Isomorphism of "Functional" Intersection Types[*]

Mario Coppo, Mariangiola Dezani-Ciancaglini, Ines Margaria, and Maddalena Zacchi

Dipartimento di Informatica, Università di Torino
corso Svizzera 185, 10149 Torino, Italy
`{coppo, dezani, ines, zacchi}@di.unito.it`

Abstract

Type isomorphism for intersection types is quite odd, since it is not a congruence and it does not extend type equality in the standard interpretation of types. The lack of congruence is due to the proof theoretic nature of the intersection introduction rule, which requires the same term to be the subject of both premises. A partial congruence can be recovered by introducing a suitable notion of type similarity. Type equality in standard models becomes included in type isomorphism whenever atomic types have "functional" interpretations, i.e. they are equivalent to arrow types. This paper characterises type isomorphism for a type system in which the equivalence between atomic types and arrow types is induced by the initial projections of the Scott D_∞ model via the correspondence between inverse limit models and filter λ-models.

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.3.3 Studies of Program Constructs, D.1.1 Applicative (Functional) Programming

Keywords and phrases type isomorphism, lambda calculus, intersection types

Digital Object Identifier 10.4230/LIPIcs.TYPES.2013.129

1 Introduction

The notion of *type isomorphism* is a particularisation of the general notion of isomorphism as defined in category theory. Two objects a and b are *isomorphic* if there exist two morphisms $f : a \to b$ and $g : b \to a$ such that $f \circ g = id_b$ and $g \circ f = id_a$:

$$\begin{tikzcd} a \arrow[r, bend left, "f"] \arrow[loop left, "id_a"] & b \arrow[l, bend left, "g"] \arrow[loop right, "id_b"] \end{tikzcd}$$

Analogously, two *types* σ and τ in some typed λ-calculus, are *isomorphic* if there are two λ-terms f and g of types $\sigma \to \tau$ and $\tau \to \sigma$, respectively, such that $f \circ g$ is $\beta\eta$ equal to the identity at type τ and $g \circ f$ is $\beta\eta$ equal to the identity at type σ.

In a recent paper [15], isomorphic types are identified. So λ-terms getting a type σ have also all types isomorphic to σ. This is useful both in looking for proofs of formulas through the Curry-Howard correspondence and in searching functions by type in program libraries.

Bruce and Longo proved in [5] that only one equation, namely the **swap** equation:

$$\sigma \to \tau \to \rho \;\approx\; \tau \to \sigma \to \rho$$

is needed for characterising isomorphism in the simply typed λ-calculus.

[*] This work was partially supported by EU Collaborative project ASCENS 257414, ICT COST Action IC1201 BETTY, MIUR PRIN Project CINA Prot. 2010LHT4KM and Torino University/Compagnia San Paolo Project SALT.

19th International Conference on Types for Proofs and Programs (TYPES 2013).
Editors: Ralph Matthes and Aleksy Schubert; pp. 129–149
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Later, the study has been directed toward richer λ-calculi, obtained from the simply typed λ-calculus in an incremental way, by adding some other type constructors (like product [22, 4, 23]) or by allowing higher-order types (System F [5, 13]). Di Cosmo summarised in [14] the equations characterising type isomorphisms in different type systems. The set of equations grows incrementally in the sense that the set of equations for a typed λ-calculus, obtained by adding a primitive to a given typed λ-calculus, is an extension of the set of equations of the λ-calculus without that primitive.

In the presence of intersection, this incremental approach does not work, as pointed out in [12]; the isomorphism is no longer a congruence and type equality in the standard models of intersection types does not entail type isomorphism. Notice that both features hold also in the very tricky case of the sum types [17].

The lack of congruence can be shown considering, for instance, the types $\varphi_1 \to \varphi_2 \to \sigma$ and $\varphi_2 \to \varphi_1 \to \sigma$. They are isomorphic (by argument swapping), while their intersections with the same type $(\varphi_3 \to \varphi_4 \to \tau)$, i.e.

$$(\varphi_1 \to \varphi_2 \to \sigma) \wedge (\varphi_3 \to \varphi_4 \to \tau) \text{ and } (\varphi_2 \to \varphi_1 \to \sigma) \wedge (\varphi_3 \to \varphi_4 \to \tau),$$

are not. It is interesting to note that the lack of congruence prevents to give a finitary axiomatisation of the type isomorphism studied in this paper.

The standard models of intersection types map types to subsets of any domain that is a model of the untyped λ-calculus, with the conditions that the arrow is interpreted as the function space constructor and the intersection as the set-theoretic intersection [2]. For example, $\sigma \wedge \tau \to \rho$ is isomorphic (and equal in all standard models) to $\tau \wedge \sigma \to \rho$, but they are no longer isomorphic when intersected with an atomic type φ, i.e. $(\sigma \wedge \tau \to \rho) \wedge \varphi$ is not isomorphic to $(\tau \wedge \sigma \to \rho) \wedge \varphi$ (although their interpretations remain equal).

In place of congruence one can use a suitable notion of type similarity, as done in [12] for characterising isomorphism. Instead, the existence of non-isomorphic types, which are equal in all standard interpretations, reveals that the type assignment system considered in [12] can be improved. The problem is caused by the absence of a functional behaviour for atomic types. This is quite odd for the pure λ-calculus, where everything is a function.

The present paper proposes a type system whose isomorphisms contain type equality in standard intersection models. This is achieved by assuming that each atomic type is equivalent to a functional type. In particular the type system is sound for a type interpretation in which each atomic type is interpreted as the set of constant functions returning values belonging to the set itself. Notably, this choice takes inspiration from the properties of the standard projections of Scott's D_∞ λ-model [21] and from the relations between inverse limit models and filter models [6]. As proved in [6], in fact, D_∞ is isomorphic to a filter λ-model built from a set of atomic types which correspond to elements of the initial domain D_0. In this model a type interpretation in which all types have a functional character is obtained in a natural way by taking the open sets in the Scott topology.

A strongly related paper is [9], where the functional interpretation of atomic types is considered in a type system with also union types. Type similarity is extended to union types and proved to be sound for type isomorphism, while its completeness is only conjectured.

Summary. Section 2 presents the type assignment system with its properties, notably Subject Reduction and Subject Expansion. Section 3 discusses some basic isomorphisms which entail equality in standard models (Theorem 18). The main result of this paper is the characterisation of type isomorphism (Theorem 37) given in Section 5, using the type normalisation presented in Section 4. As a consequence type isomorphism turns out to be decidable (Theorem 39). Section 6 concludes with some directions for further studies.

2 Type Assignment System

Let $\mathtt{A}$ be a denumerable set of atomic types ranged over by φ, ψ and ω an atom not in $\mathtt{A}$. The syntax of types is given by:

$$\sigma \quad ::= \quad \varphi \quad | \; \omega \quad | \; \sigma \to \sigma \; | \; \sigma \wedge \sigma \,.$$

As usual, parentheses are omitted according to the precedence rule " $\wedge$ over $\to$" and $\to$ associates to the right. It is useful to distinguish between different kinds of types. So in the following:

- σ, τ, ρ, θ range over arbitrary types;
- α, β, γ range over atomic and arrow types, defined as $\alpha \; ::= \varphi \; | \; \omega \; | \; \sigma \to \sigma$.

The following equivalence asserts the functional character of atomic types, by equating them to arrow types. It also agrees with the interpretation of type ω as the whole domain of elements (see Definition 17).

▶ **Definition 1** (Semantic type equivalence)**.** The *semantic equivalence relation* $\cong$ *on types* is defined as the minimal congruence such that:

$$\varphi \cong \omega \to \varphi \qquad \omega \cong \omega \to \omega \qquad \sigma \cong \sigma \wedge \omega \qquad \sigma \cong \omega \wedge \sigma.$$

The congruence allows one to state that $\sigma \cong \sigma'$ and $\tau \cong \tau'$ imply $\sigma \wedge \tau \cong \sigma' \wedge \tau'$. Moreover $\sigma \to \tau \cong \sigma' \to \tau'$ iff $\sigma \cong \sigma'$ and $\tau \cong \tau'$. Note that no other equivalence is assumed between types, for instance $\sigma \wedge \tau$ is different from $\tau \wedge \sigma$.

The equivalence of Definition 1 is dubbed semantic since it is derived by the relation between D_∞ λ-models and filter λ-models, see [1] and [3] (Section 16.3). Briefly, each inverse limit model built from an ω-algebraic lattice D_0 with order $\sqsubseteq$ is isomorphic to a filter λ-model with subtyping $\leq_\infty$ when:

- the intersections of atomic types are in one-to-one correspondence γ with the compact elements of D_0 (ω corresponds to $\bot$);
- each type corresponds to a compact element of D_∞;
- each arrow type corresponds to a step function between compact elements of D_∞;
- each intersection type corresponds to the join between compact elements of D_∞;
- the subtype relation $\leq_\infty$ mimics
  - the (reverse) partial order on the compact elements of D_0, i.e. $d, d' \in D_0$ and $d \sqsubseteq d'$ imply $\gamma^{-1}(d') \leq_\infty \gamma^{-1}(d)$, and
  - the initial projection from the compact elements of D_0 to the set of continuous functions mapping D_0 in D_0, i.e. if $d \in D_0$ is mapped to the step function $d_1 \Rightarrow d_2$, then $$\gamma^{-1}(d) \leq_\infty \gamma^{-1}(d_1) \to \gamma^{-1}(d_2) \leq_\infty \gamma^{-1}(d).$$

The standard initial projection ι of Scott's model [21] maps each element of D_0 in the constant function returning that element, i.e. $\iota(d)$ is equal to the step function $\bot \Rightarrow d$ for all $d \in D_0$ (including $d = \bot$). It is then easy to verify that the first two equivalences of Definition 1 are induced by associating $\bot$ with type ω, by taking as D_0 the lattice obtained by join completion of a domain with a denumerable set of incomparable elements (corresponding to the types in $\mathtt{A}$) and by using the standard initial projection. The last two equivalences of Definition 1 agree with the facts that $\bot$ is the least element of D_∞ and that the intersection corresponds to the join.

In the type assignment system considered in this paper, types can be assigned only to linear λ-terms. A λ-term is *linear* if each free or bound variable occurs exactly once in it.

$$(Ax)\quad x\!:\!\sigma \vdash x\!:\!\sigma \qquad\qquad (\cong)\quad \frac{\Gamma \vdash M\!:\!\sigma \quad \sigma \cong \tau}{\Gamma \vdash M\!:\!\tau}$$

$$(\to I)\quad \frac{\Gamma, x\!:\!\sigma \vdash M\!:\!\tau}{\Gamma \vdash \lambda x.M\!:\!\sigma \to \tau} \qquad\qquad (\to E)\quad \frac{\Gamma_1 \vdash M\!:\!\sigma \to \tau \quad \Gamma_2 \vdash N\!:\!\sigma}{\Gamma_1, \Gamma_2 \vdash MN\!:\!\tau}$$

$$(\wedge I)\quad \frac{\Gamma \vdash M\!:\!\sigma \quad \Gamma \vdash M\!:\!\tau}{\Gamma \vdash M\!:\!\sigma \wedge \tau} \qquad\qquad (\wedge E)\quad \frac{\Gamma \vdash M\!:\!\sigma \wedge \tau}{\Gamma \vdash M\!:\!\sigma} \qquad \frac{\Gamma \vdash M\!:\!\sigma \wedge \tau}{\Gamma \vdash M\!:\!\tau}$$

Figure 1 Typing rules.

This is justified by the observation that type isomorphisms are realised by finite hereditarily permutators which are linear λ-terms (see Definitions 10 and 12). This is not restrictive since it is easy to prove that the full system without linearity restriction [6] is conservative over the present one. Therefore the types that can be derived for the finite hereditarily permutators are the same in the two systems, so the present study of type isomorphism holds for the full system too.

Figure 1 gives the typing rules. As usual, *environments* associate variables to types and contain at most one type for each variable. The environments are relevant, i.e. they contain only the used premises. The domain of the environment Γ is denoted by $dom(\Gamma)$. When writing Γ_1, Γ_2 one convenes that $dom(\Gamma_1) \cap dom(\Gamma_2) = \emptyset$. It is easy to verify that $\Gamma \vdash M\!:\!\sigma$ implies $dom(\Gamma) = FV(M)$ ($FV(M)$ denotes the set of free variables of M). An example of derivation is shown in Figure 2.

Some useful admissible rules are:

$$(L)\quad \frac{x\!:\!\sigma \vdash x\!:\!\tau \quad \Gamma, x\!:\!\tau \vdash M\!:\!\rho}{\Gamma, x\!:\!\sigma \vdash M\!:\!\rho} \qquad\qquad (\omega)\quad \frac{dom(\Gamma) = FV(M)}{\Gamma \vdash M\!:\!\omega}$$

In order to state and prove the Inversion Lemma (Lemma 4) it is handy to introduce a pre-order on types (Definition 2), which is induced by the typing rules (Lemma 3(2)).

▶ **Definition 2** (Identity pre-order on types). **1.** *The set $\mathcal{A}$ of atomic and arrow types of a type σ* (notation $\mathcal{A}(\sigma)$) is inductively defined by:

$$\mathcal{A}(\alpha) = \{\alpha, \omega\} \qquad \mathcal{A}(\sigma \wedge \tau) = \mathcal{A}(\sigma) \cup \mathcal{A}(\tau)$$

2. *The identity pre-order relation* $\precsim$ on types is defined by:

$$\sigma \precsim \tau \text{ if for all } \alpha \in \mathcal{A}(\tau) \text{ there is } \beta \in \mathcal{A}(\sigma) \text{ such that } \beta \cong \alpha.$$

It is easy to verify that $\sigma \precsim \omega$ and $\sigma \precsim \omega \to \omega$ for all types σ. Clearly, whereas $\sigma \cong \tau$ implies $\sigma \precsim \tau$, the inverse does not hold since for example $\varphi \wedge \psi \precsim \omega \to \varphi$, but $\varphi \wedge \psi \not\cong \omega \to \varphi$.

