Call-by-Value Non-determinism in a Linear Logic Type Discipline

Alejandro Díaz-Caro[1,⋆], Giulio Manzonetto[1,2], and Michele Pagani[1,2]

[1] Université Paris 13, Sorbonne Paris Cité, LIPN, F-93430, Villetaneuse, France
[2] CNRS, UMR 7030, F-93430, Villetaneuse, France

Abstract. We consider the call-by-value λ-calculus extended with a may-convergent non-deterministic choice and a must-convergent parallel composition. Inspired by recent works on the relational semantics of linear logic and non-idempotent intersection types, we endow this calculus with a type system based on the so-called Girard's second translation of intuitionistic logic into linear logic. We prove that a term is typable if and only if it is converging, and that its typing tree carries enough information to give a bound on the length of its lazy call-by-value reduction. Moreover, when the typing tree is minimal, such a bound becomes the exact length of the reduction.

1 Introduction

The intersection type discipline provides logical characterisations of operational properties of λ-terms, namely of various notions of termination, like head-, weak- and strong-normalisation (see [1, 2], and [3] as a reference). The basic idea is to look at types as the set of terms having a given computational property — the type $\alpha \cap \beta$ being the set of those terms enjoying both properties α and β. With this intuition in mind, the intersection is naturally idempotent ($\alpha \cap \alpha = \alpha$).

Another way to understand the intersection type discipline is as a deductive system for presenting the compact elements of a specific reflexive Scott domain (see e.g. [4, §3.3]). The set of types assigned to a closed term captures the interpretation of such a term in the associated domain. Intersection types are then a powerful tool for enlightening the relations between denotational semantics, syntactical types and computational properties of programs.

Intersection types have been recently revisited in the setting of the relational semantics **Rel** of Linear Logic (LL). **Rel** is a semantics providing a more *quantitative* interpretation of the λ-calculus than Scott domains. Loosely speaking, the relational interpretation of a λ-term M not only tells us whether M converges on an argument, but in case it does, it also provides information on the number of times M needs to call[3] its argument to converge. Just like the intersection type discipline captures Scott domains, *non-idempotent* intersection type systems represent relational models. In this framework the type $\alpha_1 \cap \cdots \cap \alpha_k$ may

⋆ Partially supported by grants from DIGITEO and Région Île-de-France.

[3] The notion of *calling an argument* should be made precise by specifying an operational semantics, which is usually achieved through an evaluating machine.

be more accurately represented as the finite multiset $[\alpha_1, \ldots, \alpha_k]$. The lack of idempotency is the key ingredient to model the resource sensitiveness of **Rel** — while in the usual systems $M : \alpha \cap \beta$ stands for "M can be used either as data of type α or as data of type β", when the intersection is not idempotent the meaning of $M : [\alpha, \beta]$ becomes "M will be called *once* as data of type α and *once* as data of type β". Hence, types should no longer be understood as *sets of terms*, but rather as *sets of calls* to terms.

The first intersection type system based on **Rel** has been presented in [5], where de Carvalho introduced system R, a type discipline capturing the relational version of Engeler's model. More precisely, he proved that system R, beyond characterising converging terms, carries information on the evaluation sequence as well — the size of a derivation tree typing a term is a bound on the number of steps needed to reach a normal form. Similar results are obtained in [6] for a variant of system R characterising strong normalisation and giving a bound to the longest β-reduction sequence. More recently, Ehrhard introduced a non-idempotent intersection type system characterising the convergence in the call-by-value λ-calculus [7]. Also in this case, the size of a derivation tree bounds the length of the lazy (i.e. no evaluation under λ's) call-by-value β-reduction sequence. Our goal is to extend Ehrhard's system with non-determinism.

Our starting point is [8], where it is shown that the relational model $\mathcal{D}$ of the call-by-name λ-calculus provides a natural interpretation of both may and must non-determinism. Since **Rel** interprets λ-terms as relations, the *may*-convergent non-deterministic choice can be expressed in the model as the set-theoretical union. The *must*-convergent parallel composition, instead, is interpreted by using the operation $\mathcal{D} \otimes \mathcal{D} \multimap \mathcal{D}$ obtained by combining the mix rule $\mathcal{D} \otimes \mathcal{D} \multimap \mathcal{D} \invamp \mathcal{D}$ with the contraction rule $\mathcal{D} \invamp \mathcal{D} \multimap \mathcal{D}$, this latter holding since the call-by-name model $\mathcal{D}$ has shape $?A$ for $A = \mathcal{D}^{\mathbb{N}} \multimap \perp$. We will show that the same principle (*may*-convergence as *union* of interpretations and *must*-convergence as *mix* rule plus contraction) still works in the call-by-value setting.

Ehrhard's call-by-value type system is based on the so-called "second Girard's translation" of intuitionistic logic into LL [9, 10]. The translation of a type α is actually given by two mutually defined mappings ($\alpha \mapsto \alpha^v$ and $\alpha \mapsto \alpha^c$) reflecting the two sorts (*values* and *computations*) at the basis of the call-by-value λ-calculus:

$$\iota^v = \iota, \qquad (\alpha \to \beta)^v = \alpha^c \multimap \beta^c, \qquad \alpha^c = !\alpha^v,$$

where ι is an atom. Hence, the relational model described by Ehrhard's typing system yields a solution to the equation $\mathcal{V} \simeq !\mathcal{V} \multimap !\mathcal{V}$ in **Rel**. Since in this semantics $\multimap$ is interpreted by the cartesian product and ! by finite multisets, a functional type for a value in this system is a pair (p, q) of types for computations, and a type for a computation is a multiset $[\alpha_1, \ldots, \alpha_n]$ of value types (representing n calls to a single value that must behave as $\alpha_1, \ldots, \alpha_n$).

In order to deal with the must non-determinism, namely the parallel composition, we must add to the translation considered by Ehrhard a further exponential level, called here the *parallel sort*:

$$\iota^v = \iota, \qquad (\alpha \to \beta)^v = \alpha^c \multimap \beta^{\|}, \qquad \alpha^c = !\alpha^v, \qquad \alpha^{\|} = ?\alpha^c. \qquad (1)$$

This translation enjoys the nice property of mapping the call-by-value λ-calculus into the polarised fragment of LL, as described by Laurent in [11]. Then, our typing system is describing an object in **Rel** satisfying the equation $\mathcal{V} \simeq !\mathcal{V} \multimap ?!\mathcal{V}$, where the ? connective is interpreted by the finite multiset operator. In this setting a value type is a pair $(p, [q_1, \dots, q_n])$ of a computational type p and a parallel type, that is a multiset of computations $q_1, \dots, q_n$. Intuitively, a value of that type needs a computation of type p to create a parallel composition of n computations of types $q_1, \dots, q_n$, respectively. Notice that, following [8], the composition of the mix rule and the contraction one yields an operation $?!\mathcal{V} \otimes ?!\mathcal{V} \multimap ?!\mathcal{V}$ which is used to interpret the parallel composition.

To avoid a clumsy notation with multisets of multisets, we prefer to denote a !-multiset $[\alpha_1, \dots, \alpha_m]$ (the type of a computation) with the linear logic multiplicative conjunction $\alpha_1 \otimes \cdots \otimes \alpha_m$, a ?-multiset $[q_1, \dots, q_n]$ (the type of a parallel composition of computations) with the multiplicative disjunction $q_1 \parr \cdots \parr q_n$, and finally a pair $(p, [q_1, \dots, q_n])$ with the linear implication $p \multimap (q_1 \parr \cdots \parr q_n)$. Such a notation stresses the fact that the non-idempotent intersection type systems issued from **Rel** are essentially contained in the multiplicative fragment of LL (modulo the associativity, commutativity and neutrality equivalences).

Contents. Several non-deterministic extensions of the λ-calculus have been proposed in the literature, both in the call-by-name (e.g. [8, 12]) and in the call-by-value setting (e.g. [13, 14]). In the present paper we focus on the call-by-value λ-calculus, first introduced in [15], endowed with two binary operators $+$ and $\|$ representing non-deterministic choice and parallel composition, respectively. The resulting calculus, denoted here $\Lambda_{+\|}$, is quite standard and its operational semantics is given in Section 2 through a machine performing lazy call-by-value reduction. Following [8], we model non-deterministic choice as *may* non-determinism and parallel composition as *must.* This is reflected in our reduction and in our notion of convergence. Indeed, every time the machine encounters $M + N$ in active position it actually performs a choice, while encountering $M \parallel N$ it interleaves reductions in M and in N; finally a term M converges when there is a reduction of the machine from M to a normal form.

Section 3 is devoted to provide the type discipline for $\Lambda_{+\|}$, based on the multiplicative fragment of LL (as discussed above), and to define a measure $|\cdot|$ associating a number with every type derivation. Such a measure "extracts" from the information present in the typing tree of a term, a bound on the length of its evaluation. In Section 4 we show that our type system satisfies good properties like subject reduction and expansion. We also prove that the measure associated with the typing tree of a term decreases by 1 at every reduction step, giving thus a proof of weak normalisation in ω for typable terms. From these properties it ensues directly that a term is typable if and only if it converges. Moreover, thanks to the resource consciousness of our type system, we are able to strengthen such a result — we prove that, whenever M converges, there is a type derivation $\vdash M : \alpha$

β_v-reduction	*$+$-reductions*	*$\parallel$-reductions*
$(\lambda x.M)V \to M[V/x]$	$M + N \to M$	$(M \parallel N)P \to MP \parallel NP$
	$M + N \to N$	$V(M \parallel N) \to VM \parallel VN$

Contextual rules

$$\frac{M \to M'}{M \parallel N \to M' \parallel N} \qquad \frac{N \to N'}{M \parallel N \to M \parallel N'} \qquad \frac{M \to M' \quad (*)}{MN \to M'N} \qquad \frac{M \to M' \quad (*)}{VM \to VM'}$$

Fig. 1: Reduction semantics for $\Lambda_{+\parallel}$. The condition $(*)$ stands for "$M \neq P \parallel Q$".

(with α satisfying a suitable minimality condition) such that the associated measure provides the exact number of steps reducing M to a normal form.

Finally, in Section 5 we discuss the properties of the model in **Rel** underlying our system. As expected, the interpretation turns out to be adequate, i.e. a term converges if and only if its interpretation is non-empty. On the other hand such a model is not fully abstract — there are terms having different interpretations and that cannot be (semi-)separated using applicative contexts. Our counterexample does not rely on the presence of $+$ and $\parallel$.

2 The call-by-value non-deterministic machine

We consider the call-by-value λ-calculus [15], extended with non-deterministic and parallel operators in the spirit of [8]. The set $\Lambda_{+\parallel}$ of *terms* and the set $\mathrm{V}_{+\parallel}$ of *values* are defined by mutual induction as follows (where x ranges over a countable set Var of variables):

Terms:	$M, N, P, Q ::=$	$V \mid MN \mid M + N \mid M \parallel N$	$\Lambda_{+\parallel}$
Values:	$V ::=$	$x \mid \lambda x.M$	$\mathrm{V}_{+\parallel}$

Intuitively, $M+N$ denotes the *non-deterministic choice* between M and N, while $M \parallel N$ stands for their parallel composition. Such operators are not required to be associative nor commutative. As usual, we suppose that application associates to the left and λ-abstraction to the right. Moreover, to lighten the notation, we assume that application and λ-abstraction take precedence over $+$ and $\parallel$.

The *α-conversion* and the set $\mathrm{FV}(M)$ of *free variables of* M are defined as usual in λ-calculus [16, §2.1]. A term M is *closed* whenever $\mathrm{FV}(M) = \emptyset$.

Given $M \in \Lambda_{+\parallel}$ and $V \in \mathrm{V}_{+\parallel}$, we denote by $M[V/x]$ the term obtained by simultaneously substituting the value V for all free occurrences of x in M, subject to the usual proviso about renaming bound variables in M to avoid capture of free variables in V. Hereafter terms are considered up to α-conversion.

Definition 1 (Operational semantics). *The operational semantics of $\Lambda_{+\parallel}$ is given in Figure 1. We denote by $\to^*$ the transitive and reflexive closure of $\to$.*

The side condition $(*)$ on the context rules for the application avoids critical pairs with the $\parallel$-rules: this is not actually needed but it simplifies some proofs.

A term M is called a *normal form* if there is no $N \in \Lambda_{+\|}$ such that $M \to N$. In particular, all (parallel compositions of) values are normal forms. Note that when M is closed then either it is a parallel composition of values or it reduces.

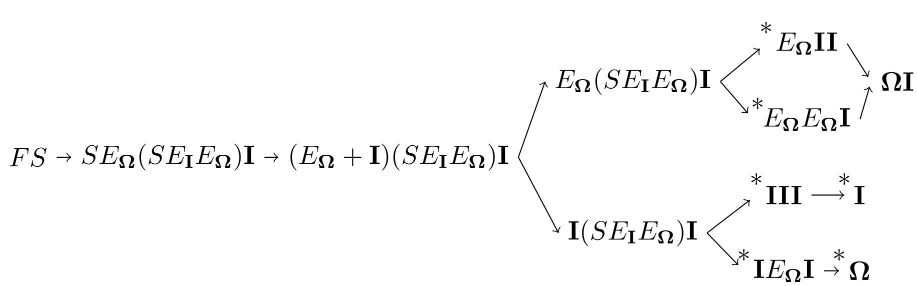
Definition 2. *A closed term* $M \in \Lambda_{+\|}$ converges *if and only if there exists a reduction* $M \to^* V_1 \| \cdots \| V_n$ *for some* $V_i \in \mathrm{V}_{+\|}$.

The intuitive idea underlying the above notion of convergence is the following:

- The non-deterministic choice $M + N$ is treated as *may*-convergent, either of the alternatives may be chosen during the reduction and the sum converges if either M or N does.
- The parallel composition $M \| N$ is modelled as *must*-convergent, the reduction forks and the parallel composition converges if both M and N do.

