Proof-functional connectives and realizability

FRANCO BARBANERA* SIMONE MARTINI†

January 1993

Abstract

The meaning of a formula built out of proof-functional connectives depends in an essential way upon the intensional aspect of the proofs of the component subformulas. We study three such connectives, strong equivalence (where the two directions of the equivalence are established by mutually inverse maps), strong conjunction (where the two components of the conjunction are established by the same proof) and relevant implication (where the implication is established by an identity map). For each of these connectives we give a type-assignment system, a realizability semantics, and a completeness theorem. This form of completeness has a corollary the semantic completeness of the type-assignment system.

1 Introduction

Usual connectives of classical propositional logic are *truth-functional*, the meaning of a compound formula being dependent only on the truth value of its subformulas. In the constructive analysis of logical constants, on the other hand, it is the concept of *proof* (or *justification*, or *reason*, or *realizer*) for a formula to become of paramount importance. In the Brouwer-Heyting-Kreisel interpretation, for instance, a realizer for A & B is given once a realizer for A and a realizer for B are given, while a realizer for $A \rightarrow B$ is given once we have an (effective) way of transforming any realizer for A into a realizer for B. Several variations on this paradigm have been studied, trying to better specify what has to be counted as a valid "realizer" (for instance, which operations are available on them), or, once fixed a formal notion of realizer, trying to identify the relations between constructive proofs and properties of their (formal) realizers. The *intensional character* of realizers, however, has no part in these analysis. Beside the fact that it has to be assumed that we are able to recognize that a realizer realizes something, no other requirement is usually issued. In particular no commitment is made on the ability to recognize whether two realizers are equal.

Provability for *proof-functional connectives* [Lopez-Escobar 85], on the other hand, depends on the actual shape of the proofs (realizers) for the components of a complex formula. This connectives arise as an attempt to clarify the logical status of constructions introduced, with different aims, in λ-calculus and theoretical computer science. It is well known that, by the Curry-Howard isomorphism, any type of (first or second order) λ-calculus corresponds to a formula of (first or second order) intuitionistic logic, whose proofs correspond to the terms of that type. In [Coppo & Dezani 80] a new type system for λ-calculus was introduced, whose notable characteristic was the presence of an *intersection* ($\wedge$) type constructor. It is a very important system for the study of typed and untyped λ-calculus (some of its features will be listed in Section 4), but it does not fit into the Curry-Howard paradigm. While the elimination rule for intersection types exactly matches the elimination rule for conjunction, the introduction rule

$$\frac{M : \alpha \quad M : \beta}{M : \alpha \wedge \beta} \tag{1}$$

*Università di Torino, Dipartimento di Informatica, Corso Svizzera 185, 10149 Torino, Italy. E-mail: barba@di.unito.it

†Università di Pisa, Dipartimento di Informatica, Corso Italia 40, 56125 Pisa, Italy. E-mail: martini@di.unipi.it

prevents the interpretation of $\alpha \wedge \beta$ as "α *and* β", since its realizer M has to be the same realizer of both α and β. It is for giving a logical account of this type constructor that [Pottinger 80] introduces the proof-functional connective of *strong conjunction*, with the informal realizability analysis "to assert A & B is to assert that one has a reason for asserting A which is also a reason for asserting B."

Proof-functional connectives can be used to study several other constructions. In this paper, in addition to *strong conjunction* (Section 4), we will study *strong equivalence* (Section 3), a connective arising from the study of provable isomorphisms in λ-theories, and *relevant implication* (Section 5), a well-known connective with also some computer science motivations in the background.

Main focus of the paper will be the realizability analysis of such connectives, outlined in the following section. Indipendently of one's opinion about their relevance *per se*, however, proof-functional connectives bears an interest also from a foundational point of view, especially for the relation between mathematical concepts and their formalization, a subject we will briefly touch in the rest of this section.

As soon as the (informal) definition of a proof-functional connective comes into one's mind, indeed, it is clear that even the problem of its "soundness" — that is, of spelling it out in a mathematically acceptable way — reduces to the possibility of considering decidable relations, like (extensional or intensional) equality, between constructions (representatives of constructive proofs). If one agrees with Brouwer's viewpoint on the non formalizability of mathematical thought, one could simply get rid of the problems by simply stating that one cannot go beyond recognizing that a construction (realizer) justifies a formula. If, however, we divert from Brouwer's extremist point of view, and build formal systems — which we believe embodying our logical reasoning — relational aspects between proofs have to be given full citizenship into our logical investigations. Accepting this last fact leads immediately to Lopez-Escobar's consideration [Lopez-Escobar 85] that

> accepting proof-functional connectives, such as strong conjunction, requires rejecting the assumption that a construction proves a unique sentence and thus forces us to distinguish between a construction as an object and a construction as a method.

To see constructions as methods means to consider them separately from the sentences they prove;hence they are formalized separately and separately they are included in rules for proof-functional connectives. In this way, such formal systems come to be more strongly dependent than other sorts of logical systems on how one decides to formalize constructions and the intended relations between them. This is well exemplified by the rule rule (1), which will be shown to be a sound introduction rule for strong conjunction. In formalizing with this rule the notion of a construction proving two different sentences, one makes the choice of formalizing equality between constructions as *syntactical* equality between λ-terms. Moreover, one makes the further choice, among the many possible and different ones, of considering the logical steps regarding strong conjunction as being, in a sense, non-existent from the point of view of constructions, since in the term corresponding to the construction for $\alpha \wedge \beta$ no information is contained about the fact that to get to this point we passed through one and the same construction for α and β. This makes us recall that formalization, in general, necessarily brings with it elements external to what we are trying to formalize and consequently enforces, somewhat paradoxically, Brouwer's viewpoint on unformalizability of mathematical reasoning seen as pure mental process in its widest sense. Hoping things to be otherwise would lead to the well known paradox of the dove dreaming of freely flying into a sky where the air which obstacles its flight is absent, so forgetting that it is the air which enables its flight.

Thus, even if not supporting the formalistic viewpoint, one could hardly disregard the fact that the interest of logic, in particular if one deals with proof-functional notions, is somewhat interleaved between the mathematical concept and its formalization. About the latter, in a proof-functional context, the notion of "formal" rule is relevant and surely cannot be an absolute one, being tightly tied at least to the rest of the formalization choices such as, in the previously considered case, how to formalize constructions and equality between them. It is then misleading to follow the (frequent) judgment of people who have a (too) clear (to be true) idea of what a logical rule must be. Rules like (1), probably are not rules at all for them. But, then, can one deny that refusing them would considerably limit the scope of the logical investigation and debate?

We shall refrain, in the rest of the paper, from this sort of considerations, but stress how they testify how the notion of proof functional connective, besides its relations with computer science, far from being a notion without citizenship into the world of logic investigation, can, in the worst of cases, make debates arise. Debates that have hardly been unfruitful for logics.

Section 2 rests on [Martini 92], while Section 3 borrows results and techniques from [Alessi & Barbanera 91].

2 Provable realizability

The informal discussion of realizability of the introduction can be pushed one step further, by formalizing the requirements on the provability of compound propositional formulas inside a formal logical system. Following Kleene's approach, this is achieved by associating to any propositional formula A a predicate $\mathbf{r}_A[x]$ together with suitable axioms formalizing that x realizes the formula A. The definition of $\mathbf{r}_A[x]$ is given inductively on the structure of A, while the intended domain for x is some universe of "realizers" for which the notions of applying a proof (of an implication) to another, or splitting a proof (of a conjunction) in the proofs of its components make sense. One obvious choice is to take any (partial) combinatory algebra $\mathcal{A} = (A, \cdot)$. Notice that we did not require a typed combinatory algebra; realizers come from an untyped universe, and, as a consequence, a single element may realize different formulas.

The following definition introduces realizability predicates for minimal propositional logic. Extensions to logics with other connectives will be given in the following sections. Assume first that for any propositional variable ϕ there exists a corresponding (atomic) realizability predicate P_ϕ. Application between realizers will be denoted by juxtaposition.

Definition 2.1 *For any positive propositional formula A, its* realizability predicate*, $\mathbf{r}_A$, is inductively defined as follows.*

- $\mathbf{r}_\phi[x] \equiv P_\phi(x)$
- $\mathbf{r}_{\sigma\to\tau}[x] \equiv \forall y(\mathbf{r}_\sigma[y] \to \mathbf{r}_\tau[xy])$

The analysis of provable realizability for a propositional formal system $\mathcal{P}$, then, consists in studying the relation between formal provability in $\mathcal{P}$ of a formula A and the provability of the formula $\exists x \mathbf{r}_A[x]$, in an another formal system $\mathcal{K}$. A standard completeness result for $\mathcal{P}$ with respect to provable realizability in $\mathcal{K}$ assumes then the form

$$\text{For all formulas } A: \ \vdash_{\mathcal{P}} A \iff \vdash_{\mathcal{K}} \exists x \mathbf{r}_A[x].$$

A result of this kind can be interpreted as *evidence* of the fact that $\mathcal{P}$ embodies (from the "point of view" of system $\mathcal{K}$) our informal specification of the connectives composing the well formed formulas of $\mathcal{P}$. Suppose, now, that $\mathcal{P}$ has connectives somewhat related to "constructivism" and, in particular, to the intensional behaviour of proofs, like the proof functional connectives; it is then natural to require $\mathcal{P}$ to be a system dealing, in some way, with "representations" of proofs, or proof-*skeletons*. If such proofs are encoded in a (partial) combinatory algebra structure (e.g. λ-terms representing skeletons of proofs in natural deduction), then completeness can be expressed more perspicuously as

$$\text{For all formulas } A: \ \vdash_{\mathcal{P}} t : A \iff \vdash_{\mathcal{K}} \mathbf{r}_A[t], \tag{2}$$

where we have denoted with $\vdash_{\mathcal{P}} t : A$ the judgment that t codes (part of) a proof in $\mathcal{P}$ of A.

Given a system $\mathcal{P}$, for which we want to investigate realizability, we have a choice both on the system $\mathcal{K}$ and on the algebra of realizers $\mathcal{A}$.