▶ **Lemma 3.** **1.** *$\Gamma \vdash M\!:\!\sigma$ iff $\Gamma \vdash M\!:\!\alpha$ for all $\alpha \in \mathcal{A}(\sigma)$.*
2. *If $\Gamma \vdash M\!:\!\sigma$ and $\sigma \precsim \tau$, then $\Gamma \vdash M\!:\!\tau$.*

Proof. (1). By structural induction on σ. If $\sigma = \alpha$ and $\Gamma \vdash M : \sigma$, the rule (ω) derives $\Gamma \vdash M\!:\!\omega$. Let $\sigma = \sigma_1 \wedge \sigma_2$. By rules $(\wedge I)$ and $(\wedge E)$ $\Gamma \vdash M\!:\!\sigma$ iff $\Gamma \vdash M\!:\!\sigma_1$ and $\Gamma \vdash M\!:\!\sigma_2$, so the induction hypothesis applies.
(2). By definition for all $\alpha \in \mathcal{A}(\tau)$ there is $\beta \in \mathcal{A}(\sigma)$ such that $\beta \cong \alpha$. Point (1) implies that $\Gamma \vdash M\!:\!\beta$ for all $\beta \in \mathcal{A}(\sigma)$. Then by rule $(\cong)$ $\Gamma \vdash M\!:\!\alpha$ for all $\alpha \in \mathcal{A}(\tau)$, so again by point (1) $\Gamma \vdash M\!:\!\tau$. ◀

In the following, $\bigwedge_{i\in\{1,\ldots,n\}} \tau_i$ is used to denote any type obtained by multiple applications of the intersection type constructor to the types $\tau_1, \ldots, \tau_n$.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{x:\sigma\vdash x:\sigma\ \ (Ax)}{x:\sigma\vdash x:\varphi_1\to\varphi_1}\,(\wedge E)
}{x:\sigma\vdash x:\varphi_1\to\omega\to\varphi_1}\,(\cong)
\qquad z:\varphi_1\vdash z:\varphi_1\ \ (Ax)
}{x:\sigma,z:\varphi_1\vdash xz:\omega\to\varphi_1}\,(\to E)
\qquad y:\omega\vdash y:\omega\ \ (Ax)
}{x:\sigma,y:\omega,z:\varphi_1\vdash xzy:\varphi_1}\,(\to E)
}{x:\sigma,y:\omega\vdash\lambda z.xzy:\varphi_1\to\varphi_1}\,(\to I)
}{x:\sigma\vdash\lambda yz.xzy:\omega\to\varphi_1\to\varphi_1}\,(\to I)
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{x:\sigma\vdash x:\sigma\ \ (Ax)}{x:\sigma\vdash x:\varphi_2\to\varphi_3\to\varphi_2}\,(\wedge E)
\qquad z:\varphi_2\vdash z:\varphi_2\ \ (Ax)
}{x:\sigma,z:\varphi_2\vdash xz:\varphi_3\to\varphi_2}\,(\to E)
\qquad y:\varphi_3\vdash y:\varphi_3\ \ (Ax)
}{x:\sigma,y:\varphi_3,z:\varphi_2\vdash xzy:\varphi_2}\,(\to E)
}{x:\sigma,y:\varphi_3\vdash\lambda z.xzy:\varphi_2\to\varphi_2}\,(\to I)
}{x:\sigma\vdash\lambda yz.xzy:\varphi_3\to\varphi_2\to\varphi_2}\,(\to I)
}{
\dfrac{x:(\varphi_1\to\varphi_1)\wedge(\varphi_2\to\varphi_3\to\varphi_2)\vdash\lambda yz.xzy:(\omega\to\varphi_1\to\varphi_1)\wedge(\varphi_3\to\varphi_2\to\varphi_2)}{\vdash\lambda xyz.xzy:(\varphi_1\to\varphi_1)\wedge(\varphi_2\to\varphi_3\to\varphi_2)\to(\omega\to\varphi_1\to\varphi_1)\wedge(\varphi_3\to\varphi_2\to\varphi_2)}\,(\to I)
}\,(\wedge I)
$$

Figure 2 Derivation of $\vdash\lambda xyz.xzy:(\varphi_1\to\varphi_1)\wedge(\varphi_2\to\varphi_3\to\varphi_2)\to(\omega\to\varphi_1\to\varphi_1)\wedge(\varphi_3\to\varphi_2\to\varphi_2)$, where $\sigma=(\varphi_1\to\varphi_1)\wedge(\varphi_2\to\varphi_3\to\varphi_2)$.

▶ **Lemma 4** (Inversion Lemma). **1.** *If* $x:\sigma \vdash x:\tau$, *then* $\sigma \precsim \tau$.
2. *If* $\Gamma \vdash \lambda x.M:\tau$ *and* $\tau \precsim \rho \to \sigma$, *then* $\Gamma, x:\rho \vdash M:\sigma$.
3. *If* $\Gamma \vdash MN:\tau$, *then there are* $\Gamma_1, \Gamma_2, \sigma_i, \tau_i$ $(1 \leq i \leq n)$ *such that* $\Gamma = \Gamma_1, \Gamma_2$ *and* $\Gamma_1 \vdash M:\sigma_i \to \tau_i$, *and* $\Gamma_2 \vdash N:\sigma_i$ *for* $1 \leq i \leq n$ *and* $\bigwedge_{i\in\{1,\dots,n\}} \tau_i \precsim \tau$.
4. *If* $\Gamma \vdash MN:\alpha$, *then there are* $\Gamma_1, \Gamma_2, \sigma, \tau$ *such that* $\Gamma = \Gamma_1, \Gamma_2$ *and* $\Gamma_1 \vdash M:\sigma \to \tau$, $\Gamma_2 \vdash N:\sigma$ *and* $\tau \precsim \alpha$.

Proof. Points (1), (2) and (3) are proved by induction on derivations. Only the non-standard cases are presented.
For point (1), if the last applied rule is ($\cong$), observe that $\sigma \precsim \tau'$ and $\tau' \cong \tau$ imply $\sigma \precsim \tau$. If the last applied rule is ($\wedge I$) or ($\wedge E$), observe that $\sigma \precsim \tau_1$ and $\sigma \precsim \tau_2$ iff $\sigma \precsim \tau_1 \wedge \tau_2$.
For point (2), if the last applied rule is ($\cong$), observe that $\tau' \cong \tau$ and $\tau \precsim \sigma \to \rho$ imply $\tau' \precsim \sigma \to \rho$. If the last applied rule is ($\wedge I$) or ($\wedge E$), observe that $\tau_1 \wedge \tau_2 \precsim \sigma \to \rho$ iff $\tau_1 \precsim \sigma \to \rho$ or $\tau_2 \precsim \sigma \to \rho$.
The proof of point (3), if the last applied rule is ($\cong$) or ($\wedge E$), is the same as that of point (1). If the last applied rule is ($\wedge I$) by the induction hypothesis one has $\bigwedge_{i\in\{1,\dots,n\}} \tau_i^{(1)} \precsim \tau_1$ and $\bigwedge_{i\in\{1,\dots,m\}} \tau_i^{(2)} \precsim \tau_2$, which imply $(\bigwedge_{i\in\{1,\dots,n\}} \tau_i^{(1)}) \wedge (\bigwedge_{i\in\{1,\dots,m\}} \tau_i^{(2)}) \precsim \tau_1 \wedge \tau_2$.
Point (4) follows from point (3) and the definition of $\precsim$. In fact point (3) gives $\Gamma = \Gamma_1, \Gamma_2$ such that $\Gamma_1 \vdash M:\sigma_i \to \tau_i$, $\Gamma_2 \vdash N:\sigma_i$ and $\bigwedge_{i\in\{1,\dots,n\}} \tau_i \precsim \tau$, for some $\Gamma_1, \Gamma_2, \sigma_i, \tau_i$ $(1 \leq i \leq n)$. In this case $\tau = \alpha$ and $\bigwedge_{i\in\{1,\dots,n\}} \tau_i \precsim \alpha$ implies that there is $\beta \in \mathcal{A}(\bigwedge_{i\in\{1,\dots,n\}} \tau_i) = \bigcup_{i\in\{1,\dots,n\}} \mathcal{A}(\tau_i)$ such that $\beta \cong \alpha$. So there is an i_0 $(1 \leq i_0 \leq n)$ such that $\beta \in \mathcal{A}(\tau_{i_0})$ and $\beta \cong \alpha$, that give $\tau_{i_0} \precsim \alpha$. One can then choose $\sigma = \sigma_{i_0}$ and $\tau = \tau_{i_0}$. ◀

The following characterisation of the arrow types of the identity $\lambda x.x$ justifies the name of the pre-order relation in Definition 2.

▶ **Corollary 5.** $\vdash \lambda x.x:\sigma \to \tau$ *iff* $\sigma \precsim \tau$.

Proof. Easy from Lemmas 4(2), 4(1) and 3(2). ◀

The Inversion Lemma allows one to show some useful properties of arrow types derivable for λ-abstractions.

▶ **Lemma 6.** **1.** *If* $\Gamma \vdash \lambda x.M:\sigma \to \tau$ *and* $\Gamma \vdash \lambda x.M:\rho \to \theta$, *then* $\Gamma \vdash \lambda x.M:\sigma \wedge \rho \to \tau \wedge \theta$.
2. *If* $\Gamma \vdash \lambda x.M:\sigma \to \tau$ *and* $\Gamma \vdash \lambda x.M:\sigma \to \rho$, *then* $\Gamma \vdash \lambda x.M:\sigma \to \tau \wedge \rho$.

Proof. (1). By Lemma 4(2) $\Gamma, x:\sigma \vdash M:\tau$ and $\Gamma, x:\rho \vdash M:\theta$, which imply $\Gamma, x:\sigma \wedge \rho \vdash M:\tau$ and $\Gamma, x:\sigma \wedge \rho \vdash M:\theta$ by rules ($\wedge E$) and (L). Rule ($\wedge I$) derives $\Gamma, x:\sigma \wedge \rho \vdash M:\tau \wedge \theta$. Rule ($\to I$) concludes the proof.
(2). By Lemma 4(2) and rules ($\wedge I$), ($\to I$). ◀

This section ends with the proofs of Subject Reduction (Theorem 8) and Subject Expansion (Theorem 9). As usual a Substitution Lemma is required.

▶ **Lemma 7** (Substitution Lemma). *If* $\Gamma, x:\sigma \vdash M:\tau$ *and* $\Gamma' \vdash N:\sigma$ *and* $dom(\Gamma) \cap dom(\Gamma') = \emptyset$, *then* $\Gamma, \Gamma' \vdash M[N/x]:\tau$.

Proof. The proof is by structural induction on M. ◀

▶ **Theorem 8** (Subject Reduction). *If* $\Gamma \vdash M:\tau$ *and* $M \longrightarrow_\beta^* N$, *then* $\Gamma \vdash N:\tau$.

Proof. It is enough to show that $\Gamma \vdash (\lambda x.M)N : \tau$ implies $\Gamma \vdash M[N/x] : \tau$. By Lemma 4(3) there are $\Gamma_1, \Gamma_2, \sigma_i, \tau_i$ $(1 \leq i \leq n)$ such that $\Gamma = \Gamma_1, \Gamma_2$ and $\Gamma_1 \vdash \lambda x.M : \sigma_i \to \tau_i$, $\Gamma_2 \vdash N : \sigma_i$ for $1 \leq i \leq n$ and $\bigwedge_{i\in\{1,\ldots,n\}} \tau_i \precsim \tau$. By Lemma 4(2) $\Gamma_1, x : \sigma_i \vdash M : \tau_i$, which implies $\Gamma_1, \Gamma_2 \vdash M[N/x] : \tau_i$ by Lemma 7 for $1 \leq i \leq n$. By applications of rule $(\wedge I)$ one has $\Gamma \vdash M[N/x] : \bigwedge_{i\in\{1,\ldots,n\}} \tau_i$ and, by Lemma 3(2), one obtains $\Gamma \vdash M[N/x] : \tau$. ◀

Types are not preserved by η-reduction, for example $x : \varphi \to \varphi \vdash \lambda y.xy : \varphi \wedge \psi \to \varphi$, while $x : \varphi \to \varphi \not\vdash x : \varphi \wedge \psi \to \varphi$.

Subject expansion holds for both β and η-expansions.

