Embedding intersection types into multiplicative linear logic

Jean-Marie Madiot, LIPN

June 6, 2010

Abstract

Intersection types characterize properties on lambda-terms such as head, weak and strong normalization. One can establish relations between intuitionistic logic and intersection types with idempotence. Here we consider a type system without idempotence and we underline common traits with intuitionistic multiplicative linear logic. We analyse head normalization to try and get properties such as completeness, soundness and subject reduction or expansion.

1 Introduction

λ-calculus. λ-calculus is a functional notion of computability: one can simulate a Turing machine by rewriting over the set Λ of λ-terms which are defined by the a context-free grammar (1). The rewriting system is called β-reduction ($\to_\beta$) and is described in (2), where $t[u/x]$ is t in which u as replaced x. We extend this rule into a reduction relation satisfying $t \to t'$ then $tu \to t'u$, $ut \to ut'$ and $\lambda x.t \to \lambda x.t'$.

$$\Lambda \ni t ::= x \mid \lambda x.t \mid tt \tag{1}$$

$$(\lambda x.t)u \to_\beta t[u/x] \tag{2}$$

In the following we will also consider the head-reduction $\to_h$ (which only reduces the redexes in head position) which is a deterministic reduction (included in the usual β-reduction) defined as the smallest relation such that $(\lambda x.t)u \to_h t[u/x]$ and that if $t \to_h t'$ then $tu \to_h t'u$ (if t is not of the form $\lambda y.m$) and $\lambda x.t \to_h \lambda x.t'$.

We also consider α-equivalence by saying $\lambda x.u =_\alpha \lambda y.u[y/x]$ (if y is not free in u). This notion of equivalence comes naturally with the notion of computation which does not consider the actual name of the variable in substitution.

Normalization. Turing-completeness is an important property but having computations to terminate is also interesting. The λ-calculus has simple ways to guarantee computations termination − which in terms of rewriting systems is called normalization. Indeed the simply typed λ-calculus $\Lambda^{\to}$ guarantees that a λ-term simply typed is normalizable. However, not all normalizable terms are simply typable. On the contrary intersection type systems provide means to type every normalizable term and guarantee typed terms are normalizable. More precisely they can characterize several computational behaviours as head-, weak or strong normalization.

Idempotence. The operation of intersection $\wedge$ on types can have different properties such as commutativity ($A \wedge B = B \wedge A$), associativity ($(A \wedge B) \wedge C = A \wedge (B \wedge C)$) or idempotence ($A \wedge A = A$). This last equality is satisfied in most intersection type systems and it can be interesting to consider a non idempotent intersection.

In fact recent works show that non-idempotent intersection type systems can be used to catch quantitative properties of λ-terms, such as the number of steps needed by a suitable machine to compute the head-normal form of a term [Car07]. We can wonder if behind this type system one can establish relations with another logic. In the following we consider the relation between this kind of typing and multiplicative linear logic (MLL).

Underlying logics The system $D\Omega$ [CDC78, CDC80, Sal80] defines the intersection types which characterize (head-, strongly, and weakly) normalizable λ-terms [Kri93]. Intuitionistic logic (NJ) – in its conjunctive version – can be seen as a logic containing the logic underlying to $D\Omega$. Here we present a type system decorating intuitionistic multiplicative linear logic (IMLL) natural deduction which we will call $\mathcal{M}\Omega$. The underlying idea is that $\mathcal{M}\Omega$ is to $IMLL$ what $D\Omega$ is to NJ. In fact the system $D\Omega$ authorizes contraction and weakening respectively by handling contexts additively and with the axiom rule. Allowing contraction is a fundamental difference between the two systems.

2 Definitions

2.1 Normalization

Given a reduction relation $\rightarrow \in P(\Lambda^2)$ a term t is said :

- *in normal form* iff $t \not\rightarrow$ (*i.e.* $\forall u \in \Lambda\ t \not\rightarrow u$),
- *weakly normalizable* iff there exists a reduction sequence starting from t ending on a normal form,
- *strongly normalizable* iff there is no infinite reduction sequence starting from t

The head normal forms are the normal forms for $\rightarrow_h$ so they are the λ-terms of the form: $\lambda x_1 \cdots x_n.y\ u_1 \cdots u_n$. We define the following subsets of Λ: HN (respectively WN, SN) as the sets of all head-(respectively weakly, strongly) normalizable terms. We also note NF the set of normal forms and HNF the set of head normal forms.

2.2 Type system

A typing judgment is an element of $\mathcal{J} = \mathcal{C} \times (\Lambda \times T)$ where T is the set of all possible types, Λ the set of all λ-terms, V the set of variables and $\mathcal{C} = \mathcal{P}(V \times T)$ a set of ordered pairs (x, A) of a variable x and a type A which we call "context" from now. The notation $\Gamma \vdash_S t : A$ where Γ is a context, t a λ-term, A a type and S a set of typing judgments means that $(\Gamma, (t, A)) \in S$. The notation $x \# \Gamma$ means that x is not a variable typed in the context Γ : $\forall (y, A) \in \Gamma\ \ x \neq y$.

In the system $D\Omega$ the contexts are constrained to have each variable appears at most only once as in most type systems, they can be seen as partial functions from V to T with a finite domain. In the systems farther qualified of multiplicative ($\mathcal{M}\Omega, \lambda L\text{-}IMLL, \lambda L\text{-}IMLL^*, \mathcal{M}\Omega^*$) the contexts will be *multisets* ($\mathcal{C} = \mathcal{M}_{V \times T}$) and will authorize multiple appearances of the same variable, as the multiplicity of each variable in the context has a meaning in those type systems.

A type system S is described as a set of inference rules $\mathcal{R}$ of typing judgments of $\mathcal{P}(\mathcal{J}^n \times \mathcal{J})$ which is presented as a rule of arity n which usually are parametrized by terms, types and contexts. The type system then designates also the set of all typing judgments which can be deduced inductively by the inference rules. (*i.e.* the smallest set s such that for all n-ary rule r, $\forall((h_1, \ldots, h_n), c) \in r\ \ (\forall i\ h_i \in s) \Rightarrow c \in s$). Each typing judgment $\Gamma \vdash t : A$ is proven by at least one proof tree π of the inference rules which led to the typing judgment. We will note this $\pi :: \Gamma \vdash t : A$.

3 The system $D\Omega$

The terms of the system $D\Omega$ [CDC78, Sal80] are those of the standard λ-calculus. The types of the system $D\Omega$ are defined by the grammar 3 where α is a type variable. The rules of $D\Omega$ are described in figure 1.

$$T \ni A ::= \alpha \mid A \rightarrow A \mid A \wedge A \mid \Omega \tag{3}$$

In all rules the context Γ of the conclusion can appear in each premise even if there are several premises: the system is context-additive. The system handles contraction thanks to the $\wedge_I$ rule and the @ rule and weakening thanks to the axiom rule, the Ω rule and the $\wedge_{E_i}$ rules. For instance we can inhabit the type $A \rightarrow A \wedge A$ thanks to the proof in figure 2. This is of course a desired property as a term can have several times his initial types. The idempotence of the intersection is intimately related

$$\frac{}{\Gamma, x : A \vdash x : A}\ ax \qquad \frac{}{\Gamma \vdash t : \Omega}\ \Omega$$

$$\frac{\Gamma, x : A \vdash t : B \qquad x \# \Gamma}{\Gamma \vdash \lambda x.t : A \to B}\ \to_I \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash tu : B}\ \to_E$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash t : B}{\Gamma \vdash t : A \wedge B}\ \wedge_I \qquad \frac{\Gamma \vdash t : A_1 \wedge A_2}{\Gamma \vdash t : A_i}\ \wedge_{E_i}$$

Figure 1: Inference rules of $D\Omega$

to this type. As $\wedge$ is a syntactical binary operator we need to have a different notion of equality. We will talk about equality between two types A and B in this way : A is equal to B ($A = B$) if we can inhabit both $A \to B$ and $B \to A$ in the logic of the type system S.

$$\cfrac{\cfrac{x : A \vdash x : A \qquad x : A \vdash x : A}{x : A \vdash x : A \wedge A}\ \wedge_I}{\vdash \lambda x.x : A \to A \wedge A}\ \to_I$$

Figure 2: Idempotence in $D\Omega$

3.1 The system D

The types of the system D are those of the system $D\Omega$ but without Ω. The terms are also those of the lambda-calculus, and the rules are those of the system $D\Omega$ without the rule Ω.

4 The system R

The system R has been introduced in [Car07] – using multisets and pairs instead of $\otimes/1/\multimap$ – in a context speaking of relational semantics of linear logic, as an intersection type system. In this system we will denote the intersection $\otimes$ to underline its non idempotence as in linear logic this symbol designates an operation in which $A \otimes A \neq A$. Accordingly we will use $\multimap$ to denote implication. The neutral element for $\otimes$ is 1 and corresponds to the Ω type in $D\Omega$. We define the following type grammar:

$$A ::= \alpha \mid A_\otimes \multimap A$$

$$A_\otimes ::= 1 \mid A_\otimes \otimes A$$

We will note $(\dots(1 \otimes A_1)\dots) \otimes A_n$ by $A_1 \otimes \dots \otimes A_n$ or also by $\bigotimes_{i \leq n} A_i$. We can syntactically embed this grammar into the grammar of the types of $D\Omega$ by translating the derivation tree (and replacing $\otimes$ by $\wedge$, 1 by Ω). But here we insist on the fact that intersection (and thus 1) can only be in the left part of function types $\multimap$. This has a non trivial effect on properties the system will have.

The rules of the system $R = (ax, \lambda, (@_n)_{n \in \mathbb{N}})$ are described in figure 3. We remark this is an infinite set of rules as there is a rule $@_n$ for each $n \in \mathbb{N}$.

$$\frac{}{x : A \vdash x : A}\ ax \qquad \frac{\Gamma, x : A_1, \dots x : A_n \vdash t : B \qquad x \# \Gamma}{\Gamma \vdash \lambda x.t : A_1 \otimes \dots \otimes A_n \multimap B}\ \lambda$$

$$\frac{\Gamma \vdash t : A_1 \otimes \dots \otimes A_n \multimap B \qquad \Delta_i \vdash u : A_i \qquad \forall i \in [1, n]}{\Gamma, \Delta_1, \dots, \Delta_n \vdash tu : B}\ @_n\ (n \geq 0)$$

Figure 3: Inference rules of R

4.1 Non idempotence

The grammar does not authorize the previous notion of equality such as $A \otimes B = B \otimes A$ since we cannot have intersection types $\otimes$ in the right part of the function types $\multimap$. We then define a different notion of equality. We could say $A = B$ iff we can replace any occurrence of A by B (and B by A) in any formula provable in the system. Even if this property is difficult to work with, the system R has not an idempotent intersection anymore, as $\nvdash \lambda fx.fx : (A \multimap X) \multimap (A \otimes A \multimap X)$ and $\nvdash \lambda fx.fx : (A \otimes A \multimap X) \multimap (A \multimap X)$ (whereas $\nvdash \lambda fx.fx : (A \multimap X) \multimap (A \multimap X)$).