For what concerns $\mathcal{A}$, a standard and classical choice is to take Kleene's partial applicative structure $(\omega, \cdot)$ where $n \cdot m = \phi_n(m)$ for some suitable enumeration ϕ_n of the partial recursive functions. Another possible choice, the one used in the present paper, is to consider the total combinatory algebra of untyped λ-terms modulo β-equality. Recall that λ-terms are those expressions generated by the following grammar

$$t ::= x \mid \lambda x.t \mid (tt)$$

modulo the equality given by the axiom

$$(\lambda x.t)s =_\beta t[x \leftarrow s] \qquad (\beta)$$

and the other standard rules for making $=_\beta$ a congruence; we shall omit the subscript β when there is no danger of confusion. As a variant of this approach, in Section 4.4 we will also deal with Combinatory Logic terms equipped with the so-called combinatory β-equality.

For what concerns the system $\mathcal{K}$, its choice essentially depends on the notions $\mathcal{P}$ is intended to formalize and on the conceptual use one wish to make of a completeness result.

For instance, the standard choice for the investigation of provable realizability for intuitionistic propositional logic is to take the classical predicate calculus (thus, $\mathcal{P}$ is **IL**, $\mathcal{K}$ is **CL** and $\mathcal{A}$ is Kleene's structure), for which [Mints 89] proved completeness. The more $\mathcal{K}$ is powerful and "far" from $\mathcal{P}$, the stronger a completeness result appears as evidence that the formalization given by $\mathcal{P}$ to some logical concept is sound and robust. The choice of classical logic for $\mathcal{K}$, however, does not give any additional bonus over the choice of intuitionistic logic. Due to a result of Mints, in fact, for judgments of the form of those involved in the completeness property (equivalence (2) above), the two systems coincide (see 3.2). This is the reason for which, in the following, we will always take intuitionistic predicate calculus for $\mathcal{K}$.

We will carry over this kind of analysis for three type-assignment systems, proving for each of them a completeness theorem of the form (2). This result has as a corollary, the semantical completeness of the type-assignment system we started with (system $\mathcal{P}$ in (2)). This is detailed in section 4.4.2 for strong conjunction, where we obtain an easy proof of the completeness of the intersection type discipline.

In the following sections we will find useful to switch between different formulation of the same calculus. We denote with **LK** (**LJ**, respectively) Gentzen sequent calculus for classical (intuitionistic) logic with equality, where untyped λ-terms are used as terms and equality is β-conversion; with **NK** and **NJ** we denote the corresponding natural deduction systems. The following is a standard result, the presence of λ-terms being irrelevant.

Proposition 2.2 *(i)* $\Gamma \vdash_{\mathbf{LK}} A \iff \Gamma \vdash_{\mathbf{NK}} A$.
(ii) $\Gamma \vdash_{\mathbf{LJ}} A \iff \Gamma \vdash_{\mathbf{NJ}} A$.

Notation We will use & for intuitionistic conjunction and $\wedge$ for strong conjunction (or intersection for types).

3 Strong Equivalence

We consider in this section a proof functional connective for double implication. Although many connectives of this kind could be defined (e.g. requiring the two directions of the implication to be obtained by the same proof, cf. [Lopez-Escobar 85]), the one we are interested in, and that we call *strong equivalence* and denote with $\cong$, comes from independent work on *provable isomorphisms* in typed λ-calculi ([Bruce & Longo 85], [Bruce *et al.* 92] and [Di Cosmo 91]).

A justification for the propositional formula $A \cong B$ is given iff we have a justification t for $A \rightarrow B$ (that is a way of mapping any justification a for A into a justification ta for B), a justification s for $B \rightarrow A$, and moreover for any justification a for A, and any justification b for B, we are able to recognize that a and $s(ta)$ are the same justification for A, and that b and $t(sb)$ are the same justification for B. Strong equivalences, thus, are those double implications for which we can give proofs of the two "directions" that are one the inverse of the other.

The reader will have noticed how isomorphisms come in. Indeed, given a typed λ-calculus $\boldsymbol{\lambda_T}$, two types A and B are provably isomorphic iff there are two terms $f : A \rightarrow B$ and $g : B \rightarrow A$ such that $\boldsymbol{\lambda_T} \vdash f \circ g = id_B$ and $\boldsymbol{\lambda_T} \vdash g \circ f = id_A$. The problem attacked in the cited papers is to characterize by means of an equational theory all the isomorphisms provable in a specific $\boldsymbol{\lambda_T}$. ([Bruce & Longo 85] deals with simply typed λ-calculus; [Bruce *et al.* 92] with the simply typed λ-calculus extended with (surjective) pairs and a terminal type; [Di Cosmo 91] with second order λ-calculus.) From the computer science point of view, these results help to analyse extensions to typed functional languages, where in some cases it may be useful to relax the constraint on the application of a function to its arguments, allowing arguments from *isomorphic* types. (Indeed some preliminary studies [Rittri 90] on this subject had a key retrieval problem for library functions as explicit motivation.). From the logical point of view, by the Curry-Howard isomorphism, these results exactly characterize strong equivalences in fragments of intuitionistic logic. It appeared then natural to investigate how this new connective behaves with respect to provable realizability.

The informal analysis of the realizability of strong equivalence we have just made can be formalized along the lines of Section 2. Note, first, that our algebra of realizers has to have a way of pairing realizers, since a realizer for $A \cong B$ is given via two different realizers, one for $A \rightarrow B$ and one for $B \rightarrow A$. Instead of taking pure λ-terms, then, it is convenient to extend our combinatory algebra with explicit pairs and projections:

$$t ::= x \mid \lambda x.t \mid (tt) \mid < t,t > \mid \mathsf{f}\, t \mid \mathsf{s}\, t.$$

Application associates to the left. Conversion between λ-terms is given by the following rules

$$\begin{array}{lcll} (\lambda x.t)s & = & t[x \leftarrow s] & (\beta) \\ \mathsf{f} < s,t > & = & s & (\pi_1) \\ \mathsf{s} < s,t > & = & t & (\pi_2)\ , \end{array}$$

together with rules for making $=$ a congruence; we will sometime write $=_{\beta,\pi}$ for this relation. Composition of terms is defined as usual: $M \circ N \equiv \lambda x.M(Nx)$; id denotes the term $\lambda x.x$.

With this machinery, we can extend Definition 2.1 with the following.

Definition 3.1 *The realizability predicate for strong equivalence is defined as*

$$\mathbf{r}_{A\cong B}[x] \equiv \mathbf{r}_{A\rightarrow B}[\mathsf{f}\, x] \ \&\ \mathbf{r}_{B\rightarrow A}[\mathsf{s}\, x] \ \&\ ((\mathsf{f}\, x) \circ (\mathsf{s}\, x) = id) \ \&\ ((\mathsf{s}\, x) \circ (\mathsf{f}\, x) = id)$$

Next sections will introduce a very natural formal system for strong equivalence and then show its completeness.

3.1 A Type Assignment System for Strong Equivalence

Strong equivalence can be seen as a new type constructor for λ-calculus, embodying into type assignment rules the realizability analysis discussed above.

Types are given by the following grammar (ϕ ranges over atomic types)

$$T ::= \phi \mid T \rightarrow T \mid T \cong T. \tag{3}$$

A *basis* is a finite sequence $x_1 : A_1, \ldots, x_n : A_n$, where the x's are distinct variables and the A's are types; we usually denote a basis with Γ.

Definition 3.2 *The type assignment system for Strong Equivalence is given in Figure 1, defining by mutual induction the relations $\Gamma \vdash t : A$ (a* type judgment*, to be read "t has type A in basis Γ") and $\Gamma \vdash t = s$ (an* equality judgment*, to be read "t and s are equal terms in basis Γ).*

Remark 3.1 *1. In force of rules (Eq), the relation $t : A$ holds up to $=_{\beta,\pi}$, a fact which is essential for completeness.*

2. A type $A \cong B$ is inhabited *(that is, $\vdash s : A \cong B$ for some term s), iff A and B are provable isomorphic, iff $A = B$ is derivable in Bruce and Longo's equality theory for simply typed λ-calculus.*

$$\Gamma\ x : A \vdash x : A \quad (\mathrm{Ax})$$

$$\frac{\Gamma \vdash s : A \to B \quad \Gamma \vdash t : B \to A \quad \Gamma \vdash s \circ t = id \quad \Gamma \vdash t \circ s = id}{\Gamma \vdash \langle s, t\rangle : A \cong B} \cong\mathcal{I}$$

$$\frac{\Gamma \vdash s : A \cong B}{\Gamma \vdash \mathsf{f}\, s : A \to B} \cong\mathcal{E}^f \qquad \frac{\Gamma \vdash s : A \cong B}{\Gamma \vdash \mathsf{s}\, s : B \to A} \cong\mathcal{E}^s$$

$$\frac{\Gamma \vdash t : A \cong B}{\Gamma \vdash (\mathsf{f}\, t) \circ (\mathsf{s}\, t) = id} \cong\mathcal{E}^1_= \qquad \frac{\Gamma \vdash t : A \cong B}{\Gamma \vdash (\mathsf{s}\, t) \circ (\mathsf{f}\, t) = id} \cong\mathcal{E}^2_=$$

$$\frac{\Gamma\ x : A \vdash b : B}{\Gamma \vdash \lambda x.b : A \to B} \to\mathcal{I} \qquad \frac{\Gamma \vdash b : A \to B \quad \Gamma \vdash a : A}{\Gamma \vdash b(a) : B} \to\mathcal{E}$$

$$\frac{\Gamma\ x : A \vdash tx : B}{\Gamma \vdash t : A \to B} (\eta) \qquad \frac{\begin{array}{c}\Gamma \vdash \mathsf{f}\, t : A \to B \quad \Gamma \vdash \mathsf{s}\, t : B \to A \\ \Gamma \vdash (\mathsf{f}\, t) \circ (\mathsf{s}\, t) = \mathrm{id} \quad \Gamma \vdash (\mathsf{s}\, t) \circ (\mathsf{f}\, t) = \mathrm{id}\end{array}}{\Gamma \vdash t : A \cong B} (SP)$$

$$\frac{s =_{\beta,\pi} t}{\Gamma \vdash s = t} (\mathrm{Eq}_1) \qquad \frac{\Gamma \vdash s : A \quad \Gamma \vdash s = t}{\Gamma \vdash t : A} (\mathrm{Eq}_2)$$

$$\frac{\Gamma \vdash s = u \quad \Gamma \vdash u = t}{\Gamma \vdash s = t} (\mathrm{Eq\!-\!t}) \qquad \frac{\Gamma \vdash s = t}{\Gamma \vdash t = s} (\mathrm{Eq\!-\!s})$$

Figure 1: TYPE-ASSIGNMENT SYSTEM FOR STRONG EQUIVALENCE

3.2 Provable realizability for strong equivalence

Following [Mints 89], and according to completeness (2), we will show that, for any A, $\mathbf{r}_A[t]$ is provable in intuitionistic predicate calculus **LJ** iff the type assignment system introduced in Definition 3.2 proves that t can be given the type A.