▶ **Theorem 9** (Subject Expansion). *If M is a linear λ-term and $M \longrightarrow^*_{\beta\eta} N$ and $\Gamma \vdash N : \tau$, then $\Gamma \vdash M : \tau$.*

Proof. For β-expansion it is enough to show that $\Gamma \vdash M[N/x] : \tau$ implies $\Gamma \vdash (\lambda x.M)N : \tau$. The proof is by structural induction on M, observing that the linearity condition implies that there is exactly one occurrence of x in M.
For η-expansion let $\Gamma \vdash M : \tau$ and $\alpha \in \mathcal{A}(\tau)$. By Lemma 3(1) it is enough to show that $\Gamma \vdash \lambda x.Mx : \alpha$, where x is fresh. Let $\alpha \cong \sigma \to \rho$. By Lemma 3(1) and rule $(\cong)$ $\Gamma \vdash M : \sigma \to \rho$. By rules $(\to E)$ and $(\to I)$ one has $\Gamma \vdash \lambda x.Mx : \sigma \to \rho$. Rule $(\cong)$ implies $\Gamma \vdash \lambda x.Mx : \alpha$. Lemma 3(1) concludes. ◀

3 Isomorphism and Equality in Models

The study of type isomorphism in λ-calculus is based on the characterisation of λ-term invertibility. A λ-term P is *invertible* if there exists a λ-term P^{-1} such that $P \circ P^{-1} =_{\beta\eta} P^{-1} \circ P =_{\beta\eta} \lambda x.x$. The paper [11] completely characterises the invertible λ-terms in $\lambda\beta\eta$-calculus: the invertible terms are all and only the *finite hereditary permutators*.

▶ **Definition 10** (Finite Hereditary Permutator). A *finite hereditary permutator* (FHP for short) is a λ-term of the form (modulo β-conversion)

$$\lambda x y_1 \ldots y_n.x(P_1 y_{\pi(1)}) \ldots (P_n y_{\pi(n)}) \quad (n \geq 0)$$

where π is a permutation of $1, \ldots, n$, and $P_1, \ldots, P_n$ are FHPs.

Note that the identity is trivially an FHP (take $n = 0$). Another example of an FHP is

$$\lambda x y_1 y_2.x\, y_2\, y_1 \;{}_{\beta}\!\overset{*}{\longleftarrow}\; \lambda x y_1 y_2.x\,((\lambda z.z)\, y_2)\,((\lambda z.z)\, y_1),$$

which proves the swap equation. It is easy to show that FHPs are closed on composition.

▶ **Theorem 11.** *A λ -term is invertible iff it is a finite hereditary permutator.*

This result, obtained in the framework of the untyped λ-calculus, has been the basis for studying type isomorphism in different type systems for the λ-calculus. Note that every FHP has, modulo $\beta\eta$-conversion, a unique inverse P^{-1}. Even if in the type free λ-calculus FHPs are defined modulo $\beta\eta$-conversion [11], in this paper each FHP is considered only modulo β-conversion, because types are not invariant under η-reduction. Taking into account these properties, the definition of type isomorphism can be stated as follows:

▶ **Definition 12** (Type isomorphism). Two types σ and τ are isomorphic $(\sigma \approx \tau)$ if there exists a pair $< P, P^{-1} >$ of FHPs, inverse of each other, such that $\vdash P : \sigma \to \tau$ and $\vdash P^{-1} : \tau \to \sigma$. The pair $< P, P^{-1} >$ *proves* the isomorphism.

When $P = P^{-1}$ one can simply write “P proves the isomorphism”.

It is immediate to verify that type isomorphism is an equivalence relation.

Clearly semantic type equivalence implies type isomorphism, i.e.

$$\sigma \cong \tau \text{ implies } \sigma \ \approx\ \tau$$

The inverse does not hold, for example $\lambda xyz.xzy$ proves $\omega \to \varphi \to \varphi \ \approx\ \varphi \to \varphi$ (note that $\varphi \to \varphi \cong \varphi \to \omega \to \varphi$), but $\omega \to \varphi \to \varphi \ncong \varphi \to \varphi$.

It is useful to consider some basic isomorphisms, which are directly related to set theoretic properties of intersection and to standard properties of functional types. It is interesting to remark that all these isomorphisms are provable equalities in the system $\mathbf{B}_+$ of relevant logic [20].

$$\begin{array}{ll}
\textbf{idem}. & \sigma \wedge \sigma \approx \sigma \\
\textbf{comm}. & \sigma \wedge \tau \approx \tau \wedge \sigma \\
\textbf{assoc}. & (\sigma \wedge \tau) \wedge \rho \approx \sigma \wedge (\tau \wedge \rho) \\
\textbf{split}. & \sigma \to \tau \wedge \rho \approx (\sigma \to \tau) \wedge (\sigma \to \rho)
\end{array}$$

The identity $\lambda x.x$ proves the first three isomorphisms, and its η-expansion $\lambda xy.xy$ proves the fourth one.

An intersection $\sigma \wedge \tau$ is set-theoretically equal to σ if σ is included in τ. So, it is handy to introduce a pre-order on types which formalises set-theoretic inclusion taking into account the meaning of the arrow type constructor and the semantic type equivalence given in Definition 1. This pre-order is dubbed normalisation pre-order being used in the next section to define normalisation rules (Definition 19).

▶ **Definition 13** (Normalisation pre-order on types). The *normalisation pre-order* $\leq$ is the pre-order relation on types defined by:

$$\begin{array}{rl}
\sigma \leq \omega & \quad \sigma \wedge \tau \leq \sigma \quad \sigma \wedge \tau \leq \tau \\
\varphi \leq \sigma \to \varphi & \quad \omega \leq \sigma \to \omega \\
\sigma \leq \tau,\ \sigma \leq \rho \Rightarrow \sigma \leq \tau \wedge \rho & \quad \sigma' \leq \sigma,\ \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'
\end{array}$$

Notice that $\sigma \leq \omega$ derives from $\sigma \wedge \omega \cong \sigma$. Moreover $\varphi \leq \sigma \to \varphi$ and $\omega \leq \sigma \to \omega$ are justified by $\varphi \cong \omega \to \varphi$, $\omega \cong \omega \to \omega$ and the contra-variance of $\leq$ for arrow types.

The identity pre-order and the normalisation pre-order are incomparable, for example $\omega \to \varphi \precsim \varphi$, $\omega \to \varphi \nleq \varphi$ and $\sigma \to \tau \leq \sigma \wedge \rho \to \tau$, $\sigma \to \tau \not\precsim \sigma \wedge \rho \to \tau$.

The soundness of the normalisation pre-order follows from the following lemma, which shows the expected isomorphisms. This lemma uses particular forms of FHPs defined as follows.

▶ **Definition 14** (Finite Hereditary Identity). A *finite hereditary identity* (FHI) is a β-normal form obtained from $\lambda x.x$ through a finite (possibly zero) number of η-expansions.

It is easy to verify that, for each FHI different from the identity, one gets

$$\mathsf{Id} \, {}_{\beta}^{*}\!\!\longleftarrow \lambda xy.\mathsf{Id}_1(x(\mathsf{Id}_2 y))$$

for unique FHIs $\mathsf{Id}_1, \mathsf{Id}_2$. For example, for $\mathsf{Id} = \lambda xy_1y_2y_3.x(\lambda t.y_1t)y_2(\lambda u_1u_2.y_3u_1u_2)$ one has $\mathsf{Id}_1 = \lambda xy_2y_3.xy_2(\lambda u_1u_2.y_3u_1u_2)$ and $\mathsf{Id}_2 = \lambda xt.xt$.

▶ **Lemma 15.** 1. *Let* Id *be an FHI, then* $\vdash \mathsf{Id}:\sigma \to \sigma$ *for every type* σ.

2. *If* $\sigma \leq \tau$*, then there is an FHI* Id *such that* $\vdash \mathsf{Id}:\sigma \to \tau$.

3. *If* $\sigma \leq \tau$*, then* $\sigma \wedge \tau \ \approx\ \sigma$.

Proof. (1). The proof is trivial observing that the identity $\lambda x.x$ has type $\sigma \to \sigma$ for all σ and types are preserved by η-expansions (Theorem 9).
(2). The proof is by induction on the definition of $\leq$. Only interesting cases are considered. If $\sigma \leq \rho$ and $\rho \leq \tau$ imply $\sigma \leq \tau$, then by the induction hypothesis there are FHIs Id_1, Id_2 such that $\vdash \mathsf{Id}_1 : \sigma \to \rho$ and $\vdash \mathsf{Id}_2 : \rho \to \tau$. This implies $\vdash \lambda x.\mathsf{Id}_2(\mathsf{Id}_1 x) : \sigma \to \tau$. It is easy to verify that $\lambda x.\mathsf{Id}_2(\mathsf{Id}_1 x)$ β-reduces to an FHI.
If $\sigma \leq \tau$ and $\sigma \leq \rho$ imply $\sigma \leq \tau \wedge \rho$, then by the induction hypothesis there are FHIs Id_1, Id_2 such that $\vdash \mathsf{Id}_1 : \sigma \to \tau$ and $\vdash \mathsf{Id}_2 : \sigma \to \rho$. By definition of FHI there is an FHI Id such that $\mathsf{Id} \longrightarrow^*_\eta \mathsf{Id}_1$ and $\mathsf{Id} \longrightarrow^*_\eta \mathsf{Id}_2$. By Subject Expansion (Theorem 9) $\vdash \mathsf{Id} : \sigma \to \tau$ and $\vdash \mathsf{Id} : \sigma \to \rho$, which imply $\vdash \mathsf{Id} : \sigma \to \tau \wedge \rho$ by Lemma 6(2).
If $\varphi \leq \sigma \to \varphi$ one can derive $y : \sigma \vdash y : \omega$ by rule (ω), and $x : \varphi \vdash x : \omega \to \varphi$ by rule $(\cong)$. Then $\vdash \lambda xy.xy : \varphi \to \sigma \to \varphi$ holds by rules $(\to E)$ and $(\to I)$.
If $\sigma' \leq \sigma$ and $\tau \leq \tau'$ imply $\sigma \to \tau \leq \sigma' \to \tau'$, then by the induction hypothesis there are FHIs Id_1, Id_2 such that $\vdash \mathsf{Id}_2 : \sigma' \to \sigma$ and $\vdash \mathsf{Id}_1 : \tau \to \tau'$. This implies

$$\vdash \lambda xy.\mathsf{Id}_1(x(\mathsf{Id}_2 y)) : (\sigma \to \tau) \to \sigma' \to \tau'$$

and $\lambda xy.\mathsf{Id}_1(x(\mathsf{Id}_2 y))$ β-reduces to an FHI.
(3). By point (2) there is an FHI Id such that $\vdash \mathsf{Id} : \sigma \to \tau$. By point (1) one has $\vdash \mathsf{Id} : \sigma \to \sigma$. Lemma 6(2) gives $\vdash \mathsf{Id} : \sigma \to \sigma \wedge \tau$. Lastly $\vdash \lambda x.x : \sigma \wedge \tau \to \sigma$. ◀

For example $\lambda xyz.xyz$ has type $(\varphi \to \varphi) \to (\varphi \to \varphi) \wedge (\varphi \to \psi \to \varphi)$. Notice that $\varphi \to \varphi \cong \varphi \to \omega \to \varphi \leq \varphi \to \psi \to \varphi$.

Lemma 15 proves the validity of the basic isomorphism:

$$\textbf{erase}. \qquad \text{if } \sigma \leq \tau \text{ then} \qquad \sigma \wedge \tau \;\approx\; \sigma$$

The following lemma assures that one can consider types modulo idempotence, commutativity, associativity, splitting and erasure in every type context $C[]$. A *type context* is defined as usual:

$$\mathcal{C}[\,] \;::=\; [\,] \;\mid\; \mathcal{C}[\,] \to \sigma \;\mid\; \sigma \to \mathcal{C}[\,] \;\mid\; \sigma \wedge \mathcal{C}[\,] \;\mid\; \mathcal{C}[\,] \wedge \sigma$$

▶ **Lemma 16.** *If $\sigma \approx \tau$ is proved by reflexive and transitive application of the basic isomorphisms* (**idem**)*,* (**comm**)*,* (**assoc**)*,* (**split**)*, and* (**erase**)*, then $\mathcal{C}[\sigma] \approx \mathcal{C}[\tau]$.*

Proof. As the isomorphism is reflexive and transitive, it is enough to consider the case in which $\sigma \approx \tau$ is proved by one application of (**idem**), (**comm**), (**assoc**), (**split**), and (**erase**). The proof is by structural induction on type contexts. For any context $\mathcal{C}[\,]$, an FHI $\mathsf{Id}_{\mathcal{C}[\,]}$ that proves the isomorhism $\mathcal{C}[\sigma] \approx \mathcal{C}[\tau]$ is provided.

- $\mathsf{Id}_{[\,]} = \lambda x.x$ for (**idem**), (**comm**), (**assoc**); $\mathsf{Id}_{[\,]} = \lambda xy.xy$ for (**split**); $\mathsf{Id}_{[\,]}$ is given by Lemma 15(3) for (**erase**).
- $\mathsf{Id}_{\mathcal{C}[\,]\to\rho} \;{}_\beta\!\longleftarrow \lambda xy.x(\mathsf{Id}_{\mathcal{C}[\,]} y)$.
- $\mathsf{Id}_{\rho\to\mathcal{C}[\,]} \;{}_\beta\!\longleftarrow \lambda xy.\mathsf{Id}_{\mathcal{C}[\,]}(xy)$.
- $\mathsf{Id}_{\rho\wedge\mathcal{C}[\,]} = \mathsf{Id}_{\mathcal{C}[\,]\wedge\rho} = \mathsf{Id}_{\mathcal{C}[\,]}$. ◀

For example $\lambda xy.x(\lambda zt.yzt)$ proves $(\varphi \to \varphi) \to \psi \;\approx\; (\varphi \to \varphi) \wedge (\varphi \to \psi \to \varphi) \to \psi$. In fact, $\lambda wzt.wzt$, having the type $(\varphi \to \varphi) \to (\varphi \to \varphi) \wedge (\varphi \to \psi \to \varphi)$, proves the isomorphism by erasure (Lemma 15(3)). Moreover $\mathsf{Id}_{[\,]\to\varphi} \;{}^{*}_{\beta}\!\longleftarrow \lambda xy.x(\mathsf{Id}_{[\,]} y) \;= \lambda xy.x((\lambda wzt.wzt)y)$ since the isomorphism used in the empty context is the (**erase**).

