# Lambda Calculus and Intuitionistic Linear Logic

*Simona Ronchi della Rocca*
Università degli studi di Torino
Dipartimento di Informatica
C.so Svizzera n.185 - 10149 TORINO - ITALY.
E-mail: ronchi@di.unito.it

*Luca Roversi*
Università degli studi di Pisa
Dipartimento di Informatica
C.so Italia n.40 - 56125 PISA - ITALY.
E-mail: rover@di.unipi.it

## 1 Introduction

The *Curry-Howard isomorphism* [1] is the basis of typed functional programming. By means of this isomorphism, the intuitionistic proof of a formula can be seen as a functional program, whose type is the formula itself. In this way, the computation process has its logic realization in the proof normalization procedure. Both the implicative fragment of the intuitionistic propositional logic together with the simply typed $\lambda$-calculus [3], and the second order propositional logic together with the second order $\lambda$-calculus of Girard and Reynold [8, 19] are examples of such an isomorphism. The *linear logic*, introduced by Girard [9], seems to be particularly interesting from the computational point of view. It is more refined than the classical logic: the use of the structural rules is explicitly controlled through a modal connective, denoted by the unary operator !. In other words, weakening and contraction can be applied only to modal formulas. Since weakening and contraction rules are naturally related to the operations of erasing and copying information, respectively, the linear logic can be seen as a model for a computational environment with an explicit control of the resource management. These features can be effectively studied using a language corresponding to the intuitionistic fragment of the linear logic, through the Curry-Howard isomorphism.

Until now, some languages inspired by this isomorphism have been designed. First of all, Lafont [13] defined a calculus of combinators, corresponding to the intuitionistic linear logic (ILL in the following), where combinators were suggested from the categorical interpretation of the logic. Then, he defined a *linear abstract machine* for the evaluation of his calculus. Abramsky [1] has been the first one proposing a "linear $\lambda$-calculus". His language is inspired to the classical $\lambda$-calculus and it is defined using ILL as type assignment for it. The functional language is obtained by "decorating" with terms the rules of the sequent calculus for ILL. Abramsky also proposed an extension of the SECD machine for

[1] or "formulas-as-types" principle

evaluating his calculus in a *call-by-value setting*. However, the Abramsky's language does not realize the desired isomorphism. The language does not allow the correspondence *reduction rules/cut-elimination* because normal terms which encode non-cut-free deductions do exist. This implies the existence of some term representing different deductions which can not be reduced each other by any cut-elimination step. All this comes from the syntax chosen to represent the rule introducing ! on the right.

As pointed out by Lincoln and Mitchell in [14], a type assignment system for a functional language should be expressed in a natural deduction way. In such a case, there is a simple relation between the syntactic shape of a term and the structure of the deduction proving its type. The correspondence is lost in the sequent calculus because it is a formula-driven definition of the logic. Moreover, a natural deduction formulation naturally implies the *subject reduction theorem* which is a direct consequence of the closure of the (natural) deductions with respect to substitution. Such a theorem is significant because of its computational meaning: it says that types are preserved under reductions. Consequently, Lincoln and Mitchell proposed a linear $\lambda$-calculus based on a natural deduction formulation of ILL. Their proposal is based on the idea of Prawitz on logic S4 [18] and is essentially the formulation of Troelstra [22]. The language by Lincoln and Mitchell, when restricted to the multiplicative fragment of ILL, is linear in the sense that every free variable occurs exactly once in a term. This makes it very difficult to deal with operations of copying and erasing: their execution might modify the term so that the syntactic constraint fails. For these reasons, in [14], copying and erasing operations can be executed only at the top level. An unpleasant consequence is that the operational semantics is no more a congruence, making it difficult to look for operational properties of the language. For solving the problem, a further proposal has been made in [4] where only the multiplicative fragment of ILL is studied. This new functional language is linear and, for preserving linearity constraints, the term encoding the introduction of the modality implies that only closed terms can be *completely* copied/erased. The obtained operational semantics is a congruence. However, there is a price to pay. The operational behaviour of the language is not easily understandable because of the structural complexity of the reduction rule involving the modality. Moreover, the big number of "commuting conversions" (originated by computationally meaningless cuts of the logic) let the definition of an operational semantics difficult. An interesting approach to the definition of the language in [4] is that both its syntax and its operational properties are suggested by a categorical semantics for ILL.

In this paper we present a further proposal of a language related to ILL by the Curry-Howard isomorphism. The need of a new proposal follows from some, very general, consideration.

The $\lambda$-calculus is interesting and useful, as paradigmatic language, because of its simplicity. It has just abstraction and application as formation rules, and a single evaluation rule ($\beta$-rule) which, essentially, formalizes the substitution. In the same line, a language can be used as a paradigm for the linear intuitionistic computation, only if it preserves both the syntactical and the computational simplicity of the $\lambda$-calculus.
The new language $\Lambda_!$ we are going to propose has both a simpler syntax and an easier operational semantics than all the previous listed languages.

$\Lambda_!$ is a fully typed language. It is obtained as a "decoration" of the intuitionistic linear proofs, given in natural deduction style. The most peculiar syntactical feature of $\Lambda_!$ is that terms are built starting from two different sorts of variables, and that not only variables, but also *patterns*, can be bound, in the spirit of [5]. A pattern is an expression formed by a logic connective applied to variables [2]. The two different sorts of variables correspond respectively to modal and non modal premises. The fact that structural rules can be applied only to modal premises reflects that terms must obey to linearity constraints only on one sort of variables. Thanks to the use of patterns, the reduction rules, induced by the cuts of the logics, have a uniform definition: they are all extensions of the classical $\beta$-rule, based on pattern-matching. Consequently the computational behaviour of $\Lambda_!$ is very simple, and properties of the language, like the strong normalization, for example, can be easily proved by standard techniques.

We start from a natural deduction formulation of ILL which differs from those already introduced in the literature. We have rules introducing two connectives at once. Because of the behavior of our introduction rule for the modality, the substitution property holds only partially. It can only be stated under some conditions on subderivations. The terms of $\Lambda_!$ are defined as "decorations" of the proofs in this natural deduction system. The use of variables of two sorts ensures the correct behavior of the substitution, making the language normalizable with respect to a suitable notion of cut. The existence of two kinds of variables allows to get a computationally sound and untyped version $\Lambda_!^-$ of $\Lambda_!$, following what can be done in the intuitionistic case. Namely, the Curry type assignment system can be obtained by a "forgetful function" which erases the type information from the terms of the Church typed $\lambda$-calculus. In the same way, $\Lambda_!^-$ can be written forgetting the types information of $\Lambda_!$. The two sorts of variable encode in the syntax the lost type information. Hence, the reduction rules of $\Lambda_!$ can be safely updated to $\Lambda_!^-$.

For studying the semantics of $\Lambda_!$, a natural approach is the categorical one. A general definition of the properties that a category must have, in order to be a model for linear sequents, is both in [2] and [4]. The category we shall define is slightly different. We want to model the natural deductions defining $\Lambda_!$. The differences originate from the syntax of $\Lambda_!$ which allows to get rid of syntactical heavyness of the previous mentioned languages. We shall define what "intuitionistic linear category" means and we prove that every of such categories is a model for $\Lambda_!$. It turns out that every model of this kind induces an extensional theory, also closed under a "commuting relation" which involves normal forms. Such a relation is defined just by two clauses, because both erasure and duplication are implicit into our language. Hence, the commuting relation is significatively simpler than the one induced by the commuting conversions of [4].

A language similar to $\Lambda_!$ was presented in [23]. The language in [23] has two classes of variables: intuitionistic and linear ones. However, the Wadler's approach is different from ours since his language is untyped, it encodes sequent calculus derivations and it is linear in all variables.

The contents of the paper follow. Once recalled what the sequent calculus for ILL is in Section 2, Section 3 introduces our system for ILL which is both in natural deduction style and equivalent to the sequent calculus with respect to the set of provable formulas. Section 4

[2] $X \otimes Y$ is an example of pattern

is devoted to the definition of the language $\Lambda_!$. Further it lists the computational properties of $\Lambda_!$. Section 5 deals with the categorical model for $\Lambda_!$ and the properties it induces on the language. In Section 6 a short discussion on the untyped version of $\Lambda_!$ and its relation with $\Lambda_!$ is given. Finally, Appendix A contains the machinery needed to prove both strong normalization and Church-Rosser theorems for $\Lambda_!$.

The contents of this paper were the subject of an invited talk at the Logic Colloquium 1994 [20].

# 2 Intuitionistic linear logic

In this section we define the intuitionistic fragment of linear logic, using a sequent calculus formulation.

**Definition 2.1** *i) Let $\mathcal{A}_{ILL}$ be a set of atomic formulas of ILL containing at least the constant* 1*. The set $\mathcal{F}_{ILL}$ of well formed formulas of ILL is given as follows:*

$$\frac{a \in \mathcal{A}_{ILL}}{a \in \mathcal{F}_{ILL}} \quad \frac{A, B \in \mathcal{F}_{ILL} \quad \Box \in \{\oplus, \otimes, \&, \multimap\}}{A \Box B \in \mathcal{F}_{ILL}} \quad \frac{A \in \mathcal{F}_{ILL}}{!A \in \mathcal{F}_{ILL}}$$

*A formula of the shape $!A$, for some $A \in \mathcal{F}_{ILL}$, will be called !-formula or modal formula; $!\mathcal{F}_{ILL} \subseteq \mathcal{F}_{ILL}$ is the set of all the !-formulas.*

*ii) Let $\equiv$ be the syntactic identity on $\mathcal{F}_{ILL}$.*

**Definition 2.2** *i) A context is a finite sequence of $\mathcal{F}_{ILL}$. Contexts are ranged over by greek capital letters $\Gamma$,...,$\Delta$, ...,$\Theta$. If a context $\Gamma$ contains only formulas of $!\mathcal{F}_{ILL}$ it will be named by $!\Gamma$. The concatenation of $\Gamma$ and $\Delta$ is denoted by $\Gamma, \Delta$.*

*ii) The Sequent Calculus for ILL proves sequents $\Gamma \vdash_{\mathcal{S}} A$ where $\Gamma$ is a (possibly empty) context and $A \in \mathcal{F}_{ILL}$. It consists of the following rules:*

$$\frac{\Gamma \vdash_{\mathcal{S}} A \quad A, \Delta \vdash_{\mathcal{S}} B}{\Gamma, \Delta \vdash_{\mathcal{S}} B}(Cut) \qquad \frac{}{A \vdash_{\mathcal{S}} A}(Id)$$

$$\frac{\Gamma, A, B, \Delta \vdash_{\mathcal{S}} C}{\Gamma, B, A, \Delta \vdash_{\mathcal{S}} C}(Exchange)$$

$$\frac{\Gamma \vdash_{\mathcal{S}} A \quad B, \Delta \vdash_{\mathcal{S}} C}{\Gamma, A \multimap B, \Delta \vdash_{\mathcal{S}} C}(\multimap L) \qquad \frac{\Gamma, A \vdash_{\mathcal{S}} B}{\Gamma \vdash_{\mathcal{S}} A \multimap B}(\multimap R)$$

$$\frac{\Gamma, A, B \vdash_{\mathcal{S}} C}{\Gamma, A \otimes B \vdash_{\mathcal{S}} C}(\otimes L) \qquad \frac{\Gamma \vdash_{\mathcal{S}} A \quad \Delta \vdash_{\mathcal{S}} B}{\Gamma, \Delta \vdash_{\mathcal{S}} A \otimes B}(\otimes R)$$

$$\frac{\Gamma, A \vdash_{\mathcal{S}} C}{\Gamma, A \& B \vdash_{\mathcal{S}} C}(\& L_l) \qquad \frac{\Gamma \vdash_{\mathcal{S}} A \quad \Gamma \vdash_{\mathcal{S}} B}{\Gamma \vdash_{\mathcal{S}} A \& B}(\& R)$$

$$\frac{\Gamma, B \vdash_{\mathcal{S}} C}{\Gamma, A \& B \vdash_{\mathcal{S}} C}(\& L_r)$$

$$\frac{\Gamma \vdash_{\mathcal{S}} A}{\Gamma \vdash_{\mathcal{S}} A \oplus B}(\oplus R_l)$$

$$\frac{\Gamma, A \vdash_{\mathcal{S}} C \quad \Gamma, B \vdash_{\mathcal{S}} C}{\Gamma, A \oplus B \vdash_{\mathcal{S}} C}(\oplus L)$$

$$\frac{\Gamma \vdash_{\mathcal{S}} B}{\Gamma \vdash_{\mathcal{S}} A \oplus B}(\oplus R_r)$$

$$\frac{\Gamma, A \vdash_{\mathcal{S}} B}{\Gamma, !A \vdash_{\mathcal{S}} B}(!L) \qquad \frac{!\Gamma \vdash_{\mathcal{S}} A}{!\Gamma \vdash_{\mathcal{S}} !A}(!R)$$

$$\frac{\Gamma \vdash_{\mathcal{S}} B}{\Gamma, !A \vdash_{\mathcal{S}} B}(W) \qquad \frac{\Gamma, !A, !A \vdash_{\mathcal{S}} B}{\Gamma, !A \vdash_{\mathcal{S}} B}(C)$$

