $r\overrightarrow{P}$: C where r is a variable and C a type containing an occurrence of* **T** *then either r:D and* **T** *occurs in D or some $P_i = [D]$ and* **T** *occurs in D.*

Proof. By induction on the length n of $\overrightarrow{P}$.

- (Base) for $n = 0$ then r:C and C contains **T** by hypothesis.
- (Inductive step) let the lemma be true for $n \leq k$ and prove it for $n = k + 1$.
  By cases on P_{k+1}:
  - it is a term. Then (r $\overrightarrow{P_{1...k}}$)$P_{k+1}$:C and (r $\overrightarrow{P_{1...k}}$): $Type(P_{k+1}) \to C$ so we can apply the induction hypothesis and get either r:D and **T** occurs in D or some $P_i = [D]$ and **T** occurs in D for $i \leq k$, hence for $i \leq k + 1$ too.
  - it is a type $[E]$. Then (r $\overrightarrow{P_{1...k}}$): $\forall X.T$ with T[E/X] = C. Now we have again two cases: either T does not contain **T**, but then E must, or T contains **T** and we can apply induction as in the previous case.

□

Lemma C.4 *Let $z{:}A_0 \vdash (z\ \overrightarrow{P_{1\ldots n}})$: C. If no product or* **T** *appears in the types A_0, C and in the P_i's that are types, then no product or* **T** *appears in the types of the P_i's that are terms either.*

Proof. By induction on n.
Base case: $n = 0$ trivial.
Inductive step: let the lemma be true for $n \leq k$ and prove it for $n = k + 1$
By cases on P_{k+1}:

- it is a term.

  By cases on the type E of P_{k+1}:

  - it has no occurrence of products or **T**: then the type of $(\mathrm{r}\ \overrightarrow{P_{1\ldots k}}) = E \to C$ has no occurrence of products or **T** too, so we can apply the induction hypothesis.
  - it has occurrences of products or **T**: this is impossible as $(\mathrm{r}\ \overrightarrow{P_{1\ldots k}})$: $E \to C$ contains products or **T** so by lemma C.2 or lemma C.3 either r has a type containing products or **T** or some P_i's that is a type must contain a product or **T**, in contradiction with the hypothesis.

- it is a type [E]. Then $(\mathrm{r}\ \overrightarrow{P_{1\ldots k}})$: $\forall X.T$ and in $\forall X.T$ there are no occurrences of products or **T** (as there are not in E by hypothesis and T[E/X] = C). So we can apply the induction hypothesis.

□

Lemma 4.11 (Main Lemma)
Let $\overrightarrow{N}$, $\overrightarrow{M}$ (with $m = \mid \overrightarrow{N} \mid$ and $n = \mid \overrightarrow{M} \mid$) be sequences of terms and/or types and $\circ$ and $\bullet$ be head-free substitutions such that $DistInv(\overrightarrow{M}, \overrightarrow{x}, \overrightarrow{N}, \overrightarrow{y}))^{\circ\bullet}$, then the following hold:

1. *the substitutions $\bullet$ and $\circ$ are idempotent, i.e. $\bullet\bullet = \bullet$ and $\circ\circ = \circ$*
2. *when M_i and N_j are terms,*

$$M_i = \lambda\overrightarrow{v_i}.x_{\sigma(i)}\overrightarrow{P_i}, \quad N_j = \lambda\overrightarrow{u_j}.y_{\pi(j)}\overrightarrow{Q_j}$$

   where $\sigma{:}n \to m$, $\pi{:}m \to n$ are integer functions s.t. $\sigma(\pi(i))=i$ if M_i is a term and $\pi(\sigma(j))=j$ if N_j is a term