5 Normalizations in R

All standard normalization proofs for intersection types [CDC78, CDC80, Sal80, Kri93] use computability, realizability or induction[1] over ω^2 arguments. In the case of the system R, we can prove normalization using an explicit bound on the length of reductions and then a simple induction on ω.

5.1 Head-normalization in R

In this section we will develop the fact that every term typable in R is head-normalizable. In this section every typing judgment ($\vdash$) is relative to the system R ($\vdash_R$).

We define a function m of measure of the proof trees of R into the natural numbers, representing the number of uses of the application rule $@_n$ in a proof tree. More formally, if $r(\rho_1, \ldots, \rho_m)$ designates any proof consisting of a rule named r (which can be ax, λ, or $@_n$) applied to the proofs $\rho_1, \ldots, \rho_m$:

$$m(ax()) \quad := \quad 0 \tag{4}$$

$$m(\lambda(\rho)) \quad := \quad m(\rho) \tag{5}$$

$$m(@_n(\rho, \delta_1, \ldots, \delta_n)) \quad := \quad 1 + m(\rho) + \sum m(\delta_i) \tag{6}$$

In order to prove the desired property of subject reduction with a decreasing measure m to bound the length of the reduction sequences we describe the behaviour of the typing of the result of a redex by the substitution lemma, which allows us to derive a type for $u[v/x]$ given a typing of u and a certain number of typings of v.

Lemma 1 (Extended substitution lemma for R). *If π is a proof and $(\delta_i)_{i\in[1,n]}$ a family of proofs such that $\pi :: \Gamma, x : A_1 \ldots, x : A_n \vdash u : B$ ($x \# \Gamma$) and $\forall i \leq n$ $\delta_i :: \Delta_i \vdash v : A_i$ then we can build a proof $S(\pi, (\delta_i)_{i\in[1,n]}) :: \Gamma, \Delta_1, \ldots, \Delta_n \vdash u[v/x] : B$ verifying the following equation:*

$$m(S(\pi, \delta_i)) = m(\pi) + \sum_{i=1}^{n} m(\delta_i) \tag{7}$$

Proof. We proceed by induction on the structure of u:

Case 1 (variable). $u = y \neq x$ (so $u[v/x] = y$):
Here π is an axiom of the form $\Gamma', x : A_1 \ldots, x : A_n, y : B \vdash y : B$. As there is no weakening in the axiom rule we have $n = 0$ and $\Gamma' = \emptyset$. We now build the proof $S(\pi, (\delta_i)) = \pi$ which is the axiom proving $y : B \vdash y : B$. The equation (7) follows straightforwardly.

Case 2 (variable). $u = x$ (so $u[v/x] = v$) :
Here π is an axiom of the form $\Gamma', x : A_1 \ldots, x : A_n \vdash y : B$ so $n = 1$ and $\Gamma' = \emptyset$. The proof $\delta_1 :: \Delta_1 \vdash v : A_1$ is sufficient, so we take $S(\pi, (\delta_i)) = \delta_1$ and the equation (7) follows: $m(S(\pi, (\delta_j)_j)) = m(\delta_1) = 0 + \sum_{j=1}^{1} m(\delta_j) = m(\pi) + \sum_j m(\delta_j)$.

Case 3 (application). $u = tw$

$$\pi = \cfrac{\begin{matrix} \vdots\ \rho \\ P \vdash t : C_1 \otimes \cdots \otimes C_p \multimap B \end{matrix} \qquad \begin{matrix} \vdots\ \nu_j \\ \Upsilon_j \vdash w : C_j \end{matrix} \qquad j \in [1..p]}{P, \Upsilon_1 \ldots, \Upsilon_p \vdash tw : B}\ @_p$$

[1] not just on natural numbers, but on a measure whose image is in $\mathbb{N}^2$ for instance

Then we recursively apply the lemma on ρ, $\nu_1, \dots, \nu_p$ by carefully distributing the indices among $[1, n]$. Indeed $\Gamma, x : A_1, \dots, x : A_n = P, \Upsilon_1, \dots, \Upsilon_p$. We define a notation $i|\Psi$ designating the indexing $\{i \mid x : A_i \in \Psi\}$ where Ψ can be either P or any of the Υ_j.

By induction we get the following typing proofs:

$$S(\rho, (\delta_i)_{i|P}) \quad :: \quad P', (\Delta_i)_{i|P} \vdash t[v/x] : C_1 \otimes \dots \otimes C_p \multimap B \tag{8}$$

$$\forall j \in [1, p] \quad S(\nu_j, (\delta_i)_{i|\Upsilon_j}) \quad :: \quad \Upsilon'_j, (\Delta_i)_{i|\Upsilon_j} \vdash w[v/x] : C_j \tag{9}$$

(where P' and Υ'_j are P and Υ_j without any type assignment for x.)

The following equation holds ($\subseteq$: for all j, A_j is either in Γ or in one of the Δ_i; $\supseteq$: each $(\Delta_i)_{i|-}$ is already a subset of $(\Delta_i)_i$) :

$$(\Delta_i)_i = (\Delta_i)_{i|P} \cup \bigcup_{j \in [1,p]} (\Delta_i)_{i|\Upsilon_j}$$

so we can build with the rule $@_p$ a proof $S(\pi, (\delta_i)_i) :: P', \Upsilon'_1 \dots, \Upsilon'_p, \Delta_1, \dots, \Delta_n \vdash t[v/x]w[v/x] : B$.

The equation (7) holds:

$$\begin{aligned} m(S(\pi, (\delta_i)_i)) &= 1 + m(S(\rho, (\delta_i)_{i|P})) + \sum_{j=1}^{p} m(S(\nu_j, (\delta_i)_{i|\Upsilon_j})) \\ &= 1 + m(\rho) + \sum_{i|P} m(\delta_i) + \sum_{j=1}^{p} \left(m(\nu_j) + \sum_{i|\Upsilon_j} m(\delta_i) \right) \\ &= \left(1 + m(\rho) + \sum_{j=1}^{p} m(\nu_j) \right) + \sum_{i=1}^{n} m(\delta_i) \\ &= m(\pi) + \sum_{i=1}^{n} m(\delta_i) \end{aligned}$$

Case 4 (abstraction). $u = \lambda y.t$

$$\pi = \dfrac{\begin{matrix} \vdots\ \rho \\ \Gamma', x : A_1, \dots, x : A_n, y : C_1, \dots y : C_p \vdash t : B \end{matrix}}{\Gamma', x : A_1, \dots, x : A_n \vdash \lambda y.t : C_1 \otimes \dots \otimes C_n \multimap B}\ \lambda$$

By induction, we get $S(\rho, (\delta_i)) :: \Gamma', \Delta_1, \dots, \Delta_n, y : C_1, \dots y : C_p \vdash t : B$ and by applying the rule λ again we obtain a correct proof $S(\pi, (\delta_i)) = \lambda S(\rho, (\delta_i))$.

$$m(S(\pi, (\delta_i))) = m(S(\rho, (\delta_i))) = m(\rho) + \sum m(\delta_i) = m(\pi) + \sum m(\delta_i)$$

□

The subject reduction lemma is true but we first describe the behaviour in the head reduction which will help to prove a stronger property: the head reduction is finite.

Lemma 2 (Subject head reduction). *For all λ-terms t and t' such that $t \to_h t'$ and $\pi :: \Gamma \vdash_R t : A$, there exists a proof tree π' such that $\pi' :: \Gamma \vdash_R t' : A$ and $m(\pi) = m(\pi') - 1$.*

Proof. We prove the statement by induction on the structure of the λ-term t.

Case 1 (variable). $t = x$: $t \not\to_h$ (contradiction)

Case 2 (abstraction). $t = \lambda x.u$: then $t' = \lambda x.u'$ with $u \to_h u'$ (by definition of head-reduction) and we have a proof $\tilde{\pi} :: \Gamma' \vdash u : A'$ because the only way to type $\lambda x.u$ in $\mathcal{M}\Omega$ is the λ-rule, so $\tilde{\pi}$ is basically the remaining of π after removing this rule.

By induction we get a proof $\tilde{\pi}' :: \Gamma' \vdash u' : A'$ s.t. $m(\tilde{\pi}') = m(\tilde{\pi}) - 1$ and by applying the λ-rule to π we get π' and $m(\pi') = m(\tilde{\pi}') = m(\tilde{\pi}) - 1 = m(\pi) - 1$

Case 3 (application). $t = wv$

Subcase 3.1 (variable)**.** $w = x$ (t' cannot exist)

Subcase 3.2 (application)**.** $w = uv$
$t = uv$ where u is not an abstraction. Then $t' = u'v$ and we proceed the same way to get a proof $\tilde{\pi}$ typing u (which is always possible because in the system R to type an application it is necessary to type the left part) then by induction $\tilde{\pi}'$ typing u' and then π' typing t'.

$$m(\pi') = m(\tilde{\pi}') + 1 = m(\tilde{\pi}) - 1 + 1 = m(\pi) - 1$$

Subcase 3.3 (head-redex)**.** $w = \lambda x.u$
If $t = (\lambda x.u)v$ then $t' = u[v/x]$, this is the interesting case for which we will use the previously proved extended substitution lemma. π has the following form :

$$\pi = \cfrac{\cfrac{\cfrac{\vdots\, \rho}{\Gamma', x : \alpha_1, \ldots x : \alpha_n \vdash u : A}}{\Gamma' \vdash \lambda x.u : \alpha_1 \otimes \cdots \otimes \alpha_n \multimap A}\,\lambda \qquad \cfrac{\vdots\, \delta_i}{\Delta_i \vdash v : \alpha_i} \qquad i \in [1, n]}{\Gamma \vdash (\lambda x.u)v : A}\,@_n$$

$$m(\pi) = m(\rho) + \sum m(\delta_i) + 1$$

Thanks to the extended substitution lemma, $\pi' := S(\pi, \delta_i)$ is a proof of $\Gamma \vdash u[v/x] : A$ and the equation (7) gives the relation

$$m(\pi') = m(S(\pi, (\delta_i))) = m(\rho) + \sum m(\delta_i) = m(\pi) - 1$$

□

Lemma 3 (Subject reduction)**.** *For all λ-terms t and t' such that $t \to t'$ and $\pi :: \Gamma \vdash_R t : A$ we can prove $\pi' :: \Gamma \vdash_R t' : A$ with $m(\pi') \leq m(\pi)$ (m can be decreased not just by 1, but by any natural integer r, including 0)*

Proof. In the case of any β-reduction we proceed the same way except for the case study:

- $t = x$ (impossible)
- $t = uv$ and $t' = u'v$ (same as previous lemma, decreased by $r \geq 0$)
- $t = (\lambda x.u)v$ and $t' = u[v/x]$ (same as previous lemma, $r = 1$)
- $t = \lambda x.u$ (so $t' = \lambda x.u'$) (same as previous lemma, $r \geq 0$)
- $t = uv$ and $t' = uv'$: the last rule has to be $@_n$. We can apply the induction hypothesis on all proofs typing v to obtain proofs typing v' (even if $n = 0$) to build the proof tree typing uv'. (If $n = 0$ then $r = 0$ otherwise r is the sum of all of the r obtained by induction.)