**Notation:** We shall use the greek capital letters $\Pi, \Pi_1, \Pi', \ldots$ for ranging over the well formed derivations of the system $\vdash_{\mathcal{S}}$. In particular, a derivation $\Pi$ proving the sequent $\Gamma \vdash_{\mathcal{S}} A$ will be indicated by $\Pi : \Gamma \vdash_{\mathcal{S}} A$.

The connectives of ILL can be divided into two groups: the *multiplicative* ($\multimap$ , $\otimes$, !) and the *additive* ($\oplus$, &).

The system $\vdash_{\mathcal{S}}$ is normalizing: if $\Pi : \Gamma \vdash_{\mathcal{S}} A$, then there exists a derivation $\Pi' : \Gamma \vdash_{\mathcal{S}} A$, in which instances of the $(Cut)$-rule are never used.

# 3 Intuitionistic linear logic in natural deduction style

In the introduction we pointed out that the definition of a term language corresponding (through the Curry-Howard isomorphism) to ILL is strongly related to its formulation in a natural deduction style. The main problem in designing such a formulation is the modal connective !.

The rule $(!R)$ of the sequent calculus is not sound for a natural deduction formulation of ILL, since if we used it for introducing !, then the substitution property would not hold. Consequently the natural deduction would not be normalizable. Let see an example. Suppose to have a system $\vdash_{\mathcal{N}}$ which is a natural deduction version of $\vdash_{\mathcal{S}}$. Assume that the introduction of ! in $\vdash_{\mathcal{N}}$ is

$$(!I)\frac{!\Gamma \vdash_{\mathcal{N}} A}{!\Gamma \vdash_{\mathcal{N}} !A},$$

namely $(!I)$ is $(!R)$ of $\vdash_{\mathcal{S}}$. Let observe the deduction here below:

$$(\multimap E)\dfrac{(\multimap I)\dfrac{(!I)\dfrac{!A \vdash_{\mathcal{N}} !A}{!A \vdash_{\mathcal{N}} !!A}}{\vdash_{\mathcal{N}} !A \multimap !!A} \quad (\multimap E)\dfrac{A \vdash_{\mathcal{N}} A \quad A \multimap !A \vdash_{\mathcal{N}} A \multimap !A}{A, A \multimap !A \vdash_{\mathcal{N}} !A}}{A, A \multimap !A \vdash_{\mathcal{N}} !!A}$$

The deduction is not normal because of the conclusion of $(\multimap I)$ is the *major premise* of $(\multimap E)$. This is a "detour" which, in the intuitionistic case, is eliminated (i) replacing the premise $!A \vdash_{\mathcal{N}} !A$ by the conclusion of the uppermost $(\multimap E)$, (ii) dropping both $(\multimap I)$ and the lowermost $(\multimap E)$. However, ILL does not admit the simplified deduction

$$(!I)\dfrac{(\multimap E)\dfrac{A \multimap !A \vdash_{\mathcal{N}} A \multimap !A \quad A \vdash_{\mathcal{N}} A}{A, A \multimap !A \vdash_{\mathcal{N}} !A}}{A, A \multimap !A \vdash_{\mathcal{N}} !!A}$$

because $(!I)$ needs the impossible situation $A, A \multimap !A \in !\mathcal{F}_{ILL}$ to be applied. In [14] the problem has been solved by designing an $(!I)$ rule explicitly encoding the substitution for the corresponding term formation. [4] has almost the same solution. Radical different proposals are in [15] and [16]. They are based on a notion of level to express the modal dependencies among formulas.

Nevertheless, we will follow a further different approach. We leave the $(!R)$ rule unchanged, passing from the ILL expressed as a sequent calculus to our natural-deduction-like definition, and we call it $(!I)$. We get a system where the substitution property partially holds: it can be stated only for derivations satisfying a particular condition. We define a suitable notion of "cut", taking into account this condition. The result is a normalizable system, where the normalization is reached in an "artificial" way. This choice is made on purpose, in view of a future decoration of the system by the language $\Lambda_!$. The choice of leaving $(!R)$ unchanged, will allow a very natural definition of the reduction rules. Another main difference between our formulation of ILL and the previous ones, is that our system contains rules introducing two connectives at once, namely the pairs $(\otimes, \multimap)$ and $(\oplus, \multimap)$. Therefore, both for $\otimes$ and $\oplus$ there is no an elimination rule, since this role is played, in both cases, by the rule eliminating the connective $\multimap$. They have been designed for having a uniform definition of reduction in the term language.

**Definition 3.1** *i) Let a context be a multiset of formulas of $\mathcal{F}_{ILL}$. By abuse of notation, we will use greek capital letters for ranging over this new notion of context. Let $\Gamma$ be a context. We shall denote by $\Gamma^*$ its restriction to $\mathcal{F}_{ILL} \backslash !\mathcal{F}_{ILL}$ and by $!\Gamma$ the restriction of $\Gamma$ to $!\mathcal{F}_{ILL}$. By $\Gamma, \Delta$ we denote the multiset union, i.e., the union with the sum of multiplicities.*

*ii) The system $\vdash_{\mathcal{N}}$ proves judgments $\Gamma \vdash_{\mathcal{N}} A$, where $A \in \mathcal{F}_{ILL}$, and $\Gamma$ is a context. It consists of the following rules:*

$$\dfrac{A \in \mathcal{F}_{ILL}}{!\Gamma, \{A\} \vdash_{\mathcal{N}} A}(Id)$$

$$\frac{\Gamma^*, !\Theta \vdash_{\mathcal{N}} A \multimap B \quad \Delta^*, !\Theta \vdash_{\mathcal{N}} A}{\Gamma^*, \Delta^*, !\Theta \vdash_{\mathcal{N}} B}(\multimap E) \quad \frac{\Gamma, \{A\} \vdash_{\mathcal{N}} B}{\Gamma \vdash_{\mathcal{N}} A \multimap B}(\multimap I)$$

$$\frac{\Gamma, \{A\}, \{B\} \vdash_{\mathcal{N}} C}{\Gamma \vdash_{\mathcal{N}} (A \otimes B) \multimap C}(\otimes \multimap I) \quad \frac{\Gamma^*, !\Theta \vdash_{\mathcal{N}} A \quad \Delta^*, !\Theta \vdash_{\mathcal{N}} B}{\Gamma^*, \Delta^*, !\Theta \vdash_{\mathcal{N}}: A \otimes B}(\otimes I)$$

$$\frac{\Gamma, \{A\} \vdash_{\mathcal{N}} C \quad \Gamma, \{B\} \vdash_{\mathcal{N}} C}{\Gamma \vdash_{\mathcal{N}} (A \oplus B) \multimap C}(\oplus \multimap I) \quad \frac{\Gamma \vdash_{\mathcal{N}} A}{\Gamma \vdash_{\mathcal{N}} A \oplus B}(\oplus I_l) \quad \frac{\Gamma \vdash_{\mathcal{N}} B}{\Gamma \vdash_{\mathcal{N}} A \oplus B}(\oplus I_r)$$

$$\frac{\Gamma \vdash_{\mathcal{N}} A\&B}{\Gamma \vdash_{\mathcal{N}} A}(\&E_l) \quad \frac{\Gamma \vdash_{\mathcal{N}} A\&B}{\Gamma \vdash_{\mathcal{N}} B}(\&E_r) \quad \frac{\Gamma \vdash_{\mathcal{N}} A \quad \Gamma \vdash_{\mathcal{N}} B}{\Gamma \vdash_{\mathcal{N}} A\&B}(\&I)$$

$$\frac{\Gamma \vdash_{\mathcal{N}} !A}{\Gamma \vdash_{\mathcal{N}} A}(!E) \quad \frac{!\Gamma \vdash_{\mathcal{N}} A}{!\Gamma \vdash_{\mathcal{N}} !A}(!I)$$

Let notice that the system $\vdash_{\mathcal{N}}$ does not explicitly contain structural rules. Both the weakening and the contraction are derived rules, and this property is reached by the additive behavior of the context, when restricted to its modal part. Moreover, $\vdash_{\mathcal{N}}$ and $\vdash_{\mathcal{S}}$ are equivalent, as they essentially prove the same set of judgments. Let introduce some further notation before demonstrating such property. Let $\Gamma$ be a natural deduction context, i.e, a multiset of formulas. $\Gamma^+$ denotes a sequence built on $\Gamma$. Therefore, $(\ )^+$ makes $\Gamma$ into a sequent calculus context.

**Property 3.1** *i) If* $\Gamma \vdash_{\mathcal{N}} A$*, then* $\Gamma, !\Delta \vdash_{\mathcal{N}} A$*, for every* $!\Delta$*.*

*ii) If* $\Gamma, \{!B\}, \{!B\} \vdash_{\mathcal{N}} A$*, then* $\Gamma, \{!B\} \vdash_{\mathcal{N}} A$*.*

*iii)* $\Gamma \vdash_{\mathcal{N}} A$ *if, and only if,* $\Gamma^+ \vdash_{\mathcal{S}} A$*, for every* $\Gamma^+$*.*

**Proof.**

i) and ii) By induction on the length of the derivation. The arbitrariness of the context of $(Id)$, in its modal part, and the additive nature of the modal contexts constitute the keypoints to conclude the proof.

iii) ($\Rightarrow$) By induction on the length of the derivation of $\Gamma \vdash_{\mathcal{N}} A$.

($\Leftarrow$) By induction on the length of the derivation of $\Gamma \vdash_{\mathcal{S}} A$, using *i)* and *ii)*. Since derivations in $\vdash_{\mathcal{S}}$ are normalizing, we can consider only derivations without applications of the ($Cut$)-rule. In case the last applied rule is a structural rule, use i) and ii). Cases where the last applied rule is an introduction on the right are easy. In the cases the last applied rule is an introduction on the left, the corresponding derivation in the system $\vdash_{\mathcal{N}}$ can be built without applying the substitution on the deductions of the system $\vdash_{\mathcal{N}}$. For example, let the last applied rule be:

$$\frac{\Gamma \vdash_{\mathcal{S}} A \quad B, \Delta \vdash_{\mathcal{S}} C}{\Gamma, A \multimap B, \Delta \vdash_{\mathcal{S}} C}(\multimap L)$$

By induction, $\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} A$ and $\{B\}, \Delta_{\mathcal{N}} \vdash_{\mathcal{N}} C$, $\Gamma_{\mathcal{N}}, \Delta_{\mathcal{N}}$ being the multisets corresponding to $\Gamma$ and $\Delta$. By point i), $\Gamma_{\mathcal{N}}, !\Delta_{\mathcal{N}} \vdash_{\mathcal{N}} A$ and $\{B\}, \Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} C$. Then the corresponding derivation in $\vdash_{\mathcal{N}}$ is:

$$\frac{\dfrac{\Delta^*_{\mathcal{N}}, \{B\}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} C}{\Delta^*_{\mathcal{N}}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} B \multimap C}(\multimap I) \quad \dfrac{\{A \multimap B\}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} A \multimap B \quad \Gamma^*_{\mathcal{N}}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} A}{\Gamma^*_{\mathcal{N}}, \{A \multimap B\}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{N}} B}(\multimap E)}{\Delta^*_{\mathcal{N}}, \Gamma^*_{\mathcal{N}}, \{A \multimap B\}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}} \vdash_{\mathcal{S}} C}(\multimap E)$$

where $\Delta^*_{\mathcal{N}}, \Gamma^*_{\mathcal{N}}, !\Delta_{\mathcal{N}}, !\Gamma_{\mathcal{N}}$ is $\Delta, \Gamma$.