3. *$n = m$*
4. *every P_{i_k} (Q_{j_h}) that is a type expression is just a type variable*
5. *every variable free in P_{i_k} (Q_{j_h}) has no occurrence of $\times$ or* **T** *in the type expressions occurring in it*
6. *P_{i_k} (Q_{j_h}) has no occurrence of $\times$ or* **T** *in its type if P_{i_k} (Q_{j_h}) is a term*
7. *Define $s_1 = \mid \overrightarrow{P}_i \mid, r_2 = \mid \overrightarrow{u}_{\sigma(i)} \mid, r_1 = \mid \overrightarrow{v}_i \mid, s_2 = \mid \overrightarrow{Q}_{\sigma(i)} \mid$. Without loss of generality, suppose $s_1 \geq r_2$. Then $r_1 \geq s_2$. Furthermore, $\overrightarrow{Q}_{\sigma(i)}$ can be extended with a sequence of type and/or term variables $[u''_1, \ldots, u''_{r_1-s_2}]$ to a sequence $\overrightarrow{Q'}_{\sigma(i)}$ such that*

   $DistInv(\overrightarrow{P}_i, \overrightarrow{v}_i, \overrightarrow{Q'}_{\sigma(i)}, \overrightarrow{u'}_{\sigma(i)})^{\triangleright\triangleleft}$ holds, where $\overrightarrow{u'}_{\sigma(i)} = \overrightarrow{u}_{\sigma(i)} \cup [u''_1, \ldots, u''_{r_1-s_2}]$ and $\triangleleft$, $\triangleright$ are suitable head-free substitutions.

Proof. We will show properties 1 - 7 in order, but we factor out here a remark that is needed along with most of the proof. We will very often make use of the Church-Rosser rewrite system associated to the calculus in order to exclude possible reductions. In all these cases the reduction (*gentop*) will not be possible, as in order to be applied to a term M *gentop* needs that M has a type in $Iso(\mathbf{T})$, which will never be the case in the following as no $\mathbf{T}$ type will be involved, and any type in $Iso(\mathbf{T})$ contains at least one occurrence of $\mathbf{T}$. We will not state this fact explicitly all the time.

1. is trivial due to the requirements 1 and 2 on the domain and codomain of the head-free substitutions $\circ$ and $\bullet$ in the definition 4.10 of $DistInv(,,,)$.

2. since the M_i in $\overrightarrow{M}$ and the N_i in $\overrightarrow{N}$ are terms are in normal form, we know that

$$M_i = \lambda \overrightarrow{v_i}.R_i \overrightarrow{P_i}$$

and

$$N_i = \lambda \overrightarrow{u_i}.S_i \overrightarrow{Q_i}$$

Where R_i, $\overrightarrow{P_i}$ (respectively S_i, $\overrightarrow{Q_i}$) are terms in normal form and R_i (respectively S_i) is not an abstraction nor an application (first or second order).

Now notice that R_i (respectively S_i)

- cannot be a pair $\langle R_{i_1}, R_{i_2}\rangle$, as otherwise, for typing reasons, $M_i = \lambda \overrightarrow{v_i}.\langle R_{i_1}, R_{i_2}\rangle$ has type $A \to \ldots \to B \times C$ that is not an arrow-only type;
- cannot be $*{:}\mathbf{T}$, as otherwise the type of the term would contain $\mathbf{T}$;
- cannot be a bound variable, as otherwise $M_i^\bullet[\overrightarrow{N}^\circ / \overrightarrow{x}]$ could not reduce to the free variable y_i. Indeed, if it is bound, then $M_i^\bullet[\overrightarrow{N}^\circ / \overrightarrow{x}]$ is $\lambda \overrightarrow{v_i}.R_i \overrightarrow{P^\bullet}_i[\overrightarrow{N}^\circ / \overrightarrow{x}]$, since a substitution affects only free variables. So R_i is still bound. Now, y_i is in normal form, and the equality $M_i^\bullet[\overrightarrow{N}^\circ / \overrightarrow{x}] = y_i$ can be turned, by the Church-Rosser property, into a reduction sequence $M_i^\bullet[\overrightarrow{N}^\circ / \overrightarrow{x}] \twoheadrightarrow y_i$. But no reduction rule can make disappear from a term in head normal form a bound variable that is in the head position.