For the purpose of this section, axioms of **LJ** are of the form $\vdash_{LJ} s = t$ together with a justification for $s =_{\beta,\pi} t$, or of the form $s = t, P(s) \vdash_{LJ} P(t)$. Recall (e.g. [Takeuti 75]) that for this system we can prove the elimination of *essential* cuts, a cut being inessential if the cut formula is an equation.

All proofs and techniques are modifications of Mints' ones, those for strong equivalence being similar to those for (standard) conjunction. The novelty is the presence of equations in the premises of sequents (coming from left introduction of strong equivalences), that forces us to choose a less compact form for the axioms and to consider normal proofs with inessential cuts (axioms in [Mints 89] are sequents $\Delta, P(s) \vdash P(s')$, together with a justification for $s = s'$.) In the following we will only give the cases for axioms and strong equivalences, all the other being either identical to Mints' treatment, or trivially modifiable.

The main idea for Mints' result is a careful analysis of the structure of **LJ** proofs of sequents expressing provable realizability. The crucial points are expressed by the following proposition, showing that any sequent occurring in a proof of a formula of the form $\mathbf{r}_A[t]$ has a very special shape (thus allowing an easy inductive proof of theorem 3.4.)

The following proposition considers (essential-) cut-free proofs of sequents of the form $\vdash_{LJ} \mathbf{r}_F[t]$. By the subformula property, such derivations will only contain either formulas of the form $\mathbf{r}_A[t]$, with A subformula of F, or equations $t = s$ (coming from formulas $\mathbf{r}_{A\cong B}[t]$), or instances

$$\mathbf{r}_A[s] \to \mathbf{r}_B[ts] \tag{4}$$

(coming from formulas of the kind $\forall y(\mathbf{r}_A[y] \to \mathbf{r}_B[ty])$.) By the permutability of inferences in sequent proofs ([Kleene 52], theorem 2, pag. 18) we can assume that in any proof of the kind at hand, any $\to$-rule (introducing a formula of the shape 4) is immediately followed by a $\forall$-rule on the same side, thus producing proof structures of the shape

$$\frac{\Gamma, \mathbf{r}_A[y] \vdash_{LJ} \mathbf{r}_B[ty]}{\Gamma \vdash_{LJ} \mathbf{r}_{A\to B}[t]} \vdash\to \quad \text{and} \quad \frac{\Gamma \vdash_{LJ} \mathbf{r}_A[y] \quad \Gamma, \mathbf{r}_B[ty] \vdash_{LJ} C}{\Gamma, \mathbf{r}_{A\to B}[t] \vdash_{LJ} C} \to\vdash .$$

Similarly, we can always rearrange a proof in such a way that all & -rules introducing (on the left or on the right) a formula $\mathbf{r}_{A\cong B}[t]$ occur together, giving rise to proof structures we will call $\vdash\cong$ and $\cong\vdash$. Once these proof structures have been isolated, the proof of the following proposition is straightforward; in its statement, these proof structures have to be counted as single inferences .

Notation: (i) $\mathbf{t}$ denotes a sequence of terms; $\mathbf{E}$ denotes a sequence of equations.

(ii) π denotes a finite (possibly empty) sequence of projections (e.g. $\mathsf{f\,s\,f\,f}$).

(iii) If $\pi \equiv p_1 \dots p_n$, then $\pi(\mathbf{t})$ stands for $p_1(\dots(p_n(\mathbf{t}))\dots)$.

(iv) A term is a *head projection term* iff it is of the shape $\pi_n(\dots\pi_1(\pi(x\mathbf{t})\mathbf{t}_1)\dots\mathbf{t}_n)$, for $n \geq 0$.

(v) We will write $\mathbf{r}_{\mathbf{A}}[\mathbf{t}] \vdash \mathbf{r}_B[s]$ for $\mathbf{r}_{A_1}[t_1], \dots, \mathbf{r}_{A_n}[t_n] \vdash \mathbf{r}_B[s]$

Proposition 3.3 *Let F be any propositional formula. Any sequent occurring in a cut-free derivation of the sequent $\vdash_{LJ} \mathbf{r}_F[t]$ is of the form*

$$\mathbf{r}_{F_1}[h_1], \dots, \mathbf{r}_{F_n}[h_n], E_1, \dots, E_k \vdash_{LJ} \mathbf{r}_G[s] \tag{5}$$

or of the form

$$\mathbf{r}_{F_1}[h_1], \dots, \mathbf{r}_{F_n}[h_n], E_1, \dots, E_k \vdash_{LJ} Q \tag{6}$$

where $n, k \geq 0$, s is a term, the h_i's are head projection terms, and the E_i's and Q are equations between terms.

Proof: By induction on the length of the derivation. The basis and the case of an inessential cut are obvious.

Case $\cong\vdash$: We have four cases:

$$\frac{\mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{r}_{A\to B}[\mathsf{f}\,h'], \mathbf{E} \vdash \mathbf{r}_G[s]}{\mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{r}_{A\cong B}[h'], \mathbf{E} \vdash \mathbf{r}_G[s]}$$

and its symmetric; or

$$\frac{\mathbf{r}_{\mathbf{F}}[\mathbf{h}], (\mathsf{f}\,h') \circ (\mathsf{s}\,h') = id, \mathbf{E} \vdash \mathbf{r}_G[s]}{\mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{r}_{A\cong B}[h'], \mathbf{E} \vdash \mathbf{r}_G[s]}$$

and its symmetric. In all the cases the premises are of the required form.

Case $\vdash\cong$:

$$\frac{\begin{array}{cc}\mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{E} \vdash \mathbf{r}_{A\to B}[\mathsf{f}\,t] & \mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{E} \vdash (\mathsf{f}\,t) \circ (\mathsf{s}\,t) = id \\ \mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{E} \vdash \mathbf{r}_{B\to A}[\mathsf{s}\,t] & \mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{E} \vdash (\mathsf{s}\,t) \circ (\mathsf{f}\,t) = id\end{array}}{\mathbf{r}_{\mathbf{F}}[\mathbf{h}], \mathbf{E} \vdash \mathbf{r}_{A\cong B}[t]}\ . \tag{7}$$

□

Remark 3.2 *Although we have stated the previous proposition in terms of the intuitionistic system* **LJ**, *the completeness theorem holds also for classical predicate calculus. Following [Mints 89], in fact, in [Martini 92] it is proved that*

$$\mathbf{r}_{\mathbf{A}}[\mathbf{t}], \mathbf{E} \vdash_{LJ} \mathbf{r}_B[s] \Leftrightarrow \mathbf{r}_{\mathbf{A}}[\mathbf{t}], \mathbf{E} \vdash_{LK} \mathbf{r}_B[s].$$

Theorem 3.4 *(Completeness)*
(i) $x_1 : A_1, \dots, x_n : A_n \vdash_{\cong} t : B \iff r_{A_1}[x_1], \dots, r_{A_n}[x_n] \vdash_{LJ} r_B[t]$
(ii) $x_1 : A_1, \dots, x_n : A_n \vdash_{\cong} s = t \iff r_{A_1}[x_1], \dots, r_{A_n}[x_n] \vdash_{LJ} s = t$

Proof: (i) and (ii) are proved by combined induction on the derivations; for **LJ** it will be convenient to argue instead in its natural deduction version, **NJ**.
($\Rightarrow$):
Basis: If $x_1 : A_1, \ldots, x_n : A_n \vdash_{\cong} x_i : A_i$, then an inessential cut from $\vdash_{LJ} x_i = x_i$ and $x_i = x_i, r_{A_i}[x_i] \vdash_{LJ} r_{A_i}[x_i]$, followed by $n-1$ weakenings gives the thesis.

Case $\cong \mathcal{I}$: By inductive hypothesis, for $\Gamma' \equiv r_{A_1}[x_1], \ldots, r_{A_n}[x_n]$, we have $\Gamma' \vdash_{LJ} r_{A\to B}[s]$, $\Gamma' \vdash_{LJ} r_{B\to A}[t]$, $\Gamma' \vdash_{LJ} s \circ t = id$, and $\Gamma' \vdash_{LJ} t \circ s = id$. By the equality rules (which are derivable in **LJ**) we obtain

$$\Gamma' \vdash_{LJ} r_{A\to B}[\mathsf{f} < s, t >] \text{ and } \Gamma' \vdash_{LJ} r_{B\to A}[\mathsf{s} < s, t >].$$

Noting that $(\mathsf{f} < s, t >) \circ (\mathsf{s} < s, t >) = s \circ t$, one $(\cong \mathcal{I})$ (in **NJ**) allows to conclude.

Case (SP): Similarly.

Cases $\cong \mathcal{E}$: Immediate, by induction and $\& \mathcal{E}$.

Case (Eq$_1$): If $s =_{\beta,\pi} t$, then $\vdash_{LJ} s = t$ is an axiom. Some weakenings give the thesis.

Case (Eq$_2$): By induction and the equality rules in **LJ**.
($\Leftarrow$):
Assume $r_{\mathbf{A}}[\mathbf{x}] \vdash_{LJ} r_B[t]$. We argue by induction on a normal natural deduction proof in **NJ**.

Basis: The proof consists of the single premise $r_B[x]$. Then $x : B \vdash_{\cong} x : B$ by (Ax).

Equality rules: If the conclusion is an equality, the induction hypothesis and the corresponding equality rule give the thesis. If the conclusion is a predicate

$$\frac{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} s = t \qquad r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_B[s]}{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_B[t]}$$

conclude by induction hypothesis and rule (Eq$_2$).