The other cases of introduction on the left are similar.

□

A short remark is in order here. As Property 3.1.iii) states, $\vdash_{\mathcal{S}}$ and $\vdash_{\mathcal{N}}$ prove the same judgments (modulo the different representations of the contexts). There is not a one-one correspondence between derivations in the two systems. More precisely, a derivation in $\vdash_{\mathcal{N}}$ encodes a set of derivations in $\vdash_{\mathcal{S}}$. This because $\vdash_{\mathcal{N}}$ does not explicitly contain structural rules. A deduction of $\vdash_{\mathcal{N}}$ represents derivations of $\vdash_{\mathcal{S}}$ either differing from each other in the order of applications of structural rules, or obtained by eliminating cuts involving structural rules.

We have repeatedly stated that the substitution property partially holds for $\vdash_{\mathcal{N}}$. In the following we shall make this claim clear. Let remind what the substitution property is for a deductive system, say $\rhd$. Let be given a *main* derivation $\Pi_1 : \Gamma, B \rhd A$ and a *secondary* derivation $\Pi_2 : \Delta \rhd B$. If, replacing each instance of the axioms of $\Pi_1$, proving $B$, by $\Pi_2$ we always get a legal derivation $\Pi : \Gamma, \Delta \rhd A$, then we say that the deductive system $\rhd$ enjoys the substitution property.

**Definition 3.2** *$\Gamma \vdash^!_{\mathcal{N}} A$ if, and only if, $\Gamma \vdash_{\mathcal{N}} A$ and, if $A \equiv !B$, for some $B$, then $\Gamma = !\Gamma$. A judgment of the shape $\Gamma \vdash^!_{\mathcal{N}} A$ will be called $!$-judgment.*

**Property 3.2 (Substitution property)** *The system $\vdash_{\mathcal{N}}$ enjoys the substitution property in case the secondary derivations proves a $!$-judgment.*

**Proof.** By induction on the length of the main derivation. □

**Definition 3.3** *The* cuts *in the system* $\vdash_{\mathcal{N}}$ *are deductions of the following shape:*

$\multimap$ *-cut*

$$\dfrac{\dfrac{\Gamma^*, \{A\}, !\Theta \vdash_{\mathcal{N}} B}{\Gamma^*, !\Theta \vdash_{\mathcal{N}} A \multimap B}(\multimap I) \quad \Delta^*, !\Theta \vdash^!_{\mathcal{N}} A}{\Gamma^*, \Delta^*, !\Theta \vdash_{\mathcal{N}} B}(\multimap E)$$

$\otimes$*-cut*

$$\dfrac{\dfrac{\Gamma_1^*, \{A\}, \{B\}, !\Theta \vdash_{\mathcal{N}} C}{\Gamma_1^*, !\Theta \vdash_{\mathcal{N}} (A \otimes B) \multimap C}(\otimes \multimap I) \quad \dfrac{\Gamma_2^*, !\Theta \vdash^!_{\mathcal{N}} A \quad \Delta^*, !\Theta \vdash^!_{\mathcal{N}} B}{\Gamma_2^*, \Delta^*, !\Theta \vdash_{\mathcal{N}} A \otimes B}(\otimes I)}{\Gamma_1^*, \Gamma_2^*, \Delta^*, !\Theta \vdash_{\mathcal{N}} C}(\multimap E)$$

$\oplus$*-cut*

$$\dfrac{\dfrac{\Gamma^*, \{A\}, !\Theta \vdash_{\mathcal{N}} C \quad \Gamma^*, \{B\}, !\Theta \vdash_{\mathcal{N}} C}{\Gamma^*, !\Theta \vdash_{\mathcal{N}} (A \oplus B) \multimap C}(\oplus \multimap I) \quad \dfrac{\Delta^*, !\Theta \vdash^!_{\mathcal{N}} A}{\Delta^*, !\Theta \vdash_{\mathcal{N}} A \oplus B}(\oplus I)}{\Delta^*, \Gamma^*, !\Theta \vdash_{\mathcal{N}} C}(\multimap E)$$

$\&_l$*-cut*

$$\dfrac{\dfrac{\Gamma \vdash_{\mathcal{N}} A \quad \Gamma \vdash_{\mathcal{N}} B}{\Gamma \vdash_{\mathcal{N}} A \& B}(\& I)}{\Gamma \vdash_{\mathcal{N}} A}(\& E_l)$$

$\&_r$*-cut*

$$\dfrac{\dfrac{\Gamma \vdash_{\mathcal{N}} A \quad \Gamma \vdash_{\mathcal{N}} B}{\Gamma \vdash_{\mathcal{N}} A \& B}(\& I)}{\Gamma \vdash_{\mathcal{N}} B}(\& E_r)$$

*!-cut*

$$\dfrac{\dfrac{!\Gamma \vdash_{\mathcal{N}} A}{!\Gamma \vdash_{\mathcal{N}} !A}(!I)}{!\Gamma \vdash_{\mathcal{N}} A}(!E)$$

*A deduction is* normal *if, and only if, it has no cuts.*

**Theorem 3.1** *The system* $\vdash_{\mathcal{N}}$ *is normalizing, i.e., if* $\Gamma \vdash_{\mathcal{N}} A$*, then it can be proved by a normal derivation.*

**Proof.** The proof is quite standard. Let the degree of a cut be the number of symbols of the formula in its major premise. Let the degree of a derivation be the pair $(d, n)$, where $d$ is the maximum degree of a cut in it, and $n$ is the number of cuts of degree $d$. A normal derivation has the degree $(0, 0)$. Let $\Pi : \Gamma \vdash_{\mathcal{N}} A$, and let $\Pi$ be not normal. It must be proved that there is $\Pi' : \Gamma \vdash_{\mathcal{N}} A$, such that the degree of $\Pi'$ is smaller than the degree of $\Pi$. It is easy to design an effective procedure eliminating at each step a cut of maximum degree in $\Pi$ not below cuts of maximum degree. $\square$

# 4 The language $\Lambda_!$

In this section we introduce the typed language $\Lambda_!$ in the spirit of the Curry-Howard isomorphism. The system $\vdash_{\mathcal{N}}$ is used as generator of the terms of $\Lambda_!$. Namely, $\Lambda_!$ is defined through a typed system $\vdash$, whose rules are those of $\vdash_{\mathcal{N}}$ "decorated" by terms. Every term will encode a derivation in $\vdash_{\mathcal{N}}$, proving its type.

**Definition 4.1** *i) Let $Var$ and $!Var$ be two distinguished countable set of variables, ranged over, respectively, by $x$, $y$, $z$, and $!x$, $!y$, $!z$. $X, Y, Z$ will range over $Var \cup !Var$. A basis $\Gamma$ is a partial function from $Var \cup !Var$ to $\mathcal{F}_{ILL}$, satisfying the following constraint: $\Gamma(X)$ is a modal formula if and only if $X \in !Var$. Moreover , if $\Gamma$ is a basis, $\Gamma^*$ will be its restriction to $Var$, and $!\Gamma$ its restriction to $!Var$. If $\Gamma$ and $\Delta$ are two basis with disjoint domain,then $\Gamma$, $\Delta$ is the basis such that:*

$$\Gamma, \Delta(X) = \begin{cases} \Gamma(X) & \textit{if } X \in Dom(\Gamma) \\ \Delta(X) & \textit{if } X \in Dom(\Delta) \end{cases}$$

*Notice that we still make an abuse of notation, using greek capital letters for ranging over the set of basis.*

*ii) The system $\vdash$ proves judgments of the shape $\Gamma \vdash M : A$, where $\Gamma$ is a basis and $A \in \mathcal{F}_{ILL}$. It consists of the following rules:*

$$\frac{A \in \mathcal{F}_{ILL}}{!\Gamma, \{X : A\} \vdash X : A}(Id)$$

$$\frac{\Gamma^*, !\Theta \vdash M : A \multimap B \quad \Delta^*, !\Theta \vdash N : A \quad Dom(\Gamma^*) \cap Dom(\Delta^*) = \emptyset}{\Gamma^*, \Delta^*, !\Theta \vdash (MN) : B}(\multimap E)$$

$$\frac{\Gamma, \{X : A\} \vdash M : B}{\Gamma \vdash (\lambda X : A.M) : A \multimap B}(\multimap I)$$

$$\frac{\Gamma, \{X : A\}, \{Y : B\} \vdash M : C}{\Gamma \vdash (\lambda X \otimes Y : A \otimes B.M) : (A \otimes B) \multimap C}(\otimes \multimap I)$$

$$\frac{\Gamma^*, !\Theta \vdash M : A \quad \Delta^*, !\Theta \vdash N : B \quad Dom(\Gamma^*) \cap Dom(\Delta^*) = \emptyset}{\Gamma^*, \Delta^*, !\Theta \vdash (M \otimes N) : A \otimes B}(\otimes I)$$

$$\frac{\Gamma, \{X : A\} \vdash M : C \quad \Gamma, \{Y : B\} \vdash N : C}{\Gamma \vdash (\lambda X \oplus Y : A \oplus B.M|N) : (A \oplus B) \multimap C}(\oplus \multimap I)$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash (M \oplus \_) : A \oplus B}(\oplus I_l)$$

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash (\_ \oplus M) : A \oplus B}(\oplus I_r)$$

$$\frac{\Gamma \vdash M : A\&B}{\Gamma \vdash \mathbf{l}(M) : A}(\&E_l) \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M\&N) : A\&B}(\&I)$$

$$\frac{\Gamma \vdash M : A\&B}{\Gamma \vdash \mathbf{r}(M) : B}(\&E_r)$$

$$\frac{\Gamma \vdash M :!A}{\Gamma \vdash \mathbf{d}(M) : A}(!E) \qquad \frac{!\Gamma \vdash M : A}{!\Gamma \vdash !M :!A}(!I)$$

**Definition 4.2** *i) $M \in \Lambda_!$ if, and only if, there are a basis $\Gamma$ and a type $A$ such that $\Gamma \vdash M : A$. $!\Lambda_!$ is the subset of $\Lambda_!$ whose elements either are in $!Var$ or have the shape $!M$, for some $M \in \Lambda_!$.*

*ii) For any $M \in \Lambda_!$, the set $FV(M)$ of its* free variables *is inductively given on the structure of the terms in the following way:*

$$\begin{array}{lcl}
FV(X) & = & \{X\} \\
FV(\lambda X : A.M) & = & FV(M) \setminus \{X\} \\
FV(MN) & = & FV(M) \cup FV(N) \\
FV(\lambda X \otimes Y : A \otimes B.M) & = & FV(M) \setminus \{X, Y\} \\
FV(M \otimes N) & = & FV(M) \cup FV(N) \\
FV(\lambda X \oplus Y : A \oplus B.M|N) & = & FV(M) \setminus \{X, Y\} \\
FV(M \oplus _) & = & FV(M) \\
FV(_ \oplus M) & = & FV(M) \\
FV(M\&N) & = & FV(M) \cup FV(N) \\
FV(\mathbf{l}(M)) & = & FV(M) \\
FV(\mathbf{r}(M)) & = & FV(M) \\
FV(!M) & = & FV(M) \\
FV(\mathbf{d}(M)) & = & FV(M)
\end{array}$$

**Property 4.1** *i) If $\Gamma \vdash M : A$, then $FV(M) \subseteq Dom(\Gamma)$.*

*ii) If $\Gamma \vdash M : A$, then there exists $\Delta$ such that $\Delta \vdash M : A$ and $Dom(\Delta) = FV(M)$.*

**Proof.** Both i) and ii) by induction on the length of the derivation of $\Gamma \vdash M : A$. □

The following property assures us that a term of $\Lambda_!$ correctly encodes a proof of $\vdash_{\mathcal{N}}$.