We want to show that R_i cannot be a projection $\mathrm{p}_k\mathrm{Q}$ either. Suppose R_i is $\mathrm{p}_k\mathrm{Q}$ (k = 1,2), then

$$M_i = \lambda \overrightarrow{v_i}.\mathrm{p}_k Q P_{i_1} \ldots P_{i_l}$$

with Q in normal form. More precisely, $\mathrm{Q} = \mathrm{O}\overrightarrow{R_i}$, where O is in normal form and is not an abstraction or an application[8]. Again, O cannot be a pair $\langle O_1, O_2\rangle$ (as otherwise $\mathrm{Q} = \langle O_1, O_2\rangle$ and the subterm $\mathrm{p}_k\mathrm{Q} = \mathrm{p}_k\langle O_1, O_2\rangle$ of M_i would be a π redex, while we know that M_i is in n.f.) nor a bound variable (for the same reasons shown above for R_i). This argument can be iterated to show that

$$M_i = \lambda \overrightarrow{v_i}.\mathrm{p}_{k_1}(\ldots(\mathrm{p}_{k_z}(r\,\overrightarrow{O}_{z+1})\,\overrightarrow{O}_z)\ldots\,\overrightarrow{O}_1)P_{i_2}\ldots P_{i_l}$$

with r a free variable.

With the same argument we show that

$$N_i = \lambda \overrightarrow{u_i}.\mathrm{p}_{s_1}(\ldots(\mathrm{p}_{s_w}(s\,\overrightarrow{U}_{w+1})\,\overrightarrow{U}_w)\ldots\,\overrightarrow{U}_1)Q_{i_2}\ldots Q_{i_h}$$

Now we can use the property $\overrightarrow{M}_i^\bullet[\overrightarrow{N}^\circ / \overrightarrow{x}] = y_i$ to show that $r \in \overrightarrow{x}$ considering the following cases:

[8] If it is an abstraction, then the term is not in normal form, while if it is an application O_1O_2 we would take O_1 instead. Notice also that Q itself is not an abstraction for typing reasons.

(a) If $\bullet$ does not affect r, then r must be one of the $\overrightarrow{x}$, otherwise it is impossible to reduce the sequence of projections in front of r or (if the sequence is empty) to reduce $\overrightarrow{M}_i^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}]$ to a term not containing r, as r is the head variable in $\overrightarrow{M}_i^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}]$ too.

(b) Otherwise, notice that $\bullet$ is a head-free substitution and cannot, by hypothesis, generate any y_i, so that again $\overrightarrow{M}_i^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}]$ has a head free variable w that is not any y_i and cannot be erased by reductions, no matter if the sequence of projections in front of it is empty or not.

So r must be some $x_{\sigma(i)}$ for some integer function $\sigma : n \to m$. Analogously, s must be some $y_{\pi(i)}$ for some integer function $\pi : m \to n$. This means that

$$\overrightarrow{M}_i^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}] = \lambda\overrightarrow{v_i}.\mathrm{p}_{k_1}(\ldots(\mathrm{p}_{k_z}(\mathrm{N}^{\circ}_{\sigma(i)}\,\overrightarrow{O}^{\triangleright}_{z+1})\,\overrightarrow{O}^{\triangleright}_{z})\ldots\overrightarrow{O}^{\triangleright}_{1})\,\overrightarrow{P}^{\triangleright}_{i_2\ldots i_l}$$

where $\triangleright$ is the substitution $\bullet[\overrightarrow{N}^{\circ}/\overrightarrow{x}]$. But since

$$N_{\sigma(i)} = \lambda\overrightarrow{u_{\sigma(i)}}.\mathrm{p}_{s_1}(\ldots(\mathrm{p}_{s_w}(y_{\pi(\sigma(i))}\,\overrightarrow{U}_{w+1})\,\overrightarrow{U}_{w})\ldots\overrightarrow{U}_{1})\,\overrightarrow{Q}_{\sigma(i)_2\ldots\sigma(i)_h}$$

then if $|\,\overrightarrow{u_{\sigma(i)}}\,| \leq |\,\overrightarrow{O}^{\triangleright}_{z+1}\,|$ we have