□

Lemma 4 (finite head reduction in R)**.** *For all λ-term t, for all context Γ and for all type A: if $\Gamma \vdash_R t : A$ then the head reduction of t is finite.*

Proof. The result is straightforward. In fact we prove a stronger property : if $\pi :: \Gamma \vdash_R t : \tau$ then the length of all the sequences of head reductions starting from t is bounded by $m(\pi)$. □

Corollary 5 ($R \subseteq HN$)**.** *For all λ-term t, for all context Γ and for all type A: if $\Gamma \vdash_R t : A$ then t is head-normalizable.*

5.2 Strong normalization in R^*

We have a similar result for strong normalization corresponding to the system R^*.

Lemma 6 (Subject reduction in R^*)**.** *For all λ-terms t and t' such that $t \to t'$ and $\pi :: \Gamma \vdash_{R^*} t : A$, there exists a proof tree π' such that $\pi' :: \Gamma \vdash_{R^*} t' : A$ and $m(\pi) > m(\pi')$.*

Proof. The proof does look like the one of lemma 3 for the subject reduction in R except for the application case and for the statement of the substitution lemma.

Case 1 (variable)**.** $t = x$: there is no t'.

Case 2 (abstraction)**.** $t = \lambda x.u$:
$t' = \lambda x.u'$ where $u \to u'$. The proof of lemma 2 is still convenient as when reusing the proof subtree in the induction we do not add any rule $@_0$.

Case 3 (application)**.** $t = wv$

Subcase 3.1 (left)**.** $t = uv, t' = u'v$
The original proof holds – even if u is an abstraction – thanks to the fact that in R as well as in R^* typing uv needs typing u.

Subcase 3.2 (right)**.** $t = uv, t' = uv'$
We have $v \to v'$. Thanks to the absence of the rule $@_0$, when typing uv one has to type also v at least once, so the application of the rule $@_n$ $(n \geq 1)$ provides us with n proof trees δ_i of $\Delta_i \vdash_{R^*} v : A_i$, which can be recursively transformed into $\delta'_i :: \Delta_i \vdash_{R^*} v' : A_i$ where $\forall i\ m(\delta'_i) < m(\delta_i)$ which can be used to prove $\pi' :: \Gamma, \Delta_1, \ldots, \Delta_n \vdash_{R^*} uv' : B$ with the following property:

$$m(\pi') = m(\rho) + \sum_{i=1}^{n} m(\delta'_i) \leq m(\rho) + \sum_{i=1}^{n} (m(\delta_i) - 1) \leq m(\pi) - n < m(\pi)$$

Subcase 3.3 (redex)**.** $t = (\lambda x.u)v$, $t' = u[v/x]$
The exact original proof holds as the substitution lemma do not change any arity of rules $@_n$, so work in R^* as well as in R. However the substitution lemma is slightly different since we have to consider the weakening in the axiom rule; the equation is then an inequation:

$$m(S(\pi, \delta_i)) \leq m(\pi) + \sum_{i=1}^{n} m(\delta_i)$$

The proof of the substitution lemma is modified in the axiom case ($0 \leq$anything), in the abstraction case (simple induction) and in the application case ($\leq$ instead of $=$ between the first and the second line of equalities)

□

Theorem 7 ($R^* \subseteq SN$)**.** *For all λ-term t, for all context Γ and for all type A if $\Gamma \vdash_{R^*} t : A$ then t is strongly normalizable.*

Proof. Thanks to $\pi :: \Gamma \vdash_{R^*} t : A$ the previous lemma bounds the length of all reduction sequences starting from t by $m(\pi)$. Like in lemma 5 as its proof does not use the fact $\to_h$ is a strategy but any reduction making m decrease. □

5.3 Weak normalization in R

We will show that for any term typable in R with both type and context not containing 1 in respectively positive and negative position is weakly normalizable. We use the normal order *i.e.* the leftmost, outermost reduction $\to_l$ as a normalizing strategy[2]. We will first state a slightly different subject reduction lemma.

Positive $\in_+$ and negative $\in_-$ positions describe the number of times we go left on a $\multimap$ in a path in a type:

[2] the normal order is the smallest relation $\to_l$ such that $(\lambda x.t)u \to_l t[u/x]$ and that if $t \to_l t'$ then $tu \to_l t'u$, $xt \to_l xt'$ and $\lambda x.t \to_l \lambda x.t'$

- $A \in_+ A$
- $A \in_s B \otimes C$ if $(A \in_s B)$ or $(A \in_s C)$ where s can be whether $+$ of $-$
- $A \in_s B \multimap C$ if $(A \in_{-s} B)$ or $(A \in_s C)$
- $A \in_s (\Gamma, x : B)$ if $A \in_s B$
- $A \in_s (\Gamma \vdash B)$ if $A \in_s B$ or $A \in_{-s} \Gamma$

Lemma 8 (Subject normal reduction in $R\backslash 1+$)**.** *For all λ-terms t not in normal form and $\pi :: \Gamma \vdash_R t : A$ and $1 \notin_+ \Gamma \vdash A$, there exists a proof tree π' such that $\pi' :: \Gamma \vdash_R t' : A$ and $m(\pi) > m(\pi')$, where $t \to_l t'$.*

Proof. We proceed by induction on t.

Case 1 (Not in HNF)**.** If t is not in head normal form we apply lemma 6 to obtain $t \to_h t'$and $\pi' :: \Gamma \vdash t' : A$ with $m(\pi') < m(\pi)$. The head reduction is the first step of the leftmost reduction, so the only remaining case is when t is in head normal form (but not in normal form).

Case 2 (HNF, λ)**.** $t = \lambda x.u$
$t' = \lambda x.u'$ where $u \to_l u'$. A is of the form $B_1 \otimes \cdots \otimes B_n \multimap C$ so $1 \notin_+ C$ and $1 \notin_- B_i$. The proof tree contain also the typing judgment $\Gamma' = \Gamma, x : B_1, \ldots, x : B_n \vdash u : C$ we indeed have $1 \notin_- \Gamma'$ so we can use the induction property and the remaining of the proof is the same as in original lemma 2.

Case 3 (HNF, $\neg\lambda$)**.** $t = xu_1 \ldots u_m$ This is a significant difference with head-normalization. $t' = xu_1 \ldots u'_k \ldots u_m$ where $u_k \to_l u'_i$ (where k is the smallest j for which u_j is not in normal form). For all j, the proof tree contains typing of the subterms:

$$\Gamma_j \vdash xu_1 \ldots u_j : \bigotimes_{i \leq n_{j+1}} B_i^{j+1} \multimap \cdots \multimap \bigotimes_{i \leq n_m} B_i^m \multimap A$$

notably:

$$\Gamma_0 \vdash x : \bigotimes_{i \leq n_1} B_i^1 \multimap \cdots \multimap \bigotimes_{i \leq n_m} B_i^m \multimap A$$

As this last typing judgment can only be established thanks to the axiom rule, Γ_0 is exactly the right part of the typing judgment. Since $1 \notin_- \Gamma = \Gamma_m \supseteq \cdots \supseteq \Gamma_0$ we can deduce $1 \notin_- \Gamma_0$ so $\forall i, j\ A \notin_+ B_i^j$. Moreover as $1 \in_+ 1 = \bigotimes_{i \leq 0} C_i$ we can also deduce $\forall j\ n_j \geq 1$.

The proof derivation π for $\Gamma \vdash_R xu_1 \ldots u_k : A'$ where $A' = \bigotimes_{i \leq n_{k+1}} B_i^{k+1} \multimap \cdots \multimap \bigotimes_{i \leq n_m} B_i^m \multimap A$ has the form:

$$\pi = \cfrac{\cfrac{}{\Gamma_{k-1} \vdash u_1 \ldots u_{k-1} : B_1^k \otimes \cdots \otimes B_n^k \multimap A'}\ ax \qquad \cfrac{\vdots\ \delta_i^k}{\Delta_i^k \vdash v : B_i^k} \qquad i \in [1, n_k]}{\Gamma_k \vdash xu_1 \ldots u_k : A'}\ @_{n_k}$$

Since for all i we have $1 \notin_- \Delta_i^k$ (because $\Delta_i^k \subseteq \Gamma_k$), we can use the induction hypothesis to get for each $i \in [1, n]$ a proof $\delta_i'^k :: \Delta_i^k \vdash_R v' : B_i^k$.

We can apply on all δ_i^k the induction hypothesis and the equation in lemma 6 holds because m decrease for a positive number n_k of subproofs.

□

Theorem 9 ($R\backslash 1+ \subseteq WN$)**.** *For all λ-term t, for all context Γ and for all type A, if $\Gamma \vdash_R t : A$ and $1 \notin_+ \Gamma \vdash A$ then t is weakly normalizable (with strategy $\to_l$).*

Proof. Same as before using lemma 8. For all $\pi :: \Gamma \vdash_R t : A$ there is a $\to_l$ reduction sequence of length $\leq m(\pi)$ starting from t whose last term is in normal form. □

Both conditions $1 \notin_+ A$ and $1 \notin_- \Gamma$ are necessary R is not weakly normalizable if we only have the condition $1 \notin_+ A$. Indeed the λ-term $x(\delta\delta)$ is not weakly normalizing and neither is $\lambda x.x(\delta\delta)$ because the only redex in each term gives the term itself after reduction. However $x(\delta\delta)$ is typable without the condition $1 \notin_- \Gamma$ and $\lambda x.x(\delta\delta)$ is without $1 \notin_+ A$.