Let us provide some examples. We set $\mathbf{I} = \lambda x.x$, $\mathbf{\Delta} = \lambda x.xx$ and we denote by $\mathbf{\Omega}$ the paradigmatic non-converging term $\mathbf{\Delta\Delta}$, which reduces to itself as $\mathbf{\Delta}$ is a value. The reduction is *lazy*, i.e. it does not reduce under abstractions, so for example $\lambda y.\mathbf{\Omega}$ is a normal form. In fact, when considering closed terms, the parallel compositions of values are exactly the normal forms, thus justifying Definition 2. We would like to stress that our system is designed in such a way that a parallel composition of values is not a value. As a consequence, the term $P = \lambda k.\mathbf{\Delta} \| \mathbf{\Delta}$ is not a value, so the term $(\lambda x.x\mathbf{I}x)P$ is converging. Indeed, it reduces to $(\lambda x.x\mathbf{I}x)(\lambda k.\mathbf{\Delta}) \| (\lambda x.x\mathbf{I}x)\mathbf{\Delta} \to^* \mathbf{\Delta} \| \mathbf{\Delta}$. Notice that, if we consider P as a value, then $(\lambda x.x\mathbf{I}x)P$ would diverge since it would reduce to $P\mathbf{I}P \to^* (\mathbf{\Delta} \| \mathbf{I})P \to^* \mathbf{\Delta}P \| P$ and one can check easily that $\mathbf{\Delta}P$ diverges.

The presence of the non-deterministic choice $+$ enlightens a typical feature of the call-by-value λ-calculus: application is bilinear (i.e. it commutes with $+$) while abstraction is not linear. Indeed, one can prove that $(M+M')(N+N')$ and $MN+MN'+M'N+M'N'$ are operationally indistinguishable, while $\lambda x.(M+N)$ and $\lambda x.M + \lambda x.N$, in general, are not. For example, take $S = \lambda x.(x + \mathbf{I})$, $S' = \lambda x.x + \lambda x.\mathbf{I}$, $E_{\mathbf{I}} = \lambda x.\mathbf{I}$, $E_{\mathbf{\Omega}} = \lambda x.\mathbf{\Omega}$, and $F = \lambda b.bE_{\mathbf{\Omega}}(bE_{\mathbf{I}}E_{\mathbf{\Omega}})\mathbf{I}$. Now observe that FS is converging to the value $\mathbf{I}$, while FS' diverges. Indeed, remarking that $SE_{\mathbf{I}}E_{\mathbf{\Omega}}$ reduces non-deterministically to $\mathbf{I}$ and to $E_{\mathbf{\Omega}}$, we have:



while FS' has two reducts, either $F\mathbf{I}$ reducing to $\mathbf{\Omega I}$, or $FE_{\mathbf{I}}$ reducing to $\mathbf{\Omega}$.

Finally, we give two examples mixing $+$ and $\|$. The term $(\lambda x.(x \| x))(V+V')$ converges either to $V \| V$ or to $V' \| V'$, while the term $(\lambda x.(x + x))(V \| V')$ converges to $V \| V'$, only.

$$\frac{}{x:\tau\vdash x:\tau}\,ax \qquad \frac{\Delta_i, x:\tau_i\vdash M:\alpha_i \qquad 1\le i\le n}{\bigotimes_{i=1}^{n}\Delta_i\vdash \lambda x.M:\bigotimes_{i=1}^{n}(\tau_i\multimap\alpha_i)}\multimap_I \quad n\ge 0$$

$$\frac{\Delta\vdash M:\bigparr_{i=1}^{k}\bigotimes_{j=1}^{n_i}(\tau_{ij}\multimap\alpha_{ij}) \qquad \Gamma_i\vdash N:\bigparr_{j=1}^{n_i}\tau_{ij} \quad 1\le i\le k}{\Delta\otimes\bigotimes_{i=1}^{k}\Gamma_i\vdash MN:\bigparr_{i=1}^{k}\bigparr_{j=1}^{n_i}\alpha_{ij}}\multimap_E \quad \begin{array}{c}k\ge 1\\ n_i\ge 1\end{array}$$

$$\frac{\Delta\vdash M:\alpha}{\Delta\vdash M+N:\alpha}+_\ell \qquad \frac{\Delta\vdash N:\alpha}{\Delta\vdash M+N:\alpha}+_r \qquad \frac{\Delta\vdash M:\alpha_1 \qquad \Gamma\vdash N:\alpha_2}{\Delta\otimes\Gamma\vdash M\parallel N:\alpha_1\parr\alpha_2}\parallel_I$$

Fig. 2: Type system: the inference rules.

3 Linear Logic Based Type System

In this section we introduce our type system based on linear logic. The set $\mathbb{T}$ of *(parallel) types* and the set $\mathbb{C}$ of *computational types* are generated by the following grammar:

parallel-types:	$\alpha,\beta ::=$	$\alpha\parr\beta \mid \tau$	$\mathbb{T}$
computational-types:	$\tau,\rho ::=$	$\mathbf{1} \mid \tau\otimes\rho \mid \tau\multimap\alpha$	$\mathbb{C}$

For the sake of simplicity, types are considered up to associativity and commutativity of the tensor $\otimes$ and the par $\parr$. The type $\mathbf{1}$, which is the only atomic type, represents the empty tensor and is therefore its neutral element (i.e. $\tau\otimes\mathbf{1}=\tau$). Accordingly, we write $\otimes_{i=1}^{n}\tau_i$ for $\tau_1\otimes\cdots\otimes\tau_n$ when $n\ge 1$, and for $\mathbf{1}$ when $n=0$. Similarly, when $n\ge 1$, $\parr_{i=1}^{n}\alpha_i$ stands for $\alpha_1\parr\cdots\parr\alpha_n$. We do not allow the empty par as it would correspond to an empty sum of terms, that would be delicate to treat operationally [17]. Note that neither $\otimes$ nor $\parr$ is idempotent.

Definition 3. *A* context Γ *is a total map from* Var *to* $\mathbb{C}$*, such that* $\mathrm{dom}(\Gamma)=\{x\mid\Gamma(x)\neq\mathbf{1}\}$ *is finite. The tensor of two contexts Γ and Δ, written $\Gamma\otimes\Delta$, is defined pointwise.*

As a matter of notation, we write $x_1:\tau_1,\dots,x_n:\tau_n$ for the context Γ such that $\Gamma(x_i)=\tau_i$ and $\Gamma(y)=\mathbf{1}$ for all $y\notin\vec{x}$. The context mapping all variables to $\mathbf{1}$ is denoted by $\emptyset$; note that $\Gamma\otimes\emptyset=\Gamma$.

Definition 4.

- *The* type system *for $\Lambda_{+\parallel}$ is defined in Figure 2.* Typing judgements *are of the form $\Gamma\vdash M:\alpha$; when $\Gamma=\emptyset$ we simply write $\vdash M:\alpha$.* Derivation trees *will be denoted by π.*

– *A term* $M \in \Lambda_{+\|}$ *is* typable *if there exist* $\alpha \in \mathbb{T}$ *and a context* Γ *such that* $\Gamma \vdash M : \alpha$.

The rules for typing non-deterministic choice and parallel composition reflect their operational behaviour. Non-deterministic choice is may-convergent, thus it is enough to ask that one of the terms in a sum is typable; on the other hand parallel composition is must-convergent, we therefore require that all its components are typable. Intuitively, when dealing with closed terms, the $⅋$ operator can be only introduced to type a parallel composition, and gives an account of the number of its components. In fact, for closed regular λ-terms, the type system looses the $⅋$-level and collapses to the one presented in [7].

The $\multimap_E$ rule reflects the distribution of the parallel operator over the application. For example, take $M = x \parallel x'$ and $N = y \parallel y'$ in the premises of $\multimap_E$, then we have $k = 2$ and $n_1 = n_2 = 2$ so that the type of the term MN is a $⅋$ of four types, which is in accordance with $(x \parallel x')(y \parallel y') \to^* (xy \parallel xy') \parallel (x'y \parallel x'y')$.

Remark 5 *For every* $V \in \mathrm{V}_{+\|}$ *we can derive* $\vdash V : \mathbf{1}$*. Indeed, if* V *is a variable, then the derivation follows by* ax*; if* V *is an abstraction, then it follows by* $\multimap_I$ *using* $n = 0$*. As a simple consequence we get* $\vdash V_1 \parallel \cdots \parallel V_k : \mathbf{1} ⅋ \cdots ⅋ \mathbf{1}$ *(k times) for all* $V_1, \ldots, V_k \in \mathrm{V}_{+\|}$.

Concerning the possible types of values, the next more general lemma holds.

Lemma 6. *Let* $V \in \mathrm{V}_{+\|}$*. If* $\Delta \vdash V : \alpha$ *then* $\alpha \in \mathbb{C}$.

Proof. A proof of $\Delta \vdash V : \alpha$ ends in either a ax or a $\multimap_I$ rule. In both cases α is a computational-type. □

To help the reader to get familiar with the type system, we provide some examples of typable and untypable terms.

Example 7. Recall that $\mathbf{I} = \lambda x.x$, $\mathbf{\Delta} = \lambda x.xx$ and $\mathbf{\Omega} = \mathbf{\Delta\Delta}$.

1. $\vdash \mathbf{I} : \bigotimes_{i=1}^{n}(\tau_i \multimap \tau_i)$ and $\vdash \lambda x.\mathbf{I} : \bigotimes_{i=1}^{n}(\mathbf{1} \multimap \bigotimes_{j=1}^{k_i}(\tau_{ij} \multimap \tau_{ij}))$.
2. $\vdash \mathbf{\Delta} : \bigotimes_{i=1}^{n}((\tau_i \multimap \alpha_i) \otimes \tau_i) \multimap \alpha_i$.
3. $\mathbf{\Omega}$ is not typable. By contradiction, suppose $\vdash \mathbf{\Omega} : \alpha$. By ($\multimap_E$) and (2) there is a type τ such that $\vdash \mathbf{\Delta} : \tau \multimap \alpha$ and $\vdash \mathbf{\Delta} : \tau$. Let us choose such a τ with minimal size. Applying (2) to $\vdash \mathbf{\Delta} : \tau \multimap \alpha$, we get $\tau = (\tau' \multimap \alpha) \otimes \tau'$, from which one can deduce (see Lemma 9, below) that $\vdash \mathbf{\Delta} : \tau' \multimap \alpha$ and $\vdash \mathbf{\Delta} : \tau'$, thus contradicting the minimality of τ.
4. However, $\vdash \lambda x.\mathbf{\Omega} : \mathbf{1}$, so $\vdash \lambda x.\mathbf{\Omega} + \mathbf{\Omega} : \mathbf{1}$, but $\lambda x.\mathbf{\Omega} \parallel \mathbf{\Omega}$ is not typable.
5. From (1) and (4) we get: $\vdash \mathbf{I} \parallel \lambda x.\mathbf{\Omega} : (\bigotimes_{i=1}^{n}(\tau_i \multimap \tau_i)) ⅋ \mathbf{1}$.

We now define a measure associating a natural number with every derivation tree. In Section 4.1 we prove that such a measure decreases along the reduction. In the next definition we follow the notation of Figure 2, in particular in the $\multimap_E$-case the parameter n_i refers to the arity of the $⅋$ in the conclusion of π_i.

Definition 8. *The* measure $|\pi|$ *of a derivation tree* π *is defined inductively as:*

$$\pi = \frac{}{S}\ ax \qquad |\pi| = 0$$

$$\pi = \frac{\pi_1 \ \cdots \ \pi_n}{S} \multimap_I \qquad |\pi| = \sum_{i=1}^{n} |\pi_i|$$

$$\pi = \frac{\pi_0 \quad \pi_1 \dots \pi_k}{S} \multimap_E \ \begin{array}{c} k \geq 1 \\ n_i \geq 1 \end{array} \qquad |\pi| = \sum_{i=0}^{k} |\pi_i| + (\sum_{i=1}^{k} 2n_i) - 1$$

$$\pi = \frac{\pi'}{S} +_\ell \quad \textit{or} \quad \pi = \frac{\pi'}{S} +_r \qquad |\pi| = |\pi'| + 1$$

$$\pi = \frac{\pi_1 \qquad \pi_2}{S} \parallel_I \qquad |\pi| = |\pi_1| + |\pi_2|$$

Hereafter, we may slightly abuse the notation and write $\pi = \Gamma \vdash M : \alpha$ *to refer to a derivation tree* π *ending by the sequent* $\Gamma \vdash M : \alpha$*.*

The measure of a derivation only depends on its rules of type $\multimap_E$, $+_\ell$ and $+_r$. These are in fact the kinds of rules that can type a redex (β_v and $\parallel$ redexes are typed by $\multimap_E$ rules, $+$ redexes by $+_\ell$, $+_r$ rules). Each occurrence of a $+_\ell$ or $+_r$ rule counts for one, because a $+$-reduction does not create new rules in the derivation typing the contractum (see the proof of Theorem 11 for more details). An occurrence of a $\multimap_E$ counts for the number of "active" connectives appearing in the principal premise, i.e. the number of the connectives that are underlined in the left-most premise of the $\multimap_E$ rule in Figure 2, indeed

$$\underbrace{\sum_{i=1}^{k} n_i}_{\multimap\text{'s}} + \underbrace{\sum_{i=1}^{k} (n_i - 1)}_{\otimes\text{'s}} + \underbrace{(k-1)}_{⅋\text{'s}} = (\sum_{i=1}^{k} 2n_i) - 1.$$

Such a weight is needed since the $\parallel$-reduction creates two new $\multimap_E$ rules in the derivation typing the contractum. The measure decreases however, since the sum of the weight of the two new rules is less than the weight of the eliminated rule.