$$\begin{aligned}
&\overrightarrow{M}_i^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}] = \\
&= \lambda\overrightarrow{v_i}.\mathrm{p}_{k_1}(\ldots \\
&\qquad(\mathrm{p}_{k_z}(((\mathrm{p}_{s_1}(\ldots(\mathrm{p}_{s_w}(y_{\pi(\sigma(i))}\,\overrightarrow{U}^{\circ\diamond}_{w+1})\,\overrightarrow{U}^{\circ\diamond}_{w})\ldots\overrightarrow{U}^{\circ\diamond}_{1}))\,\overrightarrow{Q}^{\circ\diamond}_{\sigma(i)_2\ldots\sigma(i)_h})\overrightarrow{O'}^{\triangleright}_{z+1})\,\overrightarrow{O}^{\triangleright}_{z}) \\
&\quad\ldots\overrightarrow{O}^{\triangleright}_{1})\,\overrightarrow{P}^{\triangleright}_{i_2\ldots i_l}
\end{aligned}$$

(where $\overrightarrow{O'}^{\triangleright}_{z+1}$ is what is left of $\overrightarrow{O}^{\triangleright}_{z+1}$ after the β reductions on $(\overrightarrow{N}^{\circ}_{\sigma(i)}\ \overrightarrow{O}^{\triangleright}_{z+1})$ and $\diamond$ is the substitution $[\overrightarrow{O}^{\triangleright}_{z+1}/\overrightarrow{u_{\sigma(i)}}]$)

Otherwise

$$\begin{aligned}
&\overrightarrow{M}_i^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}] = \\
&= \lambda\overrightarrow{v_i}.\mathrm{p}_{k_1}(\ldots \\
&\qquad(\mathrm{p}_{k_z}(\lambda\overrightarrow{u'}_{\sigma(i)}.((\mathrm{p}_{s_1}(\ldots(\mathrm{p}_{s_w}(y_{\pi(\sigma(i))}\,\overrightarrow{U}^{\circ\diamond}_{w+1})\,\overrightarrow{U}^{\circ\diamond}_{w})\ldots\overrightarrow{U}^{\circ\diamond}_{1}))\,\overrightarrow{Q}^{\circ\diamond}_{\sigma(i)_2\ldots\sigma(i)_h}))\,\overrightarrow{O}^{\triangleright}_{z}) \\
&\quad\ldots\overrightarrow{O}^{\triangleright}_{1})\,\overrightarrow{P}^{\triangleright}_{i_2\ldots i_l}
\end{aligned}$$

(where $\overrightarrow{u'}_{\sigma(i)}$ is what is left of $\overrightarrow{u}_{\sigma(i)}$ after the β reductions on $(\overrightarrow{N}^{\circ}_{\sigma(i)}\ \overrightarrow{O}^{\triangleright}_{z+1})$ (actually, for typing reasons, this sequence must be empty: we have a projection p_{k_z} applied to this term, that cannot have, then, a functional type!) and $\diamond$ is the substitution $[\overrightarrow{O}^{\triangleright}_{z+1}/\overrightarrow{u_{\sigma(i)}}]$)

In any case, these terms must be equal to y_i, which is a normal form, and equality can be turned into reduction due to the Church-Rosser property. Now, this reduction is possible iff $z = w = 0$ and $y_{\pi(\sigma(i))} = y_i$, i.e. $\pi(\sigma(i)) = i$: there is no reduction rule that allows us to get rid of the sequence of projections, because they are blocked by the variable $y_{\pi(\sigma(i))}$ (that is free and is already in normal form). So r_i must be $x_{s(i)}$ and $\pi(\sigma(i)) = i$ for i s.t. M_i is a term. Analogously we can proceed for $\overrightarrow{N}$ and get that s_j must be $y_{\pi(j)}$ and $\sigma(\pi(j)) = j$ for j s.t. N_j is a term.

3. Since we already know that the number of type variables in $\overrightarrow{M}$, $\overrightarrow{N}$, $\overrightarrow{x}$ and $\overrightarrow{y}$ is equal (Lemma C.1) it is possible to extend the previous functions π and σ in order to get $\pi(\sigma(i)) = i$ and $\sigma(\pi(j)) = j$ for all i and j: just let σ map indexes of different type variables in $\overrightarrow{M}$ to indexes of different type variables in $\overrightarrow{x}$ and π be the inverse of σ on the indexes of type variable of $\overrightarrow{N}$.