$R\backslash 1+$ is not strongly normalizable Because we can type a term without typing each subterm in R, we can build a term with a non normalizing part (such as $\delta\delta$). Such a term cannot strongly normalize. For exemple $\vdash_{R\backslash 1+} (\lambda x.\lambda y.y)(\delta\delta) : \alpha \multimap \alpha$.

5.4 Typing head normal forms

Lemma 10 ($HNF \subseteq R$)**.** *For all term t in head normal form, there exists a type A and a context Γ such as $\Gamma \vdash_R t : A$.*

Proof. Firstly, if t is in head normal form since it has the form ($n, m \geq 0$, y is a term variable and u_i are arbitrary terms) $t = \lambda x_1 \dots \lambda x_n.yu_1 \dots u_m$. When y is not a x_i, we can build a proof tree proving the typing judgment $y : 1 \multimap \cdots \multimap 1 \multimap A \vdash t : 1 \multimap \cdots \multimap 1 \multimap A$ where there is n occurrences of 1 in the type of y in the context and m occurrences of 1 in the type of t. (To do so, we apply m times the $@_0$ rule on the axiom rule typing y and then n times the λ rule).

If $y = x_i$ (for the greatest i possible) then the i^{th} rule λ (starting from the root of the proof) actually has x_i in the context so with the same rules we get the following typing judgment where the i^{th} 1 is replaced: $\vdash \lambda x_1 \dots \lambda x_n.yu_1 \dots u_m : 1 \multimap \cdots \multimap (1 \multimap \cdots \multimap 1 \multimap A) \multimap \cdots \multimap 1 \multimap A$. □

5.5 Typing normal forms

Lemma 11 ($NF \subseteq R^* \cap R\backslash 1+$)**.** *For all term t in normal form, there exists a type A and a context Γ such that $\Gamma \vdash_{R^*} t : A$ and $\Gamma \vdash_R t : A$ with no 1 in positive position in $\Gamma \vdash A$ (i.e. $1 \notin_+ A$ and $1 \notin_- \Gamma$).*

Proof. In fact use an even more restricted system with only three rules $R^1 = \{\lambda, ax, @_1\}$ where ax is the axiom rule without weakening. Each rule of R^1 is provable both in R and in R^*.

We proceed by induction on t. Because t is in normal form $t = \lambda x_1 \dots x_n.yu_1 \dots u_m$ where the u_i are in normal forms so we get $\delta_i :: \Delta_i \vdash u_i : B_i$ with $1 \notin_+ B_i$ and $1 \notin_- \Delta_i$. To the axiom rule $y : B_1 \multimap \cdots \multimap B_m \multimap \alpha$ we apply m times the $@_1$ rule to get the following typing judgment: $y : B_1 \multimap \cdots \multimap B_m \multimap \alpha, \Delta_1, \dots, \Delta_m = \Gamma \vdash yu_1 \dots u_m : \alpha$.

We remark that $1 \notin_- \Gamma$. We can now apply n times the λ rule with the variables $x_n, x_{n-1}, \dots, x_1$ and obtain: $\Gamma' \vdash \lambda x_1 \dots x_n.yu_1 \dots u_m : C_1 \multimap \cdots \multimap C_n \multimap \alpha$ where Γ' and all the C_i come from Γ so $1 \notin_- \Gamma'$ and $1 \notin_- C_i$ and eventually $1 \notin_+ C_1 \multimap \cdots \multimap C_n \multimap \alpha$. □

5.6 Typing normalizable terms

Lemma 12 (Expansion substitution lemma in R)**.** *If $\pi :: \Sigma \vdash t[u/x] : A$ then there exists $n \in \mathbb{N}$, some types $B_1, \dots, B_n$, some contexts $\Gamma, \Delta_1, \dots, \Delta_n$ such that:*
$\delta_i :: \Delta_i \vdash u : B_i$
$\gamma :: \Gamma, x : B_1, \dots, x : B_n \vdash t : A$
$\Sigma = \Gamma, \Delta_1, \dots, \Delta_n$

Proof. By induction on the structure of π.

Case 1. $t = x$ then $t[u/x] = u$ we take $n = 1, B_1 = A, \Gamma = \emptyset, \gamma = ax, \Delta_1 = \Sigma, \delta_1 = \pi$

Case 2. $t = y \neq x$ then $t[u/x] = y$ we take $n = 0, \Gamma = \Sigma, \gamma = \pi$

Case 3. $t = \lambda y.v$ then $t[u/x] = \lambda y.v[u/x]$ where $y \notin FV(u)$ without loss of generality.

$$\frac{\Sigma, y : C_1, \dots, y : C_m \vdash v[u/x] : D}{\Sigma \vdash \lambda y.v[u/x] : C_1 \otimes \cdots \otimes C_m \multimap D}\ \lambda$$

By induction $\Sigma' = \Sigma, y : C_1, \dots, y : C_m$ is splitted into $\Gamma', \Delta_1, \dots, \Delta_n$ and $\Delta_i \vdash u : B_i$ so if y is in one of the Δ_i then $y \in FV(u)$ which is impossible. So $\Gamma' = \Gamma, y : C_1, \dots, y : C_m$ and we can apply the λ rule to obtain the judgment $\Gamma, x : B_1, \dots, x : B_n \vdash \lambda y.v : C_1 \otimes \cdots \otimes C_m \multimap D$

Case 4. $t = vw$ then $t[u/x] = v[u/x]w[u/x]$. Applications on the induction hypothesis on the left premise typing $v[u/x]$ and each right premises typing $w[u/x]$ gives us respectively $n, (B_i), (\Gamma_i)$ and $m_j, (C_i)_j, (\Delta_i)_j$. We straightforwardly build $n + m, ((B_i), (C_i)_1, \dots, (C_i)_p)$ which corresponds to the typing of $vw = t$.

□

Lemma 13 (Subject expansion in R)**.** *If $t \to t'$ and $\pi' :: \Gamma \vdash_R t' : A$ then there exists $\pi :: \Gamma \vdash_R t : A$*

Proof. By induction on the structure of π

Case 1. $t = x$ (impossible)

Case 2. $t = \lambda y.u$ then $t' = \lambda y.u'$ with $u \to u'$.
$\pi = \lambda(\rho)$ with $\rho' :: \Gamma, x : B \vdash u' : C$ applying the induction hypothesis on ρ' we obtain $\rho :: \Gamma, x : B \vdash u : C$ from we conclude $\pi = \lambda(\rho)$.

Case 3. $t = uv$ and $t' = u'v$. Then $\pi' = @_n(\gamma', (\delta_i))$, we keep (δ_i) and we apply the IH on γ'

Case 4. $t = uv$ and $t' = uv'$. Then $\pi' = @_n(\gamma, (\delta'_i))$, we keep (γ) and we apply the IH on all the (δ'_i) (even if $n = 0$).

Case 5. $t = (\lambda x.u)v$ and $t' = u[v/x]$. We use the expansion substitution lemma to get from $\Gamma \vdash u[v/x] : A$ the types $B_1, \dots, B_n$ and the proofs γ and (δ_i) to build the following proof tree:

$$\dfrac{\dfrac{\dfrac{\vdots\ \gamma}{\Gamma, x : B_1, \dots, x : B_n \vdash u : A}}{\Gamma \vdash \lambda x.u : B_1 \otimes \cdots \otimes B_n \multimap A}\ \lambda \qquad \dfrac{\vdots\ \delta_i}{\Delta_i \vdash v : B_i}}{\Gamma, \Delta_1, \dots, \Delta_n \vdash (\lambda x.u)v : A}\ @_n$$

□

Note the subject expansion is true for any reduction $\to$ and not only the head-reduction $\to_h$.

Theorem 14 ($HN \subseteq R$)**.** *For all head-normalizable term t, there exists π, Γ, and A such that $\pi :: \Gamma \vdash_R t : A$.*

Proof. We can define head-normalizability with an inductive property: t is head-normalizable iff t is in head normal form or $t \to t'$ and t' is head-normalizable. Therefore lemma 10 typing head normal forms in R and lemma 13 typing predecessors in R are enough to complete the proof. □

Theorem 15 ($WN \subseteq R\backslash 1+$)**.** *For all weakly normalizable term t, there exists π, Γ, and A such that $\pi :: \Gamma \vdash_R t : A$ with 1 not in positive position in $\Gamma \vdash A$*

Proof. We can define weak normalizability with an inductive property: t is weakly normalizable iff t is in normal form or $t \to t'$ and t' is weakly normalizable. Therefore lemma 11 typing normal forms in $R\backslash 1+$ and lemma 13 typing predecessors in R (while preserving both A and Γ) are enough to complete the proof. □

Theorem 16 ($SN \subseteq R^*$)**.** *For all strongly normalizable term t, there exists π, Γ, and A such that $\pi :: \Gamma \vdash_{R^*} t : A$.*

Proof. The proof is quite similar to the previous completeness theorems (14, 15) but we need slightly different lemmas:

- Expansion substitution lemma:
  If $\pi :: \Sigma \vdash_{R^*} t[u/x] : A$ and $x \in FV(t)$ then there exists $n \in \mathbb{N}^*$, some types $B_1, \dots, B_n$, some contexts $\Gamma, \Delta_1, \dots, \Delta_n$ such that:
  $\delta_i :: \Delta_i \vdash_{R^*} u : B_i$
  $\gamma :: \Gamma, x : B_1, \dots, x : B_n \vdash_{R^*} t : A$
  $\Sigma = \Gamma, \Delta_1, \dots, \Delta_n$

- Typing predecessors:
  If $t \to t'$ and t strongly normalizable and $\pi' :: \Gamma' \vdash_R t' : A'$ then there exists $\pi :: \Gamma \vdash_R t : A$

$\square$

Finally we have proven the system R characterizes all the head-normalizable terms ($R = HN$) that R with no 1 in positive position in the final typing characterizes weakly normalizable terms ($R\backslash 1+ = WN$) and that R^* characterizes strongly normalizable terms ($SN = R^*$). Proofs of normalization were done by bounding the length of the reductions. We have also proven the subject reduction for R, $R\backslash 1+$ and R^* and the subject expansion for R, $R\backslash 1+$. (There is no such thing for R^* because $KI(\delta\delta) \notin WN \to_\beta I \in WN$).