$\& \mathcal{I}$:

$$\frac{\begin{array}{c} r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_{B\to C}[\mathsf{f}\, t] \quad r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_{C\to B}[\mathsf{s}\, t] \\ r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} (\mathsf{f}\, t) \circ (\mathsf{s}\, t) = id \quad r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} (\mathsf{s}\, t) \circ (\mathsf{f}\, t) = id \end{array}}{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_{B\cong C}[t]} \; \& \mathcal{I}$$

Induction hypothesis and rule (SP) give the thesis.

$\& \mathcal{E}$: We have two main cases, one where the conclusion is a realizability predicate

$$\frac{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_{B\to C}[s] \;\&\; \psi}{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_{B\to C}[s]} \; \& \mathcal{E}^l$$

and one where the conclusion is an equality

$$\frac{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} s = s' \;\&\; \psi}{r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} s = s'} \; \& \mathcal{E}^l$$

In both cases, by Proposition 3.3 the premise has to be of the form $r_{\mathbf{A}}[\mathbf{x}] \vdash_{LJ} r_{B\cong C}[t]$ for some term t (the right-hand side cannot be a single equation because we are applying an &-elimination rule). By the definition of the realizability predicate, the premise is thus

$$r_{\mathbf{A}}[\mathbf{x}] \vdash_{NJ} r_{B\to C}[\mathsf{f}\, t] \;\&\; r_{C\to B}[\mathsf{s}\, t] \;\&\; (\mathsf{f}\, t) \circ (\mathsf{s}\, t) = id \;\&\; (\mathsf{s}\, t) \circ (\mathsf{f}\, t) = id,$$

where $s = \mathsf{f}\, t$, in the first case, and $s = (\mathsf{f}\, t) \circ (\mathsf{s}\, t)$, $s' = id$ in the second case. Conclude by induction and rule $(\cong \mathcal{E}^f)$ (in the first case) or $(\cong \mathcal{E}^1_=)$ (in the second). The two symmetrical cases are analogous. $\Box$

4 Strong Conjunction

The first logical system for strong conjunction is due to Lopez-Escobar [Lopez-Escobar 85], who first studied also its completeness with respect to provable realizability. The system can be seen as a variant of the type assignment system for λ-terms we will introduce in Definition 4.1. As it was shown in [Mints 89], however, Lopez-Escobar's system is *not* complete with respect to provable realizability. The formula $((\alpha \to \beta) \wedge \delta) \to (((\alpha \wedge \gamma) \to \beta) \wedge \delta)$ is not provable in that system, yet it is realized by (any representative of) the identity function (any realizer for $\alpha \to \beta$ is also a realizer for $(\alpha \wedge \gamma) \to \beta)$. This counterexample relies on the fact that the system lacks a rule of extensionality for proofs, which has then to be added in order to achieve completeness (this is the role of rule (η), cf. Lemma 4.2).

A second system for strong conjunction is the one given in [Mints 89], which contains an extensionality rule, but which uses *typed* λ-terms as notation for proofs, taking then their *type erasures* (i.e. the untyped terms obtained by forgetting all type decorations) as realizers. In [Mints 89] it is further proved that the given system is complete with respect to provable realizability. The proof, however, does not work properly, for the presence of these typed terms. The difference between typed systems and type assignment systems is often overlooked and this can make problems arise.

A typed system consists in a set of terms each possessing a precise, unique type; types are not, so to speak, autonomous objects, but just decorations recalling what the functionality of a term is, and they are used as a means of partitioning the set of terms. In type assignment systems, instead, types are objects themselves, and belong to a universe strictly separated from that of (*untyped*) terms. Types can be looked at as predicates denoting functional properties; the rules of the type assignment system allow one to infer which predicates hold for a given term. While typed terms have a unique type, in a type assignment system a term can have more than a functionality, that is, it can be given more than one type.

Let us informally explain how having overlooked these differences prevented the completeness proof in [Mints 89] to work properly.

It is not difficult to see that the untyped λ-term $\lambda x.x$ (i.e. a representative for the identity function) is a realizer for the formula $(\alpha \to \alpha) \wedge (\beta \to \beta)$. In fact, in the notation of Section 2, it is *provable* that

$$\vdash_{LJ} \mathbf{r}_{(\alpha\to\alpha)\wedge(\beta\to\beta)}[\lambda x.x].$$

However, it is not possible to have a *typed* term t such that the statement $(\alpha \to \alpha) \wedge (\beta \to \beta)(t)$ is derivable, even if it is possible to derive the statements $(\alpha \to \alpha)(t_1)$ and $(\beta \to \beta)(t_2)$ for terms t_1 and t_2 such that their type erasures are both equal to $\lambda x.x$ (we have used the notation $\gamma(t)$ for the judgment "γ is the type of the typed term t", in order to stress the difference of this approach with a type-assignment system). This is because two typed terms with the same "structure", but with different types (even if the difference between them consists only in the names of type variables), are two distinct terms, a fact that prevents Mints' rule

$$\frac{\Gamma \vdash \gamma(t) \qquad \Gamma \vdash \delta(t)}{\Gamma \vdash \gamma \wedge \delta(t)}$$

from applying in case we have $\alpha \to \alpha(t_1)$ and $\beta \to \beta(t_2)$ and we only know that t_1 and t_2 have the same type erasure, as it should be in the proof of Lemma 2.3 in [Mints 89].

To amend the completeness result we have to recast the system in a type assignment framework instead that in a typed one ([Mints 91] agrees with our remarks). With this modification, and with essentially the same rules and similar proofs, we will be able to show the completeness of the system (Theorem 4.13 below). The system we obtain, in conclusion, is an already existing one: the type assignment system for intersection types introduced in [Barendregt *et al.* 83].

4.1 A Type Assignment System for Strong Conjunction

Definition 4.1 *(i) The set of* intersection types *is the set of types built out of a denumerable set of type variables and the type constant ω by means of two type constructors: $\to$ and $\wedge$.*

$$x : \sigma \vdash^{\wedge} x : \sigma \quad (\mathrm{Ax})$$

$$B \vdash^{\wedge} M : \omega \quad (\mathrm{Ax}-\omega)$$

$$\frac{B \vdash^{\wedge} M : \tau}{B, x : \sigma \vdash^{\wedge} M : \tau} \ (weak)$$

$$\frac{B, x : \sigma \vdash^{\wedge} M : \tau}{B \vdash^{\wedge} \lambda x.M : \sigma \to \tau} \ (\to I) \quad (*)$$

$$\frac{B \vdash^{\wedge} M : \sigma \to \tau \quad B \vdash^{\wedge} N : \sigma}{B \vdash^{\wedge} (MN) : \tau} \ (\to E)$$

$$\frac{B \vdash^{\wedge} M : \sigma \quad B \vdash^{\wedge} M : \tau}{B \vdash^{\wedge} M : \sigma \wedge \tau} \ (\wedge I)$$

$$\frac{B \vdash^{\wedge} M : \sigma \wedge \tau}{B \vdash^{\wedge} M : \sigma} \ (\wedge E^{l}) \qquad \frac{B \vdash^{\wedge} M : \sigma \wedge \tau}{B \vdash^{\wedge} M : \tau} \ (\wedge E^{r})$$

$$\frac{B \vdash^{\wedge} M : \sigma \quad \sigma \leq \tau}{B \vdash^{\wedge} M : \tau} \ (\leq)$$

($*$) if x does not occur in B.

Figure 2: THE TYPE ASSIGNMENT SYSTEM FOR STRONG CONJUNCTION

(ii) *A* (type assignment) statement *is an expression of the form $M : \sigma$ where σ is a type and M is a λ-term. M is called the* subject *of the statement. A* basis *B is a set of statements with only variables as subjects. The same variable can occur several times in a basis B.*

(iii) *The relation $\leq$ among types is the smallest relation satisfying the following clauses:*

1.	$\tau \leq \tau$	2.	$\tau \leq \tau \wedge \tau$
3.	$\sigma \leq \omega$	4.	$\omega \leq \omega \to \omega$
5.	$\sigma \wedge \tau \leq \sigma$	6.	$\sigma \wedge \tau \leq \tau$
7.	$(\sigma \to \rho) \wedge (\sigma \to \tau) \leq \sigma \to (\rho \wedge \tau)$	8.	$\sigma \leq \tau \leq \rho \Rightarrow \sigma \leq \rho$
9.	$\sigma \leq \sigma', \tau \leq \tau' \Rightarrow \sigma \wedge \tau \leq \sigma' \wedge \tau'$		
10.	$\sigma' \leq \sigma, \tau \leq \tau' \Rightarrow (\sigma \to \tau) \leq \sigma' \to \tau'$		

$\sigma \sim \tau$ is short for ($\sigma \leq \tau$ and *$\tau \leq \sigma$).*

(iv) *The type assignment system $\vdash^{\wedge}$ is defined in Figure 2.*

Remark 4.1 *It is easy to check that if $B \vdash^{\wedge} M : \alpha$ then $B \cup B' \vdash^{\wedge} M : \alpha$ for any basis B'.*

Lemma 4.2 *[Barendregt* et al. *83] Rule*

$$\frac{B \vdash \lambda x.Mx : \sigma}{B \vdash M : \sigma} \ (\eta) \quad \text{if } x \notin \mathrm{FV}(M)$$

is admissible in $\vdash^{\wedge}$.

Lemma 4.3 *(Subject conversion) [Coppo & Dezani 80] [$B \vdash^{\wedge} M : \alpha$* and *$M =_{\beta} M'$] $\Rightarrow B \vdash^{\wedge} M' : \alpha$.*

It is beyond the scope of this paper to give the detailed motivations and the main achievements obtained by means of system $\vdash^{\wedge}$ and its variants. To show its relevance for the syntactical theory of pure λ-calculus we just quote the following theorem, where intersection types are used to characterize relevant classes of λ-terms.

Theorem 4.4 *[Barendregt* et al. *83, Coppo* et al. *82]*

(i) M is normalizable iff there exist B and α (B,α ω-free) such that $B \vdash^{\wedge} M : \alpha$.

(ii) M is strongly normalizable iff there exist B and α (B,α ω-free) such that $B \vdash^{\wedge}_{-\omega} M : \alpha$, where $\vdash^{\wedge}_{-\omega}$ is the subsystem obtained from $\vdash^{\wedge}$ by taking out the constant ω and axiom (Ax-ω).