Lemma 16 justifies the notation $\bigwedge_{i \in I} \sigma_i$ with finite I, where a single atomic or arrow type is seen as an intersection (in this case I is a singleton).

The standard models of intersection types map types to subsets of any domain that is a model of the untyped λ-calculus, with the condition that the arrow is interpreted as the function space constructor and the intersection as the set-theoretic intersection. More formally using $\mathcal{P}$ to denote the power-set:

▶ **Definition 17.** Let $\mathcal{D}$ be the domain of a λ-model and $\mathcal{V} : \mathtt{A} \to \mathcal{P}(\mathcal{D})$ a mapping from atomic types to subsets of $\mathcal{D}$. The *standard interpretation of types* is given by:

$$\begin{array}{ll} [\![\varphi]\!]_{\mathcal{V}} = \mathcal{V}(\varphi) & [\![\omega]\!]_{\mathcal{V}} = \mathcal{D} \\ [\![\sigma \to \tau]\!]_{\mathcal{V}} = \{d \in \mathcal{D} \mid \forall d' \in [\![\sigma]\!]_{\mathcal{V}} :\ d \cdot d' \in [\![\tau]\!]_{\mathcal{V}}\} & [\![\sigma \wedge \tau]\!]_{\mathcal{V}} = [\![\sigma]\!]_{\mathcal{V}} \cap [\![\tau]\!]_{\mathcal{V}} \end{array}$$

The equalities corresponding to the contextual closure of the basic type isomorphisms (**idem**), (**comm**), (**assoc**), (**split**), and (**erase**), include the ones of [2], which are proved to be the equalities valid in all standard models. Therefore all types equal in all standard models are isomorphic in the system of Figure 1.

▶ **Theorem 18.** *Type* equality *in the standard models of intersection types entails type isomorphism.*

Instead, the standard type interpretation does not validate a pre-order which includes the clause $\varphi \leq \omega \to \varphi$ or $\omega \to \varphi \leq \varphi$ or both, unless the mapping from atomic types to subsets of $\mathcal{D}$ enjoys particular properties. In particular a mapping $\mathcal{V}_0$ from atomic types to subsets of $\mathcal{D}$ such that

$$\mathcal{V}_0(\varphi) = \{d \in \mathcal{D} \mid \forall d' \in \mathcal{D} :\ d \cdot d' \in \mathcal{V}_0(\varphi)\}$$

validates the semantic type equivalence given in Definition 1, i.e. it gives the same interpretation to equivalent types. If $\mathcal{D}$ is the domain of the inverse limit model discussed after Definition 1, then a valid interpretation is that of taking as $\mathcal{V}_0(\varphi)$ the set of all the elements of $\mathcal{D}$ greater than or equal to the finite element corresponding to φ. Notably $\mathcal{V}_0(\varphi)$ is an open set in the Scott topology over $\mathcal{D}$. More generally, the interpretation of each type σ is the open set of all elements of $\mathcal{D}$ greater than or equal to the finite element corresponding to σ, when mapping arrow types in step functions and intersection types in joins.

4 Normalisation

To investigate type isomorphism, following a common approach [4, 14, 12, 7, 8], a notion of *normal form* of types is introduced. *Normal type* is short for type in normal form. The notion of normal form is effective, since an algorithm to find the normal form of an arbitrary type is given.

Type normalisation rules are introduced together with the proof of their soundness.

▶ **Definition 19** (Type normalisation rules)**.** The type normalisation rules are:

$$\begin{array}{llll} (\varphi \Rightarrow) & \omega \to \varphi \Longrightarrow \varphi & (\omega \Rightarrow) & \omega \leq \sigma \text{ and } \sigma \neq \omega \text{ imply } \sigma \Longrightarrow \omega \\ (\wedge \Rightarrow) & \sigma \to \tau \wedge \rho \Longrightarrow (\sigma \to \tau) \wedge (\sigma \to \rho) & (\leq\Rightarrow) & \sigma \leq \tau \text{ implies } \sigma \wedge \tau \Longrightarrow \sigma \end{array}$$

$$(\text{ctx} \Rightarrow) \quad \sigma \Longrightarrow \tau \text{ implies } \mathcal{C}[\sigma] \Longrightarrow \mathcal{C}[\tau]$$

The first two rules follow immediately from semantic type equivalence, the following two rules correspond to the **split** and **erase** basic isomorphisms, respectively. Since $\omega \leq \sigma \to \omega$, an admissible rule is $\sigma \to \omega \Longrightarrow \omega$.

For example by rules $(\wedge \Rightarrow)$ and $(\leq\Rightarrow)$, taking into account that $\wedge$ is considered modulo commutativity:

$$(\varphi_1 \to \varphi_2 \wedge \varphi_3) \wedge \varphi_3 \Longrightarrow (\varphi_1 \to \varphi_2) \wedge (\varphi_1 \to \varphi_3) \wedge \varphi_3 \Longrightarrow (\varphi_1 \to \varphi_2) \wedge \varphi_3$$

since $\varphi_3 \leq \varphi_1 \to \varphi_3$.

A *normal type* ξ is either ω or a normal intersection type. A *normal intersection type* ζ is either a normal singleton type or an intersection of normal intersection types, which cannot be reduced by rule $(\leq\Rightarrow)$. A *normal singleton type* ν is either an atomic type different from ω or an arrow type from a normal intersection type to a normal singleton type, which cannot be reduced by rule $(\varphi \Rightarrow)$. Formally:

$$\xi ::= \omega \mid \zeta \qquad \zeta ::= \nu \mid \zeta \wedge \zeta \qquad \nu ::= \varphi \mid \xi \to \nu$$

where an intersection is allowed only if rule $(\leq\Rightarrow)$ cannot be applied at top level and an arrow is allowed only if rule $(\varphi \Rightarrow)$ cannot be applied at top level. So a normal type is either ω or $\bigwedge_{i\in I} \nu_i$ for some I and ν_i with $i \in I$.

For example $(\varphi \to \varphi) \wedge \psi$ is a normal type, but not a normal singleton type, while $\varphi \to \varphi$ is a normal singleton type.

The type $(\omega \to \varphi \to \varphi) \wedge \psi \to \psi$ is a normal singleton type, because $(\omega \to \varphi \to \varphi) \wedge \psi$ is a normal intersection type, being $\omega \to \varphi \to \varphi$ a normal singleton type. On the contrary $(\varphi \to \omega \to \varphi) \wedge \psi \to \psi$ is not a normal singleton type, because $\varphi \to \omega \to \varphi$ is not so.

▶ **Theorem 20** (Soundness of the normalisation rules)**.** *If $\sigma \Longrightarrow \tau$, then there are FHIs* Id, Id' *such that* $\vdash \mathsf{Id} : \sigma \to \tau$, $\vdash \mathsf{Id}' : \tau \to \sigma$.

Proof. Rule $(\varphi \Rightarrow)$ is obtained by orienting the equivalence relation between types, so it is sound since equivalent types are shown isomorphic by the identity. Rule $(\omega \Rightarrow)$ is sound because, by Lemma 15(2), there is an FHI Id such that $\vdash \mathsf{Id} : \omega \to \sigma$, and obviously $\vdash \mathsf{Id} : \sigma \to \omega$. Rule $(\wedge \Rightarrow)$ is sound by the isomorphism (**split**). Lemma 15(3) implies the soundness of rule $(\leq\Rightarrow)$. Lemma 16 implies the soundness of rule $(\text{ctx} \Rightarrow)$. ◀

For example $(\varphi_1 \to \varphi_2 \wedge \varphi_3) \wedge \varphi_3 \Longrightarrow^* (\varphi_1 \to \varphi_2) \wedge \varphi_3$ as shown before, and $\lambda xy.xy$ proves

$$(\varphi_1 \to \varphi_2 \wedge \varphi_3) \wedge \varphi_3 \;\approx\; (\varphi_1 \to \varphi_2) \wedge \varphi_3.$$

In fact, both

$$x : (\varphi_1 \to \varphi_2 \wedge \varphi_3) \wedge \varphi_3 \vdash \lambda y.xy : (\varphi_1 \to \varphi_2) \wedge \varphi_3$$

and

$$x : (\varphi_1 \to \varphi_2) \wedge \varphi_3 \vdash \lambda y.xy : (\varphi_1 \to \varphi_2 \wedge \varphi_3) \wedge \varphi_3$$

are derivable.

The following theorem shows the existence and uniqueness of the normal forms, i.e. that the normalisation rules are terminating and confluent.

▶ **Theorem 21** (Uniqueness of normal form)**.** *The normalisation rules of Definition 19 are terminating and confluent.*

Proof. The *termination* follows from an easy adaptation of the recursive path ordering method [10]. The partial order on operators is defined by: $\to \succ \wedge$. Notice that the induced recursive path ordering $\succ^*$ has the subterm property. This solves the case of all rules but $(\wedge \Rightarrow)$. For rule $(\wedge \Rightarrow)$, since $\to \succ \wedge$, it is enough to observe that $\sigma \to \tau \wedge \rho \succ^* \sigma \to \tau$ and $\sigma \to \tau \wedge \rho \succ^* \sigma \to \rho$.

For *confluence*, thanks to the Newman Lemma [18], it is sufficient to prove the convergence of the critical pairs. Figure 3 shows the diamonds for the only three interesting cases, where $\sigma \to \tau \wedge \rho \leq \theta$, $\omega \to \varphi \leq \sigma$, $\omega \leq \sigma$, respectively. ◀

$$\begin{tikzcd}
 & (\sigma \to \tau \wedge \rho) \wedge \theta \arrow[dl, Rightarrow, "(\leq\Rightarrow)"'] \arrow[dr, Rightarrow, "(\wedge\Rightarrow)"] & \\
\sigma \to \tau \wedge \rho \arrow[dr, Rightarrow, "(\wedge\Rightarrow)"'] & & (\sigma \to \tau) \wedge (\sigma \to \rho) \wedge \theta \arrow[dl, Rightarrow, "(\leq\Rightarrow)"] \\
 & (\sigma \to \tau) \wedge (\sigma \to \rho) &
\end{tikzcd}$$

$$\begin{tikzcd}
 & (\omega \to \varphi) \wedge \sigma \arrow[dl, Rightarrow, "(\leq\Rightarrow)"'] \arrow[dr, Rightarrow, "(\varphi\Rightarrow)"] & \\
\omega \to \varphi \arrow[dr, Rightarrow, "(\varphi\Rightarrow)"'] & & \varphi \wedge \sigma \arrow[dl, Rightarrow, "(\leq\Rightarrow)"] \\
 & \varphi &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \sigma \wedge \omega \arrow[dl, Rightarrow, "(\leq\Rightarrow)"'] \arrow[dr, Rightarrow, "(\omega\Rightarrow)"] & \\
\sigma \arrow[dr, Rightarrow, "(\omega\Rightarrow)"'] & & \omega \wedge \omega \arrow[dl, Rightarrow, "(\leq\Rightarrow)"] \\
 & \omega &
\end{tikzcd}$$

Figure 3 Critical pairs and their diamonds.

The unique (modulo idempotence, commutativity and associativity of $\wedge$) normal form of σ is denoted by $\sigma{\downarrow}$. The soundness of the normalisation rules (Theorem 20) implies that each type is isomorphic to its normal form.

▶ **Corollary 22.** $\sigma \approx \sigma{\downarrow}$.

As expected, semantic equivalent types have the same normal form. Clearly the inverse is false, since $(\sigma \to \tau \wedge \rho){\downarrow}= (\sigma \to \tau) \wedge (\sigma \to \rho)$, but $\sigma \to \tau \wedge \rho \ncong (\sigma \to \tau) \wedge (\sigma \to \rho)$.