6 NJ and $D\Omega$

The intuitionistic logic – in its conjunctive fragment (see right) – can be decorated by λ-terms to obtain the type system $D\Omega$ if we replace T by Ω (see figure 1). The only difference with the usual decoration is for the rules T, $\wedge_I$ and $\wedge_E$.

$$\frac{}{\Gamma, A \vdash A}\ ax \qquad \frac{}{\Gamma \vdash T}\ T$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to_I \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ \to_E$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge_I \qquad \frac{\Gamma \vdash A_1 \wedge A_2}{\Gamma \vdash A_i}\ \wedge_{E_i}$$

6.1 Aside from Curry-Howard

Decorating NJ with a conjunction seen as an intersection implies moving away from the Curry-Howard correspondence. In fact whereas in the Curry-Howard correspondence we associate to each rule a constructor (x, $\star$, λ, application, $\langle,\rangle$, π_1, π_2) in the intersection types decoration we lose constructor for both Ω and $\wedge$ (leaving only x, λ and application). Of course this way of using conjunction corresponds to the intersection of types as typing of type $A \wedge B$ means typing the same term with both types A and B.

6.2 Not quite NJ

As described in [Hin84] not all formulas provable in NJ are provable in $D\Omega$. We can give a more immediate counter example: $A \to B \to A \wedge B$ is provable in NJ but the immediate proof cannot be decorated in $D\Omega$. In fact, this formula is not provable in $D\Omega$ [BDC95].

7 $\mathcal{M}\Omega$

In order to establish a relation between intersection type systems without idempotence and multiplicative linear logic we develop a partial decoration of the intuitionistic multiplicative linear logic, the type system $\mathcal{M}\Omega$. It characterizes the head-normalizable terms but does not enjoy other interesting properties of a common type system as the subject reduction or the subject expansion[3].

[3] as seen farther we do not know whether the property of subject expansion is true

7.1 Natural deduction for IMLL

Using $IMLL$ – the intuitionistic version of MLL – let us be able to consider only operators $\otimes, 1, \multimap$. The intuitionistic role of the other connectives $\wp$ and $\perp$ is completely fulfilled by the left side of the sequents $(1\vdash, \otimes\vdash)$.
We present the natural deduction version of $IMLL$ [BBdPH93, Tro95] on the right. We remark the multiplicative concatenation of contexts in the case of several premises, and the absence of weakening in every rule.

$$\dfrac{}{A \vdash A}\ ax$$

$$\dfrac{}{\vdash 1}\ 1_I \qquad \dfrac{\Gamma \vdash 1 \qquad \Delta \vdash C}{\Gamma, \Delta \vdash C}\ 1_E$$

$$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \multimap_I \qquad \dfrac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ \multimap_E$$

$$\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \otimes_I \qquad \dfrac{\Gamma \vdash A \otimes B \qquad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C}\ \otimes_E$$

7.2 $\mathcal{M}\Omega$: a decoration of N-$IMLL$

The figure 4 gives the inference rules for a decoration of N-$IMLL$ in a comparable way to $D\Omega$ seen as a decoration NJ. We will note this decoration $\mathcal{M}\Omega$ as the concatenation of the contexts make them behave like multisets – which we usually note $\mathcal{M}$. We remark there is no structural rule (contraction, weakening) in $\mathcal{M}\Omega$. However the number of occurrences of a same variable in the contexts can be greater than one (zero, respectively) thanks to the tensor elimination rule$\otimes_E$ (the 1 elimination rule, respectively). The figure 5 shows typings of $\delta = \lambda x.xx$ and $K = \lambda xy.x$.

$$\dfrac{}{x : A \vdash x : A}\ ax$$

$$\dfrac{}{\vdash t : 1}\ 1_I \qquad \dfrac{\Gamma \vdash u : 1 \qquad \Delta \vdash t : C}{\Gamma, \Delta \vdash t : C}\ 1_E$$

$$\dfrac{\Gamma, x : A \vdash t : B \qquad x\#\Gamma}{\Gamma \vdash \lambda x.t : A \multimap B}\ \multimap_I \qquad \dfrac{\Gamma \vdash t : A \multimap B \qquad \Delta \vdash u : A}{\Gamma, \Delta \vdash tu : B}\ \multimap_E$$

$$\dfrac{\Gamma \vdash t : A \qquad \Delta \vdash t : B}{\Gamma, \Delta \vdash t : A \otimes B}\ \otimes_I \qquad \dfrac{\Gamma \vdash u : A \otimes B \qquad \Delta, x : A, y : B \vdash t : C \qquad x, y\#\Delta}{\Gamma, \Delta \vdash t[u/x, y] : C}\ \otimes_E$$

Figure 4: $\mathcal{M}\Omega$: decoration of N-$IMLL$

$$\dfrac{\dfrac{\dfrac{}{x : A \otimes (A \multimap B) \vdash x : \sim}\ ax \qquad \dfrac{}{y : A, z : A \multimap B \vdash zy}\ \multimap_E (ax, ax)}{x : A \otimes (A \multimap B) \vdash xx : B}\ \otimes_E}{\vdash \delta : A \otimes (A \multimap B) \multimap B}\ \multimap_I \qquad \dfrac{\dfrac{\dfrac{}{x : A \vdash x : A}\ ax}{x : A, y : 1 \vdash x : A}\ 1_E}{\vdash K : A \multimap 1 \multimap A}\ \multimap_I \times 2$$

Figure 5: Contraction and weakening in $\mathcal{M}\Omega$

7.2.1 A partial decoration

This decoration is not complete, as there is formulas provable in $IMLL$ which are not inhabited in the type assignment described above. For instance $A \multimap B \multimap A \otimes B$ is provable in $IMLL$ (see right).

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\ ax \qquad \dfrac{}{B \vdash B}\ ax}{A, B \vdash A \otimes B}\ \otimes_I}{A \vdash B \multimap A \otimes B}\ \multimap_I}{\vdash A \multimap B \multimap A \otimes B}\ \multimap_I$$

In the following attempt to decorate the previous proof tree we must satisfy some equation we annotate the proof tree with. We see that we end up in a impossible situation ($x = y \neq x$). More generally $A \multimap B \multimap A \otimes B$ is not a inhabited type in this type system.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : A \vdash u : A}\ ax(\Rightarrow u = x) \qquad \dfrac{}{y : B \vdash v : B}\ ax(\Rightarrow v = y)}{x : A, y : B \vdash t : A \otimes B}\ \otimes_I(\Rightarrow t = u = v)}{x : A \vdash \lambda y.v : B \multimap A \otimes B}\ \multimap_I (\Rightarrow x \neq y)}{\vdash \lambda x.\lambda y.v : A \multimap B \multimap A \otimes B}\ \multimap_I$$

7.3 Completeness of $\mathcal{M}\Omega$

The system $\mathcal{M}\Omega$ is complete as it can type with a non trivial type any head-normalizable term (*i.e.* $HN \subseteq \mathcal{M}\Omega$). The way we prove that here is that $\mathcal{M}\Omega$ is a supersystem of R ($R \subseteq \mathcal{M}\Omega$) by operating renaming in the proofs of R. The completeness of $\mathcal{M}\Omega$ is immediate as R is already complete ($HN \subseteq R$). The reason we use R is that $\mathcal{M}\Omega$ seems not to have the usual property used for proving completeness (*i.e.* subject expansion).

Lemma 17 (Customization lemma)**.** *If $\pi :: \Gamma, x : B_1, \ldots, x : B_n \vdash_R t : A$ then there exists π', t' and fresh variables $x_1, \ldots, x_n$ such that $\pi' :: \Gamma, x_1 : B_1, \ldots, x_n : B_n \vdash_R t' : A$ and $t = t'[x/x_1 \ldots x_n]$ with $size(\pi') = size(\pi)$.*

Proof. By induction on π:

Case 1. $\pi = ax$ (then $t = y$). It depends whether $y = x$ or not but the result is straightforward.

Case 2. $\pi = \lambda(\rho)$ (then $t = \lambda y.u$ and $y \neq x$ without loss of generality)
By induction we obtain free variables $x_1, \ldots, x_n$ and u' such that $u = u'[x/x_1 \ldots x_n]$. With $t' = \lambda y.u'$,

$$t = \lambda y.u = \lambda y.(u'[x/x_1 \ldots x_n]) = (\lambda y.u')[x/x_1 \ldots x_n] = t'[x/x_1 \ldots x_n]$$

Case 3. $\pi = @_m(\rho, (\delta))$ (then $t = uv$)
$\Gamma, x : B_1, \ldots, x : B_n = \Sigma, \Delta_1, \ldots \Delta_n$ so the B_is are splitted into $m+1$ parts into each premise : $c^i_1, \ldots, c^i_{k_i}$. for $i \in [0, m]$:

$$\rho :: \Sigma', x : B_{c^0_1}, \ldots, x : B_{c^0_{k_0}} \vdash u : D_1 \otimes \cdots \otimes D_m \multimap A$$

$$\delta_i :: \Delta'_i, x : B_{c^i_1}, \ldots, x : B_{c^i_{k_i}} \vdash u : D_i$$

With $\Sigma = \Sigma', x : B_{c^0_1}$ and $\Delta_i = \Delta'_i, x : B_{c^i_1}$. By concatenating the contexts $\Sigma' \cup \bigcup_i \Delta'_i$ we obtain Γ and concatenating all recursively obtained c^i_j gives $[1, n]$ so applying $@_m$ to the newly obtained ρ', $(\delta')_i$ sum up to the exact wanted context.

□

Lemma 18 ($R \subseteq \mathcal{M}\Omega$)**.** *If $\Gamma \vdash_R t : A$ then $\Gamma \vdash_{\mathcal{M}\Omega} t : A$*

Proof. We cannot directly prove the rules of R in the system $\mathcal{M}\Omega$ because $\mathcal{M}\Omega$ deals with intersection of only two types at a time, but lemma 17 solves this problem. We prove by induction on the size of π that if $\pi :: \Gamma \vdash_R t : A$ then $\Gamma \vdash_{\mathcal{M}\Omega} t : A$.