**Property 4.2 (Unicity property)** *Let $\Gamma \vdash M : A$. There is a unique derivation proving this judgment.*

**Proof.** Let proceed by induction on the length of the deduction of $\Gamma \vdash M : A$, once proved in the same way the lemma

**Lemma 4.1** *Let $\Gamma \vdash M : A$. For every $A'$ and $\Gamma'$ such that $\forall X \in FV(M).\Gamma(X) = \Gamma'(X)$, if $\Gamma' \vdash M : A'$, then $A$ is $A'$.*

$\Box$

We think that some remark on the use of two sorts of variables is in order here. From a logical point of view, a variable in $Var$ corresponds to a premise which is used exactly once. On the contrary, a variable in $!Var$ corresponds to a premise either used more than once, or not used at all. This means that, in the corresponding derivation of the system $\vdash_{\mathcal{S}}$, a weakening or a contraction rule has been applied. In this way the language $\Lambda_!$ has a partial linearity constraint on $Var$. For formally expressing such a constraint, we need to introduce some definitions.
The set $\{(Id), (-\circ\ I), (-\circ\ E), (\otimes\ -\circ\ I), (!I), (!E)\}$ of rules constitutes the multiplicative fragment of $\vdash$. Let $\Gamma \vdash M : A$ be derived in the multiplicative fragment. Then $M$ will be called a *multiplicative term*. It is easy to verify that $M$ is a multiplicative term if, and only if, it is of the shape $X$, $P \otimes Q$, $\lambda X \otimes Y : A \otimes B.P$, $\lambda X : A.P$, $PQ$, $!P$, where $P$ and $Q$ are, in turn, multiplicative. Otherwise a term is called *additive*.

**Definition 4.3** *Two subterms $P$ and $Q$ of a term $M \in \Lambda_!$ are said* disjoint terms in $M$ *if, and only if, both*

- *exists a subterm $N$ of $M$ having either the shape $\lambda X \oplus Y : A \oplus B.P'|Q'$ or the shape $P'\&Q'$,*
- *$P$ is a subterm of $P'$ and $Q$ is a subterm of $Q'$.*

**Property 4.3** *i) Let $M$ be a multiplicative term and $X \in Var$.*
*If $X \in FV(M)$, then $X$ occurs exactly once in $M$.*
*If $X$ is bound, then $X$ occurs exactly once in its scope.*

*ii) Let $M$ be additive and $X \in Var$.*
*If $X \in FV(M)$ and $X$ occurs $n > 1$ times in $M$, then exist $n$ pairwise disjoint terms of $M$ where $X$ occurs exactly once.*
*If $X$ is bound and $X$ occurs $n > 1$ times in $M$, then exist $n$ pairwise disjoint terms in its scope where $X$ occurs exactly once.*

*iii) $FV(!M) \subset !Var$.*

**Proof.** All points can be proved by induction on the length of the derivation. $\Box$

Now we introduce the reduction rules of our language. The $\alpha$-rule, i.e. the rule for renaming bound variables, is shown in the next definition. Observe that a variable can be replaced only by another variable of the same sort.

**Definition 4.4** *Let $M[Y/X]$ denote the term $M$ where all free occurrences of the variable $X$ are replaced by occurrences of $Y$.*

*i)* $\to_\alpha$ *is the reduction rule defined as follows:*

- $(\lambda X : A.M) \to_\alpha (\lambda Y : A.M[Y/X]) \Leftrightarrow (Y \notin FV(M) \wedge (X \in Var \Leftrightarrow Y \in Var))$
- $(\lambda X \otimes Y : A \otimes B.M) \to_\alpha (\lambda X' \otimes Y' : A \otimes B.M[X'/X][Y'/Y]) \Leftrightarrow ((X', Y' \notin FV(M) \wedge (X \in Var \Leftrightarrow X' \in Var) \wedge (Y \in Var \Leftrightarrow Y' \in Var))$
- $(\lambda X \oplus Y : A \oplus B.P|Q) \to_\alpha (\lambda X' \oplus Y' : A \oplus B.P[X'/X]|Q[Y'/Y]) \Leftrightarrow ((X' \notin FV(P) \wedge (Y' \notin FV(Q) \wedge (X \in Var \Leftrightarrow X' \in Var) \wedge (Y \in Var \Leftrightarrow Y' \in Var))$

*ii)* $\to^*_\alpha$ *is the reflexive, transitive and contextual closure of* $\to_\alpha$. $=_\alpha$ *is the reflexive, transitive, contextual, and symmetric closure of* $\to_\alpha$.

For defining the computational rules, first we need to define the operation of substitution of a variable by a term. Again, it is necessary to take into account that variables can be of different sorts. So a variable in $!Var$, representing a reusable resource, can be replaced only by a modal term: the substitution turns out to be a partial function.

**Definition 4.5** *The substitution of a term* $N \in \Lambda_!$ *for a free variable* $X \in (Var \cup !Var)$ *in a term* $M \in \Lambda_!$ *(notation* $M[N/X]$*) is a partial function so defined:*

$M[N/X] =$ *if* $(X \in !Var \Leftrightarrow N \in !\Lambda_!)$
*then the usual capture free substitution, using the* $\alpha$*-rule of the definition 4.4*
*else undefined.*

*We will denote by* $M[N_1/X_1, \ldots, N_n/X_n]$ *the simultaneous substitution of every* $N_i$ *for all the free occurrences of* $x_i$ *in* $M$*, minding the agreement of the sorts.*

The substitution property holds for $\Lambda_!$.

**Property 4.4 (Substitution property)** *If* $\Gamma^*, !\Theta, \{X : A\} \vdash M : B$ *and* $\Delta^*, !\Theta \vdash N : A$ *and* $M[N/X]$ *is defined, then* $\Gamma^*, \Delta^*, !\Theta \vdash M[N/X] : B$.

**Proof.** By induction on the length of the derivation of $\Gamma^*, !\Theta, \{X : A\} \vdash M : B$. □

The reduction rules for $\Lambda_!$ are given in next definition. As it can be easily checked, every reduction corresponds to a cut, according to Definitions 3.3 and 4.5.

**Definition 4.6** *i) Let* $\to$ *denote the reduction rule defined as follows:*

- $(\lambda X : A.M_1)M_2 \to M_1[M_2/X]$ *if* $M_1[M_2/X]$ *is defined;*
- $(\lambda X_1 \otimes X_2 : A \otimes B.M)(M_1 \otimes M_2) \to M[M_1/X_1, M_2/X_2]$ *if* $M[M_1/X_1, M_2/X_2]$ *is defined;*
- $(\lambda X \oplus Y : A \oplus B.M_1|M_2)(M \oplus \_) \to M_1[M/X]$ *if* $M_1[M/X]$ *is defined;*
- $\mathbf{l}(M_1 \& M_2) \to M_1$*;*
- $\mathbf{r}(M_1 \& M_2) \to M_2$*;*

- $\mathbf{d}(!M) \to M$;

*ii)* $\to^*$ *is the contextual, transitive and reflexive closure of* $\to$. $\approx$ *is the contextual, transitive, reflexive, and symmetric closure of* $\to$. *The contextual and transitive closure of* $\to$ *is* $\to^+$.

The subject reduction is the key property for a type system of a language.

**Theorem 4.1** *i)* **(Subject reduction)** *If* $\Gamma \vdash M : A$ *and* $M \to N$, *then* $\Gamma \vdash N : A$.

*ii) The terms of* $\Lambda_!$ *have a normal form.*

**Proof.**

i) By induction on the the structure of $M$, using both Substitution and Unicity properties.

ii) The proof follows from the normalizability of $\vdash_{\mathcal{N}}$ and the definition of the reduction rule for $\Lambda_!$.

□

Moreover, $\Lambda_!$ has all the good properties we would like it had.

**Theorem 4.2** *i) The terms of* $\Lambda_!$ *are strongly normalizable.*

*ii) The rewriting system* $\to$ *enjoys the* $Church - Rosser$ *property.*

**Proof.** Both i) and ii) will be proved in Appendix A. □

# 5 Categorical model for $\Lambda_!$

In this section we will give the denotational semantics of $\Lambda_!$ in a categorical setting. Because of the way $\Lambda_!$ was constructed, a model for it is a model for derivations in the system $\vdash_{\mathcal{N}}$. Let define a suitable category for $\Lambda_!$, based on [4].

**Definition 5.1** *An* intuitionistic linear category $\mathbf{C}$ *is a category such that:*

- $\mathbf{C}$ *is monoidal symmetric with respect to the bifunctor* $\otimes$. *The unit of* $\otimes$ *is 1.*
- $\mathbf{C}$ *is closed with respect to the bifunctor* $\multimap$ *, i.e., for all* $a, b, c \in Obj_{\mathbf{C}}$, *there exists a natural isomorphism* $\Lambda : Hom_{\mathbf{C}}(a \otimes b, c) \to Hom_{\mathbf{C}}(a, b \multimap c)$.
- $\mathbf{C}$ *is cartesian and co-cartesian.*
- $\mathbf{C}$ *is enriched by a symmetric monoidal comonad* $(!, \delta :! \to !!, \varepsilon :! \to \mathbf{Id}_{\mathbf{C}}, m_{A,B}, m_1)$ *such that:*

– *if* $1 : \mathbf{C} \to \mathbf{C}$ *denotes the constant functor, then there exist two natural transformations* $E_{\_} : !\_ \to 1$ *and* $D_{\_} : !\_ \to !\_ \otimes !\_$ *such that, for all* $a \in Obj_{\mathbf{C}}$, $(!a, D_a, E_a)$ *is a commutative comonoid.*

– $\delta$ *is an element of* $(\_ \otimes \_)$-$\mathbf{coalg}_{\mathbf{C}}((!a, D_a), (!!a, D_{!a}))$ *and* $1$-$\mathbf{coalg}_{\mathbf{C}}((!a, D_a), (!!a, D_{!a}))$.

In the following, for a better reading, we shall drop subscripts and superscripts on the morphisms of $\mathbf{C}$, when they are clear from the context.

Let us notice that the linear category just introduced is slightly different from the category defined in [4] for interpreting sequents of ILL. As already remarked in the introduction, the differences originate from the syntax of $\Lambda_!$ which is simpler than those already proposed.