▶ **Lemma 23.** *If $\sigma \cong \tau$, then $\sigma{\downarrow}= \tau{\downarrow}$.*

Proof. The proof is by cases on Definition 1. For the equivalences $\varphi \cong \omega \to \varphi$ and $\omega \cong \omega \to \omega$, rules $(\varphi \Rightarrow)$ and $(\omega \Rightarrow)$ give $(\omega \to \varphi){\downarrow}= \varphi$ and $(\omega \to \omega){\downarrow}= \omega$, respectively. For the equivalences $\sigma \cong \omega \wedge \sigma$ and $\sigma \cong \sigma \wedge \omega$, rule $(\wedge \Rightarrow)$ with $\sigma \leq \omega$ gives $(\omega \wedge \sigma){\downarrow}= (\sigma \wedge \omega){\downarrow}= \sigma$. The congruence follows from the applicability of the normalisation rules in any type context. ◀

This section ends showing some properties of normal types for FHPs. The main result is that isomorphic normal types different from ω are intersections with the same number of normal singleton types, which are pairwise isomorphic (Theorem 27). Lemmas 24, 25 and 26 show preliminary results. In the following ξ, χ *range over normal types* and ν, μ, λ *range over normal singleton types.*

▶ **Lemma 24.** **1.** *If $\omega \precsim \sigma \to \tau$, then $\omega \cong \tau$.*

2. *If $\mu \precsim \sigma \to \tau$ and $\tau \ncong \omega$, then $\mu \cong \sigma \to \nu$ and $\tau \cong \nu$ for some ν.*

Proof. (1). Immediate by definition of $\precsim$ (Definition 2).
(2). By definition of $\precsim$ and of $\cong$ (Definition 1). ◀

▶ **Lemma 25.** *Let $\lambda x y_1 \ldots y_n.x Q_1 \ldots Q_n$ be an FHP.*

1. *If $x : \bigwedge_{i\in I} \mu_i \vdash \lambda y_1 \ldots y_n.x Q_1 \ldots Q_n : \bigwedge_{j\in J} \nu_j$, then for every $j \in J$ there is a $i_j \in I$ such that $x : \mu_{i_j} \vdash \lambda y_1 \ldots y_n.x Q_1 \ldots Q_n : \nu_j$.*

2. *If $x : \omega \vdash \lambda y_1 \ldots y_n.x Q_1 \ldots Q_n : \xi$, then $\xi = \omega$.*

Proof. (1). Take an arbitrary $j \in J$. Without loss of generality assume

$$\nu_j \cong \xi_1 \to \cdots \to \xi_n \to \nu.$$

This is not a restriction since $\varphi \cong \underbrace{\omega \to \cdots \to \omega}_{m} \to \varphi$ for all m. By rules $(\wedge E)$ and $(\cong)$

$$x{:}\bigwedge_{i\in I}\mu_i \vdash \lambda y_1 \dots y_n.xQ_1 \dots Q_n{:}\bigwedge_{j\in J}\nu_j$$

implies $x : \bigwedge_{i\in I}\mu_i \vdash \lambda y_1 \dots y_n.xQ_1 \dots Q_n : \xi_1 \to \cdots \to \xi_n \to \nu$. Then by Lemma 4(2) it follows

$$x{:}\bigwedge_{i\in I}\mu_i, y_1{:}\xi_1, \dots, y_n{:}\xi_n \vdash xQ_1 \dots Q_n{:}\nu.$$

By repeated applications of Lemma 4(4) there are $\sigma_1, \dots, \sigma_n, \tau_1, \dots, \tau_n$ such that

$$x{:}\bigwedge_{i\in I}\mu_i, y_{\pi(1)}{:}\xi_{\pi(1)}, \dots, y_{\pi(h-1)}{:}\xi_{\pi(h-1)} \vdash xQ_1 \dots Q_{h-1}{:}\sigma_h \to \tau_h \text{ and}$$
$$y_{\pi(h)}{:}\xi_{\pi(h)} \vdash Q_h{:}\sigma_h,$$

where $y_{\pi(h)}$ is the head variable of Q_h for $1 \le h \le n$. Moreover $\tau_k \precsim \sigma_{k+1} \to \tau_{k+1}$ for $1 \le k \le n-1$ and $\tau_n \precsim \nu$. By Lemma 4(1) $x{:}\bigwedge_{i\in I}\mu_i \vdash x{:}\sigma_1 \to \tau_1$ implies $\bigwedge_{i\in I}\mu_i \precsim \sigma_1 \to \tau_1$. Then there is $i_j \in I$ such that $\mu_{i_j} \precsim \sigma_1 \to \tau_1$ by definition of $\precsim$. Lemma 24(2) applied to $\mu_{i_j} \precsim \sigma_1 \to \tau_1$ gives $\mu_{i_j} \cong \sigma_1 \to \nu_1$ and $\tau_1 \cong \nu_1$ for some ν_1. This together with $\tau_1 \precsim \sigma_2 \to \tau_2$ implies $\nu_1 \precsim \sigma_2 \to \tau_2$. Again by Lemma 24(2) one has $\nu_1 \cong \sigma_2 \to \nu_2$ and $\tau_2 \cong \nu_2$ for some ν_2. By iterating one gets $\nu_k \cong \sigma_{k+1} \to \nu_{k+1}$ and $\tau_{k+1} \cong \nu_{k+1}$ for some ν_{k+1} $(1 \le k \le n-1)$. Lastly $\tau_n \cong \nu_n$ and $\tau_n \precsim \nu$ imply $\nu_n \cong \nu$. Taking into account that $\nu_k \cong \sigma_{k+1} \to \nu_{k+1}$ and $\nu_{k+1} \cong \sigma_{k+2} \to \nu_{k+2}$ imply $\nu_k \cong \sigma_{k+1} \to \sigma_{k+2} \to \nu_{k+2}$, $(1 \le k \le n-2)$, one can conclude $\mu_{i_j} \cong \sigma_1 \to \cdots \to \sigma_n \to \nu$. Notice that this implies $\mu_{i_j} \not\cong \omega$ whenever $\nu_j \not\cong \omega$.
Rules $(\to E)$ and $(\to I)$ applied to $x{:}\mu_{i_j} \vdash x{:}\sigma_1 \to \cdots \to \sigma_n \to \nu$ and $y_{\pi(h)}{:}\xi_{\pi(h)} \vdash Q_h{:}\sigma_h$ for $1 \le h \le n$ derive $x{:}\mu_{i_j} \vdash \lambda y_1 \dots y_n.xQ_1 \dots Q_n{:}\nu_j$.
(2). Toward a contradiction assume $\xi = \bigwedge_{j\in J}\nu_j$. Let $\nu_j \cong \xi_1 \to \cdots \to \xi_n \to \nu$ for an arbitrary $j \in J$, as in the proof of point (1). One gets $\omega \precsim \sigma_1 \to \tau_1$ and

$$\tau_k \precsim \sigma_{k+1} \to \tau_{k+1} \text{ for } 1 \le k \le n-1,\ \tau_n \precsim \nu.$$

Lemma 24(1) implies $\omega \cong \nu$, which is impossible. ◀

▶ **Lemma 26.** *If* $\vdash \mathsf{Id}{:}\xi \to \chi$*, then* $\xi \le \chi$.

Proof. If $\chi = \omega$ the proof is trivial. If $\xi = \omega$, then $\chi = \omega$ by Lemma 25(2). Let $\xi = \bigwedge_{i\in I}\mu_i$, $\chi = \bigwedge_{j\in J}\nu_i$. By Lemma 25(1) for all $j \in J$ there is $i_j \in I$ such that $\vdash \mathsf{Id}{:}\mu_{i_j} \to \nu_j$. Then it is enough to show $\mu_{i_j} \le \nu_j$. The proof is by structural induction on Id. If $\mathsf{Id} = \lambda x.x$ by Lemma 4(1) $\mu_{i_j} \precsim \nu_j$; since both these types are normal singleton types, $\mu_{i_j} \cong \nu_j$, and Lemma 23 implies $\mu_{i_j} = \nu_j$. Otherwise let $\mathsf{Id}\ {}_{\beta}^{*}\!\!\longleftarrow \lambda xy.\mathsf{Id}_1(x(\mathsf{Id}_2 y))$ and $\mu_{i_j} \cong \xi' \to \mu$, $\nu_i \cong \chi' \to \nu$. By Lemma 4 $\vdash \mathsf{Id}_1 : \mu \to \nu$ and $\vdash \mathsf{Id}_2 : \chi' \to \xi'$. By the induction hypothesis $\mu \le \nu$ and $\chi' \le \xi'$, which imply $\mu_{i_j} \le \nu_j$. ◀

One can use the previous lemmas to prove that if an FHP P has the type $\bigwedge_{i\in I}\mu_i \to \bigwedge_{j\in J}\nu_j$ and its inverse P^{-1} has the type $\bigwedge_{j\in J}\nu_j \to \bigwedge_{i\in I}\mu_i$, then not only for every $j \in J$ there is a $i_j \in I$ such that $\vdash P{:}\mu_{i_j} \to \nu_j$, but P^{-1} precisely maps each component ν_j of the target intersection to its corresponding μ_{i_j} in the source intersection.

▶ **Theorem 27.** *If* $\bigwedge_{i\in I}\mu_i \approx \bigwedge_{j\in J}\nu_j$ *and* $< P, P^{-1} >$ *proves this isomorphism, then there is a permutation* π *between* I *and* J *such that* $< P, P^{-1} >$ *proves* $\mu_i \approx \nu_{\pi(i)}$ *for all* $i \in I$.

Proof. By Lemma 25(1), for all $j \in J$ there is $i_j \in I$ such that $\vdash P{:}\mu_{i_j} \to \nu_j$. Again by Lemma 25(1) there is $j' \in J$ such that $\vdash P^{-1}{:}\nu_{j'} \to \mu_{i_j}$. Let us suppose $j' \neq j$ towards a contradiction. One gets $x{:}\nu_{j'} \vdash P(P^{-1}x){:}\nu_j$ and by rule $(\to I)$ $\vdash \lambda x.P(P^{-1}x){:}\nu_{j'} \to \nu_j$, which implies that $\nu_{j'} \leq \nu_j$ by Lemma 26, since $\lambda x.P(P^{-1}x)$ β-reduces to an FHI. So $\bigwedge_{j\in J} \nu_j$ would not be a normal type, since rule $(\leq\Rightarrow)$ could be applied. ◀

5 Characterisation of Isomorphism

This section shows the main result of the paper, i.e. that two types are isomorphic iff their normal forms are “similar” (Definition 28). The basic aim of the similarity relation is that of formalising isomorphism determined by argument permutations (as in the swap equation). This relation has to take into account the fact that, for two types to be isomorphic, it is not sufficient that they coincide modulo permutations of types in the arrow sequences, as in the case of cartesian products. Indeed the same permutation must be applicable to all types in the corresponding intersections. The key notion of similarity exactly expresses such a condition.

▶ **Definition 28** (Similarity). The *similarity* relation between two sequences of normal types $\langle \xi_1, \dots, \xi_m \rangle$ and $\langle \chi_1, \dots, \chi_m \rangle$, written $\langle \xi_1, \dots, \xi_m \rangle \sim \langle \chi_1, \dots, \chi_m \rangle$, is the smallest equivalence relation such that:

1. $\langle \xi_1, \dots, \xi_m \rangle \sim \langle \xi_1, \dots, \xi_m \rangle$;
2. if $\langle \xi_1, \dots, \xi_i, \xi_{i+1}, \dots, \xi_m \rangle \sim \langle \chi_1, \dots, \chi_i, \chi_{i+1}, \dots, \chi_m \rangle$, then
$$\langle \xi_1, \dots, (\xi_i \wedge \xi_{i+1})\downarrow, \dots, \xi_m \rangle \sim \langle \chi_1, \dots, (\chi_i \wedge \chi_{i+1})\downarrow, \dots, \chi_m \rangle;$$
3. if $\langle \xi_i^{(1)}, \dots, \xi_i^{(m)} \rangle \sim \langle \chi_i^{(1)}, \dots, \chi_i^{(m)} \rangle$ for $1 \leq i \leq n$, then
$$\langle (\xi_1^{(1)} \to \dots \to \xi_n^{(1)} \to \nu_1)\downarrow, \dots, (\xi_1^{(m)} \to \dots \to \xi_n^{(m)} \to \nu_m)\downarrow \rangle \sim$$
$$\langle (\chi_{\pi(1)}^{(1)} \to \dots \to \chi_{\pi(n)}^{(1)} \to \nu_1)\downarrow, \dots, (\chi_{\pi(1)}^{(m)} \to \dots \to \chi_{\pi(n)}^{(m)} \to \nu_m)\downarrow \rangle,$$
where π is a permutation of $1, \dots, n$.

Similarity between normal types is trivially defined as similarity between unary sequences:
$$\xi \sim \chi \text{ if } \langle \xi \rangle \sim \langle \chi \rangle.$$

For example, from $\langle \omega \rangle \sim \langle \omega \rangle$ and $\langle \varphi \rangle \sim \langle \varphi \rangle$ one obtains, by Definition 28(3),
$$\langle (\omega \to \varphi \to \varphi)\downarrow \rangle \sim \langle (\varphi \to \omega \to \varphi)\downarrow \rangle,$$
that is $\omega \to \varphi \to \varphi \sim \varphi \to \varphi$. Moreover $\langle \psi, \omega \to \varphi \to \varphi \rangle \sim \langle \psi, \varphi \to \varphi \rangle$ gives
$$\psi \wedge (\omega \to \varphi \to \varphi) \sim \psi \wedge (\varphi \to \varphi).$$

The soundness of similarity can be shown without difficulties.

▶ **Theorem 29** (Soundness). *If $\langle \xi_1, \dots, \xi_m \rangle \sim \langle \chi_1, \dots, \chi_m \rangle$, then there is a pair of FHPs that proves $\xi_j \approx \chi_j$, for $1 \leq j \leq m$.*

Proof. By induction on the definition of $\sim$ (Definition 28).
(1). $\langle \xi_1, \dots, \xi_m \rangle \sim \langle \xi_1, \dots, \xi_m \rangle$. The identity proves the isomorphism.
(2). $\langle \xi_1, \dots, \xi_i, (\xi_i \wedge \xi_{i+1})\downarrow, \dots, \xi_m \rangle \sim \langle \chi_1, \dots, \chi_i, (\chi_i \wedge \chi_{i+1})\downarrow, \dots, \chi_m \rangle$ since
$$\langle \xi_1, \dots, \xi_i, \xi_{i+1}, \dots, \xi_m \rangle \sim \langle \chi_1, \dots, \chi_i, \chi_{i+1}, \dots, \chi_m \rangle.$$
By the induction hypothesis there is a pair $< P, P^{-1}>$ that proves $\xi_j \approx \chi_j$, for $1 \leq j \leq m$. By Lemma 6(1), the same pair proves $\xi_i \wedge \xi_{i+1} \approx \chi_i \wedge \chi_{i+1}$. By Theorem 20 there are FHIs $\mathsf{Id}_1, \mathsf{Id}_2, \mathsf{Id}'_1, \mathsf{Id}'_2$ such that $< \mathsf{Id}_1, \mathsf{Id}_2>$ proves $\xi_i \wedge \xi_{i+1} \approx (\xi_i \wedge \xi_{i+1})\downarrow$ and $< \mathsf{Id}'_1, \mathsf{Id}'_2>$ proves $\chi_i \wedge \chi_{i+1} \approx (\chi_i \wedge \chi_{i+1})\downarrow$. By Lemma 15(1) $\vdash \mathsf{Id}_\ell : \xi_j \to \xi_j$ and $\vdash \mathsf{Id}'_\ell : \chi_j \to \chi_j$ for $1 \leq j \leq m$ and $1 \leq \ell \leq 2$. Then the pair $< \lambda x.\mathsf{Id}'_1(P(\mathsf{Id}_2 x)), \lambda x.\mathsf{Id}_1(P^{-1}(\mathsf{Id}'_2 x)) >$ proves the required isomorphism.