(iii) M has head normal form iff there exist B and α (α tail proper) such that $B \vdash^{\wedge} M : \alpha$, where a type is tail proper *if it is of the form $\rho_1 \rightarrow \ldots \rightarrow \rho_n \rightarrow \phi$ with ϕ type variable.*

Systems with intersection types have been used also in the construction of (*filter*) models of the type-free λ-calculus, see [Barendregt *et al.* 83, Coppo *et al.* 82].

To prove that $\vdash^{\wedge}$ is a good formalization for the notion of strong conjunction, i.e. a sound and complete system for provable realizability, we shall first prove the equivalence of $\vdash^{\wedge}$ to the type assignment system $\vdash_{\mathcal{M}}$ defined below. Then, by using this equivalence, a proof will be given of the fact that a statement $\vdash^{\wedge} M : \alpha$ is derivable iff it is possible to prove in intuitionistic first order logic that M is a realizer for α.

Definition 4.5 *The system $\vdash_{\mathcal{M}}$ is defined from $\vdash^{\wedge}$ by taking out the constant ω, axiom (Ax-ω) and rule ($\leq$), and by adding the following rules.*

$$\frac{B, x : \alpha \vdash_{\mathcal{M}} Mx : \beta}{B \vdash_{\mathcal{M}} M : \alpha \rightarrow \beta}\ (\eta_1)$$

$$\frac{B \vdash_{\mathcal{M}} M : \alpha \quad M =_{\beta} N}{B \vdash_{\mathcal{M}} N : \alpha.}\ (eq_{\beta})$$

Note that $\vdash_{\mathcal{M}}$ is an *untyped* version of the system for provable realizability given in [Mints 89] (the subscript $\mathcal{M}$ is used to recall this fact). The completeness result of next section could have been proved more directly, thus avoiding system $\vdash_{\mathcal{M}}$. However, our formalization, which closely match Mints' proof, helps to note how the *untypedness* of $\vdash_{\mathcal{M}}$ is essential in the proof, where Mints' approach would break.

Lemma 4.6 *Rule ($\leq$) is admissible in $\vdash_{\mathcal{M}}$.*

Proof: By induction on the derivation of $\sigma \leq \tau$. □

Lemma 4.7 *Let $\omega \notin \alpha, B$. Then $B \vdash^{\wedge} M : \alpha \Leftrightarrow B \vdash_{\mathcal{M}} M : \alpha$.*

Proof: ($\Leftarrow$) It is enough to prove that rules (eq_{β}) and (η_1) are admissible in $\vdash^{\wedge}$.

(eq_{β}) By Lemma 4.3.

(η_1) By Lemma 4.2 the admissibility of this rule can be easily proved in the following way.

$$\frac{\dfrac{B, x : \alpha \vdash^{\wedge} Mx : \beta}{B \vdash^{\wedge} \lambda x.Mx : \alpha \rightarrow \beta}\ (\rightarrow I)}{B \vdash^{\wedge} M : \alpha \rightarrow \beta}\ (\eta)$$

($\Rightarrow$) $B \vdash^{\wedge} M : \alpha$	$\Rightarrow$	$B \vdash^{\wedge} \overline{M} : \alpha$	by Lemma 4.3 $\overline{M}$ is the normal form of M (which exists by Theorem 4.4)
	$\Rightarrow$	$B \vdash^{\wedge}_{-\omega} \overline{M} : \alpha$	by the *subformula principle*, cf. Lemma 4.5 of [Barendregt *et al.* 83]
	$\Rightarrow$	$B \vdash_{\mathcal{M}} \overline{M} : \alpha$	by Lemma 4.6
	$\Rightarrow$	$B \vdash_{\mathcal{M}} M : \alpha$	by rule (eq_{β}).

□

4.2 Provable Realizability for Strong Conjunction

We will show in this section that the type assignment system of Definition 4.1 is complete with respect to provable realizabilty. We extend, first, the definition of the realizability predicate, Definition 2.1, to encompass strong conjunction.

Definition 4.8 *The realizability predicate for strong conjunction is defined as*

$$\mathbf{r}_{A\wedge B}[x] \equiv \mathbf{r}_A[x] \ \& \ \mathbf{r}_B[x].$$

Realizers are here pure λ-terms as defined in Section 2, modulo β-equality. As for **LJ**, we can use here the more compact form having as axioms sequents $\Delta, P(s) \vdash P(s')$, together with a justification for $s = s'$. Recall, now, from Section 3.2, that proofs in **LJ** of formulas of the form $\mathbf{r}_A[t]$ can be seen, by the subformula principle and the permutability of rules, as consisting of special proof structures, called there $\vdash\rightarrow$ and $\rightarrow\vdash$. (Recall also the notation we used there: $\mathbf{M}$ stands for a sequence of terms, while $\mathbf{r}_\alpha[\mathbf{M}] \vdash \mathbf{r}_B[N]$ stands for $\mathbf{r}_{\alpha_1}[M_1], \ldots, \mathbf{r}_{\alpha_n}[M_n] \vdash \mathbf{r}_B[N]$

Proposition 3.3 specializes now to the following proposition [Mints 89], showing that if we are interested only in deriving formulas of the form $\mathbf{r}_A[t]$, then it is enough to use a restriction of **LJ** where the sequences of formulas on the left of the entailment are of a special shape and the rules are only those for conjunction ($\& \vdash_{LJ}$ and $\vdash_{LJ} \&$) and the two "condensed" proof structures for implication ($\rightarrow\vdash$ and $\vdash\rightarrow$).

Proposition 4.9 *Any sequent occurring in the cut-free derivation of the sequent*

$$\mathbf{r}_{\alpha_1}[x_1\mathbf{M}_1], \ldots \mathbf{r}_{\alpha_h}[x_h\mathbf{M}_h] \vdash_{LJ} \mathbf{r}_\beta[M]$$

is of the form

$$\mathbf{r}_{\gamma_1}[y_1\mathbf{M}_1], \ldots \mathbf{r}_{\gamma_k}[y_k\mathbf{M}_k] \vdash_{LJ} \mathbf{r}_\delta[M].$$

In order to show the equivalence of $\vdash_{LJ}$ and $\vdash_{\mathcal{M}}$, it is handy to introduce some further notation for comparing sequences of formulas in $\vdash_{LJ}$ and basis in $\vdash_{\mathcal{M}}$.

Definition 4.10 *(i) A context Γ for $\vdash_{LJ}$ is an* r-context *if all its elements are of the form $\mathbf{r}_\alpha[M]$.*

(ii) Let Γ be an r-context in which all terms M are variables and let B be a basis for $\vdash^\wedge$ such that $\omega \notin B$.

1. *Γ° is the basis for $\vdash^\wedge$ obtained by replacing in Γ each element $\mathbf{r}_\gamma[x]$ by $x : \gamma$.*
2. *$B^\diamond$ is the r-context obtained by replacing in B each element $x : \gamma$ by $\mathbf{r}_\gamma[x]$.*

Lemma 4.11

$$B \vdash_{\mathcal{M}} M : \alpha \Rightarrow B^\diamond \vdash_{LJ} \mathbf{r}_\alpha[M].$$

Proof: By the induction on the derivation of $B \vdash_{\mathcal{M}} M : \alpha$. We shall give only the non-trivial cases.

$(\wedge E)$

$$\frac{B \vdash_{\mathcal{M}} M : \alpha \wedge \beta}{B \vdash_{\mathcal{M}} M : \alpha}$$

By the induction hypothesis $B^\diamond \vdash_{LJ} \mathbf{r}_{\alpha\wedge\beta}[M]$. By Proposition 2.2 $B^\diamond \vdash_{NJ} \mathbf{r}_{\alpha\wedge\beta}[M] \equiv \mathbf{r}_\alpha[M]\&\mathbf{r}_\beta[M]$. Then by $\&E$) in $\vdash_{NJ}$ we get $B^\diamond \vdash_{NJ} \mathbf{r}_\alpha[M]$ and, by 2.2 again, $B^\diamond \vdash_{LJ} \mathbf{r}_\alpha[M]$.

$(\rightarrow I)$

$$\frac{B, x : \alpha \vdash^\wedge M : \beta}{B \vdash^\wedge \lambda x.M : \alpha \rightarrow \beta}$$

By the induction hypothesis $B^\diamond, x : \alpha \vdash_{LJ} \mathbf{r}_\beta[N]$ and hence, since $\vdash_{LJ}$ is a system with equality and $N = (\lambda x.N)x$, we get $B^\diamond, x : \alpha \vdash_{LJ} \mathbf{r}_\beta[(\lambda x.N)x]$. By applying rule $\vdash\rightarrow$ we obtain $B^\diamond \vdash_{LJ} \mathbf{r}_{\alpha\rightarrow\beta}[N]$.

$(\rightarrow E)$

$$\frac{B \vdash^{\wedge} M : \alpha \rightarrow \beta \quad B \vdash^{\wedge} N : \alpha}{B \vdash^{\wedge} (MN) : \beta}$$

By the induction hypothesis $B^{\diamond} \vdash_{LJ} \mathbf{r}_{\alpha\rightarrow\beta}[M]$ and $B^{\diamond} \vdash_{LJ} \mathbf{r}_{\alpha}[N]$. By Proposition 2.2 $B^{\diamond} \vdash_{NJ} \mathbf{r}_{\alpha\rightarrow\beta}[M] \equiv \forall x(\mathbf{r}_{\alpha}[x] \rightarrow \mathbf{r}_{\beta}[Nx])$. Then by rules $(\forall E)$ and $(\rightarrow E)$ and 2.2 again, $B^{\diamond} \vdash_{LJ} \mathbf{r}_{\beta}[MN]$.

□

Lemma 4.12 *Let* $\Gamma \vdash_{LJ} \mathbf{r}_{\alpha}[M]$ *with* $\Gamma = \{\mathbf{r}_{\gamma_i}[P_i]\}_{i\in I}$ *such that there exist bases* B_i *such that* $B_i \vdash^{\wedge} P_i : \gamma_i$. *Then* $\bigcup_{i\in I} B_i \vdash^{\wedge} M : \alpha$.