**Notations** Let $Obj_{\mathbf{C}}$ be ranged over by $a, b, c, \ldots$.

- For all $f : c \to a$ and $g : c \to b$, we shall name $\langle f, g\rangle$ the unique arrow such that the diagram of the cartesian product commutes:

$$\begin{tikzcd}
& c \arrow[dl, "f"'] \arrow[d, dotted, "\langle f,g\rangle"] \arrow[dr, "g"] & \\
a & a \times b \arrow[l, "p^l_{a,b}"] \arrow[r, "p^r_{a,b}"'] & b
\end{tikzcd}$$

- For every $f : a \to c$ and $g : b \to c$, we shall name $f|g$ the unique arrow such that the diagram of the cartesian coproduct commutes:

$$\begin{tikzcd}
a \arrow[r, "in^l_{a,b}"'] & a + b & b \arrow[l, "in^r_{a,b}"] \\
& c \arrow[ul, "f"] \arrow[u, dotted, "f|g"] \arrow[ur, "g"'] &
\end{tikzcd}$$

- For every $f : (a \otimes b) \to c$, we name $ev_{b,c}$ the evaluation map such that the following diagram commutes:

$$\begin{tikzcd}
a \otimes b \arrow[r, "f"] \arrow[d, dotted, "\Lambda(f)\otimes id_b"'] & c \\
b \multimap c \otimes b \arrow[ur, "ev_{b,c}"'] &
\end{tikzcd}$$

- For every $a, b \in Obj_{\mathbf{C}}$, $m_{a,b} : !a \otimes !b \to !(a \otimes b)$ and $m_1 : 1 \to !1$ are the maps making ! a monoidal functor and $\delta$ and $\varepsilon$ monoidal natural transformations.
- We call $\jmath_{a\otimes(b+c)}$ the natural isomorphism between $a \otimes (b + c)$ and $(a \otimes b) + (a \otimes c)$. It exists because $a \otimes \_$ is left adjoint to $a \multimap \_$ and, therefore, $a \otimes \_$ preserves all colimits of $\mathbf{C}$.

In next definition, we introduce some morphisms useful for defining the interpretation of a term in a simple and concise way, while preserving its correctness.

**Definition 5.2**
- *For all $a_1, \ldots, a_n \in Obj_{\mathbf{C}}$ and for every permutation $\sigma$ of the sequence $1, \ldots, n$, we call $exc^{a_1\otimes\ldots\otimes a_n}_{a_{\sigma(1)}\otimes\ldots\otimes a_{\sigma(n)}}$ the natural isomorphism between $a_1 \otimes \ldots \otimes a_n$ and $a_{\sigma(1)} \otimes \ldots \otimes a_{\sigma(n)}$. The isomorphism exists because of the definition of intuitionistic linear category.*
- *Let $!Obj_{\mathbf{C}}$ be the subset of $Obj_{\mathbf{C}}$ containing objects with shape $!a$, for some $a \in Obj_{\mathbf{C}}$.*
- *Let $iso_{1\otimes 1}$ be the isomorphism between $1$ and $1 \otimes 1$. We define the map*

  *$D^*_{a_1\otimes\ldots\otimes a_n, b_1\otimes\ldots\otimes b_m} : a_1 \otimes \ldots \otimes a_n \to b_1 \otimes \ldots \otimes b_m$ as follows:*

$$
\begin{array}{ll}
D^*_{1,1\otimes 1} = iso_{1\otimes 1} & \\
D^*_{a,a} = id_a & \textit{if } a \notin (!Obj_{\mathbf{C}} \cup \{1\}) \\
D^*_{!a,!a\otimes !a} = D_a & \\
D^*_{a_1\otimes\ldots\otimes a_{n-1}\otimes a_n, b_1\otimes\ldots\otimes b_m\otimes a_n} = D^*_{a_1\otimes\ldots\otimes a_{n-1}, b_1\otimes\ldots\otimes b_m} \otimes id_{a_n} & \textit{if } a_n \notin !Obj_{\mathbf{C}} \\
D^*_{a_1\otimes\ldots\otimes a_{n-1}\otimes !a_n, b_1\otimes\ldots\otimes b_m\otimes !a_n\otimes !a_n} = D^*_{a_1\otimes\ldots\otimes a_{n-1}, b_1\otimes\ldots\otimes b_m} \otimes D_{a_n} &
\end{array}
$$

- *Let $H^{\gamma}_{\gamma'} : \gamma \to \gamma'$ be an abbreviation for $exc^{\delta}_{\gamma'} \circ D^*_{\gamma,\delta}$.*
- *Let $a_1, \ldots, a_n$ be objects of $\mathbf{C}$. Suppose that, for all $i \in I \subset \{1, \ldots, n\}$, $a_i \equiv 1$ and, $a_k \not\equiv 1$, for all $k \in K = \{1, \ldots, n\} \setminus I$.*

  *Let $iso_{\otimes_{k\in K} a_k}$ be the natural isomorphism between $a_1 \otimes \ldots \otimes a_n$ and $\bigotimes_{k\in K} a_k$.*

  *We call $\Pi^j_{a_1\otimes\ldots\otimes a_n} : a_1 \otimes \ldots \otimes a_n \to a_j$ the morphism*

  *$iso_{a_j} \circ (\bigotimes_{i=1}^{j-1} E_{a_i} \otimes id_{a_j} \otimes \bigotimes_{i=j+1}^{n} E_{a_i})$.*

  *We consider it a sort projection because, for all morphisms $f : a \to b$, the two diagrams*

$$
\begin{tikzcd}
!c \otimes a \arrow[r, "\Pi^1_{!c\otimes a}"'] \arrow[d, "id_{!c}\otimes f"] & a \arrow[d, "f"'] \\
!c \otimes b \arrow[r, "\Pi^1_{!c\otimes b}"] & b
\end{tikzcd}
\qquad
\begin{tikzcd}
a \otimes !c \arrow[r, "\Pi^2_{a\otimes !c}"'] \arrow[d, "f\otimes id_{!c}"] & a \arrow[d, "f"'] \\
b \otimes !c \arrow[r, "\Pi^2_{b\otimes !c}"] & b
\end{tikzcd}
$$

  *commute.*

Our aim is to prove that every intuitionistic linear category is a model for $\Lambda_!$. For doing that, we define the interpretation of a judgment $\Gamma \vdash M : A$ as a morphism of $\mathbf{C}$, where $\mathbf{C}$ is an intuitionistic linear category. This results in interpreting $M$ itself as morphism of $\mathbf{C}$.

We start by defining the interpretation of a formula in $\mathcal{F}_{ILL}$. Clearly, the definition of the interpretation, both of a formula and of a term, should be indexed by the particular category we are working on. For reasons of readability, we shall drop such indexes.

**Definition 5.3** *i) An environment is a map from $\mathcal{A}_{ILL}$ to $Obj_{\mathbf{C}}$. We call it $\rho$.*

*ii) The interpretation of a formula in $\mathcal{F}_{ILL}$ is the function $[\![\ ]\!] : \mathcal{F}_{ILL} \to (\mathcal{A}_{ILL} \to Obj_{\mathbf{C}}) \to Obj_{\mathbf{C}}$, defined as follows:*

$$
\begin{array}{lll}
[\![A]\!]_\rho & = \rho(A) & \textit{if } A \in \mathcal{A}_{ILL}. \\
[\![A \multimap B]\!]_\rho & = [\![A]\!]_\rho \multimap [\![B]\!]_\rho & \\
[\![A \otimes B]\!]_\rho & = [\![A]\!]_\rho \otimes [\![B]\!]_\rho & \\
[\![A \& B]\!]_\rho & = [\![A]\!]_\rho \times [\![B]\!]_\rho & \\
[\![A \oplus B]\!]_\rho & = [\![A]\!]_\rho + [\![B]\!]_\rho & \\
[\![!A]\!]_\rho & = ![\![A]\!]_\rho &
\end{array}
$$

Let assume that there is a fixed total order relation $\leq \subseteq (Var \cup !Var) \times (Var \cup !Var)$. When we write a basis $\Gamma = \{X_1 : A_1, \ldots, X_n : A_n\}$, we assume that the order on indexes is given by sorting the names of the variables, according to $\leq$.

Given an environment $\rho$, a judgment $\{X_1 : A_1, \ldots, X_n : A_n\} \vdash M : A$ will be interpreted as a morphism $1 \otimes [\![A_1]\!]_\rho \otimes \ldots \otimes [\![A_n]\!]_\rho \to [\![A]\!]_\rho$. The order between the assumptions can be obtained by using the morphisms $H^{\gamma}_{\gamma'}$, for suitable $\gamma$ and $\gamma'$. For a better reading, we shall usually drop the environment $\rho$, when writing the interpretation of a term $M$. Further, we assume the following

**Notation**

- The interpretation $[\![A]\!]_\rho$ of a formula $A \in \mathcal{F}_{ILL}$ will be denoted by the lower letter $a$. For example, $[\![A_1 \otimes \ldots \otimes A_n]\!]_\rho$ will be represented by $a_1 \otimes \ldots \otimes a_n$.
- Let $\Gamma$ be the basis $\{X_1 : A_1, \ldots, X_n : A_n\}$. Then $1 \otimes [\![A_1]\!]_\rho \otimes \ldots \otimes [\![A_n]\!]_\rho$ will be denoted by $[\![\Gamma]\!]_\rho$, and, further, abbreviated by $\gamma$.

**Definition 5.4** *Let $\rho$ be a given environment. The interpretation of a judgment $\{X_1 : A_1, \ldots, X_n : A_n\} \vdash M : A$ in $\rho$ is denoted by $[\![\{X_1 : A_1, \ldots, X_n : A_n\} \vdash M : A]\!]_\rho$, and is a morphism $1 \otimes [\![A_1]\!]_\rho \otimes \ldots \otimes [\![A_n]\!]_\rho \to [\![A]\!]_\rho$. It is defined by induction on the derivation in the following way:*

- $[\![\{X_1 : A_1, \ldots, X_n : A_n\} \vdash X_i : A_i]\!]_\rho = \Pi^{i+1}_{1 \otimes a_1 \otimes \ldots \otimes a_n}$.

- $[\![\Gamma \vdash MN : B]\!]_\rho = ev_{a,b} \circ (f \otimes g) \circ H^{1\otimes\gamma}_{1\otimes\gamma_M\otimes 1\otimes\gamma_N}$
  *where* $f \equiv [\![\Gamma_M \vdash M : A \multimap B]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma_M, a \multimap b)$,
  $g \equiv [\![\Gamma_N \vdash N : A]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma_N, a)$.

- $[\![\Gamma \vdash \lambda X : A.M : A \multimap B]\!]_\rho = \Lambda(f)$
  *where* $f \equiv [\![\Gamma, \{X : A\} \vdash M : B]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma \otimes a, b)$.

- $[\![\Gamma \vdash \lambda X \otimes Y : A \otimes B.M : (A \otimes B) \multimap C]\!]_\rho = \Lambda(f)$
  *where* $f \equiv [\![\Gamma, \{X : A\}, \{Y : B\} \vdash M : C]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma \otimes a \otimes b, c)$.

- $[\![\Gamma \vdash \lambda X \oplus Y : A \oplus B.P|Q : (A \oplus B) \multimap C]\!]_\rho = \Lambda((f \mid g) \circ \jmath_{1\otimes\gamma\otimes(a+b)})$
  *where* $f \equiv [\![\Gamma, \{X : A\} \vdash P : C]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma \otimes a, c)$
  *and* $g \equiv [\![\Gamma, \{Y : B\} \vdash Q : C]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma \otimes b, c)$.

- $[\![\Gamma \vdash M\&N : A\&B]\!]_\rho = \langle f, g\rangle$
  *where* $f \equiv [\![\Gamma \vdash M : A]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma, a)$
  *and* $g \equiv [\![\Gamma \vdash N : B]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma, b)$.

- $[\![\Gamma \vdash \mathbf{l}(M) : A]\!]_\rho = p^l_{a,b} \circ f$ *and* $[\![\Gamma \vdash \mathbf{r}(M) : B]\!]_\rho = p^r_{a,b} \circ f$
  *where* $f \equiv [\![\Gamma \vdash M : A\&B]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma, a \times b)$.

- $[\![\Gamma \vdash M \otimes N : A \otimes B]\!]_\rho = (f \otimes g) \circ H^{1\otimes\gamma}_{1\otimes\gamma_M\otimes 1\otimes\gamma_N}$
  *where* $f \equiv [\![\Gamma_M \vdash M : A]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma_M, a)$,
  $g \equiv [\![\Gamma_N \vdash N : B]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma_N, b)$.