(3).
$$\begin{array}{c}\langle(\xi_1^{(1)}\to\ldots\to\xi_n^{(1)}\to\nu_1)\!\downarrow,\ldots,(\xi_1^{(m)}\to\ldots\to\xi_n^{(m)}\to\nu_m)\!\downarrow\rangle\sim\\ \langle(\chi_{\pi(1)}^{(1)}\to\ldots\to\chi_{\pi(n)}^{(1)}\to\nu_1)\!\downarrow,\ldots,(\chi_{\pi(1)}^{(m)}\to\ldots\to\chi_{\pi(n)}^{(m)}\to\nu_m)\!\downarrow\rangle\end{array}$$

since $\langle\xi_i^{(1)},\ldots,\xi_i^{(m)}\rangle\sim\langle\chi_i^{(1)},\ldots,\chi_i^{(m)}\rangle$ for $1\le i\le n$. By the induction hypothesis, there are pairs $< P_i, P_i^{-1}>$ proving $\xi_i^{(j)}\approx\chi_i^{(j)}$ for $1\le j\le m$. Let

$$\begin{array}{lcl}P & = & \lambda xy_1\ldots y_n.x(P_1^{-1}y_{\pi^{-1}(1)})\ldots(P_n^{-1}y_{\pi^{-1}(n)})\\ P^{-1} & = & \lambda xy_1\ldots y_n.x(P_{\pi(1)}y_{\pi(1)})\ldots(P_{\pi(n)}y_{\pi(n)})\end{array}$$

It is easy to verify that

$$\begin{array}{c}\vdash P\!:\!(\xi_1^{(j)}\to\ldots\to\xi_n^{(j)}\to\mu_j)\to\chi_{\pi(1)}^{(j)}\to\ldots\to\chi_{\pi(n)}^{(j)}\to\nu_j\\ \vdash P^{-1}\!:\!(\chi_{\pi(1)}^{(j)}\to\ldots\to\chi_{\pi(n)}^{(j)}\to\nu_j)\to\xi_1^{(j)}\to\ldots\to\xi_n^{(j)}\to\mu_j\end{array}$$

for $1\le j\le m$. Notice that

$$(\xi_1\to\ldots\to\xi_h\to\mu)\!\downarrow=\begin{cases}\xi_1\to\ldots\to\xi_k\to\mu & \text{if } \xi_{k+1}=\ldots=\xi_h=\omega\\ & \quad\text{and } \mu \text{ is an atomic type,}\\ \xi_1\to\ldots\to\xi_h\to\mu & \text{otherwise}\end{cases}$$

since $\xi_1,\ldots,\xi_h$ are normal types and μ is a normal singleton type. Then

$$\xi_1\to\ldots\to\xi_h\to\mu\cong(\xi_1\to\ldots\to\xi_h\to\mu)\!\downarrow,$$

and, by the typing rule $(\cong)$:

$$\begin{array}{c}\vdash P\!:\!(\xi_1^{(j)}\to\ldots\to\xi_n^{(j)}\to\mu_j)\!\downarrow\to(\chi_{\pi(1)}^{(j)}\to\ldots\to\chi_{\pi(n)}^{(j)}\to\nu_j)\!\downarrow\\ \vdash P^{-1}\!:\!(\chi_{\pi(1)}^{(j)}\to\ldots\to\chi_{\pi(n)}^{(j)}\to\nu_j)\!\downarrow\to(\xi_1^{(j)}\to\ldots\to\xi_n^{(j)}\to\mu_j)\!\downarrow\end{array}$$

for $1\le j\le m$. So $< P, P^{-1} >$ is the required pair. ◀

An immediate implication of the Soundness Theorem is that two similar types are isomorphic.

▶ **Corollary 30.** *If $\xi\sim\chi$, then $\xi\approx\chi$.*

As an example, by $\langle\omega,\varphi_1,\omega\rangle\sim\langle\omega,\varphi_1,\omega\rangle$, $\langle\varphi_2,\varphi_3,\omega\rangle\sim\langle\varphi_2,\varphi_3,\omega\rangle$, $\langle\omega,\varphi_4,\varphi_5\rangle\sim\langle\omega,\varphi_4,\varphi_5\rangle$ and the permutation $< 3,2,1 >$, one has:

$$\begin{array}{l}\langle\omega\to\varphi_2\to\psi_1,\ \varphi_1\to\varphi_3\to\varphi_4\to\psi_2,\ \omega\to\omega\to\varphi_5\to\psi_3\rangle\sim\\ \langle\omega\to\varphi_2\to\psi_1,\ \varphi_4\to\varphi_3\to\varphi_1\to\psi_2,\ \varphi_5\to\psi_3\rangle.\end{array}$$

The isomorphism between the corresponding elements of the two sequences is proved by the FHP $\lambda xy_1y_2y_3.xy_3y_2y_1$.

As another example, by $\langle\varphi_1\rangle\sim\langle\varphi_1\rangle$, $\langle\varphi_2\rangle\sim\langle\varphi_2\rangle$, $\langle\varphi_3\rangle\sim\langle\varphi_3\rangle$, $\langle\omega\rangle\sim\langle\omega\rangle$, using the permutation $< 4,1,3,2 >$, one has

$$\varphi_1\to\varphi_2\to\varphi_3\to\psi\ \sim\ \omega\to\varphi_1\to\varphi_3\to\varphi_2\to\psi.$$

The isomorphism is proved by the pair $< \lambda xy_1y_2y_3y_4.xy_2y_4y_3y_1,\ \lambda xy_1y_2y_3y_4.xy_4y_1y_3y_2 >$.

The proof of the similarity completeness, i.e. that isomorphic types have similar normal forms (Theorem 36), is based on the isomorphism characterisation given in [12]. The type system of [12] has all the rules of Figure 1, but rule $(\cong)$. The isomorphism of [12] is called here *weak isomorphism* and it is denoted by $\approx^w$. The pre-order on types of [12] (*weak pre-order*) is a restriction of the present normalisation pre-order, since in [12] no equivalence between types is assumed. For example φ and $\sigma\to\varphi$ are unrelated in the weak pre-order. The rules for normalising types in [12] are the rules $(\wedge\Rightarrow)$, $(\le\Rightarrow)$, and $(\text{ctx}\Rightarrow)$, but the application of rule $(\text{ctx}\Rightarrow)$ is subject to some conditions. For example $(\sigma\to\tau\wedge\rho)\wedge\varphi$ is a normal form in [12]. The normal form of [12] is called here *weak normal form* and denoted by $\downarrow^w$.

The paper [12] defines a similarity between types, here dubbed weak similarity ($\sim^w$) that differs from similarity ($\sim$) since the semantic type equivalence $\cong$, introduced in the current type system, makes necessary to identify semantic equivalent types. The definition of similarity pays heed to that.

▶ **Definition 31** (Weak Similarity)**.** The *weak similarity* relation between two sequences of types $\langle\sigma_1, \dots, \sigma_m\rangle$ and $\langle\tau_1, \dots, \tau_m\rangle$, written $\langle\sigma_1, \dots, \sigma_m\rangle \sim^w \langle\tau_1, \dots, \tau_m\rangle$, is the smallest equivalence relation such that:

1. $\langle\sigma_1, \dots, \sigma_m\rangle \sim^w \langle\sigma_1, \dots, \sigma_m\rangle$;
2. if $\langle\sigma_1, \dots, \sigma_i, \sigma_{i+1}, \dots, \sigma_m\rangle \sim^w \langle\tau_1, \dots, \tau_i, \tau_{i+1}, \dots, \tau_m\rangle$, then
$$\langle\sigma_1, \dots, (\sigma_i \wedge \sigma_{i+1}), \dots, \sigma_m\rangle \sim^w \langle\tau_1, \dots, (\tau_i \wedge \tau_{i+1}), \dots, \tau_m\rangle;$$
3. if $\langle\sigma_i^{(1)}, \dots, \sigma_i^{(m)}\rangle \sim^w \langle\tau_i^{(1)}, \dots, \tau_i^{(m)}\rangle$ for $1 \le i \le n$, then
$$\langle\sigma_1^{(1)} \to \dots \to \sigma_n^{(1)} \to \rho_1, \dots, \sigma_1^{(m)} \to \dots \to \sigma_n^{(m)} \to \rho_m\rangle \sim^w$$
$$\langle\tau_{\pi(1)}^{(1)} \to \dots \to \tau_{\pi(n)}^{(1)} \to \rho_1, \dots, \tau_{\pi(1)}^{(m)} \to \dots \to \tau_{\pi(n)}^{(m)} \to \rho_m\rangle,$$
where π is a permutation of $1, \dots, n$.

Weak similarity between types is trivially defined as weak similarity between unary sequences:
$$\sigma \sim^w \tau \text{ if } \langle\sigma\rangle \sim^w \langle\tau\rangle.$$

The main difference between similarity (Definition 28) and weak similarity (Definition 31) is that the first one only relates types in normal form. As a matter of fact, similarity and weak similarity are incomparable, for example $\varphi \to \varphi \sim \omega \to \varphi \to \varphi$, $\varphi \to \varphi \not\sim^w \omega \to \varphi \to \varphi$, and
$$\varphi \to \omega \to \varphi \sim^w \omega \to \varphi \to \varphi,\ \varphi \to \omega \to \varphi \not\sim \omega \to \varphi \to \varphi,$$
since $\varphi \to \omega \to \varphi$ is not a normal type.

The characterisation of type isomorphism given in [12] can be written using the present notation as:

▶ **Theorem 32.** *$\sigma \approx^w \tau$ iff $\sigma{\downarrow}^w {\sim^w} \tau{\downarrow}^w$.*

In order to use this result for showing completeness (Theorem 36) it is handy to compare $\sim$ with $\sim^w$ (Lemma 34) and $\approx$ with $\approx^w$ (Lemma 35). The following auxiliary lemma can be shown by induction on the definition of $\sim$.

▶ **Lemma 33.** *If $\langle\xi_1, \dots, \xi_i, \xi_{i+1}, \dots, \xi_m\rangle \sim \langle\chi_1, \dots, \chi_i, \chi_{i+1}, \dots, \chi_m\rangle$, then*

1. *$\langle\xi_1, \dots, \xi_i, \xi_i, \xi_{i+1}, \dots, \xi_m\rangle \sim \langle\chi_1, \dots, \chi_i, \chi_i, \chi_{i+1}, \dots, \chi_m\rangle$;*
2. *$\langle\xi_1, \dots, \xi_i, \omega, \xi_{i+1}, \dots, \xi_m\rangle \sim \langle\chi_1, \dots, \chi_i, \omega, \chi_{i+1}, \dots, \chi_m\rangle$.*

Proof. The proof is by induction over the derivation of similarity. The only interesting case is when similarity is obtained using case 3 of Definition 28. Let
$\langle(\xi_1^{(1)} \to \dots \to \xi_n^{(1)} \to \nu_1){\downarrow}, \dots, (\xi_1^{(i)} \to \dots \to \xi_n^{(i)} \to \nu_i){\downarrow}, \dots, (\xi_1^{(m)} \to \dots \to \xi_n^{(m)} \to \nu_m){\downarrow}\rangle \sim$
$\langle(\chi_{\pi(1)}^{(1)} \to \dots \to \chi_{\pi(n)}^{(1)} \to \nu_1){\downarrow}, \dots, (\chi_{\pi(1)}^{(i)} \to \dots \to \chi_{\pi(n)}^{(i)} \to \nu_i){\downarrow}, \dots, (\chi_{\pi(1)}^{(m)} \to \dots \to \chi_{\pi(n)}^{(m)} \to \nu_m){\downarrow}\rangle$
since $\langle\xi_j^{(1)}, \dots, \xi_j^{(i)}, \dots, \xi_j^{(m)}\rangle \sim \langle\chi_j^{(1)}, \dots, \chi_j^{(i)}, \dots, \chi_j^{(m)}\rangle$ for $1 \le j \le n$.
By the induction hypothesis $\langle\xi_j^{(1)}, \dots, \xi_j^{(i)}, \xi_j^{(i)}, \dots, \xi_j^{(m)}\rangle \sim \langle\chi_j^{(1)}, \dots, \chi_j^{(i)}, \chi_j^{(i)}, \dots, \chi_j^{(m)}\rangle$ for $1 \le j \le n$, which imply by the same clause:
$$\langle(\xi_1^{(1)} \to \dots \to \xi_n^{(1)} \to \nu_1){\downarrow}, \dots, \mu, \mu, \dots, (\xi_1^{(m)} \to \dots \to \xi_n^{(m)} \to \nu_m){\downarrow}\rangle \sim$$
$$\langle(\chi_{\pi(1)}^{(1)} \to \dots \to \chi_{\pi(n)}^{(1)} \to \nu_1){\downarrow}, \dots, \mu', \mu', \dots, (\chi_{\pi(1)}^{(m)} \to \dots \to \chi_{\pi(n)}^{(m)} \to \nu_m){\downarrow}\rangle$$
and

$$\langle(\xi_1^{(1)} \to \ldots \to \xi_n^{(1)} \to \nu_1)\!\downarrow, \ldots, \mu, (\xi_1^{(i)} \to \ldots \to \xi_n^{(i)} \to \omega)\!\downarrow, \ldots, (\xi_1^{(m)} \to \ldots \to \xi_n^{(m)} \to \nu_m)\!\downarrow\rangle \sim$$
$$\langle(\chi_{\pi(1)}^{(1)} \to \ldots \to \chi_{\pi(n)}^{(1)} \to \nu_1)\!\downarrow, \ldots, \mu', (\chi_{\pi(1)}^{(i)} \to \ldots \to \chi_{\pi(n)}^{(i)} \to \omega)\!\downarrow, \ldots, (\chi_{\pi(1)}^{(m)} \to \ldots \to \chi_{\pi(n)}^{(m)} \to \nu_m)\!\downarrow\rangle$$

where $\mu = (\xi_1^{(i)} \to \ldots \to \xi_n^{(i)} \to \nu_i)\!\downarrow$, $\mu' = (\chi_{\pi(1)}^{(i)} \to \ldots \to \chi_{\pi(n)}^{(i)} \to \nu_i)\!\downarrow$. This concludes the proof by observing that $(\xi_1^{(i)} \to \ldots \to \xi_n^{(i)} \to \omega)\!\downarrow = (\chi_{\pi(1)}^{(i)} \to \ldots \to \chi_{\pi(n)}^{(i)} \to \omega)\!\downarrow = \omega$. ◀