   Namely, from Lemma C.1 we obtain also that the type variables in both $\overrightarrow{M}$ and $\overrightarrow{N}$ are just a permutation of each other, so that there exists $\sigma_{type}{:}n \to m$, $\pi_{type}{:}m \to n$ s.t. $\sigma_{type}(\pi_{type}(i)) = i$ if M_i is a type variable and $\pi_{type}(\sigma_{type}(j)) = j$ when N_j is a type variable, hence we can define $\sigma' : n \to m$, $\pi' : m \to n$ as

   - $\sigma'(i) = \sigma(i)$ if M_i is a term
   - $\sigma'(i) = \sigma_{type}(i)$ if M_i is a type variable
   - $\pi'(i) = \pi(j)$ if N_j is a term
   - $\pi'(i) = \pi_{type}(j)$ if N_j is a type variable

   with the property that $\pi'(\sigma'(i)) = i$ and $\sigma'(\pi'(j)) = j$ for all i and j, as every M_i and N_j is either a term or a type variable.

   Due to well known properties of permutations, this entails $n = m$ and furthermore π' and σ' are permutations that are each other's inverses. This says, besides, that the number of terms is the same in both sequences.

4. let's consider again $\overrightarrow{M}^{\bullet}_i\ [\overrightarrow{N}^{\circ}/\overrightarrow{x}]$. Now we know from 3 and the proof of 2 that

$$
\begin{aligned}
&\overrightarrow{M}^{\bullet}_i[\overrightarrow{N}^{\circ}/\overrightarrow{x}] = \\
&\quad= \lambda\overrightarrow{v_i}.\mathrm{N}^{\circ}_{\sigma(i)}\mathrm{P}^{\triangleright}_{i_2}\dots\mathrm{P}^{\triangleright}_{i_l} \text{ (where } \triangleright \text{ is } \bullet[\overrightarrow{N}^{\circ}/\overrightarrow{x}]) \\
&\quad= \lambda\overrightarrow{v_i}.(\lambda\overrightarrow{u_{\sigma(i)}}.y_{\pi(\sigma(i))}\,\overrightarrow{Q}^{\circ}_{\sigma(i)})\mathrm{P}^{\triangleright}_{i_2}\dots\mathrm{P}^{\triangleright}_{i_l} \\
&\quad= \lambda\overrightarrow{v_i}.(\lambda\overrightarrow{u_{\sigma(i)}}.y_i\,\overrightarrow{Q}^{\circ}_{\sigma(i)})\mathrm{P}^{\triangleright}_{i_2}\dots\mathrm{P}^{\triangleright}_{i_l} \\
&\quad= \begin{cases} \lambda\overrightarrow{v_i}.(\lambda\overrightarrow{u'_{\sigma(i)}}.y_i\,\overrightarrow{Q}^{\circ}_{\sigma(i)})[\mathrm{P}^{\triangleright}_{i_2}\dots\mathrm{P}^{\triangleright}_{i_l}/\overrightarrow{u - u'_{\sigma(i)}}] & \text{if } \mid \overrightarrow{u_{\sigma(i)}} \mid > l-1 \\ \lambda\overrightarrow{v_i}.((y_i\,\overrightarrow{Q}^{\circ}_{\sigma(i)})[\mathrm{P}^{\triangleright}_{i_2}\dots\mathrm{P}^{\triangleright}_{i_{l'}}/\overrightarrow{u_{\sigma(i)}}])\mathrm{P}^{\triangleright}_{i_{l'+1}}\dots\mathrm{P}^{\triangleright}_{i_l} & \text{otherwise} \end{cases}
\end{aligned}
$$

   Some care is needed in checking the last equality: in the case $\mid \overrightarrow{u}_{\sigma(i)} \mid > l-1$, we named $\overrightarrow{u'}_{\sigma(i)}$ the abstracted variables that are left after the β reductions, and the notation $[\mathrm{P}^{\triangleright}_{i_2}\dots\mathrm{P}^{\triangleright}_{i_l}/\overrightarrow{u - u'_{\sigma(i)}}]$ is a shorthand to indicate that the substitution is performed on the first $l-1$ variables of $\overrightarrow{u}$.
   The only way to reduce both expressions to y_i is a series of η reductions. This means that every Q_i type expression must be a type variable. The same can be shown for P_i considering $\overrightarrow{N}^{\circ}_i[\overrightarrow{M}^{\bullet}/\overrightarrow{y}]$ instead.