Case 1 (axiom)**.** $\pi = ax$ and the same rule is in $\mathcal{M}\Omega$

Case 2 (abstraction)**.** $t = \lambda x.u$. We have a proof tree of this form:

$$\dfrac{\begin{array}{c}\vdots\ \gamma \\ \Gamma, x : A_1, \ldots, x : A_n \vdash u : B \qquad x \# \Gamma\end{array}}{\Gamma \vdash \lambda x.u : A_1 \otimes \cdots \otimes A_n \multimap B}\ \lambda$$

Subcase 2.1 ($n \leq 2$)**.** We choose n fresh variables $x_1, \ldots, x_n$ and we apply lemma 17 on γ to obtain $\gamma_c :: \Gamma, x_1 : A_1, \ldots, x_n : A_n \vdash_R u' : B$ where $u'[x/x_1 \ldots x_n] = u$. We apply the induction hypothesis on γ_c ($size(\gamma_c) = size(\gamma) < size(\pi)$) we obtain $\gamma'_c :: \Gamma, x_1 : A_1, \ldots x_n : A_n \vdash_{\mathcal{M}\Omega} u' : B$ on which we successively apply, for $i \in [1, n-1]$, the rule $\otimes_E(ax, -)$:

$$P_i = (\Gamma, x : A_1 \otimes A_i, x_{i+1} : A_{i+1}, \ldots x_n : A_n \vdash_{\mathcal{M}\Omega} u'[x/x_1 \ldots x_i] : B)$$

$$\dfrac{\dfrac{}{x : (A_1 \otimes A_i) \otimes x_{i+1} \vdash x : (A_1 \otimes A_i) \otimes x_{i+1}}\ ax \qquad P_i \qquad x \# (\Gamma, x_{i+2} : A_{i+2}, \ldots, x_n : A_n)}{P_{i+1}}\ \otimes_E$$

because $u'[x/x_1 \ldots x_i][x/x, x_{i+1}] = u'[x/x_1 \ldots x_{i+1}]$. We had $\gamma'_c :: P_0$ and now we have $(\otimes_E(ax, -))^{n-1}(\gamma'_c) = \gamma' :: P_n = \Gamma, x : A_1 \otimes \cdots \otimes A_n \vdash u : B$.

Subcase 2.2 ($n = 0$)**.** By induction on $\gamma :: \Gamma \vdash_R u : B$ we obtain $\gamma' :: \Gamma \vdash_{\mathcal{M}\Omega} u : B$ on which we apply $1_E(ax, -)$ to obtain $1_E(ax, \gamma') :: \Gamma, x : 1 \vdash_{\mathcal{M}\Omega} u : B$.

Subcase 2.3 ($n = 1$)**.** By induction on $\gamma :: \Gamma, x : A_1 \vdash_R u : B$ we obtain $\gamma' :: \Gamma, x : A_1 \vdash_{\mathcal{M}\Omega} u : B$

At last we just have to apply the $\multimap_I$ rule on γ' to obtain $\Gamma \vdash_{\mathcal{M}\Omega} t : A_1 \otimes \cdots \otimes A_n \multimap B$.

Case 3 (application)**.** $t = uv$. We juste have to use the induction on all the $n + 1$ premises and then group all the proof $\Delta_i \vdash v : A_i$ under the tensor by applying $n - 1$ times the $\otimes_I$ rule (or once the 1_I rule if $n = 0$). We obtain $\Delta_1, \ldots, \Delta_n \vdash_{\mathcal{M}\Omega} v : A_1 \otimes \cdots \otimes A_n$ and then we apply $\multimap_E$ on uv.

□

Corollary 19 (Completeness of $\mathcal{M}\Omega$)**.** *If t is head-normalizable then t is typable in $\mathcal{M}\Omega$ with a non trivial type.*

Proof. $\Gamma \vdash_R t : A$ for some A, Γ by lemma 14 and by lemma 18 $\Gamma \vdash_{\mathcal{M}\Omega} t : A$. As A is a type of the grammar of R, it cannot be trivial. □

7.4 Subjection reduction in $\mathcal{M}\Omega$

We don't have the property of subject reduction in $\mathcal{M}\Omega$. Indeed, the term $t = x(Iz)(x(Iz))$ can be β-reduced in $t' = xz(x(Iz))$ (where $I = \lambda y.y$). We can see $x(Iz)$ appears twice in t so it let us typing once $x(Iz)$ with a $\otimes$-type to use it twice thanks to the proof tree described in figure 6.

$$\dfrac{\dfrac{\dfrac{}{x : D \vdash x : D}\,ax \qquad \dfrac{\multimap_E (\multimap_I (ax), ax)}{z : C \vdash Iz : C}}{z : C, x : D \vdash x(Iz) : A \otimes (A \multimap B)}\,\multimap_E \qquad \dfrac{\dfrac{}{b : A \multimap B \vdash b : A \multimap B}\,ax \qquad \dfrac{}{a : A \multimap B \vdash a : A}\,ax}{a : A, b : A \multimap B \vdash ba : B}\,\multimap_E}{z : C, x : D = C \multimap (A \otimes (A \multimap B)) \vdash x(Iz)(x(Iz)) : B}\,\otimes_E$$

Figure 6: A proof tree typing $x(Iz)(x(Iz))$

However one cannot use substitution from a smaller term to obtain $t' = xz(x(Iz))$. To type t' one has to have both x and z twice in the context (or with a $\otimes$-type) with a proof tree starting with $\otimes_E$ rules to split z then x into four distinct variables. (figure 7). However there is no proof of typing t' with the same context as in the proof typing t. However, we can type t with the same typing judgment as t' so this example is a counter example to the subject reduction property, but not to the subject expansion.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x_1 \vdash x_1}\,ax \qquad \dfrac{}{z_1 \vdash z_1}\,ax}{z_1 : C, x_1 : C \multimap A \multimap B \vdash x_1 z_1 : A \multimap B}\,\multimap_E \qquad \dfrac{\dfrac{}{x_2 \vdash x_2}\,ax \qquad \dfrac{\dfrac{\dfrac{}{y \vdash y}\,ax}{\vdash I : C \multimap C}\,\multimap_I \qquad \dfrac{}{z_2 \vdash z_2}\,ax}{z_2 : C \vdash I z_2 : C}\,\multimap_E}{z_2 : C, x_2 : C \multimap A \vdash x_2(I z_2) : A}\,\multimap_E}{z_1 : C, z_2 : C, x_1 : C \multimap A, x_2 : C \multimap A \multimap B \vdash x_1 z_1 (x_2(I z_2)) : B}\,\multimap_E}{z_1 : C, z_2 : C, x : (C \multimap A) \otimes (C \multimap A \multimap B) \vdash x z_1 (x(I z_2)) : B}\,\otimes_E(ax, -)}{z : C \otimes C, x : (C \multimap A) \otimes (C \multimap A \multimap B) \vdash xz(x(Iz)) : B}\,\otimes_E(ax, -)$$

Figure 7: A proof tree typing $xz(x(Iz))$

7.5 Soundness of $\mathcal{M}\Omega$

The system $\mathcal{M}\Omega$ is sound, that is to say it non trivially types all head-normalizable terms. In order to prove this property, we use realisability arguments by defining an interpretation for each type of $\mathcal{M}\Omega$. We use indeed the generic proof in [Kri93]. We will use the same adapted pair $(\mathcal{N}, \mathcal{N}_0)$ below.

Let $\mathcal{N} = HN$ be the set of a head-normalizable terms, and $\mathcal{N}_0$ the set of all terms in head normal form not starting with a λ: $\mathcal{N}_0 = \{y u_1 \ldots u_n \mid u_i \in \Lambda\}$.

We define inductively $[\![\ ]\!] : T \to \mathcal{P}(\Lambda)$ (where T is the set of types of $\mathcal{M}\Omega$ and $\mathcal{P}(\Lambda)$ the power set of Λ the set of λ-terms):

$$[\![1]\!] := \Lambda \tag{10}$$
$$[\![\alpha]\!] := \mathcal{N} \tag{11}$$
$$[\![A \multimap B]\!] := \{t \mid \forall a \in \Lambda\ (a \in [\![A]\!] \Rightarrow ta \in [\![B]\!])\} \tag{12}$$
$$[\![A \otimes B]\!] := [\![A]\!] \cap [\![B]\!] \tag{13}$$

Lemma 20. *For all $n \geq 0, y, u_1, \ldots, u_n, A$ we have $yu_1 \ldots u_n \in [\![A]\!]$*

Proof. By induction on A, the interesting case if when $A = B \multimap C$. Let $u_{n+1} \in [\![B]\!]$. By induction $yu_1 \ldots u_n u_{n+1} \in [\![C]\!]$ so $yu_1 \ldots u_n \in [\![B \multimap C]\!]$. □

Lemma 21. *If tx is reducible by head reduction to a head normal form then so is t.*

Proof. (It is true even if x is not a variable.) We can use the fact that we can type tx in the system R so we can type t also, and then we can use the stronger property proven in the proof of lemma 5 which bounds the length of head reduction sequences. □

Lemma 22. *For all t, A if $t \in [\![A]\!]$ and A is not trivial then t is head normalizable by head reduction.*

Proof. By induction on A.
If $A = A_1 \otimes A_2$ then there exists $i \in \{1, 2\}$ such as A_i is not trivial. $t \in [\![A_1 \otimes A_2]\!] = [\![A_1]\!] \cap [\![A_2]\!] \subseteq [\![A_i]\!]$: we use the induction hypothesis on A_i.
If $A = B \multimap C$ then C is also not trivial. Let x be a fresh variable. By lemma 20 we have $x \in [\![B]\!]$ so by definition of $[\![\ ]\!]$, $tx \in [\![C]\!]$. By induction tx is so head normalizable by head reduction, then so is t by lemma 21. □

Lemma 23 (Saturation)**.** *If $T' = t[u/x]u_1 \ldots u_n \in [\![A]\!]$ then $T = (\lambda x.t)uu_1 \ldots u_n \in [\![A]\!]$*

Proof. By induction on A (for all $t, n, (u_i)$)

- $A = 1$: T is in $[\![A]\!]$ anyway.
- $A = \alpha$: $T \to_h T'$ and $T' \in [\![\alpha]\!] = HN$ so T is also in $HN = [\![A]\!]$.
- $A = A_1 \otimes A_2$. By induction if $T' \in [\![A_i]\!]$ then $T \in [\![A_i]\!]$ so this is also true for $[\![A_1]\!] \cap [\![A_2]\!]$.
- $A = B \multimap C$: let u_{n+1} in $[\![B]\!]$. $T'u_{n+1} \in [\![C]\!]$ so by induction (with C and $n + 1$) we get $Tu_{n+1} \in [\![C]\!]$. So $T \in [\![C]\!]$

□

Lemma 24 (Adequacy)**.** *If $(x_1 : A_1^1, \ldots, x_1 : A_1^{p_1}), \ldots, (x_n : A_n^1, \ldots, x_n : A_n^{p_n}) \vdash_{\mathcal{M}\Omega} t : B$ and if $(\forall i, j\ u_i \in [\![A_i^j]\!])$ then $t[u_1/x_1, \ldots, u_n/x_n] \in [\![B]\!]$.*

Proof. By induction on π the proof tree typing t. The cases where the last rule of π is ax, 1_I, 1_E or $\otimes_I$ are straightforward.