▶ **Lemma 34.** *$\sigma \sim^w \tau$ implies $\sigma\!\downarrow \sim \tau\!\downarrow$.*

Proof. One needs to show that $\langle\sigma_1, \ldots, \sigma_m\rangle \sim^w \langle\tau_1, \ldots, \tau_m\rangle$ implies

$$\langle\sigma_1\!\downarrow, \ldots, \sigma_m\!\downarrow\rangle \sim \langle\tau_1\!\downarrow, \ldots, \tau_m\!\downarrow\rangle.$$

The proof is by induction on the definition of weak similarity.
(1). $\langle\sigma_1, \ldots, \sigma_m\rangle \sim^w \langle\sigma_1, \ldots, \sigma_m\rangle$. By case 1 of Definition 28 $\langle\sigma_1\!\downarrow, \ldots, \sigma_m\!\downarrow\rangle \sim \langle\sigma_1\!\downarrow, \ldots, \sigma_m\!\downarrow\rangle$.
(2). $\langle\sigma_1, \ldots, (\sigma_i \wedge \sigma_{i+1}), \ldots, \sigma_m\rangle \sim^w \langle\tau_1, \ldots, (\tau_i \wedge \tau_{i+1}), \ldots, \tau_m\rangle$ since $\langle\sigma_1, \ldots, \sigma_i, \sigma_{i+1}, \ldots, \sigma_m\rangle \sim^w \langle\tau_1, \ldots, \tau_i, \tau_{i+1}, \ldots, \tau_m\rangle$. By the induction hypothesis

$$\langle\sigma_1\!\downarrow, \ldots, \sigma_i\!\downarrow, \sigma_{i+1}\!\downarrow, \ldots, \sigma_m\!\downarrow\rangle \sim \langle\tau_1\!\downarrow, \ldots, \tau_i\!\downarrow, \tau_{i+1}\!\downarrow, \ldots, \tau_m\!\downarrow\rangle.$$

This implies, by case 2 of Definition 28,

$$\langle\sigma_1\!\downarrow, \ldots, (\sigma_i\!\downarrow \wedge \sigma_{i+1}\!\downarrow)\!\downarrow, \ldots, \sigma_m\!\downarrow\rangle \sim \langle\tau_1\!\downarrow, \ldots, (\tau_i\!\downarrow \wedge \tau_{i+1}\!\downarrow)\!\downarrow, \ldots, \tau_m\!\downarrow\rangle,$$

which concludes the proof since $(\rho\!\downarrow \wedge \theta\!\downarrow)\!\downarrow = (\rho \wedge \theta)\!\downarrow$ for all ρ, θ.
(3).
$$\langle\sigma_1^{(1)} \to \ldots \to \sigma_n^{(1)} \to \rho_1, \ldots, \sigma_1^{(m)} \to \ldots \to \sigma_n^{(m)} \to \rho_m\rangle \sim^w$$
$$\langle\tau_{\pi(1)}^{(1)} \to \ldots \to \tau_{\pi(n)}^{(1)} \to \rho_1, \ldots, \tau_{\pi(1)}^{(m)} \to \ldots \to \tau_{\pi(n)}^{(m)} \to \rho_m\rangle$$

since $\langle\sigma_i^{(1)}, \ldots, \sigma_i^{(m)}\rangle \sim^w \langle\tau_i^{(1)}, \ldots, \tau_i^{(m)}\rangle$ for $1 \le i \le n$, where π is a permutation of $1, \ldots, n$. By the induction hypothesis $\langle\sigma_i^{(1)}\!\downarrow, \ldots, \sigma_i^{(m)}\!\downarrow\rangle \sim \langle\tau_i^{(1)}\!\downarrow, \ldots, \tau_i^{(m)}\!\downarrow\rangle$ for $1 \le i \le n$.
Let $\sharp(\rho_j) = p_j$ for $1 \le j \le m$, where $\sharp(\theta) = \begin{cases} p & \text{if } \theta\!\downarrow = \bigwedge_{\ell\in\{1,\ldots,p\}} \nu_\ell, \\ 1 & \text{if } \theta\!\downarrow = \omega. \end{cases}$
Then $\rho_j\!\downarrow = \bigwedge_{\ell\in\{1,\ldots,p_j\}} \lambda_\ell^{(j)}$ for some $\lambda_\ell^{(j)}$ $(1 \le \ell \le p_j)$ $(1 \le j \le m)$.
By Lemma 33(1)

$$\langle\underbrace{\sigma_i^{(1)}\!\downarrow, \ldots, \sigma_i^{(1)}\!\downarrow}_{p_1}, \ldots, \underbrace{\sigma_i^{(m)}\!\downarrow, \ldots, \sigma_i^{(m)}\!\downarrow}_{p_m}\rangle \sim \langle\underbrace{\tau_i^{(1)}\!\downarrow, \ldots, \tau_i^{(1)}\!\downarrow}_{p_1}, \ldots, \underbrace{\tau_i^{(m)}\!\downarrow, \ldots, \tau_i^{(m)}\!\downarrow}_{p_m}\rangle.$$

This implies by case 3 of Definition 28

$$\langle\mu_1^{(1)}\!\downarrow, \ldots, \mu_{p_1}^{(1)}\!\downarrow, \ldots, \mu_1^{(m)}\!\downarrow, \ldots, \mu_{p_m}^{(m)}\!\downarrow\rangle \sim \langle\nu_1^{(1)}\!\downarrow, \ldots, \nu_{p_1}^{(1)}\!\downarrow, \ldots, \nu_1^{(m)}\!\downarrow, \ldots, \nu_{p_m}^{(m)}\!\downarrow\rangle$$

where $\mu_\ell^{(j)} = \sigma_1^{(j)}\!\downarrow \to \ldots \to \sigma_n^{(j)}\!\downarrow \to \lambda_\ell^{(j)}$ for $1 \le \ell \le p_j$, $\nu_\ell^{(j)} = \tau_{\pi(1)}^{(j)}\!\downarrow \to \ldots \to \tau_{\pi(n)}^{(j)}\!\downarrow \to \lambda_\ell^{(j)}$ for $1 \le \ell \le p_j$. By repeated applications of case 2 of Definition 28

$$\langle\textstyle\bigwedge_{\ell\in\{1,\ldots,p_1\}} \mu_\ell^{(1)}\!\downarrow, \ldots, \bigwedge_{\ell\in\{1,\ldots,p_m\}} \mu_\ell^{(m)}\!\downarrow\rangle \sim \langle\bigwedge_{\ell\in\{1,\ldots,p_1\}} \nu_\ell^{(1)}\!\downarrow, \ldots, \bigwedge_{\ell\in\{1,\ldots,p_m\}} \nu_\ell^{(m)}\!\downarrow\rangle$$

Notice that $\bigwedge_{\ell\in\{1,\ldots,p_j\}} \mu_\ell^{(j)}\!\downarrow$ and $\bigwedge_{\ell\in\{1,\ldots,p_j\}} \nu_\ell^{(j)}\!\downarrow$ for $1 \le j \le m$ are normal types by construction. This concludes the proof, since it is easy to verify that $(\sigma_1^{(j)} \to \ldots \to \sigma_n^{(j)} \to \rho_j)\!\downarrow = \bigwedge_{\ell\in\{1,\ldots,p_j\}} \mu_\ell^{(j)}\!\downarrow$ and $(\tau_{\pi(1)}^{(1)} \to \ldots \to \tau_{\pi(n)}^{(1)} \to \rho_1)\!\downarrow = \bigwedge_{\ell\in\{1,\ldots,p_j\}} \nu_\ell^{(j)}\!\downarrow$ for $1 \le j \le m$. ◀

▶ **Lemma 35.** *If $\xi \approx \chi$, then there are $\sigma \cong \xi$, $\tau \cong \chi$ such that $\sigma \approx^w \tau$.*

Proof. By induction on the abstraction nesting in the normal forms of P,P^{-1}, where the pair $< P, P^{-1} >$ proves the isomorphism $\xi \approx \chi$. By Lemma 25(2) and Theorem 27 either $\xi = \chi = \omega$ or $\xi = \bigwedge_{i\in I} \mu_i$, $\chi = \bigwedge_{i\in I} \nu_i$ and $\mu_i \approx \nu_i$ for all $i \in I$ (note that, since $\wedge$ is commutative, one can consider the identity permutation in Theorem 27). In the first case the proof is trivial. In the second case it is enough to show that there are $\mu'_i \cong \mu_i$, $\nu'_i \cong \nu_i$ such that $< P, P^{-1} >$ proves the isomorphism $\mu'_i \approx^w \nu'_i$ for all $i \in I$. One can assume that P, P^{-1} have the same number of initial abstractions, possibly by η-expanding (Theorem 9). Let $P = \lambda x y_1 \ldots y_n.x Q_1 \ldots Q_n$ and $P^{-1} = \lambda z t_1 \ldots t_n.z R_1 \ldots R_n$. It is easy to verify that if $y_{\pi(j)}$ is the head variable of Q_j, then t_j is the head variable of $R_{\pi(j)}$ and $\lambda y_{\pi(j)}.Q_j$ is inverse of $\lambda t_j.R_{\pi(j)}$ for $1 \leq j \leq n$. Let $\mu_i \cong \xi_1 \to \ldots \to \xi_n \to \mu$ and $\nu_i \cong \chi_1 \to \ldots \to \chi_n \to \nu$. By Lemma 4(2) $x : \mu_i, y_1 : \chi_1, \ldots, y_n : \chi_n \vdash xQ_1 \ldots Q_n : \nu$ and $z : \nu_i, t_1 : \xi_1, \ldots, t_n : \xi_n \vdash zR_1 \ldots R_n : \mu$. By means of an argument similar to that one used in the proof of Lemma 25(1) there are $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n$ such that $\mu_i \cong \sigma_1 \to \ldots \to \sigma_n \to \nu$, $\nu_i \cong \tau_1 \to \ldots \to \tau_n \to \mu$. Then $\xi_j \cong \sigma_j$, $\chi_j \cong \tau_j$ for $1 \leq j \leq n$ and $\mu \cong \nu$. Moreover $y_{\pi(j)} : \chi_{\pi(j)} \vdash Q_j : \sigma_j$ and $t_j : \xi_j \vdash R_{\pi(j)} : \tau_{\pi(j)}$ for $1 \leq j \leq n$. Therefore $y_{\pi(j)} : \chi_{\pi(j)} \vdash Q_j : \xi_j$ and $t_j : \xi_j \vdash R_{\pi(j)} : \chi_{\pi(j)}$ for $1 \leq j \leq n$. This implies $\xi_j \approx \chi_{\pi(j)}$, and then by the induction hypothesis there are $\xi'_j \cong \xi_j$, $\chi'_j \cong \chi_j$ such that $\xi'_j \approx^w \chi'_{\pi(j)}$ for $1 \leq j \leq n$. One can then choose $\mu'_i \cong \xi'_1 \to \ldots \to \xi'_n \to \mu$, $\nu'_i \cong \chi'_1 \to \ldots \to \chi'_n \to \mu$, and $\sigma = \bigwedge_{i \in I} \mu'_i$, $\tau = \bigwedge_{i \in I} \nu'_i$. ◀

▶ **Theorem 36** (Completeness). *If $\sigma \approx \tau$, then $\sigma{\downarrow} \sim \tau{\downarrow}$.*

Proof. By Corollary 22, $\sigma \approx \tau$ implies $\sigma{\downarrow} \approx \tau{\downarrow}$. So, Lemma 35 assures that there are σ', τ' such that $\sigma' \cong \sigma{\downarrow}$, $\tau' \cong \tau{\downarrow}$, and $\sigma' \approx^w \tau'$. By Theorem 32 $\sigma' \approx^w \tau'$ implies $\sigma'{\downarrow^w} \sim^w \tau'{\downarrow^w}$. Lemma 34 gives $\sigma'{\downarrow} \sim \tau'{\downarrow}$, since $(\rho{\downarrow^w}){\downarrow} = \rho{\downarrow}$ for all types ρ. Lemma 23 concludes $\sigma{\downarrow} \sim \tau{\downarrow}$. ◀

The result of the present paper is summarised in the following theorem.