For example, let us consider the derivation tree π in Figure 3, which types the $\parallel$-redex $\mathbf{\Delta}(\mathbf{I} \parallel \lambda xy.\mathbf{\Omega})$ with $\mathbf{1} ⅋ \mathbf{1}$, and has three $\multimap_E$ rules — one of weight 1 in each subtree π_1, π_2, and one of weight 3 giving the conclusion, so that $|\pi| = 5$. Now, the $\multimap_E$-rule ending π splits into two $\multimap_E$-rules in the derivation tree π' typing the contractum of $\mathbf{\Delta}(\mathbf{I} \parallel \lambda xy.\mathbf{\Omega})$, namely $\pi' = \ \vdash \mathbf{\Delta I} \parallel \mathbf{\Delta}(\lambda xy.\mathbf{\Omega}) : \mathbf{1} ⅋ \mathbf{1}$. However, $|\pi'| = |\pi| - 1$ since the number of the active connectives of the $\multimap_E$-rule concluding π is greater than the sum of the number of the active connectives of its "residuals" in π'.

Finally, note that the term $\mathbf{\Delta}(\mathbf{I} \parallel \lambda xy.\mathbf{\Omega})$ reduces to the value $\mathbf{I} \parallel \lambda y.\mathbf{\Omega}$ in $5 = |\pi|$ steps. As we will show in Theorem 13 this does not happen by chance.

$$\pi = \cfrac{\cfrac{\pi_1 = x : \tau \vdash xx : \mathbf{1} \quad \pi_2 = x : \tau \vdash xx : \mathbf{1}}{\vdash \mathbf{\Delta} : (\tau \multimap \mathbf{1}) \otimes (\tau \multimap \mathbf{1})} \multimap_I \qquad \cfrac{\vdash \mathbf{I} : \tau \quad \vdash \lambda xy.\mathbf{\Omega} : \tau}{\vdash \mathbf{I} \parallel \lambda xy.\mathbf{\Omega} : \tau \parr \tau} \parallel_I}{\vdash \mathbf{\Delta}(\mathbf{I} \parallel \lambda xy.\mathbf{\Omega}) : \mathbf{1} \parr \mathbf{1}} \multimap_E$$

$$\pi' = \cfrac{\cfrac{\cfrac{\pi_1 = x : \tau \vdash xx : \mathbf{1}}{\vdash \mathbf{\Delta} : \tau \multimap \mathbf{1}} \multimap_I \quad \vdash \mathbf{I} : \tau}{\vdash \mathbf{\Delta I} : \mathbf{1}} \multimap_E \qquad \cfrac{\cfrac{\pi_2 = x : \tau \vdash xx : \mathbf{1}}{\vdash \mathbf{\Delta} : \tau \multimap \mathbf{1}} \multimap_I \quad \vdash \lambda xy.\mathbf{\Omega} : \tau}{\vdash \mathbf{\Delta}(\lambda xy.\mathbf{\Omega}) : \mathbf{1}} \multimap_E}{\mathbf{\Delta I} \parallel \mathbf{\Delta}(\lambda xy.\mathbf{\Omega}) : \mathbf{1} \parr \mathbf{1}} \parallel_I$$

Fig. 3: Derivation trees typing, respectively, the $\parallel$-redex $\mathbf{\Delta}(\mathbf{I} \parallel \lambda xy.\mathbf{\Omega})$ and its contractum $\mathbf{\Delta I} \parallel \mathbf{\Delta}(\lambda xy.\mathbf{\Omega})$, taking $\tau = (\mathbf{1} \multimap \mathbf{1}) = (\mathbf{1} \multimap \mathbf{1}) \otimes \mathbf{1}$.

4 Properties of the Type System

We prove that the set of types assigned to a term is invariant under $\to$, in a non-deterministic setting. More precisely, Theorem 11 states that if N is the contractum of a $\{\beta_v, \parallel\}$-redex in M, then any type of M is a type of N, and if N and N' are the two possible contracta of a $+$-redex in M, then any type of M is either a type of N or of N' (*subject reduction*). On the other hand Theorem 12 shows the converse, namely that whenever $M \to N$, any type of N is a type of M (*subject expansion*). Moreover, the two theorems combined prove that the measure associated with the typing tree of a term decreases (resp. increases) of exactly one unit at each typed step of reduction (resp. expansion). This is typical of non-idempotent intersection type systems, as discussed in the introduction. As a consequence, any typable term M is normalising and the measure of specific derivation trees of M gives the length of a converging reduction sequence.

4.1 Subject reduction

In order to prove subject reduction we first need some preliminary lemmas. Their proofs are lengthy but not difficult, therefore we write explicitly only the most interesting cases — the remaining cases can be found in Appendix A.1.

Lemma 9. *We have that $\pi = \Delta \vdash V : \bigotimes_{i=1}^{n} \tau_i$ if and only if $\Delta = \bigotimes_{i=1}^{n} \Delta_i$ and $\pi_i = \Delta_i \vdash V : \tau_i$ for all $i = 1, \dots, n$. Moreover, $|\pi| = \sum_{i=1}^{n} |\pi_i|$.*

Proof. We only prove ($\Rightarrow$), the other direction being similar. Since V is a value, the last rule of π is either ax or $\multimap_I$. The first case is trivial. In the second case, $V = \lambda x.M$ and the premises of the $\multimap_I$-rule are $m \geq n$, say $\pi'_j = \Delta_j, x : \rho_j \vdash M : \alpha_j$ for $j \leq m$, and $\tau_1 = \bigotimes_{j=1}^{m_1} \rho_j \multimap \alpha_j$ and $\Delta_1 = \bigotimes_{j=1}^{m_1} \Delta_j$, $\dots$, $\tau_n = \bigotimes_{j=m_{n-1}+1}^{m_n} \rho_j \multimap \alpha_j$ and $\Delta_n = \bigotimes_{j=m_{n-1}+1}^{m_n} \Delta_j$, with $m_1 + \cdots + m_n = m$.

Notice $|\pi| = \sum_{j=1}^{m} |\pi'_j|$. Then, for every $i \leq n$, a $\multimap_I$-rule with premises $\pi'_{m_{i-1}+1}, \dots, \pi'_{m_i}$ yields $\pi_i = \Delta_i \vdash \lambda x.M : \tau_i$, with $|\pi_i| = \sum_{j=m_{i-1}+1}^{m_i} |\pi'_i|$, therefore $|\pi| = \sum_{i=1}^{n} |\pi_i|$. ☐

Lemma 10 (Substitution lemma). *If $\pi_1 = \Delta, x : \tau \vdash M : \alpha$ and $\pi_2 = \Gamma \vdash V : \tau$, then there is $\pi_3 = \Delta \otimes \Gamma \vdash M[V/x] : \alpha$. Moreover $|\pi_3| = |\pi_1| + |\pi_2|$.*

Proof. By structural induction on M. We only treat the most interesting case, namely $M = NP$. In this case, the last rule of π_1 is a $\multimap_E$-rule with $k+1$ premises, say $\pi_1^0 = \Delta_0, x : \tau_0 \vdash N : \parr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$, and for $i = 1, \dots, k$, $\pi_1^i = \Delta_i, x : \tau_i \vdash P : \parr_{j=1}^{n_i} \rho_{ij}$, where $\Delta = \bigotimes_{i=0}^{k} \Delta_i$, $\tau = \bigotimes_{i=0}^{k} \tau_i$, $\alpha = \parr_{i=1}^{k} \parr_{j=1}^{n_i} \alpha_{ij}$ and $|\pi_1| = \sum_{i=0}^{k} |\pi_1^i| + (\sum_{i=1}^{k} 2n_i) - 1$. By Lemma 9, we can split π_2 into $k+1$ derivations $\pi_2^i = \Gamma_i \vdash V : \tau_i$, for $i = 0, \dots, k$, such that $\Gamma = \bigotimes_{i=0}^{k} \Gamma_i$ and $|\pi_2| = \sum_{i=0}^{k} |\pi_2^i|$. By the induction hypothesis, there are $\pi_3^0 = \Delta_0 \otimes \Gamma_0 \vdash N[V/x] : \parr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$, with $|\pi_3^0| = |\pi_1^0| + |\pi_2^0|$, and for $i = 1, \dots, k$, $\pi_3^i = \Delta_i \otimes \Gamma_i \vdash P[V/x] : \parr_{j=1}^{n_i} \rho_{ij}$, with $|\pi_3^i| = |\pi_1^i| + |\pi_2^i|$. Hence, by rule $\multimap_E$, we have

$$\pi_3 = (\Delta_0 \otimes \Gamma_0) \otimes \bigotimes_{i=1}^{k} (\Delta_i \otimes \Gamma_i) \vdash N[V/x]P[V/x] : \parr_{i=1}^{k} \parr_{j=1}^{n_i} \alpha_{ij}$$

Notice that $(\Delta_0 \otimes \Gamma_0) \otimes \bigotimes_{i=1}^{k} (\Delta_i \otimes \Gamma_i) = \Delta \otimes \Gamma$ and $N[V/x]P[V/x] = (NP)[V/x]$. Moreover, $|\pi_3| = \sum_{i=0}^{k} |\pi_3^i| + (\sum_{i=1}^{k} 2n_i) - 1 = \sum_{i=0}^{k} (|\pi_1^i| + |\pi_2^i|) + (\sum_{i=1}^{k} 2n_i) - 1 = (\sum_{i=0}^{k} |\pi_1^i| + (\sum_{i=1}^{k} 2n_i) - 1) + \sum_{i=0}^{k} |\pi_2^i| = |\pi_1| + |\pi_2|$. ☐

We now prove the subject reduction property, which ensures that the type is preserved during reduction, while the measure of the typing is strictly decreasing.

As a matter of terminology, we say that a term M *reduces to a term* N *using* $+$*-reductions*, if $M \to N$ is derivable as a direct consequence of a $+$-reduction and (possibly) some contextual rules. In the following proof, given a set S, we denote by $\sharp S$ its cardinality.

Theorem 11 (Subject reduction). *Let $\pi = \Delta \vdash M : \alpha$.*
If $M \to N$ without using $+$-reductions, then there is $\pi' = \Delta \vdash N : \alpha$.
If $M \to N_1$ and $M \to N_2$ using $+$-reductions, then there is π' such as either $\pi' = \Delta \vdash N_1 : \alpha$ or $\pi' = \Delta \vdash N_2 : \alpha$.
Moreover, in both cases we have $|\pi'| = |\pi| - 1$.

Proof. We proceed by induction on the length of the derivation of $M \to N$. We only treat the most interesting cases.

- $(\lambda x.M')V \to M'[V/x]$. Then, the last rule of π is a $\multimap_E$-rule with $k+1$ premises, say $\pi_0 = \Delta' \vdash \lambda x.M' : \parr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for every $i = 1, \dots, k$, $\pi_i = \Gamma_i \vdash V : \parr_{j=1}^{n_i} \rho_{ij}$, with moreover $\Delta = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i$, $\alpha = \parr_{i=1}^{k} \parr_{j=1}^{n_i} \alpha_{ij}$, and $|\pi| = \sum_{i=0}^{k} |\pi_i| + (\sum_{i=1}^{k} 2n_i) - 1$. However, since

Lemma 6 entails that $k = n_1 = 1$ we get $|\pi| = |\pi_0| + |\pi_1| + 1$. In addition, the only possibility for π_0 is to come from $\pi_0' = \Delta', x : \rho \vdash M' : \alpha$, where $|\pi_0| = |\pi_0'|$. By Lemma 10, $\pi' = \Delta' \otimes \Gamma \vdash M'[V/x] : \alpha$, where $|\pi'| = |\pi_0'| + |\pi_1| = |\pi_0| + |\pi_1| = |\pi| - 1$. We conclude since $\Delta' \otimes \Gamma = \Delta$.

– Let $V(M \parallel N) \to VM \parallel VN$. Then $\pi = \Delta \otimes \bigotimes_{i=1}^{k} \Gamma_i \vdash V(M \parallel N) : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$ ends in a $\multimap_E$ rule having as premises $\pi_0 = \Delta \vdash V : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and, for $i = 1, \dots, k$, $\pi_i = \Gamma_j \vdash M \parallel N : \bigparr_{j=1}^{n_i} \rho_{ij}$. Thus, we have $|\pi| = \sum_{j=0}^{k} |\pi_i| + (\sum_{i=1}^{k} 2n_i) - 1$. However, by Lemma 6, $k = 1$, so we omit the index i where it is not needed, and $|\pi| = |\pi_0| + |\pi_1| + 2n - 1$. Then $\pi_1^1 = \Gamma_1 \vdash M : \bigparr_{j \in S} \rho_j$ and $\pi_1^2 = \Gamma_2 \vdash N : \bigparr_{j \in \bar{S}} \rho_j$, where $\Gamma = \Gamma_1 \otimes \Gamma_2$, $\emptyset \neq S \subsetneq \{1, \dots, k\}$ and $\bar{S} = \{1, \dots, k\} \setminus S$ with $|\pi_1| = |\pi_1^1| + |\pi_1^2|$. By Lemma 9, we can split π_0 into two derivations, $\pi_0^S = \bigotimes_{j \in S} \Delta_j \vdash V : \bigotimes_{j \in S} (\rho_j \multimap \alpha_j)$ and $\pi_0^{\bar{S}} = \bigotimes_{j \in \bar{S}} \Delta_j \vdash V : \bigotimes_{j \in S} (\rho_j \multimap \alpha_j)$, with $|\pi_0^S| + |\pi_0^{\bar{S}}| = |\pi_0|$. By rule $\multimap_E$, we have $\pi^1 = \bigotimes_{j \in S} \Delta_j \otimes \Gamma_1 \vdash VM : \bigparr_{j \in S} \alpha_j$ and $\pi^2 = \bigotimes_{j \in \bar{S}} \Delta_j \otimes \Gamma_2 \vdash VN : \bigparr_{j \in \bar{S}} \alpha_j$, where $|\pi^1| = |\pi_0^S| + |\pi_1^1| + 2\sharp S - 1$, and $|\pi^2| = |\pi_0^{\bar{S}}| + |\pi_1^2| + 2\sharp\bar{S} - 1$. By rule $\parallel_I$, $\pi' = \bigotimes_{j=1}^{n} \Delta_i \otimes \Gamma_1 \otimes \Gamma_2 \vdash VM \parallel VN : \bigparr_{j=1}^{n} \alpha_j$, where $|\pi'| = |\pi^1| + |\pi^2| = (|\pi_0^S| + |\pi_1^1| + 2\sharp S - 1) + (|\pi_0^{\bar{S}}| + |\pi_1^2| + 2\sharp\bar{S} - 1) = |\pi_0| + |\pi_1| + 2\sharp S + 2\sharp\bar{S} - 2 = |\pi_0| + |\pi_1| + 2n - 2 = |\pi| - 1$. ☐

4.2 Subject Expansion

The proof of the fact that our system enjoys subject expansion follows by straightforward induction, once one has proved the commutation of abstraction with abstraction, application, non-deterministic choice and parallel composition.