Case 1 ($\pi = \otimes_E$)**.** $t = v[w/x_i, x_j]$ then $p_i = p_j = 1$ the induction hypothesis on v covers both the substitution $[w/x_i, x_j]$ (from the application of the rule) and the substitutions $[u_k/x_k]_{k \neq i,j}$ (from the statement of the lemma)

Case 2 ($\pi = \multimap_I$)**.** $t = \lambda x_{n+1}.v$ where x_{n+1} is fresh. $B = C \multimap D$.
Let $c \in [\![C]\!]$.
By induction we have : $v[u_1/x_1, \ldots u_n/x_n, c/x_{n+1}] \in [\![D]\!]$
This is exactly : $v[u_1/x_1, \ldots u_n/x_n][c/x_{n+1}] \in [\![D]\!]$
By the lemma 23 of saturation: $(\lambda x_{n+1}.v[u_1/x_1, \ldots u_n/x_n])c \in [\![D]\!]$
So $\lambda x_{n+1}.v[u_1/x_1, \ldots u_n/x_n] \in [\![C \multimap D]\!]$ which is exactly $(\lambda x_{n+1}.v)[u_1/x_1, \ldots u_n/x_n]) \in [\![C \multimap D]\!]$.

Case 3 ($\pi = \multimap_E$)**.** $t = vw$. $t[\vec{u}/\vec{x}] = v[\vec{u}/\vec{x}]w[\vec{u}/\vec{x}]$. $\Gamma_1 \vdash v : C \multimap B$ and $\Gamma_2 \vdash w : C$ for some C where Γ_1, Γ_2 are contexts included in the one in the conclusion, so $v[\vec{u}/\vec{x}] \in [\![C \multimap B]\!]$ and $w[\vec{u}/\vec{x}] \in [\![C]\!]$ so $v[\vec{u}/\vec{x}]w[\vec{u}/\vec{x}] \in [\![B]\!]$.

□

Corollary 25 (Soundness of $\mathcal{M}\Omega$)**.** *If t is typable with a non trivial type in $\mathcal{M}\Omega$ then it is head-normalizable.*

Proof. For some context Γ and some non trivial type A, $\Gamma \vdash_{\mathcal{M}\Omega} t : A$. By the adequacy lemma we know that if we replace each x_i (variable in the context) with itself, we obtain that $t[x_1/x_1 \ldots x_n/x_n] = t \in [\![A]\!]$ because by lemma 20 $x_i \in [\![A_i^j]\!]$. Then by lemma 22 t is head-normalizable. □

8 $\mathcal{M}\Omega^*$

8.1 Sequent calculus for $IMLL$

We don't have the desired property of subject reduction in $\mathcal{M}\Omega$. In order to understand the behaviour of such a system we will use the sequent calculus which is in general simpler to analyse. The figure on the right describes the inference rules for N-$IMLL$ – the sequent calculus of $IMLL$. Remark the common properties of non weakening and multiplicative contexts.

$$\frac{}{A \vdash A}\, ax \qquad \frac{\Gamma \vdash A \qquad \Delta, A \vdash C}{\Gamma, \Delta \vdash C}\, cut$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\, \multimap_R \qquad \frac{\Gamma \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}\, \multimap_L$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\, \otimes_R \qquad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}\, \otimes_L$$

$$\frac{}{\vdash 1}\, 1_R \qquad \frac{\Gamma \vdash C}{\Gamma, 1 \vdash C}\, 1_L$$

8.2 λL-$IMLL$: decoration of L-$IMLL$

The figure 8 describe the type system λL-$IMLL$ decorating L-$IMLL$ with λ-terms. This system is nothing more than the sequent calculus version of the decoration $\mathcal{M}\Omega$ of the N-$IMLL$ the natural deduction of $IMLL$: the decoration is canonical with respect to $\mathcal{M}\Omega$ – and also to usual decoration of sequent calculus for the simple types – in order to eventually describe the same type system.

$$\frac{}{x : A \vdash x : A}\, ax \qquad \frac{\Gamma \vdash t : A \qquad \Delta, x : A \vdash u : C \qquad x\#\Gamma}{\Gamma, \Delta \vdash u[t/x] : C}\, cut$$

$$\frac{\Gamma, x : A \vdash t : B \qquad x\#\Gamma}{\Gamma \vdash \lambda x.t : A \multimap B}\, \multimap_R \qquad \frac{\Gamma \vdash t : A \qquad \Delta, x : B \vdash u : C \qquad x\#\Delta}{\Gamma, \Delta, y : A \multimap B \vdash u[yt/x] : C}\, \multimap_L$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash t : B}{\Gamma, \Delta \vdash t : A \otimes B}\, \otimes_R \qquad \frac{\Gamma, x : A, y : B \vdash t : C \qquad x, y\#\Gamma}{\Gamma, z : A \otimes B \vdash t[z/x, y] : C}\, \otimes_L$$

$$\frac{}{\vdash t : 1}\, 1_R \qquad \frac{\Gamma \vdash u : C}{\Gamma, x : 1 \vdash u : C}\, 1_L$$

Figure 8: λL-$IMLL$: decoration of L-$IMLL$

8.3 $\mathcal{M}\Omega = \lambda L$-$IMLL$

Lemma 26 ($\mathcal{M}\Omega \subseteq \lambda L$-$IMLL$)**.** *If $\Gamma \vdash_{\mathcal{M}\Omega} t : A$ then $\Gamma \vdash_{\lambda L\text{-}IMLL} t : A$.*

Proof. The axiom rules in both system are the same and all introduction rules in $\mathcal{M}\Omega$ are the right rules in λL-$IMLL$ with the same decoration. The figure 9 describe proofs the elimination rules of $\mathcal{M}\Omega$ in λL-$IMLL$. The inclusion is then straightforward.

$$1_E := \frac{\Gamma \vdash u : 1 \qquad \dfrac{\Delta \vdash t : C}{\Delta, x : 1 \vdash t : C}\,1_L}{\Gamma, \Delta \vdash t : C}\,\mathit{cut}$$

$$\multimap_E := \frac{\Gamma \vdash t : A \multimap B \qquad \dfrac{\Delta \vdash u : A \qquad \dfrac{}{x : B \vdash x : B}\,\mathit{ax}}{\Delta, y : A \multimap B \vdash yu : B}\,\multimap_L}{\Gamma, \Delta \vdash tu : B}\,\mathit{cut}$$

$$\otimes_E := \frac{\Gamma \vdash u : A \otimes B \qquad \dfrac{\Delta, x : A, y : B \vdash t : C \qquad x, y \# \Delta}{\Delta, z : A \otimes B \vdash t[z/x, y] : C}\,\otimes_L}{\Gamma, \Delta \vdash t[u/x, y] : C}\,\mathit{cut}$$

Figure 9: Provability of $1_E, \multimap_E, \otimes_E$ in $\lambda L\text{-}IMLL$

□

Lemma 27 ($\lambda L\text{-}IMLL \subseteq \mathcal{M}\Omega$). *If* $\Gamma \vdash_{\lambda L\text{-}IMLL} t : A$ *then* $\Gamma \vdash_{\mathcal{M}\Omega} t : A$.

Proof. The axiom rules in both type systems are the same and all right rules in $\lambda L\text{-}IMLL$ are the introduction rules in $\mathcal{M}\Omega$ with the same decoration. The figure 10 shows proof tree of $\mathcal{M}\Omega$ proving the cut rule and the left rules of $\lambda L\text{-}IMLL$.

$$\mathit{cut} := \frac{\dfrac{\Gamma \vdash t : A \qquad \dfrac{}{\vdash t : 1}\,1_I}{\Gamma \vdash t : A \otimes 1}\,\otimes_I \qquad \dfrac{\dfrac{}{x : 1 \vdash x : 1}\,\text{ax} \qquad \Delta, x : A \vdash u : C}{\Delta, x : A, x : 1 \vdash u : C}\,1_E \qquad x \# \Gamma}{\Gamma, \Delta \vdash u[t/x] : C}\,\otimes_E$$

$$\multimap_L := \frac{\dfrac{\dfrac{\dfrac{}{y : A \multimap B \vdash y : A \multimap B}\,\mathit{ax} \qquad \Gamma \vdash t : A}{\Gamma, y : A \multimap B \vdash yt : B}\,\multimap_E}{\Gamma, y : A \multimap B \vdash yt : B \otimes 1}\,\otimes_I(-, 1_I) \qquad \dfrac{\Delta, x : B \vdash u : C}{\Delta, x : B, x : 1 \vdash u : C}\,1_E(\mathit{ax}, -) \qquad x \# \Delta}{\Gamma, \Delta, y : A \multimap B \vdash u[yt/x] : C}\,\otimes_E$$

$$\otimes_L := \frac{\dfrac{}{z : A \otimes B \vdash z : A \otimes B}\,\mathit{ax} \qquad \Gamma, x : A, y : B \vdash t : C \qquad x, y \# \Gamma}{\Gamma, z : A \otimes B \vdash t[z/x, y] : C}\,\otimes_E$$

$$1_L := \frac{\dfrac{}{x : 1 \vdash x : 1}\,\mathit{ax} \qquad \Gamma \vdash u : C}{\Gamma, x : 1 \vdash u : C}\,1_E$$

Figure 10: Provability of $\mathit{cut}, \multimap_L, \otimes_L, 1_L$ in $\lambda L\text{-}IMLL$

□

8.4 $\mathcal{M}\Omega^*$ as a $\lambda L\text{-}IMLL$ subsystem

The counter example to the subject reduction in $\mathcal{M}\Omega$ suggests the free usage of $\otimes$-types can affect important properties of the type system. We then define a subsystem $\mathcal{M}\Omega^*$ of $\lambda L\text{-}IMLL$ where the left introduction of the function type has an extra constraint: B cannot be a intersection ($\otimes$ or 1).