5. Any free variable in any P_i is either free in $\overrightarrow{M}$ (and then the claim holds by hypothesis) or is one of the $\overrightarrow{v_i}$. In the second case, notice that the type of the $\overrightarrow{M}$ contains as type subexpressions the type of the $\overrightarrow{v_i}$, so that it cannot contain any product or $\mathbf{T}$ either.

6. By Lemma C.4 (lemma on type of sequences not including complex type expressions) and properties 4 and 5 (recall that $\mathbf{T}$ is a type constant, not a variable)

7. w.l.o.g., let $s_1(= \mid \overrightarrow{P}_i \mid) \geq r_2(= \mid \overrightarrow{u}_{\sigma(i)} \mid)$. Notice that this implies $r_1(= \mid \overrightarrow{v}_i \mid) \geq s_2(= \mid \overrightarrow{Q}_{\sigma(i)} \mid)$, as by inspecting both cases in the proof of 4 we get $r_1 + r_2 = s_1 + s_2$ (due to

the η reductions that must occurr). Then $\overrightarrow{Q}_{\sigma(i)}$ can be extended with a sequence of type and/or term variables $[u''_1, \ldots, u''_{r_1-s_2}]$ to a $\overrightarrow{Q'}_{\sigma(i)}$ so that (we will drop the suffix σ(i) now for clarity)

$$\begin{aligned}
&(\overrightarrow{Q}^{\circ}_{1\ldots s_2}[\overrightarrow{P}^{\triangleright}_{1\ldots r_2}/\overrightarrow{u}_{1\ldots r_2}])\,\overrightarrow{P}^{\triangleright}_{r_2+1\ldots s_1} = \\
&= (\overrightarrow{Q}^{\circ}_{1\ldots s_2}[\overrightarrow{P}^{\triangleright}_{1\ldots r_2}/\overrightarrow{u}_{1\ldots r_2}])\overrightarrow{u''}_{1\ldots r_1-s_2}[\overrightarrow{P}^{\triangleright}_{r_2+1\ldots s_1}/\overrightarrow{u''}_{1\ldots r_1-s_2}] \\
&= \overrightarrow{Q'}^{\circ}[\overrightarrow{P}^{\triangleright}_{1\ldots r_2}/\overrightarrow{u}_{1\ldots r_2}, \ldots, \overrightarrow{P}^{\triangleright}_{r_2+1\ldots s_1}/\overrightarrow{u''}_{1\ldots r_1-s_2}] \\
&= \overrightarrow{Q'}^{\circ}[\overrightarrow{P}^{\triangleright}_{1\ldots s_1}/\overrightarrow{u'}_{1\ldots s_1}] \\
&= v_1, \ldots, v_{r_1}
\end{aligned}$$

while

$$\overrightarrow{P}^{\bullet}[\overrightarrow{Q}^{\triangleleft}_{1\ldots s_2}u''_1, \ldots, u''_{r_1-s_2}/\overrightarrow{v}_{1\ldots r_1}] = u_1, \ldots, u_{r_2}, u'_1, \ldots, u'_{r_1-s_2}$$

where $\triangleleft$ is the substitution $\circ[\overrightarrow{M}^{\bullet}/\overrightarrow{y}]$ that plays the symmetric role of $\triangleright$ on the $\overrightarrow{Q}$.