Proof: By induction on the derivation of $\Gamma \vdash_{LJ} \mathbf{r}_{\alpha}[M]$, exploiting Proposition 4.9.

(Ax) $\Gamma, \mathbf{r}_{\alpha}[xM] \vdash_{LJ} \mathbf{r}_{\alpha}[N]$, with $xM = N$.
By hypothesis we know that, for some B, $B \vdash^{\wedge} xM : \alpha$; Lemma 4.3 allows to conclude.

$(\vdash_{LJ} \&)$

$$\frac{\Gamma \vdash_{LJ} \mathbf{r}_{\alpha}[M] \quad \Gamma \vdash_{LJ} \mathbf{r}_{\beta}[M]}{\Gamma \vdash_{LJ} \mathbf{r}_{\alpha\wedge\beta}[M]}$$

By the induction hypothesis we have B_1, B_2 such that $B_1 \vdash^{\wedge} M : \alpha$ and $B_2 \vdash^{\wedge} M : \beta$ and hence the thesis follows by Remark 4.1 and rule $(\wedge I)$

$(\& \vdash_{LJ})$

$$\frac{\Gamma, \mathbf{r}_{\delta}[P], \mathbf{r}_{\beta}[P] \vdash_{LJ} \mathbf{r}_{\alpha}[M]}{\Gamma, \mathbf{r}_{\delta\wedge\beta}[P] \vdash_{LJ} \mathbf{r}_{\alpha}[M]}$$

We have that there exists B' such that $B' \vdash^{\wedge} P : \delta \wedge \beta$. By rule $(\wedge E)$ it is possible to apply the induction hypothesis, which yields the thesis.

$(\rightarrow\vdash_{LJ})$

$$\frac{\Gamma \vdash_{LJ} \mathbf{r}_{\delta}[y] \quad \Gamma, \mathbf{r}_{\beta}[z\mathbf{P}y] \vdash_{LJ} \mathbf{r}_{\alpha}[M]}{\Gamma, \mathbf{r}_{\delta\rightarrow\beta}[z\mathbf{P}] \vdash_{LJ} \mathbf{r}_{\alpha}[M]}$$

We have that there exists B_1 such that $B_1 \vdash^{\wedge} z\mathbf{P} : \delta \rightarrow \beta$. As it is possible to apply the induction hypothesis to $\Gamma \vdash_{LJ} \mathbf{r}_{\delta}[y]$, there exists B_2 such that $B_2 \vdash^{\wedge} y : \delta$. The thesis then follows from the induction hypothesis on $\Gamma, \mathbf{r}_{\beta}[z\mathbf{P}y] \vdash_{LJ} \mathbf{r}_{\alpha}[M]$, which is applicable since, by $(\rightarrow E)$, $B_1 \cup B_2 \vdash^{\wedge} z\mathbf{P}y : \beta$.

$(\vdash_{LJ}\rightarrow)$

$$\frac{\Gamma, \mathbf{r}_{\alpha}[y] \vdash_{LJ} \mathbf{r}_{\beta}[My]}{\Gamma \vdash_{LJ} \mathbf{r}_{\alpha\rightarrow\beta}[M]}$$

We can apply the induction hypothesis since $y : \alpha \vdash^{\wedge} y : \alpha$. Then $\bigcup_{i\in I} B_i, y : \alpha \vdash^{\wedge} My : \beta$ and by rule $(\rightarrow I)$ $\bigcup_{i\in I} B_i \vdash^{\wedge} \lambda y.My : \alpha \rightarrow \beta$. Since $y \notin FV(M)$, by rule (η), which is admissible in $\vdash^{\wedge}$, we get $\bigcup_{i\in I} B_i \vdash^{\wedge} M : \alpha \rightarrow \beta$.

□

From the two lemmas above it is straightforward to obtain the following theorem, stating formally that $\vdash^{\wedge}$ is sound and complete with respect to provable realizability for strong conjunction, and hence a good formalization of it. Of course we have to take care that ω does not appear in the types of assumptions and conclusions.

Theorem 4.13 *Let $\omega \notin \alpha_1, \ldots, \alpha_n, \alpha$. Then*

$$x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash^{\wedge} M : \alpha \iff \mathbf{r}_{\alpha_1}[x_1], \ldots, \mathbf{r}_{\alpha_n}[x_n] \vdash_{LJ} \mathbf{r}_{\alpha}[M]$$

It is worth pointing out that it would be possible to prove, in a more direct way, that $\vdash_{NJ}$ is equivalent to $\vdash^{\wedge}$, without using Gentzen's sequent calculus and $\vdash_{\mathcal{M}}$. Their use is justified by our aim of comparing what is done in the present paper with what is done in [Mints 89].

4.3 Getting rid of ω

The type assignment system for implication and strong conjunction has the "non-logical" rules ($\leq$) and (ω), not involving the operators "$\rightarrow$" and "$\wedge$. Is it possible to have a (restricted) system without rule ($\leq$) or (ω), for which still some completeness result is provable?

Let us observe, first, that the extensionality rule (η) is necessary in the proof of Lemma 4.12, case $\vdash_{LJ}\rightarrow$ (similarly, extensionality for pairs, rule SP of definition 3.2, is necessary for the completeness of strong equivalence, Theorem 3.4). Its presence, in other words, it is not connected with strong conjunction, but with the implication connective. Rule ($\leq$), therefore, is unavoidable, in view of its equivalence with (η), or (η_1) (see Lemmas 4.2 and 4.6).

What about rule (ω)? Its role is crucial in the proof of the Subject Conversion Lemma, 4.3, which is essential for completeness, since we use a logic equipped with β-equality between λ-terms. If we want to eliminate rule (ω), therefore, we must change our combinatory algebra of realizers in such a way that the preservation of their types could be guaranteed without making use of (ω). Such a system does exist: it is the system $\vdash^{\wedge}_{-\omega}$, with the additional restriction that the terms have to be λI-terms (we will write $\vdash^{\wedge I}_{-\omega}$ for this system). Let us recall that the λI-calculus is a restriction of the pure λ-calculus, where it is possible to form a term $\lambda x.M$ only if $x \in FV(M)$. We refer to [Barendegt 1984], Cap. 9, for the precise definition and properties.

The zealous reader will enjoy to check that if we restrict realizers to be λI-terms, a good formalization for strong conjunction, i.e. a system sound and complete w.r.t. provable realizability, is exactly $\vdash^{\wedge I}_{-\omega}$.

Before starting such an exercise, however, it is likely that the reader will ask her/himself if this is not a too strong solution for getting rid of ω. Weaker restrictions, however, seem to prevent Theorem 4.13 from holding. As an example, maintain all λ-terms as realizers and restrict only the conversion among them in the following way : $M = N$ iff both M and N are strongly normalizable, $M =_{\beta} N$, and $FV(M) = FV(N) = FV(P)$ with P the normal form of M and N. With such a restriction it is still possible to prove the ($\Leftarrow$) half of Theorem 4.13, while the derivation of $(\lambda xy.x)zt : \alpha$ from the basis $z : \alpha, t : \beta$ is a counterexample for the ($\Rightarrow$) direction.

4.4 Combinatory Logic

Realizers of implicational formulas have always been, so far, λ-terms. As mentioned in the Section 2, however, other choices are possible. We will consider in this section the combinatory algebra of Combinatory Logic (CL) terms, a well known and studied alternative to λ-terms ([Lopez-Escobar 85] uses CL-terms for his study of provable realizability).

As a corollary to Theorem 4.13, by building on some known results relating λ-terms and CL-terms, we will be able to obtain a completeness result also for CL-terms. Since CL is a first-order theory, this result, by appealing to the standard (Gödel's) completeness theorem, will enable us to get an easy proof of some semantic consequences, like the equivalence of type judgement validity for CL-terms and their λ-calculus translation w.r.t. two different notions of model (λ-algebras and λ-models). It will be also possible to provide an alternative proof of system $\vdash^{\wedge}$'s semantic completeness.

Before going into the details, however, we have to decide which notion of *equality* between CL-terms to consider. Two notions of conversion are sensible in CL. The first, *weak equality*, is sufficient for combinatory completeness (and thus for considering CL as a good theory of functions, without having to deal with the notion of bound variables). It is equationally axiomatizable by

$$\mathsf{K}xy = x \qquad \mathsf{S}xyz = xz(yz).$$

The second notion of conversion, *combinatory β-equality* (whose theory will be denoted by CL_β and whose symbol will be $=_{c\beta}$), enables a good correspondence between CL-terms and λ-terms with β-equality. It is possible to define translations

$$(-)_\lambda : \mathrm{CL-terms} \longrightarrow \lambda\mathrm{-terms}$$

$$(-)_H : \lambda\mathrm{-terms} \longrightarrow \mathrm{CL-terms},$$

such that $X =_{c\beta} Y$ iff $X_\lambda =_\beta Y_\lambda$. In what follows we shall assume $(-)_H$ to be one of the most powerful translations, for instance H^η, as defined in [Curry & Feys 58]. Also CL_β is equationally axiomatizable (we refer to [Barendegt 1984] par. 7.3 for the rather long list of axioms needed); the class of its models (*λ-algebras*) is larger than the class of models of the λ-calculus, which is first-order axiomatizable, but not with equations only (rule (ξ), a principle of weak extensionality, is an implication and does not hold in λ-algebras).

4.4.1 CL-terms with Combinatory β-equality

In the rest of this section we shall deal with provable realizability using CL-terms (with K, S and I as basic combinators) and combinatory β-equality.

The definition of the realizability predicate $\mathbf{r}_\alpha[X]$ where X is a CL-term is obviously the same as the one with λ-terms. We shall use superscripts to stress the fact that we are using systems with CL-terms.

Definition 4.14 *The system $\vdash_{LJ}^{CL_\beta}$ is the intuitionistic sequent calculus with equality, where terms are CL-terms and equality is combinatory β-equality.*

The following lemma is a straightforward check.

Lemma 4.15

$$\vdash_{LJ}^{CL_\beta} \mathbf{r}_\alpha[X] \Longleftrightarrow \vdash_{LJ} \mathbf{r}_\alpha[X_\lambda].$$

The type-assignment system corresponding to this notion of provable realizability uses CL-terms and intersection types, and was introduced in [Dezani & Hindley 89].