We refer to Appendix A.2 for a detailed proof.

Theorem 12 (Subject expansion). *If $M \to N$ and $\pi = \Delta \vdash N : \alpha$, then there is $\pi' = \Delta \vdash M : \alpha$, such that $|\pi'| = |\pi| + 1$.*

Proof. By induction on the length of the derivation of $M \to N$, splitting into cases depending on its last rule. We only consider the most interesting case, i.e. $(\lambda x.M')V \to M'[V/x]$ where $M' = PQ$. One first needs to establish, by induction on π, a claim about the commutation of abstraction with application.

Claim. If $\pi = \Delta \vdash ((\lambda x.P)V)((\lambda x.Q)V) : \alpha$, where the last rule of π is a $\multimap_E$ rule having $k+1$ premises, then there exists $\pi' = \Delta \vdash (\lambda x.PQ)V : \alpha$ such that $|\pi'| = |\pi| - k$.

By definition we have $N = (PQ)[V/x] = P[V/x]Q[V/x]$. So, $\pi = \Delta \vdash N : \alpha$ ends in a $\multimap_E$-rule with $k+1$ premises $\pi_0 = \Delta' \vdash P[V/x] : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\tau_{ij} \multimap \alpha_{ij})$ and $\pi_i = \Gamma_i \vdash Q[V/x] : \bigparr_{j=1}^{n_i} \tau_{ij}$ for $i = 1, \dots, k$, with $\Delta = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i$, $\alpha = \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$ and $|\pi| = \sum_{i=0}^{k} \pi_i + (\sum_{i=1}^{k} 2n_i) - 1$. Then, by the induction hypothesis, we get $\pi_0' = \Delta' \vdash (\lambda x.P)V : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\tau_{ij} \multimap \alpha_{ij})$, and $\pi_i' = \Gamma_i \vdash (\lambda x.Q)V : \bigparr_{j=1}^{n_i} \tau_{ij}$, with $|\pi_i'| = |\pi_i| + 1$. Hence by rule $\multimap_E$

we obtain $\pi'' = \Delta' \otimes \bigotimes_{i=1}^k \Gamma_i \vdash ((\lambda x.P)V)((\lambda x.Q)V) : \parr_{i=1}^k \parr_{j=1}^{n_i} \alpha_{ij}$, with $|\pi''| = \sum_{i=0}^k |\pi'_i| + (\sum_{i=1}^k 2n_i) - 1$. By the above claim, we get $\pi' = \Delta' \otimes \bigotimes_{i=1}^k \Gamma_i \vdash (\lambda x.PQ)V : \parr_{i=1}^k \parr_{j=1}^{n_i} \alpha_{ij}$ such that $|\pi'| = |\pi''| - k = |\pi| + 1$. □

4.3 Convergence

From our "quantitative" versions of subject reduction and subject expansion one easily obtains that our type system captures exactly the weakly normalising terms, and that the size $|\pi|$ of a derivation tree $\pi = \vdash M : \alpha$ decreases along the reduction of M. However, when α satisfies in addition a suitable minimality condition (namely the fact that α is of shape $\mathbf{1} \parr \cdots \parr \mathbf{1}$), then we can be more precise and say that there exists a reduction from M to a normal form, having length *exactly* $|\pi|$.

In the following $\parr^k \mathbf{1}$, with $k > 0$, stands for $\mathbf{1} \parr \cdots \parr \mathbf{1}$ (k times).

Theorem 13. *Let M be a closed term, and $k > 0$. There is a typing tree π for $\vdash M : \parr^k \mathbf{1}$ iff there are values $V_1, \dots, V_k$ and a reduction $M \to^* V_1 \parallel \cdots \parallel V_k$ of length $|\pi|$.*

Proof. ($\Rightarrow$) Suppose $\pi = \vdash M : \parr^k \mathbf{1}$. We proceed by induction on $|\pi|$. If $M = V_1 \parallel \cdots \parallel V_{k'}$, then π must start with a tree of $k'-1$ rules $\parallel$, and then k' rules $\multimap_I$ with conclusion, respectively, $\vdash V_1 : \mathbf{1}, \dots, \vdash V_{k'} : \mathbf{1}$. We then have $k = k'$, and M trivially converges to $V_1 \parallel \cdots \parallel V_{k'}$ in $|\pi| = 0$ steps.

Otherwise, since M is closed, there exists N such that $M \to N$. By Theorem 11, such an N can be chosen in such a way $\pi' = \vdash N : \parr^k \mathbf{1}$, with $|\pi'| = |\pi| - 1$. From the induction hypothesis we know that N converges in $|\pi'|$ steps to $V_1 \parallel \cdots \parallel V_k$. Therefore, M converges in $|\pi'| + 1 = |\pi|$ steps to $V_1 \parallel \cdots \parallel V_k$.

($\Leftarrow$) Suppose that $M \to^* V_1 \parallel \cdots \parallel V_k$. By Remark 5, there is $\pi = \vdash V_1 \parallel \cdots \parallel V_k : \parr^k \mathbf{1}$ and $|\pi| = 0$. Therefore, by the subject expansion (Theorem 12) there is $\pi' = \vdash M : \parr^k \mathbf{1}$ and $|\pi'|$ is equal to the length of the reduction $M \to^* V_1 \parallel \cdots \parallel V_k$. □

Corollary 14. *Let M be closed, then M is typable if and only if M converges.*

5 Adequacy and (Lack of) Full Abstraction

The choice of presenting a model through a type discipline or a reflexive object is more a matter of taste rather than a technical decision. (Compare for instance the type system of [18] and the interpretation of [8]). The model $\mathcal{V}$ associated with our type system lives in the category **Rel** of sets and relations (refer to [7] for more details) and is defined by $\mathcal{V} = \bigcup_{n \in \mathbb{N}} \mathcal{V}_n$, with

$$\mathcal{V}_0 = \emptyset, \qquad \mathcal{V}_{n+1} = \mathcal{M}_{\mathrm{f}}(\mathcal{V}_n) \times \mathcal{M}_{\mathrm{f}}(\mathcal{M}_{\mathrm{f}}(\mathcal{V}_n)),$$

where $\mathcal{M}_{\mathrm{f}}(X)$ denotes the set of finite multisets over a set X. In fact, $\mathcal{M}_{\mathrm{f}}(X)$ interprets in **Rel** the exponentials $!X$ and $?X$, whilst the cartesian product is the linear implication $\multimap$, so that $\mathcal{V}$ is the minimal solution of the equation $\mathcal{V} \simeq\, !\mathcal{V} \multimap ?!\mathcal{V}$. Recalling Equation 1 in the introduction, this means that the object $\mathcal{V}$ represents "value types", while computational types $\mathbb{C}$ will be represented by elements of $\mathcal{C} = !\mathcal{V} = \mathcal{M}_{\mathrm{f}}(\mathcal{V})$ and parallel-types $\mathbb{T}$ as elements of $\mathcal{T} = ?\mathcal{C} = \mathcal{M}_{\mathrm{f}}(\mathcal{C})$. This intuition can be formalized by defining two injections $(\cdot)^\circ : \mathbb{T} \to \mathcal{T}$ and $(\cdot)^\bullet : \mathbb{C} \to \mathcal{C}$ by mutual induction, as follows: $\tau^\circ = [\tau^\bullet]$, $(\alpha \parr \beta)^\circ = \alpha^\circ \uplus \beta^\circ$, $\mathbf{1}^\bullet = []$, $(\tau \otimes \rho)^\bullet = \tau^\bullet \uplus \rho^\bullet$ and $(\tau \multimap \alpha)^\bullet = [(\tau^\bullet, \alpha^\circ)]$.

It is beyond the scope of the present paper to give the explicit inductive definition of the interpretation of terms. For our purpose it is enough to know that such an interpretation can be characterised (up to isomorphism) as follows.

Definition 15. *The* interpretation of a closed term M *is defined by* $[\![M]\!] = \{\alpha \mid\ \vdash M : \alpha\} \subseteq \mathbb{T}$.

The interpretations of terms are naturally ordered by set-theoretical inclusion; an interesting problem is to determine whether there is a relationship between this ordering and the following observational preorder on terms.

Definition 16 (Observational preorder). *Let $M, N \in \Lambda_{+\|}$ be closed. We set $M \sqsubseteq N$ iff for all closed terms $\vec{P}$, $M\vec{P}$ converges implies that $N\vec{P}$ converges.*

A model is called *adequate* if $[\![M]\!] \subseteq [\![N]\!]$ entails $M \sqsubseteq N$; it is called *fully abstract* if in addition the converse holds.

The adequacy of the model $\mathcal{V}$ follows easily from Theorem 13 and the monotonicity of the interpretation.

Corollary 17 (Adequacy). *For all M, N closed, if $[\![M]\!] \subseteq [\![N]\!]$ then $M \sqsubseteq N$.*

On the contrary, $\mathcal{V}$ is not fully abstract. This is due to the fact that the call-by-value λ-calculus admits the creation of an 'ogre' that is able to 'eat' any finite sequence of arguments and converge, constituting then a top of the call-by-value observational preorder. Following [13], we define the ogre as $\mathrm{Y}^\star = \boldsymbol{\Delta}^\star\boldsymbol{\Delta}^\star$ where $\boldsymbol{\Delta}^\star = \lambda xy.xx$. The ogre $\mathrm{Y}^\star$ converges since $\mathrm{Y}^\star \to \lambda y.\mathrm{Y}^\star$ and remains convergent when applied to every sequence of values, by discarding them one at time.

Lemma 18. *For all closed terms M we have $M \sqsubseteq \mathrm{Y}^\star$.*

Proof. Given a term M and a sequence $\vec{P} = P_1 \cdots P_k$ of closed terms it is easy to check that $M\vec{P}$ can converge only when $\vec{P}$ converges. In that case we have $\mathrm{Y}^\star\vec{P} \to^* (\lambda y.\mathrm{Y}^\star)(V_1 \parallel \cdots \parallel V_n)P_2 \cdots P_k \to^* \mathrm{Y}^\star P_2 \cdots P_k \parallel \cdots \parallel \mathrm{Y}^\star P_2 \cdots P_k \to^* \lambda y.\mathrm{Y}^\star \parallel \cdots \parallel \lambda y.\mathrm{Y}^\star$. Therefore $\mathrm{Y}^\star$ is maximal with respect to $\sqsubseteq$. □

On the other hand, we have the following characterisation of $[\![\mathrm{Y}^\star]\!]$.

Lemma 19. *$\alpha \in [\![\mathrm{Y}^\star]\!]$ iff $\alpha = \bigotimes_{i=0}^{n}(\mathbf{1} \multimap \alpha_i)$ with $n \geq 0$ and $\alpha_i \in [\![\mathrm{Y}^\star]\!]$ for all $i \leq n$. In particular, we have that $[\![\mathbf{I}]\!] \not\subseteq [\![\mathrm{Y}^\star]\!]$.*

Proof. The crucial point is to remark that $\mathrm{Y}^\star \to \lambda y.\mathrm{Y}^\star$, so by Theorem 11 and 12, we get $[\![\mathrm{Y}^\star]\!] = [\![\lambda y.\mathrm{Y}^\star]\!]$. Therefore we have the following chain of equivalences:

$$\begin{aligned}
&\alpha \in [\![\mathrm{Y}^\star]\!] \text{ iff } \alpha \in [\![\lambda y.\mathrm{Y}^\star]\!] \\
&\text{iff } \alpha = \otimes_{i=0}^{n}(\tau_i \multimap \alpha_i) \in [\![\lambda y.\mathrm{Y}^\star]\!] && \text{by Lemma 6, } n \geq 0 \\
&\text{iff } \alpha = \otimes_{i=0}^{n}(\tau_i \multimap \alpha_i) \text{ and } \forall i, \tau_i \multimap \alpha_i \in [\![\lambda y.\mathrm{Y}^\star]\!] && \text{by Lemma 9} \\
&\text{iff } \alpha = \otimes_{i=0}^{n}(\tau_i \multimap \alpha_i) \text{ and } \forall i, \tau_i = \mathbf{1} \text{ and } \alpha_i \in [\![\mathrm{Y}^\star]\!] && \text{since } y \notin \mathrm{FV}(\mathrm{Y}^\star).
\end{aligned}$$

We have that $[\![\mathbf{I}]\!] \not\subseteq [\![\mathrm{Y}^\star]\!]$ as, for instance, $(\mathbf{1} \multimap \mathbf{1}) \multimap (\mathbf{1} \multimap \mathbf{1}) \in [\![\mathbf{I}]\!] \setminus [\![\mathrm{Y}^\star]\!]$. □

Summing up, get that $\mathbf{I} \sqsubseteq \mathrm{Y}^\star$, while $[\![\mathbf{I}]\!] \not\subseteq [\![\mathrm{Y}^\star]\!]$.