- $[\![\Gamma \vdash M \oplus \_ : A \oplus B]\!]_\rho = in^l_{a,b} \circ f$ *and* $[\![\Gamma \vdash \_ \oplus M : B \oplus A]\!]_\rho = in^r_{b,a} \circ f$
  *where* $f \equiv [\![\Gamma \vdash M : A]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma, a)$.

- $[\![\Gamma \vdash \mathbf{d}(M) : A]\!]_\rho = \varepsilon_a \circ f$
  *where* $f \equiv [\![\Gamma \vdash M :!A]\!]_\rho \in Hom_{\mathbf{C}}(1 \otimes \gamma, !a)$.

- $[\![!\Gamma \vdash !M :!A]\!]_\rho =!f \circ m_{1,a_1,\ldots,a_n} \circ (m_1 \otimes \bigotimes_{i=1}^n \delta_{a_i})$
  *where* $f \equiv [\![!\Gamma \vdash M : A]\!]_\rho \in Hom_{\mathbf{C}}(1\otimes!a_1 \otimes \ldots \otimes!a_n, a)$ *and* $n \geq 0$.

Let $\mathbf{C}$ be a particular intuitionistic linear category. $[\![\Gamma \vdash M : A]\!]^{\mathbf{C}}_\rho$ will denote the interpretation of $\Gamma \vdash M : A$ in the category $\mathbf{C}$. The semantic equivalence among terms, induced by the previous defined categorical interpretation, is introduced, in a standard way, as follows:

**Definition 5.5** *Let* $\Gamma \vdash M : A$ *and* $\Gamma \vdash N : A$. $M \sim_{\Gamma,A} N$ *if, and only if, for all intuitionistic linear category* $\mathbf{C}$*,* $\forall\rho.[\![\Gamma \vdash M : A]\!]^{\mathbf{C}}_\rho = [\![\Gamma \vdash N : A]\!]^{\mathbf{C}}_\rho$.

The following theorem proves the adequacy of the categorical semantics, i.e., terms reducing each other are equivalent in every categorical model.

**Theorem 5.1 (Adequacy)** *Let $\Gamma \vdash M : A$. If $M \to N$, then $M \sim_{\Gamma,A} N$.*

**Proof.** The proof follows from lemma 5.1 here below, which assures that the semantics preserves the syntactic substitution.

**Lemma 5.1 (Semantic substitution)** *Let $\Gamma^*_M, \{X : A\}, !\Gamma \vdash M : B$, $\Gamma^*_N, !\Gamma \vdash N : A$ and $\Gamma = \Gamma^*_M, \Gamma^*_N, !\Gamma$. Moreover let $[\![\Gamma^*_M, \{X : A\}, !\Gamma \vdash M : B]\!]$ be a morphism $1 \otimes \gamma' \otimes a \otimes \gamma'' \to b$. Then:*

$$[\![\Gamma \vdash M[N/X] : B]\!] = \\ [\![\Gamma^*_M, \{X : A\}, !\Gamma \vdash M : B]\!] \circ (id_{1\otimes\gamma'} \otimes [\![\Gamma^*_N, !\Gamma \vdash N : A]\!] \otimes id_{\gamma''}) \circ H^{1\otimes\gamma}_{1\otimes\gamma'\otimes 1\otimes\gamma^*_N\otimes\gamma''\otimes\gamma''}.$$

Lemma 5.1 is proved by induction on the length of the derivation of $\Gamma^*_M, !\Gamma, \{X : A\} \vdash M : B$, using $[\![\Gamma \vdash N : A]\!] \circ \mathrm{iso}^{1\otimes\gamma}_{\gamma} \circ (id_\gamma \otimes E_b) = [\![\Gamma, \{!x :!B\} \vdash N : A]\!]$, for all $!x \notin FV(N)$. □

As immediate consequence of the adequacy the following holds:

**Corollary 5.1** *Let $\Gamma \vdash M : A$. If $M \approx N$, then $M \sim_{\Gamma,A} N$.*

**Proof.** We know that $\approx$ is the reflexive, transitive, symmetric, and contextual closure of $\to$. So, $\approx$ can be written as the least relation satisfying the rules

$$\frac{M \to N}{M \approx N} \qquad \frac{M \approx N}{N \approx M} \qquad \frac{M \approx P \quad P \approx M}{N \approx M}$$

$$\frac{M \approx M' \quad N \approx N'}{MN \approx M'N'} \qquad \frac{M \approx M' \quad N \approx N'}{M \otimes N \approx M' \otimes N'} \qquad \frac{M \approx M' \quad N \approx N'}{M\&N \approx M'\&N'}$$

$$\frac{M \approx N}{M \oplus \_ \approx N \oplus \_} \qquad \frac{M \approx N}{\_ \oplus M \approx \_ \oplus N}$$

$$\frac{M \approx N}{\lambda X : A.M \approx \lambda X : A.N} \qquad \frac{M \approx N}{\lambda X \otimes Y : A \otimes B.M \approx \lambda X \otimes Y : A \otimes B.N}$$

$$\frac{M \approx M' \quad N \approx N'}{\lambda X \oplus Y : A \oplus B.M|N \approx \lambda X \oplus Y : A \oplus B.M'|N'}$$

$$\frac{M \approx N}{!M \approx !N} \qquad \frac{M \approx N}{\mathbf{d}(M) \approx \mathbf{d}(N)}$$

then the proof is by structural induction on $\approx$, using the adequacy. □

Since $\Lambda_!$ is a typed language, we expect that the semantic equivalence is extensional in every suitable categorical model. Clearly, the definition of extensionality must take into account that all kinds of patterns can be bound.

**Definition 5.6** *Let $\to_\eta$ denote the reduction rule defined as follows:*

*i)* – $\lambda X : A.MX \to_\eta M$ *if* $X \notin FV(M)$;

– $\lambda X \otimes Y : A \otimes B.M(X \otimes Y) \to_\eta M$ *if* $X, Y \notin FV(M)$;

– $(\lambda X \oplus Y : A \oplus B.(X \oplus \_)|(\_ \oplus Y))M \to_\eta M$;

– $\mathbf{l}(M)\&\mathbf{r}(M) \to_\eta M$;

– $!(\mathbf{d}(M)) \to_\eta M$.

*ii)* $\to_\eta^*$ *is the reflexive, transitive and contextual closure of* $\to_\eta$. $\approx_\eta$ *is the symmetric, reflexive, transitive, and contextual closure of* $\to_\eta$.

**Theorem 5.2** *Let* $\Gamma \vdash M : A$, $\Gamma \vdash N : A$. *Then* $M \approx_\eta N$ *implies* $M \sim_{\Gamma,A} N$.

**Proof.** First prove that $M \to_\eta N$ implies $M \sim_{\Gamma,A} N$. Then, proceed by induction on definition of $\approx_\eta$. □

Finally, we introduce the relation $\leftrightarrow$ among terms of $\Lambda_!$. $\leftrightarrow$ is induced by the model; this because the model identifies two terms $M$ and $N$ corresponding to two derivations $\Pi_M$ and $\Pi_N$ of $\vdash_{\mathcal{S}}$, obtained from each other by eliminating a *secondary* cut, i.e., a cut without computational meaning in $\vdash_{\mathcal{N}}$.

**Definition 5.7** *i) For every term* $M, N, P, Q$ *of* $\Lambda_!$,

– $P[(\lambda X \otimes Y : A \otimes B.M)N/Z] \leftrightarrow (\lambda X \otimes Y : A \otimes B.P[M/Z])N$, *where* $\{X, Y\} \cap FV(P) = \emptyset$;

– $P[(\lambda X \oplus Y : A \oplus B.M|N)Q/Z] \leftrightarrow (\lambda X \oplus Y : A \oplus B.P[M/Z]|P[N/Z])Q$, *where* $\{X, Y\} \cap FV(P) = \emptyset$;

*ii)* $\approx_c$ *is the contextual, transitive and symmetric closure of* $\leftrightarrow$.

**Theorem 5.3** *Let* $\Gamma \vdash M : A$, $\Gamma \vdash N : A$. *Then* $M \approx_c N$ *implies* $M \sim_{\Gamma,A} N$.

**Proof.** First prove that $M \leftrightarrow N$ implies $M \sim_{\Gamma,A} N$. Then, proceed by induction on the definition of $\approx_c$. □

**Definition 5.8** *Let* $\approx_{\eta c}$ *be* $\approx \cup \approx_\eta \cup \approx_c$.

The soundness of the term equivalence $\approx_{\eta c}$ w.r.t. the interpretation in every intuitionistic linear category follows:

**Corollary 5.2 (Soundness)** *Assume that* $\Gamma \vdash M : A$. *If* $M \approx_{\eta c} N$, *then* $M \sim_{\Gamma,A} N$.

**Proof.** Obvious. □

## 5.1 Completeness

Once proved that arrows interpreting two terms $M$ and $N$ are the same arrow whenever $M \approx_{\eta c} N$, we would like to know if the contrary holds. Namely, we would like to state a *completeness* result:

**Theorem 5.4 (Completeness)** *Assume that* $\Gamma \vdash M : A$. *If* $M \sim_{\Gamma,A} N$, *then* $M \approx_{\eta c} N$.

The strategy for proving such theorem would be the usual one: once taken the term model induced by $\approx_{\eta c}$ it is enough to show that the term model is an intuitionistic linear category.
For doing that we would certainly need to represent (by means of an equivalence class of terms in $\Lambda_!$) the morphisms $m_1$ and $E_1$. This is not possible because we can not explicitely deal with the unity 1 of the tensor $\otimes$ in $\Lambda_!$.
For being able to manage the unity, it is necessary to extend $\Lambda_!$. Our proposal is to write a language $\Lambda_!^1$ by adding to $\vdash$ the following two rules:

$$\frac{\Gamma^*, !\Theta \vdash M : 1 \quad \Delta^*, !\Theta \vdash N : A \quad Dom(\Gamma^*) \cap Dom(\Delta^*) = \emptyset}{\Gamma^*, \Delta^*, !\Theta \vdash \mathbf{let}\ M\ \mathbf{be} * \mathbf{in}\ N : A}(1E)$$

$$\frac{\begin{array}{l}\forall 1 \leq i \neq j \leq n. Dom(\Gamma_i^*) \cap Dom(\Gamma_j^*) = \emptyset \\ (\Gamma_i^*, !\Gamma \vdash M_i :!A_i)_{0 \leq i \leq n}\end{array}}{\Gamma_1^*, \ldots, \Gamma_n^*, !\Gamma \vdash *(M_1, \ldots, M_n) : 1}(1I)$$

Of course, the updating is conservative w.r.t. all logical properties. Further, the presence of the new terms implies the definition of new cuts. We write them as reduction rules on terms:

- $\mathbf{let}\ *()\ \mathbf{be} * \mathbf{in}\ M \to M$;
- $*(M_1, \ldots, M_{i-1}, !M_i, M_{i+1}, \ldots, M_n) \to$
  $$*(M_1, \ldots, M_{i-1}, !x_1^i, \ldots, !x_m^i, M_{i+1}, \ldots, M_n),$$
  where $\{!x_1^i, \ldots, !x_m^i\} = FV(!M_i)$.

Once extended $\approx_\eta$ by the $\eta$-reduction

$$\mathbf{let}\ M\ \mathbf{be} * \mathbf{in}\ N[*()/z] \to_\eta N[M/z]$$

and $\approx_c$ by

$$P[\mathbf{let}\ M\ \mathbf{be} * \mathbf{in}\ N/Z] \leftrightarrow \mathbf{let}\ M\ \mathbf{be} * \mathbf{in}\ P[N/Z]$$

we can effectively prove the completeness result we wish.