Definition 4.16 *The system $\vdash_{CL}^{\wedge}$ is defined by the following axiom schemes and rules.*
Axioms:

- $B \vdash_{CL}^{\wedge} \mathsf{I} : \sigma \to \sigma$
- $B \vdash_{CL}^{\wedge} \mathsf{K} : \sigma \to \tau \to \sigma$
- $B \vdash_{CL}^{\wedge} \mathsf{S} : (\sigma \to \tau \to \rho) \to (\sigma \to \tau) \to \sigma \to \rho$

Rules: $(\to E)$, $(\wedge I)$, $(\wedge E)$, (ω) *and* $(\leq)$.

Lemma 4.17 *[Dezani & Hindley 89]*

$$\vdash_{CL}^{\wedge} X : \alpha \Longleftrightarrow \vdash^{\wedge} X_\lambda : \alpha$$

The presence of rule (ω) allows $\vdash_{CL}^{\wedge}$ to have rule (eq_β) admissible also for CL-terms.

Lemma 4.18 *[Dezani & Hindley 89]*

$$B \vdash^{\wedge}_{CL} X : \alpha, \quad X =_{c\beta} Y \quad \Rightarrow \quad B \vdash^{\wedge}_{CL} Y : \alpha$$

With these facts, completeness of provable realizability is now proved easily.

Theorem 4.19

$$\vdash^{CL_\beta}_{LJ} \mathbf{r}_\alpha[X] \Longleftrightarrow \vdash^{\wedge}_{CL} X : \alpha \quad (\textit{with } \omega \notin \alpha)$$

Proof:

$$\begin{array}{cccccc} & \vdash^{CL_\beta}_{LJ} \mathbf{r}_\alpha[X] & & \vdash^{\wedge}_{CL} X : \alpha & \\ \text{Lemma 4.15} & \Updownarrow & & \Updownarrow & \text{Lemma 4.17} \\ & \vdash_{LJ} \mathbf{r}_\alpha[X_\lambda] & \Longleftrightarrow & \vdash^{\wedge} X_\lambda : \alpha & \\ & & \text{Theorem 4.13} & & \end{array}$$

□

4.4.2 Some Semantic Consequences

Theorem 4.19 allows easy and elegant proofs of several semantical equivalences, exploiting the fact that the semantic completeness of $\vdash^{CL_\beta}_{LJ}$ and $\vdash_{LJ}$ is an instance of usual (Gödel) completeness.

Let us first define the notion of validity for type assignment systems.

Definition 4.20 *[Barendregt* et al. *83] Let $\mathcal{M}$ be a λ-model(λ-algebra) over the applicative structure $< D, \cdot >$.*

(i) If ξ is a valuation of variables in D, then $[\![P]\!]^{\mathcal{M}}_{\xi}$ is the interpretation of the λ-term(CL-term) P in $\mathcal{M}$ via ξ.

(ii) Let $\mathcal{V}$ be a valuation of type variable in subsets of the domain D. Then the interpretation of a type σ in $\mathcal{M}$ via $\mathcal{V}$ is defined as follows.

- $[\![\phi]\!]^{\mathcal{M}}_{\mathcal{V}} = \mathcal{V}(\phi)$ *where ϕ a type variable.*
- $[\![\omega]\!]^{\mathcal{M}}_{\mathcal{V}} = D$
- $[\![\sigma \rightarrow \tau]\!]^{\mathcal{M}}_{\mathcal{V}} = \{d \in D | \forall e \in [\![\sigma]\!]^{\mathcal{M}}_{\mathcal{V}} \; d \cdot e \in [\![\tau]\!]^{\mathcal{M}}_{\mathcal{V}}\}$
- $[\![\sigma \wedge \tau]\!]^{\mathcal{M}}_{\mathcal{V}} = [\![\sigma]\!]^{\mathcal{M}}_{\mathcal{V}} \cap [\![\tau]\!]^{\mathcal{M}}_{\mathcal{V}}$

(iii) $\mathcal{M}, \mathcal{V}, \xi \models^{\wedge}_{(CL)} P : \sigma$ *iff* $[\![P]\!]^{\mathcal{M}}_{\xi} \in [\![\sigma]\!]^{\mathcal{M}}_{\mathcal{V}}$
$\mathcal{M}, \mathcal{V}, \xi \models^{\wedge}_{(CL)} B$ *iff* $\mathcal{M}, \mathcal{V}, \xi \models^{\wedge}_{(CL)} x : \sigma$ *for all* $x : \sigma \in B$
$B \models^{\wedge}_{(CL)} P : \sigma$ *iff for all* $\mathcal{M}, \mathcal{V}, \xi \models^{\wedge}_{(CL)} B$, $\mathcal{M}, \mathcal{V}, \xi \models^{\wedge}_{(CL)} P : \sigma$.

We shall denote by $\mathrm{CL}_\beta \models^{\wedge}_{CL} X : \alpha$ (with X a CL-term) the semantic validity of $X : \alpha$ in all λ-algebras. We can now give a new proof of the fact that both systems $\vdash^{\wedge}$ and $\vdash^{\wedge}_{CL}$ are complete with respect to this notion of validity.

Lemma 4.21 *[Barendregt* et al. *83] Let M be a λ-term.*

$$B \vdash^{\wedge} M : \alpha \Longleftrightarrow B \models^{\wedge} M : \alpha.$$

We can now give a new proof of the fact that both systems $\vdash^{\wedge}$ and $\vdash^{\wedge}_{CL}$ are complete with respect to this notion of validity, provided we restrict ourselves to bases and conclusions of derivations not containing ω in their types.

Since, via Meyer-Scott axioms [Meyer 82, Barendegt 1984], the theory of β-conversion for λ-terms is first order axiomatizable, an alternative proof of the above result can be easily obtained also from Gödel completeness theorem and Theorem 4.13, observing that, by definition of realizability predicate, it is straightforward to check that:

$$\models \mathbf{r}_\alpha[M] \Longleftrightarrow \models^{\wedge} M : \alpha$$

where the symbol $\models$ to the left of the double implication denotes validity in first order models.

Since we can easily get also

$$\mathrm{CL}_\beta \models \mathbf{r}_\alpha[X] \Longleftrightarrow \mathrm{CL}_\beta \models^{\wedge} X : \alpha,$$

semantic completeness for $\vdash^{\wedge}_{CL}$ can be obtained in the same way by Gödel's completeness theorem for $\vdash^{CL_\beta}_{LJ}$ and Theorem 4.19.

Theorem 4.22

$$\vdash^{\wedge}_{CL} X : \alpha \Longleftrightarrow CL_\beta \models^{\wedge}_{CL} X : \alpha.$$

Let us remark, that, of course, there are other, more direct way of proving the semantic completeness for $\vdash^{\wedge}_{CL}$; for instance, standard semantical arguments allow to derive the completeness direction of 4.22 from 4.21, while soundness is established with a routine induction. Provable realizability, however, offers, beside a new proof, a different perspective into the result.

Some more consequences can be obtained, relating the validity of a type judgment $X : \alpha$ in all λ-algebras to the validity of the judgment $X_\lambda : \alpha$ in all λ-models, and similarly for the translation $(-)_H$, in the other direction.

Observe first that, similarly to 4.15, we can easily establish the following relation

$$\vdash_{LJ} \mathbf{r}_\alpha[M] \Longleftrightarrow \vdash^{CL_\beta}_{LJ} \mathbf{r}_\alpha[M_H]. \tag{8}$$

Corollary 4.23

$$CL_\beta \models^{\wedge}_{CL} X : \alpha \Longleftrightarrow \models^{\wedge} X_\lambda : \alpha$$

Proof:

$$\begin{array}{ccccc} & \mathrm{CL}_\beta \models^{\wedge}_{CL} X : \alpha & & \models^{\wedge} X_\lambda : \alpha & \\ \text{Lemma 4.22} & \Updownarrow & & \Updownarrow & \text{Lemma 4.21} \\ & \vdash^{\wedge}_{CL} X : \alpha & \underset{\text{Lemma 4.17}}{\Longleftrightarrow} & \vdash^{\wedge} X_\lambda : \alpha & \end{array}$$

□

Corollary 4.24

$$CL_\beta \models^{\wedge}_{CL} M_H : \alpha \Longleftrightarrow \models^{\wedge} M : \alpha.$$

Proof:

$$\begin{array}{ccccc} \mathrm{CL}_\beta \models^{\wedge}_{CL} M_H : \alpha & & & & \models^{\wedge} M : \alpha \\ {\scriptstyle \text{Gödel's completeness}} \Updownarrow & & & & \Updownarrow {\scriptstyle \text{Lemma 4.21}} \\ \vdash^{CL_\beta}_{LJ} \mathbf{r}_\alpha[M_H] & \underset{\text{Relation (8)}}{\Longleftrightarrow} & \vdash_{LJ} \mathbf{r}_\alpha[M] & \underset{\text{Theorem 4.13}}{\Longleftrightarrow} & \vdash^{\wedge} M : \alpha \end{array}$$

□

We conclude by pointing out that it is not difficult to modify the systems described above in order to take into account also the case of weak equality on CL-term instead of combinatory β-equality.

5 Relevant implication

Realizers of usual intuitionistic implicative formulas are (intensional representatives of) *functions* transforming proofs of the antecedent formula into proofs of the consequent. The intuitionistic implication can then be seen consistently as a function space constructor for which only very limited restrictions are given on the extensional behaviour of the functions belonging to it. In the spirit of the previous sections, however, it is natural to investigate if and how this notion can be restricted and still have a sound logical meaning. In this foundational study on the notion of implication, moreover, it is natural to investigate to what extent the restrictions affect the provability of implications. In other words, given a restricted function space, is it possible to characterize those implications that are realized by elements of that space? To begin tackling such a problem, one is naturally led to investigate the implication whose related function space is the simplest one, namely the one containing only the *identity function*.

Let us note, *en passant*, that here, as in Section 3, computer science motivations lurk in the background. In the context of typed functional languages with subtypes, the notion of a type A being a subtype of B is naturally formalized as a *coercion* from A into B, that is a (typed) program $c : A \to B$ whose type-erasure is convertible to the identity function.