6 Conclusion and future work

We introduced a call-by-value non-deterministic λ-calculus with a type system ensuring convergence. We proved that such a type system gives a bound on the length of the lazy call-by-value reduction sequences, which is the exact length when the typing is minimal. Finally, we show that the relational model $\mathcal{V}$ capturing our type system is adequate, but not fully abstract.

As our counterexample to full abstraction contains no non-deterministic operators, it also holds for the standard call-by-value λ-calculus and the relational model described in [7]. This is a notable difference with the call-by-name case, where the relational model is proven to be fully abstract for the pure call-by-name λ-calculus [19], while other counterexamples (see [8, 20]) break full abstraction in presence of may or must non-deterministic operators. An open problem is to find a relational model fully abstract for the call-by-value λ-calculus.

Various fully abstract models of may and must non-determinism are known in the setting of Scott domain based semantics and idempotent intersection types. In particular, for the call-by-value case we mention [13, 14]. Comparing these models and type systems with the ones issued from the relational semantics is a research direction started in [7] with some notable results. It would be interesting to reach a better understanding of the role played by intersection idempotency in the question of full abstraction.

Another axis of research is to generalize our approach to study the convergence in (call-by-name and call-by-value) λ-calculi with richer algebraic structures than simply may/must non-deterministic operators, such as [21, 17]. In these calculi the choice operator is enriched with a weight, i.e. sums of terms are of the form $\alpha.M + \beta.N$, where α, β are scalars from a given semiring, pondering the choice. We would like to design type systems characterizing convergence properties in these systems. First steps have been done in [22, 23].

Acknowledgements. We wish to thank Thomas Ehrhard and Simona Ronchi Della Rocca for interesting discussions.

References

1. Coppo, M., Dezani-Ciancaglini, M.: A new type-assignment for λ-terms. Archiv für Math. Logik **19** (1978) 139–156
2. Sallé, P.: Une généralisation de la théorie de types en λ-calcul. RAIRO: Informatique Théorique **14**(2) (1980) 143–167
3. Krivine, J.L.: Lambda-calcul: types et modèles. Études et recherches en informatique. Masson (1990)
4. Amadio, R., Curien, P.L.: Domains and Lambda-Calculi. Number 46 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (1998)
5. de Carvalho, D.: Execution time of lambda-terms via denotational semantics and intersection types. To appear in *Math. Struct. in Comp. Sci.* (2008)
6. Bernadet, A., Lengrand, S.: Complexity of strongly normalising λ-terms via non-idempotent intersection types. In: FOSSACS 2011. (2011) 88–107
7. Ehrhard, T.: Collapsing non-idempotent intersection types. In: CSL'12. Volume 16 of LIPIcs. (2012) 259–273
8. Bucciarelli, A., Ehrhard, T., Manzonetto, G.: A relational semantics for parallelism and non-determinism in a functional setting. APAL **163**(7) (2012) 918–934
9. Girard, J.Y.: Linear logic. Theoretical Computer Science **50** (1987) 1–102
10. Maraist, J., Odersky, M., Turner, D.N., Wadler, P.: Call-by-name, call-by-value, call-by-need and the linear λ-calculus. Theor. Comp. Sci. **228**(1-2) (1999) 175–210
11. Laurent, O.: Étude de la polarisation en logique. PhD thesis, Université de Aix-Marseille II, France (2002)
12. Dezani-Ciancaglini, M., de'Liguoro, U., Piperno, A.: Filter models for conjunctive-disjunctive lambda-calculi. Theor. Comp. Sci. **170**(1-2) (1996) 83–128
13. Boudol, G.: Lambda-calculi for (strict) parallel functions. Information and Computation **108**(1) (1994) 51–127
14. Dezani-Ciancaglini, M., de'Liguoro, U., Piperno, A.: A filter model for concurrent lambda-calculus. SIAM J. Comput. **27**(5) (1998) 1376–1419
15. Plotkin, G.D.: Call-by-name, call-by-value and the λ-calculus. Theor. Comp. Sci. **1**(2) (1975) 125–159
16. Barendregt, H.: The lambda calculus: its syntax and semantics. North-Holland, Amsterdam (1984)
17. Arrighi, P., Dowek, G.: Linear-algebraic lambda-calculus: higher-order, encodings, and confluence. In: RTA'08. Volume 5117 of LNCS., Springer (2008) 17–31
18. Pagani, M., Ronchi Della Rocca, S.: Linearity, non-determinism and solvability. Fundam. Inform. **103**(1-4) (2010) 173–202
19. Manzonetto, G.: A general class of models of $\mathcal{H}^\star$. In: MFCS'09. Volume 5734 of LNCS., Springer (2009) 574–586
20. Breuvart, F.: On the discriminating power of tests in the resource λ-calculus Submitted. Draft available at http://hal.archives-ouvertes.fr/hal-00698609.
21. Vaux, L.: The algebraic lambda calculus. Math. Struct. in Comp. Sci. **19**(5) (2009) 1029–1059
22. Arrighi, P., Díaz-Caro, A.: A System F accounting for scalars. Logical Methods in Computer Science **8**(1) (2012)
23. Arrighi, P., Díaz-Caro, A., Valiron, B.: A type system for the vectorial aspects of the linear-algebraic λ-calculus. In: DCM'11. Volume 88 of EPTCS. (2012) 1–15

A Technical Appendix

This appendix is devoted to provide the full details of the proofs sketched in the main text of the paper.

A.1 Omitted proofs in Section 4.1

Lemma 9. We have that $\pi = \Delta \vdash V : \bigotimes_{i=1}^n \tau_i$ if and only if $\Delta = \bigotimes_{i=1}^n \Delta_i$ and $\pi_i = \Delta_i \vdash V : \tau_i$ for all $i = 1, \dots, n$. Moreover, $|\pi| = \sum_{i=1}^n |\pi_i|$.

Proof.

($\Rightarrow$) Let $\pi = \Delta \vdash V : \bigotimes_{i=1}^n \tau_i$. We split into cases:
  - $V = x$. Then $\Delta = x : \bigotimes_{i=1}^n \tau_i = \bigotimes_{i=1}^n \Delta_i$ with $\Delta_i = x : \tau_i$. By rule ax we have $\pi_i = x : \tau_i \vdash x : \tau_i$. Moreover, for all i we have $|\pi_i| = |\pi| = 0 = \sum_{i=1}^n |\pi_i|$.
  - $V = \lambda x.M$. Given in the main text of the paper.

($\Leftarrow$)

$n = 0$**.** Then $\Delta = \emptyset$ and $|\pi| = 0$. We split into subcases:
  - $V = x$, then only possibility is $\vdash x : \mathbf{1} = \bigotimes_{i=1}^0 \tau_i$.
  - $V = \lambda x.M$, then the only possibility is $\vdash \lambda x.M : \mathbf{1} = \bigotimes_{i=1}^0 (\tau_i \multimap \alpha_i)$.

$n > 0$**.** Let $\pi_i = \Delta_i \vdash V : \tau_i$ for $i = 1, \dots, n$. We split into subcases:
  - $V = x$. Then $\Delta_i = x : \tau_i$. By rule ax, $\pi = x : \bigotimes_{i=1}^n \tau_j \vdash x : \bigotimes_{i=1}^n \tau_i$. Notice that $|\pi| = 0 = \sum_{i=1}^n |\pi_i|$.
  - $V = \lambda x.M$. Then we have $\Delta_i = \bigotimes_{j=1}^{n_i} \Gamma_{ij}$, $\tau_i = \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for $j = 1, \dots, n_i$, $\pi_{ij} = \Gamma_{ij}, x : \rho_{ij} \vdash M : \alpha_{ij}$, with $|\pi_i| = \sum_{j=1}^{n_i} |\pi_{ij}|$. Hence, by rule $\multimap_I$, $\pi = \bigotimes_{i=1}^n \bigotimes_{j=1}^{n_i} \Gamma_{ij} \vdash \lambda x.M : \bigotimes_{i=1}^n \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$. Notice that $\bigotimes_{i=1}^n \bigotimes_{j=1}^{n_i} \Gamma_{ij} = \bigotimes_{i=1}^n \Delta_i$ and $\bigotimes_{i=1}^n \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij}) = \bigotimes_{i=1}^n \tau_i$. Therefore $|\pi| = \sum_{i=1}^n \sum_{j=1}^{n_i} |\pi_{ij}| = \sum_{i=1}^n |\pi_i|$. □

Lemma 10 (Substitution lemma)**.** If $\pi_1 = \Delta, x : \tau \vdash M : \alpha$ and $\pi_2 = \Gamma \vdash V : \tau$, then $\pi_3 = \Delta \otimes \Gamma \vdash M[V/x] : \alpha$. Moreover $|\pi_3| = |\pi_1| + |\pi_2|$.

Proof. We proceed by structural induction on M.

- $M = x$, then $\Delta = \emptyset$, $\alpha = \tau$ and $|\pi_1| = 0$. Notice that $\Delta \otimes \Gamma = \Gamma$ and $M[V/x] = V$. In addition, $|\pi_3| = |\pi_2| = |\pi_1| + |\pi_2|$.
- $M = z \neq x$, then $\Delta = z : \alpha$, $\tau = \mathbf{1}$ and $|\pi_1| = 0$. Since $\tau = \mathbf{1}$, by Lemma 9, we have $\Gamma = \emptyset$ and $|\pi_2| = 0$, so $\Delta \otimes \Gamma = z : \alpha$. As $M[V/x] = z$, we get by rule ax that $z : \alpha \vdash z : \alpha$. Concerning the measure we have $|\pi_3| = 0 = |\pi_1| + |\pi_2|$.
- $M = \lambda y.N$, where $y \neq x$. Then, the last rule of π_1 is a $\multimap_I$-rule with $n \geq 0$ premises $\pi_i = \Delta_i, x : \tau_i, y : \rho_i \vdash N : \alpha_i$, for $i \leq n$ and $\alpha = \bigotimes_{i=1}^n (\rho_i \multimap \alpha_i)$, $\Delta = \bigotimes_{i=1}^n \Delta_i$ and $\tau = \bigotimes_{i=1}^n \tau_i$.
  In the case $n = 0$, we have $\pi_1 = \vdash \lambda z.N : \mathbf{1}$, $\pi_2 = \vdash V : \mathbf{1}$ and both $|\pi_1|$ and $|\pi_2|$ are 0. Then, π_3 is simply an instance of a $\multimap_I$-rule with no premise.

Otherwise, by Lemma 9 we can split π_2 into n derivations $\pi_2^i = \Gamma_i \vdash V : \tau_i$ for $i \leq n$ and $\Gamma = \bigotimes_{i=1}^n \Gamma_i$, $|\pi_2| = \sum_{i=1}^n |\pi_2^i|$. By the induction hypothesis, there exists a derivation $\pi_3^i = \Gamma_i \otimes \Delta_i, y : \rho_i \vdash N[V/x] : \alpha_i$, with $|\pi_3^i| = |\pi_1^i| + |\pi_2^i|$. Hence by a $\multimap_I$-rule, we have

$$\pi_3 = \bigotimes_{i=1}^n (\Gamma_i \otimes \Delta_i) \vdash \lambda y.N[V/x] : \bigotimes_{i=1}^n (\rho_i \multimap \alpha_i).$$

Notice that $\bigotimes_{i=1}^n (\Gamma_i \otimes \Delta_i) = \Gamma \otimes \Delta$ and $(\lambda y.N)[V/x] = \lambda y.N[V/x]$. In addition, $|\pi_3| = \sum_{i=1}^n |\pi_3^i| = \sum_{i=1}^n (|\pi_1^i| + |\pi_2^i|) = |\pi_1| + |\pi_2|$.

– $M = NP$. Given in the main text of the paper.
– $M = M_1 + M_2$. Then either $\pi_1' = \Delta, x : \tau \vdash M_1 : \alpha$ or $\pi_1' = \Delta, x : \tau \vdash M_2 : \alpha$ with $|\pi_1| = |\pi_1'| + 1$, then by the induction hypothesis, either $\pi_3' = \Delta \otimes \Gamma \vdash M_1[V/x] : \alpha$ or $\pi_3' = \Delta \otimes \Gamma \vdash M_2[V/x] : \alpha$ with $|\pi_3'| = |\pi_1'| + |\pi_2|$. In any case, using either $+_\ell$ or $+_r$, we can derive $\pi_3 = \Delta \otimes \Gamma \vdash M_1[V/x] + M_2[V/x] : \alpha$. Notice that $M_1[V/x] + M_2[V/x] = (M_1 + M_2)[V/x]$ and $|\pi_3| = |\pi_3'| + 1 = |\pi_1'| + |\pi_2| + 1 = |\pi_1| + |\pi_2|$.
– $M = M_1 \parallel M_2$. Then $\Delta = \Delta_1 \otimes \Delta_2$, $\tau = \tau_1 \otimes \tau_2$ and $\alpha = \alpha_1 \parr \alpha_2$, with $\pi_{11} = \Delta_1, x : \tau_1 \vdash M_1 : \alpha_1$ and $\pi_{12} = \Delta_2, x : \tau_2 \vdash M_2 : \alpha_2$, where $|\pi_1| = |\pi_{11}| + |\pi_{12}|$. Also $\pi_2 = \Gamma \vdash V : \tau_1 \otimes \tau_2$, so by Lemma 9, $\Gamma = \Gamma_1 \otimes \Gamma_2$ with $\pi_{21} = \Gamma_1 \vdash V : \tau_1$ and $\pi_{22} = \Gamma_2 \vdash V : \tau_2$, where $|\pi_2| = |\pi_{21}| + |\pi_{22}|$. By the induction hypothesis $\pi_{31} = \Delta_1 \otimes \Gamma_1 \vdash M_1[V/x] : \alpha_1$ and $\pi_{32} = \Delta_2 \otimes \Gamma_2 \vdash M_2[V/x] : \alpha_2$, where $|\pi_{31}| = |\pi_{11}| + |\pi_{21}|$ and $|\pi_{32}| = |\pi_{12}| + |\pi_{22}|$. Hence, by rule $\parallel_I$, $\pi_3 = \Delta_1 \otimes \Gamma_1 \otimes \Delta_2 \otimes \Gamma_2 \vdash M_1[V/x] \parallel M_2[V/x] : \alpha_1 \parr \alpha_2$. Notice that $\Delta_1 \otimes \Gamma_1 \otimes \Delta_2 \otimes \Gamma_2 = \Delta \otimes \Gamma$ and $M_1[V/x] \parallel M_2[V/x] = (M_1 \parallel M_2)[V/x]$. In addition, $|\pi_3| = |\pi_{31}| + |\pi_{32}| = |\pi_{11}| + |\pi_{21}| + |\pi_{12}| + |\pi_{22}| = |\pi_1| + |\pi_2|$. ☐

Theorem 11 (Subject reduction)**.** Let $\pi = \Delta \vdash M : \alpha$.
If $M \to N$ using any but $+$-reductions, then $\pi' = \Delta \vdash N : \alpha$.
If $M \to N_1$ and $M \to N_2$ using $+$-reductions, then either $\pi' = \Delta \vdash N_1 : \alpha$ or $\pi' = \Delta \vdash N_2 : \alpha$.
Moreover, $|\pi'| = |\pi| - 1$

Proof. We proceed by induction on the reduction relation. β_v, $+$ and $\parallel$ reductions are given in the main text of the paper. The remaining cases are the following.