Now notice that $[\overrightarrow{M}^{\bullet}/\overrightarrow{y}]$ does not affect any u's and does not create any v's, since by the variable convention they are not free in the M's. Symmetrically, $[\overrightarrow{N}^{\circ}/\overrightarrow{x}]$ does not affect the v's and does not create any u's, so we get also:

$$\begin{aligned}
\overrightarrow{Q'}^{\triangleleft}[\overrightarrow{P}^{\triangleright}_{1\ldots s_1}/\overrightarrow{u'}_{1\ldots s_1}] &= \overrightarrow{Q'}^{\circ}[\overrightarrow{M}^{\bullet}/\overrightarrow{y}][\overrightarrow{P}^{\triangleright}_{1\ldots s_1}/\overrightarrow{u'}_{1\ldots s_1}] \\
&= v_1, \ldots, v_{r_1}
\end{aligned}$$

and

$$\begin{aligned}
\overrightarrow{P}^{\triangleright}[\overrightarrow{Q}^{\triangleleft}_{1\ldots s_2}u''_1, \ldots, u''_{r_1-s_2}/\overrightarrow{v}_{1\ldots r_1}] &= \overrightarrow{P}^{\bullet}[\overrightarrow{N}^{\circ}/\overrightarrow{x}][\overrightarrow{Q}^{\triangleleft}_{1\ldots s_2}u''_1, \ldots, u''_{r_1-s_2}/\overrightarrow{v}_{1\ldots r_1}] \\
&= u_1, \ldots, u_{r_2}, u''_1, \ldots, u''_{r_1-s_2} \\
&= u'_1, \ldots, u'_{s_1}
\end{aligned}$$

so that $\mathit{DistInv}(\overrightarrow{P_i}, \overrightarrow{v_i}, \overrightarrow{Q'}_{\sigma(i)}, \overrightarrow{u'}_{\sigma(i)})^{\triangleright\triangleleft}$.

This concludes the proof of the main lemma. □

D Miscellanea

Proposition 3.2 *Each type has a unique type normal form in R.*

Proof. Here is a direct proof of the Proposition.

We show that R is a strongly normalizing Church-Rosser rewriting system. Since it is straightforward to show that the system is weakly Church-Rosser, it is enough to show SN (as, due to the well known Newman's lemma, WCR + SN ⇒ CR).

We will show strong normalization by exibiting a measure that strictly decreases each time one of the rewriting rules is applied to a formula. Let h be a complexity measure on formulae defined as follows:

$$\begin{aligned}
h(A) &= 3 \text{ if } A \text{ is atomic} \\
h(A \times B) &= h(A) * h(B)^2 + 1 \\
h(A \to B) &= h(B)^{h(A)} \\
h(\forall X.A) &= 2^{h(A)}
\end{aligned}$$

It is obvious that this measure is an integer always greater than 2. Then it is easy to show by induction that $h(C[A]) > h(C[A'])$ if $h(A) > h(A')$, where C[ ] is an arbitrary context. To show that every rewriting decreases h, it suffices now to show that every reduction rule in R strictly decreases this measure. Since 5, 6, 7, 8 and 9 trivially decrease h, we will focus only on 1, 2, 3 and 4.

$$\begin{aligned}
h(A \times (B \times C)) &= h(A) * (h(B \times C))^2 + 1 \\
&= h(A) * (h(B) * h(C)^2 + 1)^2 + 1 \\
&> h(A) * h(B)^2 * h(C)^4 + 1 \\
&= (h(A) * h(B)^2 * h(C)^2) * h(C)^2 + 1 \\
&> h(A) * h(B)^2 * h(C)^2 + h(C)^2 + 1 \\
&\qquad \text{as } h(C)^2 > 2 \text{ and } h(A) * h(B)^2 * h(C)^2 > 2 \\
&= (h(A) * h(B)^2 + 1) * h(C)^2 + 1 \\
&= h(A \times B) * h(C)^2 + 1 \\
&= h((A \times B) \times C).
\end{aligned}$$