▶ **Theorem 37** (Main). *Two types are isomorphic iff their normal forms are similar.*

A consequence of the Main Theorem is the decidability of type isomorphism. A last lemma shows the inverse of the Soundness Theorem.

▶ **Lemma 38.** *If there is a pair of FHPs that proves $\xi_j \approx \chi_j$ for $1 \leq j \leq m$, then*

$$\langle \xi_1, \ldots, \xi_m \rangle \sim \langle \chi_1, \ldots, \chi_m \rangle.$$

Proof. By induction on the abstraction nesting in the normal forms of P,P^{-1}, where the pair $< P, P^{-1} >$ proves the isomorphisms $\xi_j \approx \chi_j$ for $1 \leq j \leq m$. As in the proof of Lemma 35, one gets $P = \lambda x y_1 \ldots y_n.xQ_1 \ldots Q_n$ and $P^{-1} = \lambda z t_1 \ldots t_n.zR_1 \ldots R_n$, where $\lambda y_{\pi(i)}.Q_i$ is inverse of $\lambda t_i.R_{\pi(i)}$ for $1 \leq i \leq n$. By Lemmas 25(2) and 33(2) one can assume that all ξ_j, χ_j are different from ω for $1 \leq j \leq m$. By Theorem 27 and case 2 of Definition 28 one can consider that all ξ_j, χ_j are singleton types for $1 \leq j \leq m$. Let $\xi_j \cong \xi^{(j)}_1 \to \ldots \xi^{(j)}_n \to \mu_j$ and $\chi_j \cong \chi^{(j)}_1 \to \ldots \chi^{(j)}_n \to \nu_j$ for $1 \leq j \leq m$. As in the proof of Lemma 35 one gets $y_{\pi(i)} : \chi^{(j)}_{\pi(i)} \vdash Q_i : \xi^{(j)}_i$ and $t_i : \xi^{(j)}_i \vdash R_{\pi(i)} : \chi^{(j)}_{\pi(i)}$ for $1 \leq j \leq m$ and $1 \leq i \leq n$. By the induction hypothesis $\langle \xi^{(1)}_i, \ldots, \xi^{(m)}_i \rangle \sim \langle \chi^{(1)}_{\pi(i)}, \ldots, \chi^{(m)}_{\pi(i)} \rangle$ for $1 \leq i \leq n$, so case 3 of Definition 28 concludes the proof. ◀

▶ **Theorem 39.** *Type isomorphism is decidable.*

Proof. By Theorem 37, for deciding if two types are isomorphic it is sufficient to check if their normal forms are similar. Normal forms can be computed owing to the fact that the normalisation rules are terminating and confluent. By Definition 28, two types are similar when the unary sequences built by these types are similar, then it enough to show that similarity of type sequences is decidable. This is done by induction on the total number of symbols in the types which occur in the two sequences. Let the sequences be $\langle \xi_1, \ldots, \xi_m \rangle$ and $\langle \chi_1, \ldots, \chi_m \rangle$. Theorem 29 implies that there is a pair of FHPs that proves $\xi_i \approx \chi_i$, for $1 \leq i \leq m$. There are the following cases (leaving out the symmetric ones):

1. If one of the ξ_i is ω, then by $\xi_i \approx \chi_i$ and Lemma 25(2) χ_i must be ω and the two sequences
$$\langle \xi_1, \ldots, \xi_{i-1}, \xi_{i+1}, \ldots, \xi_m \rangle \text{ and } \langle \chi_1, \ldots, \chi_{i-1}, \chi_{i+1}, \ldots, \chi_m \rangle$$
must be similar.

2. If one of the ξ_i is an intersection $\bigwedge_{j\in\{1,\dots,n\}} \mu_j$, then by Theorems 29 and 27 χ_i must be an intersection $\bigwedge_{j\in\{1,\dots,n\}} \nu_j$, and there are a pair of FHPs and a permutation π of $\{1,\dots,n\}$ such that the pair proves $\xi_i \approx \chi_i$, for $1 \le i \le m$, and $\mu_j \approx \nu_{\pi(j)}$, for $1 \le j \le n$. Lemma 38 implies that the two sequences
$\langle \xi_1,\dots,\xi_{i-1},\mu_1,\dots,\mu_n,\xi_{i+1},\dots,\xi_m\rangle$ and $\langle \chi_1,\dots,\chi_{i-1},\nu_{\pi(1)},\dots,\nu_{\pi(n)},\chi_{i+1},\dots,\chi_m\rangle$
are similar. Note that the number of permutations is finite and all sequences to be checked have types with lower numbers of symbols.
3. If all types in the sequences are singleton types, let for $1 \le i \le m$: $\xi_i = \xi^{(i)}_1 \to \dots \xi^{(i)}_{p_i} \to \varphi_i$ and $\chi_i = \chi^{(i)}_1 \to \dots \chi^{(i)}_{q_i} \to \psi_i$, and $n = \max\{p_1,\dots,p_m,q_1,\dots,q_m\}$. Let the similarity in question be obtained by cases 1 or 3 of Definition 28. Both cases prescribe $\varphi_i = \psi_i$ for $1 \le i \le m$ and that there must exist a permutation π of $\{1,\dots,n\}$ such that the following similarities hold:
$$\langle \hat\xi^{(1)}_j,\dots,\hat\xi^{(m)}_j\rangle \sim \langle \hat\chi^{(1)}_{\pi(j)},\dots,\hat\chi^{(m)}_{\pi(j)}\rangle \text{ for } 1 \le j \le n,$$
where $\hat\xi^{(i)}_j = \begin{cases} \xi^{(i)}_j & \text{if } j \le p_i, \\ \omega & \text{otherwise.}\end{cases}$ $\quad \hat\chi^{(i)}_j = \begin{cases} \chi^{(i)}_j & \text{if } j \le q_i, \\ \omega & \text{otherwise.}\end{cases}$
It is easy to check that any pair of the so obtained sequences has a number of symbols less than the one of the original sequence.
If instead the similarity in question is obtained by case 2 of Definition 28, one has
$$\langle \xi_1,\dots,\xi_i,\xi,\xi_{i+1},\dots,\xi_m\rangle \sim \langle \chi_1,\dots,\chi_i,\chi,\chi_{i+1},\dots,\chi_m\rangle$$
and $(\xi_i \wedge \xi){\downarrow} = \xi_i$, $(\chi_i \wedge \chi){\downarrow} = \chi_i$. Then one starts from the sequences obtained by removing ξ, χ and iterate this process until the similarity is obtained by cases 1 or 3 of Definition 28. ◀

Note that in the system of [12], in which only intersection types are considered, decidability is a rather immediate consequence of the decidability of type assignment for normal forms proved in [19]. This result does not seem easily extensible to the present type assignment system.

6 Conclusion

In this paper type isomorphism is studied in the setting of an intersection type system in which all types have a functional character. An equivalence relation is introduced that equates any atomic type φ to an arrow type from a distinguished atom ω to φ itself. In the derived type system all type isomorphisms related to the set theoretic properties of intersection, in particular idempotence, commutativity and associativity, are realised by λ-terms of proper type. These isomorphisms, together with other two isomorphisms that express properties of functional interpretation and inclusion of types, are preserved by every context. It follows that semantic type equality in all standard models of intersection types entails type isomorphism.

The type equivalence defined in this paper can be validated in the model D_∞ [21] by an interpretation in which each type denotes an open set in the Scott topology. One could then use the present type system to investigate the isomorphisms between open sets in D_∞. The problem of finding a model which validates all and only the type isomorphisms studied in this paper remains open.

We plan to study type isomorphism in other theories of intersection and union types, in particular in the theories providing models of the call-by-value λ-calculus. An interesting observation is that, with the typing rules given in [16] for the type constant ν, all intersections of arrow types ending by ν are isomorphic to ν. In fact the rule $\Gamma \vdash \lambda x.M : \nu$ allows one to derive both

$$x\!:\!\nu \vdash \lambda y_1 \dots y_m.xy_1 \dots y_m\!:\!\sigma_1 \to \dots \to \sigma_n \to \nu$$
$$\text{and } x\!:\!\sigma_1 \to \dots \to \sigma_n \to \nu \vdash \lambda y.xy\!:\!\nu,$$

for any $n \le m$ and arbitrary $\sigma_1, \dots, \sigma_n$. Notably, the type theory of [16] gives a model of the call-by-value λ-calculus.

Acknowledgements. We gratefully acknowledge the anonymous referees for their careful reading of our paper and their many useful remarks and suggestions. In particular Lemma 35 strongly improved. We thank Alejandro Díaz-Caro for his useful advices.

References

1 Fabio Alessi, Mariangiola Dezani-Ciancaglini, and Furio Honsell. Inverse limit models as filter models. In D. Kesner, F. van Raamsdonk, and J. Wells, editors, *Higher-Order Rewriting*, pages 3–25. RWTH Aachen, 2004.

2 Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *The Journal of Symbolic Logic*, 48(4):931–940, 1983.

3 Henk Barendregt, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types*. Perspectives in Logic. Cambridge, 2013.

4 Kim Bruce, Roberto Di Cosmo, and Giuseppe Longo. Provable isomorphisms of types. *Mathematical Structures in Computer Science*, 2(2):231–247, 1992.

5 Kim Bruce and Giuseppe Longo. Provable isomorphisms and domain equations in models of typed languages. In R. Sedgewick, editor, *Symposium on the Theory of Computing*, pages 263–272. ACM Press, 1985.

6 Mario Coppo, Mariangiola Dezani-Ciancaglini, Furio Honsell, and Giuseppe Longo. Extended type structures and filter lambda models. In G. Lolli, G. Longo, and A. Marcja, editors, *Logic Colloquium*, pages 241–262. North-Holland, 1984.

7 Mario Coppo, Mariangiola Dezani-Ciancaglini, Ines Margaria, and Maddalena Zacchi. Towards isomorphism of intersection and union types. In S. Graham-Lengrand and L. Paolini, editors, *Intersection Types and Related Systems*, volume 121 of *EPTCS*, pages 58–80, 2013.

8 Mario Coppo, Mariangiola Dezani-Ciancaglini, Ines Margaria, and Maddalena Zacchi. Isomorphism of intersection and union types. *Mathematical Structures in Computer Science*, 2014. To appear.

9 Mario Coppo, Mariangiola Dezani-Ciancaglini, Ines Margaria, and Maddalena Zacchi. On isomorphism of "functional" intersection and union types. In J. Rehof, editor, *Intersection Types and Related Systems*, EPTCS, 2014. To appear.

10 Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

11 Mariangiola Dezani-Ciancaglini. Characterisation of normal forms possessing an inverse in the $\lambda\beta\eta$-calculus. *Theoretical Computer Science*, 2(3):323–337, 1976.

12 Mariangiola Dezani-Ciancaglini, Roberto Di Cosmo, Elio Giovannetti, and Makoto Tatsuta. On isomorphisms of intersection types. *ACM Transactions on Computational Logic*, 11(4):1–22, 2010.

13 Roberto Di Cosmo. Second order isomorphic types. A proof theoretic study on second order λ-calculus with surjective pairing and terminal object. *Information and Computation*, 119(2):176–201, 1995.

14 Roberto Di Cosmo. A short survey of isomorphisms of types. *Mathematical Structures in Computer Science*, 15:825–838, 2005.

15 Alejandro Díaz-Caro and Gilles Dowek. Simply typed lambda-calculus modulo type isomorphisms. `https://who.rocq.inria.fr/Alejandro.Diaz-Caro/stmti.pdf`, 2014.

16 Lavinia Egidi, Furio Honsell, and Simona Ronchi Della Rocca. Operational, Denotational and Logical Descriptions: a Case Study. *Fundamenta Informaticae*, 16(1):149–169, 1992.

17 Marcelo Fiore, Roberto Di Cosmo, and Vincent Balat. Remarks on isomorphisms in typed lambda calculi with empty and sum types. *Annals of Pure and Applied Logic*, 141(1–2):35–50, 2006.

18 Maxwell H. A. Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics*, 43(2):223–243, 1942.

19 Simona Ronchi Della Rocca. Principal type scheme and unification for intersection type discipline. *Theoretical Computer Science*, 59(1–2):1–29, 1988.

20 Richard Routley and Robert K. Meyer. The semantics of entailment III. *Journal of Philosophical Logic*, 1:192–208, 1972.

21 Dana Scott. Continuous lattices. In F. W. Lawvere, editor, *Toposes, Algebraic Geometry, and Logic*, number 274 in Lecture Notes in Mathematics, pages 97–136. Springer-Verlag, 1972.

22 Sergei Soloviev. The category of finite sets and cartesian closed categories. *Journal of Soviet Mathematics*, 22(3):1387–1400, 1983. English translation of the original paper in Russian published in Zapiski Nauchnych Seminarov LOMI, v.105, 1981.

23 Sergei Soloviev. A complete axiom system for isomorphism of types in closed categories. In A. Voronkov, editor, *Logic Programming and Automated Reasoning*, volume 698 of *LNCS*, pages 360–371. Springer-Verlag, 1993.