– Let $M_1 + M_2 \to M_1$ and $M_1 + M_2 \to M_2$. Then, the last rule of $\pi = \Delta \vdash M_1 + M_2 : \alpha$, is either $\pi' = \Delta \vdash M_1 : \alpha$ or $\pi' = \Delta \vdash M_2 : \alpha$, with $|\pi'| = |\pi| - 1$.
– Let $(M \parallel N)P \to MP \parallel NP$. The last rule of π is a $\multimap_E$-rule with $k+1$ premises, say $\pi_0 = \Delta_1 \otimes \Delta_2 \vdash M \parallel N : \parr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for $i = 1, \dots, k$, $\pi_i = \Gamma_i \vdash P : \parr_{j=1}^{n_i} \rho_{ij}$, such that $\pi = \Delta_1 \otimes \Delta_2 \otimes \bigotimes_{i=1}^k \Gamma_i \vdash (M \parallel N)P : \parr_{i=1}^k \parr_{j=1}^{n_i} \alpha_{ij}$ and $|\pi| = \sum_{i=0}^k |\pi_i| + (\sum_{i=1}^k 2n_i) - 1$. Moreover, the last rule of π_0 is a $\parallel_I$-rule with two premises, say $\pi_0^1 = \Delta_1 \vdash M : \parr_{i \in S} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and $\pi_0^2 = \Delta_2 \vdash N : \parr_{i \in \bar{S}} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ where $\emptyset \neq S \subsetneq \{1, \dots, k\}$ and $\bar{S} = \{1, \dots, k\} \setminus S$, with $|\pi_0| = |\pi_0^1| + |\pi_0^2|$. By

rule $\multimap_E$, we have $\pi^1 = \Delta_1 \otimes \bigotimes_{i\in S} \Gamma_i \vdash MP : \bigparr_{i\in S} \bigparr_{j=1}^{n_i} \alpha_{ij}$ and $\pi^2 = \Delta_2 \otimes \bigotimes_{i\in \bar{S}} \Gamma_i \vdash NP : \bigparr_{i\in \bar{S}} \bigparr_{j=1}^{n_i} \alpha_{ij}$, where $|\pi^1| = |\pi_0^1| + \sum_{i\in S} |\pi_i| + (\sum_{i\in S} 2n_i) - 1$ and $|\pi^2| = |\pi_0^2| + \sum_{i\in \bar{S}} |\pi_i| + (\sum_{i\in \bar{S}} 2n_i) - 1$. Then by rule $\|_I$, $\pi' = \Delta_1 \otimes \Delta_2 \otimes \bigotimes_{i=1}^k \Gamma_i \vdash MP \parallel NP : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$, with $|\pi'| = |\pi^1| + |\pi^2| = |\pi_0^1| + \sum_{i\in S} |\pi_i| + (\sum_{i\in S} 2n_i) - 1 + |\pi_0^2| + \sum_{i\in \bar{S}} |\pi_i| + (\sum_{i\in \bar{S}} 2n_i) - 1 = \sum_{i=0}^k |\pi_i| + (\sum_{i=1}^k 2n_i) - 2 = |\pi| - 1$.

- Let $M \parallel N \to M' \parallel N$ as a consequence of $M \to M'$. Then we have $\pi = \Delta_1 \otimes \Delta_2 \vdash M \parallel N : \alpha_1 \parr \alpha_2$ with $\pi_1 = \Delta_1 \vdash M : \alpha_1$ and $\pi_2 = \Delta_2 \vdash N : \alpha_2$, and where $|\pi| = |\pi_1| + |\pi_2|$. Cases:
  - Let $M \to M$ using any but $+$-reductions. Then by the induction hypothesis, $\pi_1' = \Delta_1 \vdash M' : \alpha_1$, with $|\pi_1'| = |\pi_1| - 1$. By rule $\|_I$, $\pi' = \Delta_1 \otimes \Delta_2 \vdash M' \parallel N : \alpha_1 \parr \alpha_2$, with $|\pi'| = |\pi_1'| + |\pi_2| = |\pi_1| - 1 + |\pi_2| = |\pi| - 1$.
  - Let $M \to M_1$ and $M \to M_2$ using $+$-reductions, and so $M \parallel N \to M_1 \parallel N$ and $M \parallel N \to M_2 \parallel N$ using $+$-reductions. By the induction hypothesis either $\pi_1' = \Delta_1 \vdash M_1 : \alpha_1$ or $\pi_1' = \Delta_1 \vdash M_2 : \alpha_1$, with $|\pi_1'| = |\pi_1| - 1$. Then by rule $\|_I$ either $\pi' = \Delta_1 \otimes \Delta_2 \vdash M_1 \parallel N : \alpha_1 \parr \alpha_2$ or $\pi' = \Delta_1 \otimes \Delta_2 \vdash M_2 \parallel N : \alpha_1 \parr \alpha_2$, with $|\pi'| = |\pi_1'| + |\pi_2| = |\pi_1| - 1 + |\pi_2| = |\pi| - 1$.
- Let $M \parallel N \to M \parallel N'$ as a consequence of $N \to N'$. Analogous to previous case.
- Let $PQ \to P'Q$, where P is not a parallel composition, as a consequence of $P \to P'$ using any but $+$-reductions. Then $\pi = \Delta \otimes \bigotimes_{i=1}^k \Gamma_i \vdash PQ : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$ with $\pi_0 = \Delta \vdash P : \bigparr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for $i = 1, \dots, k$, $\pi_i = \Gamma_i \vdash Q : \bigparr_{j=1}^{n_i} \rho_{ij}$, where $|\pi| = \sum_{i=0}^k |\pi_i| + (2\sum_{i=1}^k n_i) - 1$. By the induction hypothesis $\pi_0' = \Delta \vdash P' : \bigparr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ with $|\pi_0'| = |\pi_0| - 1$. Then by rule $\multimap_E$, $\pi' = \Delta \otimes \bigotimes_{i=1}^k \Gamma_i \vdash P'Q : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$ with $|\pi'| = |\pi_0'| + \sum_{i=1}^k |\pi_i| + (2\sum_{i=1}^k n_i) - 1 \sum_{i=0}^k |\pi_i| - 1 + (2\sum_{i=1}^k n_i) - 1 = |\pi| - 1$.
- Let $PQ \to P_1Q$ and $PQ \to P_2Q$ as a consequence of $P \to P_1$ and $P \to P_2$ using $+$-reductions. Then $\pi = \Delta \otimes \bigotimes_{j=1}^k \Gamma_j \vdash PQ : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$, with $\pi_0 = \Delta \vdash P : \bigparr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for $i = 1, \dots, k$, $\pi_i = \Gamma_i \vdash Q : \bigparr_{j=1}^{n_i} \rho_{ij}$, so $|\pi| = \sum_{i=0}^k |\pi_i| + (2\sum_{i=1}^k n_i) - 1$. By the induction hypothesis either $\pi_0' = \Delta \vdash P_1 : \bigparr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ or $\pi_0' = \Delta \vdash P_2 : \bigparr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ with $|\pi_0'| = |\pi_0| - 1$. Hence, by rule $\multimap_E$, we have either $\pi' = \Delta \otimes \bigotimes_{i=1}^k \Gamma_j \vdash P_1Q : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$ or $\pi' = \Delta \otimes \bigotimes_{i=1}^k \Gamma_i \vdash P_2Q : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$ where $|\pi'| = |\pi_0'| + \sum_{i=1}^k |\pi_i| + (2\sum_{i=1}^k n_i) - 1 = \sum_{i=0}^k |\pi_i| - 1 + (2\sum_{i=1}^k n_i) - 1 = |\pi| - 1$.
- Let $VP \to VP'$ as a consequence of $P \to P'$ using any but $+$-reductions. Then $\pi = \Delta \otimes \bigotimes_{i=1}^k \Gamma_j \vdash VP : \bigparr_{i=1}^k \bigparr_{j=1}^{n_i} \alpha_{ij}$ with $\pi_0 = \Delta \vdash V : \bigparr_{i=1}^k \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for $i = 1, \dots, k$, $\pi_i = \Gamma_i \vdash P : \bigparr_{j=1}^{n_i} \rho_{ij}$, where $|\pi| = \sum_{i=0}^k |\pi_i| + (2\sum_{i=1}^k n_i) - 1$. However, by Lemma 6, $k = 1$, so

we omit the index i when not needed, and $|\pi| = |\pi_0| + |\pi_1| + 2n - 1$. By the induction hypothesis, $\pi_1' = \Gamma \vdash P' : \bigparr_{j=1}^{n} \rho_j$, with $|\pi_1'| = |\pi_1| - 1$. Then by rule $\multimap_E$, $\pi' = \Delta \otimes \Gamma \vdash VP' : \bigparr_{j=1}^{n} \alpha_j$ where $|\pi'| = |\pi_0| + |\pi_1'| + 2n - 1 = |\pi_0| + |\pi_1| - 1 + 2n - 1 = |\pi| - 1$.

– Let $VP \to VP_1$ and $VP \to VP_2$ as a consequence of $P \to P_1$ and $P \to P_2$ using $+$-reductions. Then $\pi = \Delta \otimes \bigotimes_{i=1}^{k} \Gamma_i \vdash VP : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$, with $\pi_0 = \Delta \vdash V : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\rho_{ij} \multimap \alpha_{ij})$ and for $i = 1, \ldots, k$ $\pi_i = \Gamma_i \vdash P : \bigparr_{j=1}^{n_i} \rho_{ij}$. So, $|\pi| = \sum_{i=0}^{k} |\pi_i| + (2 \sum_{i=1}^{k} n_i) - 1$. However, by Lemma 6, $k = 1$, so we omit the index i when not needed, and $|\pi| = |\pi_0| + |\pi_1| + 2n - 1$. By the induction hypothesis, either $\pi_1' = \Gamma \vdash P_1 : \bigparr_{j=1}^{n} \rho_j$ or $\pi_1' = \Gamma \vdash P_2 : \bigparr_{j=1}^{n} \rho_j$, with $|\pi_1'| = |\pi_1| - 1$. Hence by rule $\multimap_E$, we have either $\pi' = \Delta \otimes \Gamma \vdash VP_1 : \bigparr_{j=1}^{n} \alpha_j$ or $\pi' = \Delta \otimes \Gamma \vdash VP_2 : \bigparr_{j=1}^{n} \alpha_j$, where $|\pi'| = |\pi_0| + |\pi_1'| + 2n - 1 = |\pi_0| + |\pi_1| - 1 + 2n - 1 = |\pi| - 1$. □

A.2 The Proof of Subject Expansion

As mentioned in Section 4.2, to prove that the subject expansion holds, we first need some technical lemmas stating the commutation of abstraction with abstraction, application, non-deterministic choice and parallel composition.

In the following proofs we silently use the fact that the type of a value must be a computational-type (Lemma 6) to simplify our derivation trees.

Lemma 20 (Abstraction commutation). *If $\pi = \Delta \vdash \lambda y.(\lambda x.M)V : \bigotimes_{i=1}^{n} (\rho_i \multimap \alpha_i)$ and $y \notin \mathrm{FV}(V)$, then there exists $\pi' = \Delta \vdash (\lambda x.\lambda y.M)V : \bigotimes_{i=1}^{n} (\rho_i \multimap \alpha_i)$ such that $|\pi'| = |\pi| - n + 1$.*

Proof. If $\Delta \vdash \lambda y.(\lambda x.M)V : \bigotimes_{i=1}^{n} (\rho_i \multimap \alpha_i)$, then we have $\Delta = \bigotimes_{i=1}^{n} (\Delta_i \otimes \Gamma_i)$, with the following π derivation (since $y \notin \mathrm{FV}(V)$):

$$\dfrac{\dfrac{\dfrac{\pi_i = \Delta_i, y : \tau_i, x : \rho_i \vdash M : \alpha_i}{\Delta_i, y : \tau_i \vdash \lambda x.M : \rho_i \multimap \alpha_i}\multimap_I \qquad \pi_i' = \Gamma_i \vdash V : \rho_i}{\Delta_i \otimes \Gamma_i, y : \tau_i \vdash (\lambda x.M)V : \alpha_i}\multimap_E \qquad i = 1, \ldots, n}{\bigotimes_{i=1}^{n} (\Delta_i \otimes \Gamma_i) \vdash \lambda y.(\lambda x.M)V : \bigotimes_{i=1}^{n} (\tau_i \multimap \alpha_i)}\multimap_I$$

So $|\pi| = \sum_{i=1}^{n} (|\pi_i| + |\pi_i'| + 1) = \sum_{i=1}^{n} |\pi_i| + \sum_{i=1}^{n} |\pi_i'| + n$.