$$\frac{\Gamma \vdash t : A \multimap B \qquad \Delta, x : B \vdash u : C \qquad x \# \Delta \quad B \neq 1, \_ \otimes \_}{\Gamma, \Delta, y : A \multimap B \vdash u[yt/x] : C}\,\multimap_L$$

8.5 Subject reduction, expansion

Definition 28. *A sequent Γ is $\otimes$-free if no formula in Γ has the tensor $\otimes$ or the unit 1 as the top-level connective.*

Lemma 29 (Subject reduction)**.** *Let $\Gamma \vdash t : A$ and $t \to_\beta t'$, then $\Gamma \vdash t' : A$.*

Proof. We prove the following lemmas by induction on the proof of the typing judgment:

Lemma 30 (arrow)**.** *If $\Gamma \vdash \lambda x.t : A \multimap B$, then $\Gamma, x : A \vdash t : B$.*

Lemma 31 (tensor)**.** *If $\Gamma, x : A \otimes B \vdash t : C$, then $\Gamma, x : A, x : B \vdash t : C$.*

Lemma 32 (splitting)**.** *If $\Gamma \vdash t : A \otimes B$ and Γ are $\otimes$-free, then there exist Γ_1 and Γ_2 with $\Gamma = \Gamma_1, \Gamma_2$ such that $\Gamma_1 \vdash t : A$ and $\Gamma_2 \vdash t : B$.*

Lemma 33 (application)**.** *If $\Gamma \vdash tu : B$ and Γ, B are $\otimes$-free, then there is a type A such that $\Gamma_1 \vdash t : A \multimap B$ and $\Gamma_2 \vdash u : A$, with $\Gamma = \Gamma_1, \Gamma_2$.*

We consider only $\otimes$-free contexts and types Γ, A. By the following properties provable by induction on the derivations on the proofs. Then we proceed by induction on t. If t is an abstraction we apply the lemma 30 and we conclude by induction hypothesis. If $t = uv$ and $t' = u'v'$ with $u \to_\beta u'$, or $t' = uv'$ with $v \to_\beta v'$, we apply the lemma 33 and we conclude easily by induction hypothesis. Finally the case $t = (\lambda x.s)u$ and $t' = s[u/x]$ is obtained applying the lemma 33 to $(\lambda x.s)u$, then the lemma 30 to $\lambda x.s$ and finally we the typing of $s[u/x]$ by a cut rule. □

Lemma 34 (Subject expansion)**.** *Let $\Gamma \vdash t : A$ and $t' \to_\beta t$, then $t' : A$.*

Proof. We prove the following lemma by induction on t. The abstraction (resp. application) case is deduced by the abstraction (resp. application) lemma and the induction hypothesis. In the application case we in fact have A_1 and A_2 by induction hypothesis, and we define $A = A_1 \otimes A_2$.

Lemma 35 (substitution)**.** *Let $\Gamma \vdash t[u/x] : B$ and Γ, B $\otimes$-free, then there is A such that $\Gamma_1, x : A \vdash t : B$ and $\Gamma_2 \vdash u : A$ and $\Gamma = \Gamma_1, \Gamma_2$.*

Then by induction on t as lemma 29, using lemma 35 for the base case ($t = u[v/x]$ and $t' = (\lambda x.u)v$). □

8.6 Completeness, soundness

Lemma 36 (Typing HNF in $\mathcal{M}\Omega^*$)**.** *If t is in head normal form, there $\Gamma \vdash_{\mathcal{M}\Omega^*} t : A$ with for some context Γ and some non trivial type A*

Proof. We use the same typing than used in the proof of the lemma 10 as there is no $\otimes/1$ types in any right part of any type. □

Theorem 37 (Completeness of $\mathcal{M}\Omega^*$)**.** *If t is head normalizable, then $\Gamma \vdash_{\mathcal{M}\Omega^*} t : A$ with for some context Γ and some non trivial type A*

Proof. This directly follows the subject expansion lemma 34 and the typing of head normal forms (lemma 36) □

Theorem 38 (Soundness of $\mathcal{M}\Omega^*$)**.** *If $\Gamma \vdash_{\mathcal{M}\Omega^*} t : A$ for some non trivial type A then t is head-normalizable.*

Proof. $\mathcal{M}\Omega^* \subseteq \lambda L\text{-}IMLL$ (subsystem) and $\lambda L\text{-}IMLL \subseteq \mathcal{M}\Omega$ by lemma 27 then t is head-normalizable (lemma 25) □

9 Conclusion

9.1 Results

Considering non idempotent intersection in intersection type systems has led to manipulating a system (R) which has some desired properties for a type system : subject expansion, subject reduction and characterization of several classes of normalization (HN, WN, SN). Then we have related non idempotent intersection with the tensor of linear logic by building a type system which is to $IMLL$ what $D\Omega$ is to NJ.

If the type system $\mathcal{M}\Omega$ indeed characterizes head-normalizable terms it does not have the expected property of subject reduction. We remark that this is due to a phenomenon of *sharing*, already observed and discussed in [BT04, CDLRDR05] for $IMELL$. In fact, as a typing system, $IMELL$ does not enjoy subject reduction either. The solution proposed by Baillot and Terui [BT04], which is inspired by [Bar96], is very similar to what happens in the system R: the type grammar is restricted so that certain types (exponentials in the case of $IMELL$, tensors in the case of $IMLL$) cannot appear on the right side of an arrow. Our system $\mathcal{M}\Omega^*$ achieves the same effect by constraining the typing rules, instead of the types.

9.2 Future work

During the internship the following problems came up among others and we were unable to solve them all.

9.2.1 Soundness for $\mathcal{M}\Omega^*$

The property of soundness in $\mathcal{M}\Omega^*$ is proven thanks to realizability arguments through another type system. Maybe there is a way to bound the length of the reduction sequences as we did for the system R.

9.2.2 Subject expansion for $\mathcal{M}\Omega$

We do not know if the subject expansion is satisfied in the system $\mathcal{M}\Omega$. The fact the subject reduction is false in $\mathcal{M}\Omega$ may question the validity of subject expansion. However these properties are independent from each other.

9.2.3 Weaker subject reduction for $\mathcal{M}\Omega$

Subject reduction is false in $\mathcal{M}\Omega$, but there is a sharing phenomenon generated by the intersection and we could wonder if a weaker subject reduction is satisfied:

Suggestion 39 (Weak subject reduction in $\mathcal{M}\Omega$)**.** *If $\Gamma \vdash_{\mathcal{M}\Omega} t : A$ and if $t \to t'$ then there exists t'' such that $t' \to^* t''$ and $\Gamma \vdash_{\mathcal{M}\Omega} t'' : A$*

9.2.4 MLL's cuts for subject reduction

We have already the following result:

Proposition 40 (Cut-free normal form)**.** *$\pi :: \Gamma \vdash_{\mathcal{M}\Omega} t : A$ and A is not trivial and π is cut-free then t is in head normal form.*

In $\mathcal{M}\Omega$, we define a cut as $\multimap_E$ (resp. $\otimes_E$) of which the left subtree is a tree of right $\otimes_E$s and 1_Es eventually followed by $\multimap_I$ (resp. $\otimes_I$).

Maybe there is a way to avoid realizability arguments to prove the soundness of $\mathcal{M}\Omega$ by linking steps of reduction to cuts in $IMLL$ and using the cut elimination in MLL.

9.2.5 MELL, MLL and approximation theorem

Suggestion 41 (Approximation theorem)**.** *If* $\Gamma \vdash_{IMELL} A$ *where there are* m *(resp.* n*) occurrences of* $!$ *in* Γ *(resp.* A*), then, for all integers* $p_1, \ldots, p_n$*, there exist integers* $q_1, \ldots, q_m$ *such that* $\Gamma' \vdash_{IMLL} A'$*, where* Γ' *(resp.* A'*) is* Γ *(resp.* A*) obtained by replacing the* i^{th} *occurrence of* $!$ *with* $!_{q_i}$ *(resp.* $!_{p_i}$*). (where* $!_n A := A \otimes \cdots \otimes A$ *(*n *times)).*

From time to time this property had been intimately near the properties of the intersection type systems we considered. We can relate $D\Omega$ (and the associated of contraction and weakening) to $IMELL$ and $\mathcal{M}\Omega$ (with no such properties) to MLL and then relate the approximation theorem above to a possible embedding of type system $D\Omega$ in $\mathcal{M}\Omega$.

References

[Bar96] Andrew Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, Dapt. of Computer Science, University of Edinburgh, 1996.

[BBdPH93] P. N. Benton, Gavin M. Bierman, Valeria de Paiva, and Martin Hyland. A term calculus for intuitionistic linear logic. In *TLCA*, pages 75–90, 1993.

[BDC95] Franco Barbanera and Mariangiola Dezani-Ciancaglini. Intersection and union types: Syntax and semantics. *Information and Computation*, 119:202–230, 1995.

[BT04] Patrick Baillot and Kazushige Terui. Light types for polynomial time computation in lambda-calculus. In *Proceedings of LICS'04*, pages 266–275, 2004.

[Car07] Daniel De Carvalho. *Sémantiques de la logique linéaire et temps de calcul.* PhD thesis, Université Aix-Marseille II, France, 2007.

[CDC78] Mario Coppo and Mariangiola Dezani-Ciancaglini. A new type assignment for λ-terms. *Archive for Mathematical Logic*, 19(1):139–156, 1978.

[CDC80] Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame J. Formal Logic*, 21(4):685–693, 1980.

[CDLRDR05] Paolo Coppola, Ugo Dal Lago, and Simona Ronchi Della Rocca. Elementary affine logic and call-by-value lambda-calculus. In *Typed Lambda Calculi and Applications, 7th International Conference, Proceedings*, 2005. To appear.

[Hin84] J. Roger Hindley. Coppo-dezani types do not correspond to propositional logic. *Theor. Comput. Sci.*, 28:235–236, 1984.

[Kri93] Jean-Louis Krivine. *Lambda-calculus, types and models.* Ellis Horwood, Upper Saddle River, NJ, USA, 1993.

[Sal80] Patrick Sallé. Une généralisation de la théorie des types en λ-calcul. *RAIRO: informatique théorique*, 14(2):143–167, 1980.

[Tro95] A. S. Troelstra. Natural deduction for intuitionistic linear logic. *Ann. Pure Appl. Logic*, 73(1):79–108, 1995.