Similarly,

$$\begin{aligned}
h((A \to B) \times C) &= h(C)^{(h(A) * h(B)^2 + 1)} \\
&> h(C)^{(h(A) * h(B))} \\
&= h(A \to (B \to C))
\end{aligned}$$

$$\begin{aligned}
h(A \to (B \times C)) &= (h(B) * h(C)^2 + 1)^{h(A)} \\
&> (h(B) * h(C)^2)^{h(A)} + 1 \\
&\qquad \text{as } h(A) > 2 \text{ and } h(B) * h(C)^2 > 2 \\
&= h(B)^{h(A)} * h(C)^{2 * h(A)} + 1 \\
&= h(B)^{h(A)} * (h(C)^{h(A)})^2 + 1 \\
&= h((A \to B) \times (A \to C))
\end{aligned}$$

$$\begin{aligned}
h(\forall X.A \times B) &= 2^{(h(A) * h(B)^2 + 1)} \\
&= 2 * 2^{h(A) * h(B)^2} \\
&= 2 * 2^{h(A)} * 2^{h(B)^2} \\
&\geq 2^{h(A)} * 2^{h(B)^2} + 1 \\
&\qquad \text{as } 2^{h(A)} * 2^{h(B)^2} \geq 1 \\
&> 2^{h(A)} * 2^{2 * h(B)} + 1 \\
&\qquad \text{as } h(B)^2 > 2 * h(B) \text{ since } h(B) > 2 \\
&= 2^{h(A)} * (2^{h(B)})^2 + 1 \\
&= h(\forall X.A \times \forall X.B)
\end{aligned}$$

□

Proposition D.1 *If A is a type not containing* **T**, *then there is no invertible term M of type* $A \to \mathbf{T}$. *Hence* **T** *is not definably isomorphic to any such type A.*

Proof. We can assume M is in normal form with respect to the confluent notion of reduction introduced in A.7. Due to its type, M is λx:A.∗, and we show that no term N in normal form

can be its inverse, considering the possible normal forms of N that would be compatible with its type. There are five cases:

- N is not x:$\mathbf{T} \to A$, as

$$\begin{aligned} x{\circ}M &= \lambda w : A.(x(Mw)) \\ &= \lambda w : A.(x((\lambda x : A.*)w)) \\ &= \lambda w : A.(x*) \end{aligned}$$

  that is in normal form and is not the identity of type A.

- N is not λx:$\mathbf{T}$.P, as

$$\begin{aligned} (\lambda x : \mathbf{T}.P){\circ}M &= \lambda w : A.((\lambda x : \mathbf{T}.P)(Mw)) \\ &= \lambda w : A.((\lambda x : \mathbf{T}.P)((\lambda x : A.*)w)) \\ &= \lambda w : A.((\lambda x : \mathbf{T}.P)*) \\ &= \lambda w : A.(P[*/x]) \\ &= \lambda w : A.P \text{ as } \lambda x : \mathbf{T}.P \text{ is in normal form, hence } x{\notin}FV(P) \\ &= \quad \text{(as otherwise a reduction } x : \mathbf{T} \xrightarrow{\mathbf{T}} * \text{ could apply).} \end{aligned}$$

  Since P is normal, λw:A.P is normal too, and due to the variable convention P cannot be w, so that λw:A.P is not the identity of type A.

- N is not (PQ), as

$$\begin{aligned} (PQ){\circ}M &= \lambda w : A.((PQ)(Mw)) \\ &= \lambda w : A.((PQ)((\lambda x : A.*)w)) \\ &= \lambda w : A.((PQ)*) \end{aligned}$$

  that is in normal form, as (PQ) is and ((PQ)$*$) is not a gentop redex as its type is A, that contains no occurrence of $\mathbf{T}$, and is not the identity of type A.

- N is not $\lambda X.P$ for typing reasons.

- N is not $P[B]$ for any type B, as

$$\begin{aligned} (P[B]){\circ}M &= \lambda w : A.((P[B])(Mw)) \\ &= \lambda w : A.((P[B])((\lambda x : A.*)w)) \\ &= \lambda w : A.((P[B])*) \end{aligned}$$

  that is in normal form, as $P[B]$ is and $((P[B])*)$ is not a *gentop* redex as its type is A, that contains no occurrence of $\mathbf{T}$, and is not the identity of type A.

It follows that M as in the hypothesis has no inverse. □