Using the same premises π_i and π'_i, we can derive π':

$$\dfrac{\dfrac{\dfrac{\Delta_i, y:\tau_i, x:\rho_i \vdash M:\alpha_i \qquad i=1,\dots,n}{\bigotimes_{i=1}^{n}\Delta_i, x:\bigotimes_{i=1}^{n}\rho_i \vdash \lambda y.M : \bigotimes_{i=1}^{n}(\tau_i \multimap \alpha_i)}\multimap_I}{\bigotimes_{i=1}^{n}\Delta_i \vdash \lambda x.\lambda y.M : (\bigotimes_{i=1}^{n}\rho_i) \multimap \bigotimes_{i=1}^{n}(\tau_i \multimap \alpha_i)}\multimap_I \qquad \dfrac{\Gamma_i \vdash V:\rho_i \qquad i=1,\dots,n}{\bigotimes_{i=1}^{n}\Gamma_i \vdash V : \bigotimes_{i=1}^{n}\rho_i}\text{Lemma 9}}{\bigotimes_{i=1}^{n}\Delta_i \otimes \bigotimes_{i=1}^{n}\Gamma_i \vdash (\lambda x.\lambda y.M)V : \bigotimes_{i=1}^{n}(\tau_i \multimap \alpha_i)}\multimap_E$$

Where $|\pi'| = \sum_{i=1}^{n}|\pi_i| + \sum_{i=1}^{n}|\pi'_i| + 1 = |\pi| - n + 1$.
We conclude since, by commutativity of the tensor, $\bigotimes_{i=1}^{n}\Delta_i \otimes \bigotimes_{i=1}^{n}\Gamma_i = \bigotimes_{i=1}^{n}(\Delta_i \otimes \Gamma_i)$. □

Lemma 21 (Application commutation). *If* $\pi = \Delta \vdash ((\lambda x.M)V)((\lambda x.N)V) : \alpha$, *where the last rule of* π *is a* $\multimap_E$ *rule having* $k+1$ *premises, then there exists* $\pi' = \Delta \vdash (\lambda x.MN)V : \alpha$ *such that* $|\pi'| = |\pi| - k$.

Proof. If $\Delta \vdash ((\lambda x.M)V)((\lambda x.N)V) : \alpha$ then we have that $\Delta = \Delta_1 \otimes \Delta_2 \otimes \bigotimes_{i=1}^{k}(\Gamma_i \otimes \Gamma'_i)$ and $\alpha = \parr_{i=1}^{k}\parr_{j=1}^{n_i}\alpha_{ij}$, with the following derivation π.

$$\dfrac{\pi_1 \qquad \pi_{2i} \qquad i=1,\dots,k}{\Delta_1 \otimes \Delta_2 \otimes \bigotimes_{i=1}^{k}(\Gamma_i \otimes \Gamma'_i) \vdash ((\lambda x.M)V)((\lambda x.N)V) : \parr_{i=1}^{k}\parr_{j=1}^{n_i}\alpha_{ij}}\multimap_E$$

where π_1 is given by:

$$\dfrac{\dfrac{\pi_{11} = \Delta_1, x:\rho_1 \vdash M : \parr_{i=1}^{k}\bigotimes_{j=1}^{n_i}(\tau_{ij} \multimap \alpha_{ij})}{\Delta_1 \vdash \lambda x.M : \rho_1 \multimap \parr_{i=1}^{k}\bigotimes_{j=1}^{n_i}(\tau_{ij} \multimap \alpha_{ij})}\multimap_I \qquad \pi_{12} = \Delta_2 \vdash V:\rho_1}{\Delta_1 \otimes \Delta_2 \vdash (\lambda x.M)V : \parr_{i=1}^{k}\bigotimes_{j=1}^{n_i}(\tau_{ij} \multimap \alpha_{ij})}\multimap_E$$

and π_{2i}, for $i = 1,\dots,k$, is given by:

$$\dfrac{\dfrac{\pi_{2i}^{1} = \Gamma_i, x:\rho_{2i} \vdash N : \parr_{j=1}^{n_i}\tau_{ij}}{\Gamma_i \vdash \lambda x.N : \rho_{2i} \multimap \parr_{j=1}^{n_i}\tau_{ij}}\multimap_I \qquad \pi_{2i}^{2} = \Gamma'_i \vdash V : \rho_{2j}}{\Gamma_i \otimes \Gamma'_i \vdash (\lambda x.N)V : \parr_{j=1}^{n_i}\tau_{ij}}\multimap_E$$

So, $|\pi| = |\pi_1| + \sum_{i=1}^{k} |\pi_{2i}| + (\sum_{i=1}^{k} 2n_i) - 1 = |\pi_{11}| + |\pi_{12}| + 1 + \sum_{i=1}^{k}(|\pi^1_{2i}| + |\pi^2_{2i}| + 1) + (\sum_{i=1}^{k} 2n_i) - 1$.
Using the same premises, we can derive, first π'_1:

$$\dfrac{\dfrac{\pi_{11} \qquad \pi^1_{2i} \quad i = 1, \ldots, k}{\Delta_1 \otimes \bigotimes_{i=1}^{k} \Gamma_i, x : \rho_1 \otimes \bigotimes_{i=1}^{k} \rho_{2i} \vdash MN : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}}\ \multimap_E}{\Delta_1 \otimes \bigotimes_{i=1}^{k} \Gamma_i \vdash \lambda x.MN : (\rho_1 \otimes \bigotimes_{i=1}^{k} \rho_{2i}) \multimap \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}}\ \multimap_I$$

Second, by Lemma 9, we have $\pi'_2 = \Delta_2 \otimes \bigotimes_{i=1}^{k} \Gamma'_i \vdash V : \rho_1 \otimes \bigotimes_{i=1}^{k} \rho_{2i}$. So, we get:

$$\pi' = \dfrac{\pi'_1 \qquad\qquad \pi'_2}{\Delta_1 \otimes \bigotimes_{i=1}^{k} \Gamma_i \otimes \Delta_2 \otimes \bigotimes_{i=1}^{k} \Gamma'_i \vdash (\lambda x.MN)V : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}}\ \multimap_E$$

Hence $|\pi'| = |\pi'_1| + |\pi'_2| + 1 = |\pi_{11}| + \sum_{i=1}^{k} |\pi^1_{2i}| + (\sum_{i=1}^{k} 2n_i) - 1 + \sum_{i=1}^{k} |\pi^2_{2i}| + |\pi_{12}| = |\pi| - k$.
We conclude since $\Delta_1 \otimes \bigotimes_{i=1}^{k} \Gamma_i \otimes \Delta_2 \otimes \bigotimes_{i=1}^{k} \Gamma'_i = \Delta_1 \otimes \Delta_2 \otimes \bigotimes_{i=1}^{k}(\Gamma_i \otimes \Gamma'_i)$. □

Lemma 22 (Sum commutation). *If $\pi = \Delta \vdash (\lambda x.M_1)V + (\lambda x.M_2)V : \alpha$, then there exists $\pi' = \Delta \vdash (\lambda x.(M_1 + M_2))V : \alpha$ such that $|\pi'| = |\pi|$.*

Proof. If $\Delta \vdash (\lambda x.M_1)V + (\lambda x.M_2)V : \alpha$ then we have the following π derivation (for $\Delta = \Delta' \otimes \Gamma$)

$$\dfrac{\dfrac{\dfrac{\pi_1 = \Delta', x : \tau \vdash M_i : \alpha}{\Delta' \vdash \lambda x.M_i : \tau \multimap \alpha}\ \multimap_I \qquad \pi_2 = \Gamma \vdash V : \tau}{\Delta' \otimes \Gamma \vdash (\lambda x.M_i)V : \alpha}\ \multimap_E \qquad i = 1 \text{ or } 2}{\Delta' \otimes \Gamma \vdash (\lambda x.M_1)V + (\lambda x.M_2)V : \alpha}\ +_\ell \text{ or } +_r$$

Then $|\pi| = |\pi_1| + |\pi_2| + 2$.
So, using the same premises, we can derive π' as follows

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Delta', x : \tau \vdash M_1 + M_2 : \alpha}\ +_\ell \text{ or } +_r}{\Delta' \vdash \lambda x.(M_1 + M_2) : \tau \multimap \alpha}\ \multimap_I \qquad \pi_2}{\Delta'_1 \otimes \Gamma \vdash (\lambda x.(M_1 + M_2))V : \alpha}\ \multimap_E$$

with $|\pi'| = |\pi|$. □

Lemma 23 (Parallel commutation). *If $\pi = \Delta \vdash (\lambda x.M_1)V \parallel (\lambda x.M_2)V : \alpha$, then there exists $\pi' = \Delta \vdash (\lambda x.(M_1 \parallel M_2))V : \alpha$ such that $|\pi'| = |\pi| - 1$.*

Proof. If $\Delta \vdash (\lambda x.M_1)V \parallel (\lambda x.M_2)V : \alpha$ then we have $\Delta = \Delta_1 \otimes \Gamma_1 \otimes \Delta_2 \otimes \Gamma_2$ and $\alpha = \alpha_1 \parr \alpha_2$, with the following π derivation.

$$\dfrac{\dfrac{\dfrac{\pi_{i1} = \Delta_i, x : \tau_i \vdash M_i : \alpha_i}{\Delta_h \vdash \lambda x.M_i : \tau_i \multimap \alpha_i}\multimap_I \qquad \pi_{i2} = \Gamma_i \vdash V : \tau_i}{\Delta_i \otimes \Gamma_i \vdash (\lambda x.M_i)V : \alpha_i}\multimap_E \qquad i = 1,2}{\Delta_1 \otimes \Gamma_1 \otimes \Delta_2 \otimes \Gamma_2 \vdash (\lambda x.M_1)V \parallel (\lambda x.M_2)V : \alpha_1 \parr \alpha_2}\parallel_I$$

$|\pi| = |\pi_{11}| + |\pi_{12}| + 1 + |\pi_{21}| + |\pi_{22}| + 1$
So, using the same premises, we can derive π' as follows

$$\dfrac{\dfrac{\dfrac{\pi_{11} \qquad \pi_{21}}{\Delta_1 \otimes \Delta_2, x : \tau_1 \otimes \tau_2 \vdash M_1 \parallel M_2 : \alpha_1 \parr \alpha_2}\parallel_I}{\Delta_1 \otimes \Delta_2 \vdash \lambda x.(M_1 \parallel M_2) : \tau_1 \otimes \tau_2 \multimap \alpha_1 \parr \alpha_2}\multimap_I \qquad \dfrac{\pi_{12} \qquad \pi_{22}}{\Gamma_1 \otimes \Gamma_2 \vdash V : \tau_1 \otimes \tau_2}\text{Lemma 9}}{\Delta_1 \otimes \Delta_2 \otimes \Gamma_1 \otimes \Gamma_2 \vdash (\lambda x.(M_1 \parallel M_2))V : \alpha_1 \parr \alpha_2}\multimap_E$$

Hence $|\pi'| = |\pi_{11}| + |\pi_{21}| + |\pi_{12}| + |\pi_{22}| + 1 = |\pi| - 1$.
We conclude as $\Delta_1 \otimes \Gamma_1 \otimes \Delta_2 \otimes \Gamma_2 = \Delta_1 \otimes \Delta_2 \otimes \Gamma_1 \otimes \Gamma_2$. □

Theorem 12 (Subject expansion)**.** If $M \to N$ and $\pi = \Delta \vdash N : \alpha$, then there is $\pi' = \Delta \vdash M : \alpha$, such that $|\pi'| = |\pi| + 1$.

Proof. We proceed by induction on the length of the derivation of $M \to N$. We split into cases, depending on its last rule.

– $(\lambda x.M')V \to M'[V/x]$. We proceed by structural induction on M'.
  - $M' = x$. Then $x[V/x] = V$ and $\pi = \Delta \vdash V : \alpha$ where, by Lemma 6, α is a computational-type. Since $\vdash \lambda x.x : \alpha \multimap \alpha$ can be inferred from a derivation of measure 0, we can define $\pi' = \Delta \vdash (\lambda x.x)V : \alpha$ as a rule $\multimap_E$ with the derivation of $\vdash \lambda x.x : \alpha \multimap \alpha$ and π as premises. Notice that $|\pi'| = |\pi| + 1$.
  - $M' = y$ for some $y \neq x$. Then $y[V/x] = y$ and $\Delta = y : \alpha$. Notice that in this case π is an *ax* rule, so its measure is 0. Now, $y : \alpha \vdash \lambda x.y : \mathbf{1} \multimap \alpha$ and, since $\vdash V : \mathbf{1}$ by Remark 5, we derive $\pi' = y : \alpha \vdash (\lambda x.y)V : \alpha$ using the rule $\multimap_E$. Remark $|\pi'| = 1$.
  - $M' = \lambda y.P$ for some $y \neq x$. Then $N = \lambda y.P[V/x]$ is also an abstraction and so $\pi = \Delta \vdash N : \alpha$ ends in a $\multimap_I$-rule with $n \geq 0$ premises $\pi_i = \Delta_i, y : \tau_i \vdash P[V/x] : \alpha_i$, for $i = 1, \dots, n$ and $\alpha = \bigotimes_{i=1}^n (\tau_i \multimap \alpha_i)$, $\Delta = \bigotimes_{i=1}^n \Delta_i$ and $|\pi| = \sum_i |\pi_i|$. By the induction hypothesis (in case $n = 0$ we do not need this passage), we get $\pi_i' = \Delta_i, y : \tau_i \vdash (\lambda x.P)V : \alpha_i$, with $|\pi_i'| = |\pi_i| + 1$, for each i. By rule $\multimap_I$, we derive $\pi'' = \bigotimes_{i=1}^n \Delta_i \vdash \lambda y.((\lambda x.P)V) : \bigotimes_{i=1}^n (\tau_i \multimap \alpha_i)$, such that $|\pi''| = |\pi| + n$ Finally, by Lemma 20, we obtain $\pi' = \bigotimes_{i=1}^n \Delta_i \vdash (\lambda x.\lambda y.P)V : \bigotimes_{i=1}^n (\tau_i \multimap \alpha_i)$ with $|\pi'| = |\pi''| - n + 1 = |\pi| + 1$.