We do not study $\Lambda_!^1$ because we want to implement a philosophy which imposes to erase and duplicate terms exclusively using modal variables. Terms joined at 1 actually modify this perspective because of their computational behaviour. The presence of $*(M_1, \ldots, M_n)$ and $\mathbf{let}\ M\ \mathbf{be} * \mathbf{in}\ N$ allows a different method for deleting terms. Just looking at their reduction rules one can see that $*(M_1, \ldots, M_n)$ acts like a "forgetful environment": the structure of modal instances of $M_1, \ldots, M_n$ can be forgotten. Namely, $*(M_1, \ldots, M_n)$ seems to formalize the notions either of *heap* or of *garbage collector*. These observations seem to suggest a three-level variable taxonomy: *erasing*, *linear* and *duplicating* variables. Let notice that we have just reached this conclusion following a purely syntactical way. Jacobs [12] suggested a similar solution starting from semantical speculations. This "coincidence" is attractive: a three-sort variables language with heap-like construct and a more syntactically compact[3] copy-operation will constitute a topic for future work. A further language already presenting some similar features is in [21].

# 6 The untyped version of $\Lambda_!$

In this last section we will complete the analysis of the Curry-Howard isomorphism for ILL, by defining the untyped version of $\Lambda_!$. We consider it an essential step in our construction, since types are essential at the compile time, while, during the evaluation steps they can be dropped. Thus, an evaluator for $\Lambda_!$ must be defined on an untyped language. The existence of two different sorts of variables is essential for the construction of the untyped language, since they allow the definition of the reduction rules independently from types.
Let $\Lambda_!^-$ be the untyped version of $\Lambda_!$. Here we just give its syntactical definition, and list its principal properties.

**Definition 6.1** *i) Let $U$ and $V$ be sub-sets of $Var \cup !Var$.*
*Let define $U \cap_! V$ as $(U \cap V) \setminus !Var$.*

*ii) The following deduction system proves statements $M \in \Lambda^-_{!U}$, saying that $M$ is a term of $\Lambda_!^-$ with free variables in $U$:*

$$\frac{X \in Var \cup !Var}{X \in \Lambda^-_{!\{X\}}} \qquad \frac{M \in \Lambda^-_{!U}}{\lambda X.M \in \Lambda^-_{!U\setminus\{X\}}}$$

$$\frac{M \in \Lambda^-_{!U} \quad N \in \Lambda^-_{!V} \quad V \cap_! U = \emptyset}{\{MN, M \otimes N\} \subset \Lambda^-_{!U\cup V}} \qquad \frac{M \in \Lambda^-_{!U}}{\{M \oplus \_, \_ \oplus M, \mathbf{l}(M), \mathbf{r}(M), \mathbf{d}(M)\} \subset \Lambda^-_{!U}}$$

$$\frac{M \in \Lambda^-_{!U}}{\lambda X \otimes Y.M \in \Lambda^-_{!U\setminus\{X,Y\}}} \qquad \frac{M \in \Lambda^-_{!U} \quad N \in \Lambda^-_{!V} \quad U' = U \setminus \{X\} = V \setminus \{Y\}}{\lambda X \oplus Y.M|N \in \Lambda^-_{!U'}}$$

$$\frac{M \in \Lambda^-_{!U} \quad U \subseteq !Var}{!M \in \Lambda^-_{!U}} \qquad \frac{M \in \Lambda^-_{!U} \quad N \in \Lambda^-_{!U}}{M\&N \in \Lambda^-_{!U}}$$

[3]commuting-conversions-free

It is possible to design a type assignment system for terms in $\Lambda_!^-$, through an erasing function $E$ which erases the type information from terms of $\Lambda_!$.

**Definition 6.2** *i) The type assignment system* $\vdash^-$ *proves judgments with shape* $\Gamma \vdash^- M : A$*, where* $\Gamma$ *is a context,* $M \in \Lambda_!^-$ *and* $A \in \mathcal{F}_{ILL}$*. The rules of* $\vdash^-$ *come from those of* $\vdash$ *to which the here below defined function* $E$ *has been applied.*

*ii) Let* $E : \Lambda_! \to \Lambda_!^-$ *be defined, by induction on terms, in the following way:*

$$
\begin{array}{lcl}
E(X) & = & X \\
E(\lambda X : A.M) & = & \lambda X.E(M) \\
E(MN) & = & E(M)E(N) \\
E(\lambda X \otimes Y : A \otimes B.M) & = & \lambda X \otimes Y.E(M) \\
E(M \otimes N) & = & E(M) \otimes E(N) \\
E(\lambda X \oplus Y : A \oplus B.M|N) & = & \lambda X \oplus Y.E(M)|E(N) \\
E(M \oplus \_) & = & E(M) \oplus \_ \\
E(\_ \oplus M) & = & \_ \oplus E(M) \\
E(M\&N) & = & E(M)\&E(N) \\
E(\mathbf{l}(M)) & = & \mathbf{l}(E(M)) \\
E(\mathbf{r}(M)) & = & \mathbf{r}(E(M)) \\
E(!M) & = & !E(M) \\
E(\mathbf{d}(M)) & = & \mathbf{d}(E(M))
\end{array}
$$

*$E$ can be extended to judgments by setting* $E(\Gamma \vdash M : A) = \Gamma \vdash^- E(M) : A$*.*

The relationship between the typed system $\vdash$ and the type assignment $\vdash^-$ is given by the following:

**Property 6.1** $\Gamma \vdash M : A$ *if and only if* $\Gamma \vdash^- E(M) : A$

From the point of view of the Curry-Howard isomorphism, a term of $\Lambda_!^-$ represents a set of derivations. More precisely, $M \in \Lambda_!^-$ corresponds to the, possibly empty, set $\{\Pi | \Pi : \Gamma \vdash_{\mathcal{N}} A \text{ and } (\exists M' \in \Lambda_!.\ \ M = E(M') \text{ and } \Gamma \vdash M' : A)\}$.

In [17] the system $\vdash^-$ is further studied. In particular, $\vdash^-$ is proved to enjoy the principal type property: if a term can be typed, then it can be assigned the *principal type*, such that all, and only, the types derivable for the term can be obtained from it by replacing types for its type variables. The reduction rules of $\Lambda_!^-$ are obtained from those of $\Lambda_!$ through the erasing function. The system $\vdash^-$ inherits from $\vdash$ both the property of subject reduction and strong normalization.

Moreover, in [17], a translation $T$ from $\lambda$-calculus to $\Lambda_!^-$ is defined. $T$ is such that, when applied to terms typable in the Curry type assignment system, $T$ is optimal, in the following sense. Let us say that $x$ occurring in $M$ has a linear behaviour if, and only if, it occurs once in every $N$ such that $M \to_\beta^* N$. The corresponding variable in $T(M)$ is not modal. This assures that modalities in $T(M)$ are used exactly on those sub-terms which are erased/duplicated during $\beta$-reductions. Such a property makes $\Lambda_!^-$ a good candidate as a meta-language for efficient implementations.

:———————

# 7 Aknowledgments

Simona Ronchi della Rocca whish to thank Yves Lafont and Laurent Regnier for their disponibility in discussing the semantics of linear logic, during her stage in Marseille from October to December 1993. Both the authors thank Simone Martini and Furio Honsell for the discussions on the problem of natural deduction in presence of modalities. Further, the authors are also grateful both to Alberto Pravato and Torben Brauner for their careful reading of some preliminar version of the work, where they outlined some inaccuracies.

# A Strong normalization and Church-Rosser theorems

## A.1 Strong normalization theorem

We adapt a technique of Gallier [7] to our case. The Gallier's machinery allows to build a model of $\Lambda_!$ using the notion of *candidates of reducibility* [6, 7, 10].

Let $\Lambda_{\Gamma;A}$ be the set of terms in $\Lambda_!$ for which $\vdash$ deduces $A$ from $\Gamma$. We shall state that, if $M \in \Lambda_{\Gamma;A}$, then $M$ belongs to the class $\mathrm{SN}_{\Gamma;A}$ of the *strongly normalizable terms* [4] having type $A$ from $\Gamma$ in $\vdash$; namely, we shall get $\Lambda_{\Gamma;A} \subseteq \mathrm{SN}_{\Gamma;A}$. Since $\mathrm{SN}_{\Gamma;A}$ is trivially a subset of $\Lambda_{\Gamma;A}$, it results $\mathrm{SN}_{\Gamma;A} = \Lambda_{\Gamma;A}$.

The main implication $(M \in \Lambda_{\Gamma;A}) \Rightarrow (M \in \mathrm{SN}_{\Gamma;A})$ is proved splitting it into a pair of simpler implications.

For giving them, we need the definition of the predicate Comp whose usefulness becomes clear looking at lemma A.1 below:

**Definition A.1**
- $Comp(\Gamma; A; M) \Leftrightarrow M \in SN_{\Gamma;A}$ *and* $A \in \mathcal{A}_{ILL}$.
- $Comp(\Gamma^*, !\Gamma; A \multimap B; M) \Leftrightarrow M \in SN_{\Gamma^*,!\Gamma;A\multimap B}$ *and* $(\forall N. Comp(\Delta^*, !\Gamma; A; N) \Rightarrow Comp(\Gamma^*, \Delta^*, !\Gamma; B; MN))$.
- $Comp(\Gamma^*, !\Gamma; A \otimes B; M) \Leftrightarrow M \in SN_{\Gamma^*,!\Gamma;A\otimes B}$ *and if* $M \to^* P \otimes Q$*, then exist* $\Gamma^*_P$ *and* $\Gamma^*_Q$ *such that* $Comp(\Gamma^*_P, !\Gamma; A; P)$ *and* $Comp(\Gamma^*_Q, !\Gamma; B; Q)$.
- $Comp(\Gamma; A \oplus B; M) \Leftrightarrow M \in SN_{\Gamma;A\oplus B}$ *and*
  *either if* $M \to^* P \oplus \_$*, then* $Comp(\Gamma; A; P)$
  *or if* $M \to^* \_ \oplus Q$*, then* $Comp(\Gamma; B; Q)$.
- $Comp(\Gamma; !A; M) \Leftrightarrow M \in SN_{\Gamma;!A}$ *and* $Comp(\Gamma; A; \mathbf{d}(M))$.

Comp defines non empty sets of terms enjoying the closure properties (R1)-(R3):

**Lemma A.1** *Let the following sets be given:*
$\mathcal{I}_{terms}$ *contains all, and only, the instances of the terms* $\lambda X : A.M$, $\lambda X \otimes Y : A \otimes B.M$, $\lambda X \oplus Y : A \oplus B.M|N$, $M\&N$, $M \otimes N$, $!M$, $M \oplus \_$, $\_ \oplus M$, *.* $\mathcal{E}_{terms}$ *is* $\Lambda_! \setminus \mathcal{I}_{terms}$*, while* $\mathcal{S}tubborn$ *is the set* $\mathcal{E}_{terms} \cap (\{M | not(\exists N.M \to^+ N)\} \cup \{M | \forall N.M \to^+ N \Rightarrow N \notin \mathcal{I}_{terms}\})$.

[4] $M \in \Lambda_!$ strongly normalizable iff does not exist any infinite reduction $\to^*$ starting from $M$

*(R1) If Comp*$(\Gamma; A; M)$*, then* $M \in SN_{\Gamma;A}$.

*(R2) If Comp*$(\Gamma; A; M)$ *and* $M \rightarrow^* N$*, then Comp*$(\Gamma; A; N)$.

*(R3) If* $M \in (SN_{\Gamma;A} \cap \mathcal{E}_{terms})$ *and* $((M \rightarrow^* N \in \mathcal{I}_{terms}) \Rightarrow Comp(\Gamma; A; N))$,
*then Comp*$(\Gamma; A; M)$.

**Proof.** The three points can be proved by *simultaneous induction* on the types, exploiting that, for every $\mathrm{SN}_{\Gamma;A}$, we have:

(P1) $X \in \mathrm{SN}_{\Gamma;A}$ if, and only if, $\Gamma(X) \equiv A$.