The problem of characterizing the implicative formulas realized by such a restricted set of functions can be rightly posed and solved in the context of proof-functionality, where the realizability analysis for a restricted implication ($\to_r$) having only the identity as realizer can be rephrased , as:

> To assert $\alpha \to_r \beta$ is to assert that any proof of α is also a proof of β.

The realizability predicate corresponding to this informal analysis is the following.

Definition 5.1 *The realizability predicate for $\to_r$ is defined as*

$$\mathbf{r}_{A \to_r B}[x] \equiv \mathbf{r}_{A \to B}[x] \ \& \ x =_\beta \lambda x.x$$

This proof-functional implication will be considered in a propositional language with strong conjunction, and we will show in Theorem 5.5 that it corresponds to the implication of *Relevant Logic* [Anderson & Belnap 75], from which the r subscript. This result not only supports and justifies the study of proof functional connectives, but shed also some light in the field of the logic of relevance, by showing a possible interpretation and formalization of what (some of) the mental processes involved in relevant reasoning look like.

The system of relevant logic we shall consider is a restriction of the system B^+ of [Meyer & Routley 72a, Meyer & Routley 72b], obtained by removing the "Church constant" t.

Definition 5.2 (Meyer, Routley) *The* Minimal Relevant Logic without constants *is defined as follows:*

Language: *The formulas of the language L_r are the propositional formulas built out of propositional variables using the connectives $\to$ and $\wedge$.*

Axioms: a1 $A \to A$

a2 $(A \wedge B) \to A$

a3 $(A \wedge B) \to B$

a4 $(A \to B) \wedge (A \to C) \to (A \to (B \wedge C)$

Rules: modus ponens $\dfrac{A \quad A \to B}{B}$

adjunction $\dfrac{A \quad B}{A \wedge B}$

suffixing $\dfrac{A \to B}{(B \to C) \to (A \to C)}$

prefixing $\dfrac{B \to C}{(A \to B) \to (A \to C)}$

We shall denote with TH_r the set of theorems of the Minimal Relevant Logic without constants. It is a very basic system, in which there is no weakening rule. As a consequence it is not possible to prove for this logic any form of the deduction theorem, which would allow to derive, from the adjunction rule, the formula $A \to B \to (A \wedge B)$ and so the exportation law (exp) $(A \wedge B \to C) \to (A \to B \to C)$.

Any formula of L_r can be equivalently read as a type of $\vdash^{\wedge}_{-\omega}$ and vice-versa (recall that $\vdash^{\wedge}_{-\omega}$ is $\vdash^{\wedge}$ without the constant ω, its related $\leq$-rules, and axiom (Ax-ω)). In what follows we shall explicitly say if an expression has to be considered as a type of $\vdash^{\wedge}$ or as a formula of L_r only when it is not clear from the context.

The reader will have already noticed the strong similarity between the axioms and rules of the minimal relevant logic and the clauses defining the relation $\leq$ among types of $\vdash^{\wedge}_{-\omega}$. This informal similarity is formally stated in Proposition 5.4, which enables the study of the relevant implication in a proof-functional context, where we will be able to apply the completeness result of the previous section.

Lemma 5.3 *Let M be a* closed *λ-term such that $M =_\beta \lambda x.x$ and α a type of $\vdash^{\wedge}_{-\omega}$.*
$\vdash^{\wedge} M : \alpha \quad \Leftrightarrow \quad \alpha \sim \bigwedge_{i=1}^{n} \gamma_i \to \beta_i, n \geq 1$ *and, for all* $1 \leq i \leq n$,
$\gamma_i \leq \beta_i$ *with* γ_i, β_i *types of* $\vdash^{\wedge}_{-\omega}$.

Proof: Since in $\vdash^{\wedge}$ typability is preserved under β-conversion (Lemma 4.3), if $\vdash^{\wedge} M : \alpha$ we get that $\vdash^{\wedge} \lambda x.x : \alpha$. Since α does not contain ω, it follows that $\vdash^{\wedge}_{-\omega} \lambda x.x : \alpha$. By induction on a derivation $\vdash^{\wedge}_{-\omega} \lambda x.x : \alpha$ we get now the required condition on α. $\Box$

Proposition 5.4 ([Venneri 92]) *Let α and β be types of $\vdash^{\wedge}_{-\omega}$.*

(i) $\alpha \leq \beta \Rightarrow \alpha \to \beta \in TH_r$

(ii) $\alpha \in TH_r \Rightarrow \alpha \sim \bigwedge_{i=1}^{n} \gamma_i \to \beta_i, n \geq 1$ *and, for all* $1 \leq i \leq n, \gamma_i \leq \beta_i$

With these two results and those of the previous section it is now easy to see that a formula $\alpha \to \beta$ is in TH_r iff its realizers are convertible to the identity function.

Theorem 5.5 *Let M be a* closed *λ-term s.t. $M =_\beta \lambda x.x$ and let α a formula of L_r. Then*

$$\vdash_{LJ} \mathbf{r}_\alpha[M] \Longleftrightarrow \alpha \in TH_r.$$

Proof: ($\Leftarrow$): By Proposition 5.4(ii) and Lemma 5.3, $\vdash^{\wedge}_{-\omega} M : \alpha$ and hence $\vdash^{\wedge} M : \alpha$. Then $\vdash_{LJ} \mathbf{r}_\alpha[M]$, by completeness of provable realizability (Theorem 4.13).
($\Rightarrow$): By completeness of provable realizability, $\vdash^{\wedge} M : \alpha$; By Lemma 5.3, $\alpha \sim \bigwedge(\gamma_i \to \beta_i)$, and $\gamma_i \leq \beta_i$ for $1 \leq i \leq n$. By Proposition 5.4(i), $\gamma_i \to \beta_i \in TH_r$, for any i. With $n-1$ applications of the adjunction rule we obtain $\bigwedge(\gamma_i \to \beta_i) \in TH_r$ and hence $\bigwedge(\gamma_i \to \beta_i) \to \alpha \in TH_r$ by Proposition 5.4(i). Then $\alpha \in TH_r$ by modus ponens (recall that $\alpha \sim \beta$ is $\alpha \leq \beta$ and $\beta \leq \alpha$). $\Box$

Acknowledgments

References

[Anderson & Belnap 75] A. Anderson and N. Belnap, *Entailment* vol.I, Princeton University Press (1975).

[Alessi & Barbanera 91] F. Alessi and F. Barbanera, "Strong conjunction and intersection types," in volume 520 of *Lecture Notes in Computer Science*, Springer Verlag, (1991).

[Barendegt 1984] H. Barendregt, *The Lambda Calculus: its Syntax and Semantics*, North Holland (1984).

[Barendregt *et al.* 83] H. Barendregt, M. Coppo and M. Dezani, "A filter lambda model and the completeness of type assignment," *The Journal of Symbolic Logic*, Vol. 48, no.4 (1983), pp. 931-940.

[Bruce & Longo 85] K. Bruce and G. Longo, "Provable isomorphisms and domain equations in models of typed languages," in *ACM Symposium on Theory of Computing* (*STOC 85*), Providence (May 1985), pp. .

[Bruce *et al.* 92] K. Bruce, R. Di Cosmo and G. Longo, "Provable isomorphisms of types," *Math. Struct. in Comp. Science*, vol. 2, no. 2 (1992), 231–247.

[Coppo & Dezani 80] M. Coppo and M. Dezani, "An extension of basic functionality theory for lambda-calculus," *Notre Dame Journal of Formal Logic*, Vol. 21, no. 4 (1980), pp. 685–693.

[Coppo *et al.* 82] M. Coppo, M. Dezani, F. Honsell and G. Longo, "Extended type structures and filter lambda models", *Proceedings of Logic Colloquium '82*, eds. G. Lolli, G. Longo, A. Marcja, 241-262. North-Holland. (1984).

[Curry & Feys 58] H.B. Curry, R. Feys. *Combinatory Logic,*. Vol. I. North Holland. (1958).

[Dezani & Hindley 89] M. Dezani, R. Hindley. "Intersection Types for Combinatory Logic", *Theoretical Computer Science*, 100(1992), 303–324.

[Di Cosmo 91] R. Di Cosmo, "Provable isomorphisms of second order types," Rapport de Recherche du Laboratoire d'Informatique de l'Ecole Normal Supérieure, Paris (1991).

[Hindley & Seldin 86] R. Hindley, J. P. Seldin, *Introduction to Combinators and λ-calculus*, Cambridge University Press. (1986).

[Kleene 52] S.C. Kleene, "Permutability of inferences in Gentzen's calculi LK and LJ," *Memoirs of the American Mathematical Society*, no. 10 (1952), pp. 1–26.

[Lopez-Escobar 85] E.G.K. Lopez-Escobar, "Proof functional connectives," in volume 1130 of *Lectures Notes in Mathematics* (1985), Springer Verlag, pp. 208–221.

[Martini 92] S. Martini, "Provable isomorphisms, strong equivalence, and realizability," Proceedings *IV Conferenza Italiana di Informatica Teorica*, M. Venturini-Zilli, A. Marchetti-Spaccamela eds., World Scientific Pub., 1992.

[Meyer 82] A. Meyer, "What is a model of the Lambda Calculus ?," *Information and Computation*, Vol. 52 (1982).

[Meyer & Routley 72a] R.K. Meyer and R. Routley, "Algebraic analysis of entailment I," *Logique et Analyse*, Vol. 15 (1972).

[Meyer & Routley 72b] R.K. Meyer and R. Routley, "Semantics of entailment III," *Journal of Philosophical Logic*, Vol. 1 (1972).

[Mints 89] G.E. Mints, "The completeness of provable realizability," *Notre Dame Journal of Formal Logic*, Vol. 30, no. 3 (1989), pp. 420–441.

[Mints 91] G.E. Mints, Personal communication, (1991).

[Pottinger 80] G. Pottinger, "A type assignment for the strongly normalizable λ-terms," in *To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalism*, R. Hindley and J. Seldin eds., Academic Press (1980), pp. 561–577.

[Rittri 90] M. Rittri, "Using types as search keys in function libraries," *Journal of Functional Programming*, vol. 1, no. 1 (1990).

[Takeuti 75] G. Takeuti, *Proof Theory*, North-Holland Publishing Company (1975).

[Venneri 92] B. Venneri, "Intersection types as logical formulas", internal report dipartimento di Informatica Universita' di Torino, submitted for publication, 1992.