- $M' = PQ$. By definition we have $N = (PQ)[V/x] = P[V/x]Q[V/x]$. So, $\pi = \Delta \vdash N : \alpha$ ends in a $\multimap_E$-rule with $k+1$ premises $\pi_0 = \Delta' \vdash P[V/x] : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\tau_{ij} \multimap \alpha_{ij})$ and $\pi_i = \Gamma_i \vdash Q[V/x] : \bigparr_{j=1}^{n_i} \tau_{ij}$ for $i = 1, \dots, k$, with $\Delta = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i$, $\alpha = \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$ and $|\pi| = \sum_{i=0}^{k} \pi_i + (\sum_{i=1}^{k} 2n_i) - 1$. Then, by the induction hypothesis, we get $\pi_0' = \Delta' \vdash (\lambda x.P)V : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i} (\tau_{ij} \multimap \alpha_{ij})$, and $\pi_i' = \Gamma_i \vdash (\lambda x.Q)V : \bigparr_{j=1}^{n_i} \tau_{ij}$, with $|\pi_i'| = |\pi_i| + 1$. Hence by rule $\multimap_E$ we obtain $\pi'' = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i \vdash ((\lambda x.P)V)((\lambda x.Q)V) : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$, with $|\pi''| = \sum_{i=0}^{k} |\pi_i'| + (\sum_{i=1}^{k} 2n_i) - 1$. By Lemma 21, we get $\pi' = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i \vdash (\lambda x.PQ)V : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$ such that $|\pi'| = |\pi''| - k = |\pi| + 1$.
- $M' = P + Q$. Now, by definition we have $N = (P+Q)[V/x] = P[V/x] + Q[V/x]$. Then either $\pi_1 = \Delta \vdash P[V/x] : \alpha$ or $\pi_1 = \Delta \vdash Q[V/x] : \alpha$, with $|\pi_1| = |\pi| - 1$. Hence by the induction hypothesis, there exists either $\pi_1' = \Delta \vdash (\lambda x.P)V : \alpha$ or $\pi_1' = \Delta \vdash (\lambda x.Q)V : \alpha$ with $|\pi_1'| = |\pi_1| + 1$. In both cases, either by rule $+_\ell$ or $+_r$, we get $\pi_2 = \Delta \vdash (\lambda x.P)V + (\lambda x.Q)V : \alpha$, which entails, by Lemma 22, $\pi' = \Delta \vdash (\lambda x.(P+Q))V : \alpha$ with $|\pi'| = |\pi| + 1$.
- $M' = P \parallel Q$. By definition $N = (P \parallel Q)[V/x] = P[V/x] \parallel Q[V/x]$, so $\pi = \Delta \vdash N : \alpha$ ends in a $\parallel_I$ rule with premises $\pi_1 = \Delta_1 \vdash P[V/x] : \alpha_1$ and $\pi_2 = \Delta_2 \vdash Q[V/x] : \alpha_2$, with $\Delta = \Delta_1 \otimes \Delta_2$, $\alpha = \alpha_1 \parr \alpha_2$ and $|\pi| = |\pi_1| + |\pi_2|$. By the induction hypothesis, we get $\pi_1' = \Delta_1 \vdash (\lambda x.P)V : \alpha_1$ and $\pi_2' = \Delta_2 \vdash (\lambda x.Q)V : \alpha_2$. Therefore, by applying the rule $\parallel_I$ to π_1' and π_2', we derive $\pi_3 = \Delta_1 \otimes \Delta_2 \vdash (\lambda x.P)V \parallel (\lambda x.Q)V : \alpha_1 \parr \alpha_2$ such that $|\pi_3| = |\pi_1'| + |\pi_2'| = |\pi_1| + 1 + |\pi_2| + 1 = |\pi| + 2$. From Lemma 23, we get $\pi' = \Delta_1 \otimes \Delta_2 \vdash (\lambda x.(P \parallel Q))V : \alpha_1 \parr \alpha_2$ with $|\pi'| = |\pi| + 1$.

– $P+Q \to P$, with $\Delta \vdash P : \alpha$. Then, by rule $+_\ell$ we get $\Delta \vdash P+Q : \alpha$. Checking the measure is trivial. Symmetrically we deduce the case $P + Q \to Q$

– $(Q_1 \parallel Q_2)P \to Q_1P \parallel Q_2P$. Then, $\pi = \Delta \vdash N : \alpha$ ends in a rule $\parallel_I$ with two premises $\pi_1 = \Delta_1 \vdash Q_1P : \alpha_1$ and $\pi_2 = \Delta_2 \vdash Q_2P : \alpha_2$, such that $\Delta = \Delta_1 \otimes \Delta_2$, $\alpha = \alpha_1 \parr \alpha_2$ and $|\pi| = |\pi_1| + |\pi_2|$. Moreover, the last rule of π_h, for $h = 1, 2$, is a $\multimap_E$ rule with $k_h + 1$ premises, say $\pi_{h0} = \Gamma_{h0} \vdash Q_h : \bigparr_{i=1}^{k_h} \bigotimes_{j=1}^{n_{h1}} (\tau_{hij} \multimap \alpha_{hij})$, for every $i = 1, \dots, k_h$, $\pi_{hi} = \Gamma_{hi} \vdash P : \bigparr_{j=1}^{n_{hi}} \tau_{hij}$, where $\Delta_h = \bigotimes_{i=0}^{k_h} \Gamma_{hi}$, $\alpha_h = \bigparr_{i=1}^{k_h} \bigparr_{j=1}^{n_{hi}} \alpha_{hij}$. Notice that $|\pi_h| = \sum_{i=0}^{k_h} |\pi_{hi}| + (\sum_{i=1}^{k_h} 2n_{hi}) - 1$.
By applying the rule $\parallel_I$ to π_{10} and π_{20}, we get a derivation $\pi_{30} = \Gamma_{10} \otimes \Gamma_{20} \vdash Q_1 \parallel Q_2 : (\bigparr_{i=1}^{k_1} \bigotimes_{j=1}^{n_{1i}} (\tau_{1ij} \multimap \alpha_{1ij})) \parr (\bigparr_{i=1}^{k_2} \bigotimes_{j=1}^{n_{2i}} (\tau_{2ij} \multimap \alpha_{2ij}))$. Therefore, by rule $\multimap_E$, we have the derivation $\pi' = \Gamma_{10} \otimes \Gamma_{20} \otimes (\bigotimes_{i=1}^{k_1} \Gamma_{1i} \otimes \bigotimes_{i=1}^{k_2} \Gamma_{2i}) \vdash (Q_1 \parallel Q_2)P : (\bigparr_{i=1}^{k_1} \bigparr_{j=1}^{n_{1i}} \alpha_{1ij}) \parr (\bigparr_{i=1}^{k_2} \bigparr_{j=1}^{n_{2i}} \alpha_{2ij})$. Notice that $|\pi'| = \sum_{i=0}^{k_1} |\pi_{1i}| + \sum_{i=0}^{k_2} |\pi_{2i}| + (\sum_{i=0}^{k_1} 2n_i + \sum_{i=0}^{k_2} 2n_i) - 1 = |\pi| - 1$.

– $V(P_1 \parallel P_2) \to VP_1 \parallel VP_2$. Then, $\pi = \Delta \vdash N : \alpha$ ends in a rule $\parallel_I$ with two premises $\pi_1 = \Delta_1 \vdash VP_1 : \alpha_1$ and $\pi_2 = \Delta_2 \vdash VP_2 : \alpha_2$, such that $\Delta = \Delta_1 \otimes \Delta_2$, $\alpha = \alpha_1 \parr \alpha_2$ and $|\pi| = |\pi_1| + |\pi_2|$. As in the previous case, for $h = 1, 2$, π_h ends is a $\multimap_E$ rule. Since V is a value, it can have only a computational type (Lemma 6), and so the last $\multimap_E$ rule of π_h has exactly two premises, say $\pi_{h0} = \Gamma_{h0} \vdash V : \bigotimes_{j=1}^{n_h}(\tau_{hj} \multimap \alpha_{hj})$ and $\pi_{h1} = \Gamma_{h1} \vdash P_h : \bigparr_{j=1}^{n_h} \tau_{hj}$, where $\Delta_h = \Gamma_{h0} \otimes \Gamma_{h1}$, $\alpha_h = \bigparr_{j=1}^{n_h} \alpha_{hj}$, and $|\pi_h| = |\pi_{h0}| + |\pi_{h1}| + 2n_h - 1$. The derivation π' is obtained in three steps. First, we apply the rule $\|_I$ to π_{11} and π_{21}, getting a derivation of $\Gamma_{11} \otimes \Gamma_{21} \vdash P_1 \parallel P_2 : (\bigparr_{j=1}^{n_1} \tau_{1j}) \parr (\bigparr_{j=1}^{n_2} \tau_{2j})$. Then, by Lemma 9, we get $\Gamma_{10} \otimes \Gamma_{20} \vdash V : (\bigotimes_{j=1}^{n_1}(\tau_{1j} \multimap \alpha_{1j})) \otimes (\bigotimes_{j=1}^{n_2}(\tau_{2j} \multimap \alpha_{2j}))$. Finally, we achieve π' by applying a rule $\multimap_E$ to the previous two. Notice that $|\pi'| = |\pi_{10}| + |\pi_{20}| + |\pi_{11}| + |\pi_{22}| + 2(n_1 + n_2) - 1 = |\pi| + 1$.

- $P \parallel Q \to P' \parallel Q$ as a consequence of $P \to P'$. Then, $\Delta = \Delta_1 \otimes \Delta_2$ and $\alpha = \alpha_1 \parr \alpha_2$, with $\pi_1 = \Delta_1 \vdash P' : \alpha_1$ and $\pi_2 = \Delta_2 \vdash Q : \alpha_2$, where $|\pi| = |\pi_1| + |\pi_2|$. By the induction hypothesis we get $\pi_1' = \Delta_1 \vdash P : \alpha_1$, with $|\pi_1'| = |\pi_1| + 1$ and by rule $\|_I$ we derive $\pi' = \Delta_1 \otimes \Delta_2 \vdash P \parallel Q : \alpha_1 \parr \alpha_2$ with $|\pi'| = |\pi_1'| + |\pi_2| = |\pi_1| + 1 + |\pi_2| = |\pi| + 1$. The case $P \parallel Q \to P \parallel Q'$ is obtained by a symmetrical reasoning.
- $PQ \to P'Q$, where P is not a parallel composition, as a consequence of $P \to P'$. Then $\Delta = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i$ and $\alpha = \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$, where $\pi_0 = \Delta' \vdash P' : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i}(\tau_{ij} \multimap \alpha_{ij})$, and for $i = 1, \dots, k$ we have $\pi_1 = \Gamma_i \vdash Q : \bigparr_{j=1}^{n_i} \tau_{ij}$, where $|\pi| = \sum_{i=0}^{k} |\pi_i| + (\sum_{i=1}^{k} 2n_i) - 1$. By the induction hypothesis we get $\pi_0' = \Delta' \vdash P : \bigparr_{i=1}^{k} \bigotimes_{j=1}^{n_i}(\tau_{ij} \multimap \alpha_{ij})$ with $|\pi_0'| = |\pi_0| + 1$. Hence by rule $\multimap_E$ we conclude $\pi' = \Delta' \otimes \bigotimes_{i=1}^{k} \Gamma_i \vdash PQ : \bigparr_{i=1}^{k} \bigparr_{j=1}^{n_i} \alpha_{ij}$, with $|\pi'| = |\pi_0'| + \sum_{i=1}^{k} |\pi_i| + (\sum_{i=1}^{k} 2n_i) - 1 = \sum_{i=0}^{k} |\pi_i| + (\sum_{i=1}^{k} 2n_i) = |\pi| + 1$.
- $VP \to VP'$, where P is not a parallel composition, as a consequence of $P \to P'$. Then we have $\Delta = \Delta' \otimes \Gamma$ and $\alpha = \bigparr_{i=1}^{n} \alpha_i$, where (using Lemma 6) $\pi_0 = \Delta' \vdash V : \bigotimes_{i=1}^{n}(\tau_i \multimap \alpha_i)$ and $\pi_1 = \Gamma \vdash P' : \bigparr_{i=1}^{n} \tau_i$, with $|\pi| = |\pi_0| + |\pi_1| + 2n - 1$. Now, by the induction hypothesis, we get $\pi_1' = \Gamma \vdash P : \bigparr_{i=1}^{n} \tau_i$ with $|\pi_1'| = |\pi_1| + 1$, and hence by rule $\multimap_E$, we conclude $\pi' = \Delta' \otimes \Gamma \vdash VP : \bigparr_{i=1}^{n} \alpha_i$, where $|\pi'| = |\pi_0| + |\pi_1'| + 2n - 1 = |\pi_0| + |\pi_1| + 2n = |\pi| + 1$. □