(P2) If $M \in \mathrm{SN}_{\Gamma;A}$ and $M \rightarrow N$, then $N \in \mathrm{SN}_{\Gamma;A}$.

(P3)(1) (a) If $M \in \mathrm{SN}_{\Gamma^*,!\Gamma;A \multimap B} \cap \mathcal{E}_{terms}$ and $N \in \mathrm{SN}_{\Delta^*,!\Gamma;A}$ and $(M \rightarrow^+ \lambda X : A.Q \Rightarrow (\lambda X : A.Q)N \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;B})$, then $MN \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;B}$.

(b) If $M \in \mathrm{SN}_{\Gamma^*,!\Gamma;A\otimes B \multimap C} \cap \mathcal{E}_{terms}$ and $N \in \mathrm{SN}_{\Delta^*,!\Gamma;A\otimes B}$ and $(M \rightarrow^+ \lambda X \otimes Y : A \otimes B.Q \Rightarrow (\lambda X \otimes Y : A \otimes B.Q)N \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C})$, then $MN \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C}$.

(c) If $M \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;A\oplus B \multimap C} \cap \mathcal{E}_{terms}$ and $N \in \mathrm{SN}_{\Delta^*,!\Gamma;A\oplus B}$ and $(M \rightarrow^+ \lambda X \oplus Y : A \oplus B.P|Q \Rightarrow (\lambda X \oplus Y : A \oplus B.P|Q)N \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C})$, then $MN \in \mathrm{SN}_{\Gamma,\Delta;C}$.

(P3)(2) If $M \in \mathrm{SN}_{\Gamma;A\&B} \cap \mathcal{E}_{terms}$ and $(M \rightarrow^+ P\&Q \Rightarrow \mathbf{l}(P\&Q) \in \mathrm{SN}_{\Gamma;A}$ and $\mathbf{r}(P\&Q) \in \mathrm{SN}_{\Gamma;B})$, then $\mathbf{l}(M) \in \mathrm{SN}_{\Gamma;A}$ and $\mathbf{r}(M) \in \mathrm{SN}_{\Gamma;B}$.

(P3)(3) If $M \in \mathrm{SN}_{\Gamma;!A} \cap \mathcal{E}_{terms}$ and $(M \rightarrow^+ \mathbf{d}(!N) \Rightarrow \mathbf{d}(!N) \in \mathrm{SN}_{\Gamma;A})$, then $\mathbf{d}(M) \in \mathrm{SN}_{\Gamma;A}$.

□

**Corollary A.1** *(R4) If* $M \in (SN_{\Gamma;A} \cap \mathcal{S}tubborn)$*, then Comp*$(\Gamma; A; M)$.

**Proof.** By (R3). □

(R1) is the second one of the two implications we need. We still lack the first implication. It is $\Gamma \vdash M : A \Rightarrow \mathrm{Comp}(\Gamma; A; M)$ and is obtained as an instance of the following

**Lemma A.2** *If* $X_1 : A_1, \ldots, X_n : A_n \vdash M : A$ *and Comp*$(\Delta_1^*, !\Delta; A_1; N_1)$, $\ldots$,
*Comp*$(\Delta_n^*, !\Delta; A_n; N_n)$*, then Comp*$(\Delta_1^*, \ldots, \Delta_n^*, !\Delta; A; M[N_1/X_1 \ldots N_n/X_n])$*, where* $\forall 1 \leq i \neq j \leq n. Dom(\Delta_i^*) \cap Dom(\Delta_j^*) = \emptyset$.

**Proof.** By induction on the length of $X_1 : A_1, \ldots, X_n : A_n \vdash M : A$, using (R1)-(R3) and the following conditions, true for every set $\mathrm{SN}_{\Gamma;A}$:

(P4)(1) If $M \in \mathrm{SN}_{X:A,\Gamma;B}$, then $\lambda X : A.M \in \mathrm{SN}_{\Gamma;A \multimap B}$.

(P4)(2) If $M \in \mathrm{SN}_{X:A,Y:B,\Gamma;C}$, then $\lambda X \otimes Y : A \otimes B.M \in \mathrm{SN}_{\Gamma;A\otimes B \multimap C}$.

(P4)(3) If $M \in \mathrm{SN}_{X:A,\Gamma;C}$ and $N \in \mathrm{SN}_{Y:B,\Gamma;C}$, then $\lambda X \oplus Y : A \oplus B.M|N \in \mathrm{SN}_{\Gamma;A\oplus B \multimap C}$.

(P4)(4) If $M \in \mathrm{SN}_{\Gamma;A}$ and $N \in \mathrm{SN}_{\Gamma;B}$, then $M\&N \in \mathrm{SN}_{\Gamma;A\&B}$.

(P4)(5) If $M \in \mathrm{SN}_{!\Gamma;A}$, then $!M \in \mathrm{SN}_{!\Gamma;!A}$.

(P4)(6) If $M \in \mathrm{SN}_{\Gamma^*,!\Gamma;A}$ and $N \in \mathrm{SN}_{\Delta^*,!\Gamma;B}$, then $M \otimes N \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;A\otimes B}$.

(P4)(7) If $M \in \mathrm{SN}_{\Gamma;A}$, then $M \oplus \_ \in \mathrm{SN}_{\Gamma;A\oplus B}$ and $\_ \oplus M \in \mathrm{SN}_{\Gamma;B\oplus A}$.

(P5)(1) If $M \in \mathrm{SN}_{X:A,\Gamma^*,!\Gamma;B}$ and
$(\forall N.N \in \mathrm{SN}_{\Delta^*,!\Gamma;A}$ and $M[N/X] \in \Lambda_! \Rightarrow M[N/X] \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;B})$,
then $(\lambda X : A.M)N \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;B}$.

(P5)(2) If $M \in \mathrm{SN}_{X:A,Y:B,\Gamma^*,!\Gamma;C}$ and $N \in \mathrm{SN}_{\Delta^*,!\Gamma;A\otimes B}$ and
$(N \rightarrow^* P \otimes Q$ and $M[P/X\ Q/Y] \in \Lambda_! \Rightarrow M[P/X\ Q/Y] \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C})$, then
$(\lambda X \otimes Y : A \otimes B.M)N \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C}$.

(P5)(3) If $P \in \mathrm{SN}_{X:A,\Gamma^*,!\Gamma;C}$ and $Q \in \mathrm{SN}_{Y:B,\Gamma^*,!\Gamma;C}$ and $M \in \mathrm{SN}_{\Delta^*,!\Gamma;A\oplus B}$ and
$(M \rightarrow^* N \oplus \_$ and $P[N/X] \in \Lambda_! \Rightarrow P[N/X] \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C})$ and
$(M \rightarrow^* \_ \oplus N$ and $Q[N/Y] \in \Lambda_! \Rightarrow Q[N/Y] \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C})$,
then $(\lambda X \oplus Y : A \oplus B.P|Q)M \in \mathrm{SN}_{\Gamma^*,\Delta^*,!\Gamma;C}$.

(P5)(4) If $M \in \mathrm{SN}_{\Gamma;A}$, then $\mathbf{l}(M\&N) \in \mathrm{SN}_{\Gamma;A}$ and $\mathbf{r}(N\&M) \in \mathrm{SN}_{\Gamma;A}$.

(P5)(5) If $M \in \mathrm{SN}_{\Gamma;!A}$ then $\mathbf{d}(M) \in \mathrm{SN}_{\Gamma;A}$.

□

The main theorem:

**Theorem A.1** *If* $\Gamma \vdash M : A$*, then* $M \in SN_{\Gamma;A}$.

**Proof.** Assume to have $X_1 : A_1, \ldots, X_n : A_n \vdash M : A$.
(R4) implies that $\mathrm{Comp}(X_1 : A_1; A_1; X_1), \ldots, \mathrm{Comp}(X_n : A_n; A_n; X_n)$ hold.
By lemma A.2, $\mathrm{Comp}(X_1 : A_1, \ldots, X_n : A_n; A; M[X_1/X_1 \ldots X_n/X_n])$ holds.
By (R1), $M \in \mathrm{SN}_{X_1:A_1 \ldots X_n:A_n;A}$. □

## A.2 Church-Rosser theorem

Once proved the strong normalization, it is trivial to prove the Church-Rosser theorem, exploiting well known results on the rewriting systems. In [11], for example, we find the statement *A noetherian rewriting system is confluent if, and only if, it is locally confluent.* It fits our purposes substituting *noetherian* by *strong normalizable*, *confluent* by *Church-Rosser*, *locally confluent* by *deterministic*, and proving that our rewriting system is deterministic.

## References


[1] S. Abramsky. Computational interpretation of linear logic. Technical Report 90/92, Department of Computing, Imperial College, London, 1990.

[2] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. Technical Report 92/24, Department of Computing, Imperial College, London, September 1992.

[3] H.P. Barendregt. *The Lambda Calculus.* North-Holland, second edition, 1984.

[4] N. Benton, G. Bierman, V. de Paiva, and M. Hyland. Term assignment for intuitionistic linear logic. Technical Report 262, Computer Laboratory, University of Cambridge, August 1990.

[5] V. Breazu-Tannen, D. Kesner, and L. Puel. A typed pattern calculus. In *Proceedings of the 8th Symposium on Logic in Computer Science LICS'93 (Montreal)*, pages 262 – 274, June 1993.

[6] J. Gallier. *Logic and Computer Science*, chapter On Girard's "candidats de reductibilités", pages 123 – 203. P. Odifreddi editor, Academic Press, 1990.

[7] J. Gallier. On the correspondence between proofs and lambda terms. Obtained by ftp, January 1993.

[8] J.Y. Girard. *Interpretation Fonctionelle et Elimination des Coupures de l'Arithmetique d'Ordre Superieur.* PhD thesis, Université Paris VII, 1972.

[9] J.Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1 – 102, 1987.

[10] J.Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[11] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of A.C.M.*, 27:797 – 821, 1980.

[12] B. Jacobs. Semantics of weakening and contraction. In *Typed Lambda Calculi and Applications TLCA'92*, volume LNCS. Springer-Verlag, 1992.

[13] Y. Lafont. The linear abstract machine. *Theoretical Computer Science*, 59:157 – 180, 1988.

[14] P. Lincoln and J. Mitchell. Operational aspects of linear lambda calculus. In *Proceedings of Symposium on Logic in Computer Science LICS'92*, pages 235 – 246, June 1992.

[15] Simone Martini and Andrea Masini. On the fine structure of the exponential rule. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 197–210. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

[16] G. Mints. Normal deductions in the intuitionistic linear logic. To appear in Archive for Mathematical Logic.

[17] A. Pravato and L. Roversi. $\Lambda_!$ considered both as a paradigmatic language and as a meta-language. In *Theoretical Computer Science: Proceedings of the Fifth Italian Conference (Salerno)*, pages 146 – 161. World Scientific, November 1995.

[18] D. Prawitz. *Natural Deduction, a Proof Theoretic Study.* Almquist and Wiksell-Amsterdam, 1965.

[19] J.A. Reynolds. *Paris Colloquium on Programming*, chapter Towards a Theory of Type Structures, pages 408–425. Springer-Verlag, 1974.

[20] S. Ronchi della Rocca and L. Roversi. Lambda calculus and intuitionistic linear logic. Invited talk at the Logic Colloquium'94 (Clermont-Ferrand), July 1994.

[21] L. Roversi. Curry-howard isomorphism for intuitionistic linear logic: proof-theorist's and computer-scientist's perspectives. Technical Report 15/94, Università degli Studi di Pisa, 1994.

[22] A.S. Troelstra. *Lectures on Linear Logic.* CSLI, 1992.

[23] P. Wadler. A syntax for linear logic. Presented at the *Mathematical Foundations of Programming Semantics*, New Orleans, April 1993.