An Abstract Notion of Application[1]

Pietro Di Gianantonio, Furio Honsell

Dipartimento di Matematica e Informatica, Università di Udine,
via Zanon 6, I-33100 Udine, ITALY
e-mail: pietro@udmi5400.cineca.it, honsell@uduniv.cineca.it

Many concrete notions of function application, suitable for interpreting typed lambda calculi with recursive types, have been introduced in the literature. These arise in different fields such as set theory, multiset theory, type theory and functor theory and are apparently unrelated. In this paper we introduce the general concept of *applicative exponential structure* and show that it subsumes all these notions. Our approach is based on a generalization of the notion of *intersection type*. We construe all these structures in a finitary way, so as to be able to utilize uniformly a general form of type assignment system for defining the interpretation function. Applicative exponential structures are just combinatory algebras, in general. Our approach suggests a wide variety of entirely new concrete notions of function application; e.g. in connection with boolean sets. Applicative exponential structures can be used for modeling various forms of non-deterministic operators.

1 Introduction

Various natural concrete models of the notion of function application arising in λ-calculus have been discovered since the early seventies. G.Plotkin [13], inspired by earlier work of Scott, was the first to define a set-theoretical notion of application. By means of it he built a set theoretical model for untyped λ-calculus. Since then, various other natural notions of application were discovered by Scott [15] and Engeler [7] in set theory, by Coppo, Dezani and Venneri [5, 3] in type theory and by Girard [8], Ore [12] and Lamarche [11] in functor theory, in the theory of multisets and in analytic function theory. All these concrete notions of application give rise to concrete structures, albeit not always categories, which can be used as domains for denotational semantics. More precisely these structures are rich enough to model the behavior of application in typed λ-calculi with recursive types and appropriate constructors, destructors and fixed point operators.

These notions of application, although apparently different, seem to share a common pattern. In this paper we try to capture this pattern by introducing a notion of algebraic structure, termed *applicative exponential structure,* which we show to be general enough subsume all the concrete notions mentioned earlier. In particular we define a general framework in which one can easily and uniformly express all classical constructions. One of the key features of our approach is the use of a generalized notion of type, inspired by that of "intersection type" [3], for providing a finitary description of the structures under consideration. This analysis allows for the use of a uniform kind of type assignment system for defining the interpretation of the λ-calculus language. Intuitively types are understood as finite elements of the domain, possibly having some coefficients; and a term has a given type if its denotation is approximated by a given type. Special care has to be taken in order to deal with the coefficients. This is a particularly interesting way of presenting the interpretation function since, besides being finitary in nature, it constitutes an endogenous logic in the sense of Abramsky [1]. Moreover it can provide a proof theoretic analysis of the fine structure of the models. This technique was initially introduced for the study of filter models [4] but was later applied to Girard's qualitative domains in coherent semantics [10] and quantitative domains [6].

[1] Paper accepted at the International Conference on Typed Lambda Calculus and Applications, March 1993 Utrecht (The Netherlands)

We think that our approach is successful and fruitful since, besides illuminating on the idea underlying so many apparently unrelated notions of function application, it suggests also a wide variety of new concrete alternatives. Particularly appealing and potentially interesting is the notion of application which arises in connection with boolean sets. This notion yields a sort of "boolean valued" model of the λ-calculus. It is closely related to that which arises if we carry out Plotkin's original construction in a Boolean-valued model of set theory. This kind of construction can prove to be quite interesting for modeling programming languages which feature non-deterministic operators. Moreover, it seems to open a new area of applications of model theoretic concepts to the semantics of programming languages.

For simplicity we shall not deal in this paper with the whole language of typed λ-calculus with reflexive types as in [14] or [12]. We will discuss only the case of the untyped λ-calculus language, as an interesting and important example of a reflexive type. All the results can be extended with little difficulty to the more general case.

Somehow unexpectedly, the abstract structure introduced in the paper, and many of the concrete examples given, do not model λ-calculus in the strongest possible way. In general the, so called, ξ-rule fails and these are only combinatory algebras and not lambda models nor lambda algebras. Surprisingly enough the general construction does not seem to be amenable to a simple categorical presentation, unless further equivalence relations are superimposed. These will be discussed in a forthcoming paper.

A final remark is in order. We could have presented these results following more closely the approach of Girard[8] and Lamarche [11]. This would amount to use as main source of inspiration the notion of analytic function in complex analysis. No substantial difference would arise. The approach via analytic functions is in fact "dual" to the one used here. In this paper we will only describe very briefly this alternative approach in Appendix A.

The paper is organized as follows. In section 2 we define the structures normally utilized for modeling λ-calculus using a style which focuses on the properties of the interpretation function. In section 3 we present some classical and new constructions of concrete models of the lambda calculus and gradually introduce our general framework. In section 4 we give the definition of applicative exponential structure and prove the main theorem of this paper, i.e.: applicative exponential structures are combinatory algebras. Finally in section 5 we give more examples of concrete applicative exponential structures yet uninvestigated and we outline a possible use of applicative exponential structures for modeling non-deterministic operators.

Finally the authors would like to gratefully acknowledge Fabio Alessi and Simona Ronchi della Rocca for helpful discussions in the early stages of this work.

2 Combinatory Structures

Throughout the paper we assume the reader familiar with standard notions and notations in Lambda Calculus and Combinatory Logic as in [2]. Several different applicative structures, i.e. structures with a binary operation defined on them, have been introduced in the literature for interpreting the language of λ-calculus: combinatory algebras, lambda algebras and lambda models. These differ by the strength of the equalities which they enforce on interpretations of λ-terms. Usually combinatory algebras are defined without any reference to the interpretation function using the standard combinators S, K and I. Contrary to this tradition will define uniformly all these structures in the style of [10]. By so doing the connection with type assignment systems in the sequel will be clearer.

The language Λ of *λ-calculus* is defined as usual by: $M ::= x \mid MN \mid \lambda x.M$. Terms which do not have abstracted subterms will be called *applicative terms.*

Definition 1 (à la Hindley Longo)
1) An applicative structure A, is an algebra $\langle A : Set, \circ : A \times A \to A \rangle$;
2) An environment is a function $\rho : Var \to A$. The set of environments is denoted by Env.
3)An interpretation of Λ is a function $[\![\ ]\!]: Env \to (\Lambda \to A)$. As usual $\xi[x/a]$ denotes the environment defined as $\xi[x/a](x)=a$ and $\xi[x/a](y)=\xi(y)$ if $x \neq y$.

4) A *combinatory structure*, c.s. for short, is a pair $\langle A, [\![\ ]\!]: Env \to (\Lambda \to A)\rangle$ consisting of an applicative structure and an interpretation function defined over it;
5) A *lambda model* is a c.s. where the interpretation function satisfies the following properties:

i) $[\![x]\!]_\rho = \rho(x)$
ii) $[\![M N]\!]_\rho = [\![M]\!]_\rho \circ [\![N]\!]_\rho$
iii) $[\![\lambda x.M]\!]_\rho \circ a = [\![M]\!]_{\rho[x/a]}$
iv) $[\![\lambda x.M]\!]_\rho = [\![\lambda y.([x/y]M)]\!]_\rho$ provided $y \notin FV(M)$
v) $[\![M]\!]_\rho = [\![M]\!]_\xi$ provided $\rho(x) = \xi(x)$ for $x \in FV(M)$
vi) $(\forall a\ .\ [\![M]\!]_{\rho[x/a]} = [\![N]\!]_{\rho[x/a]}) \Rightarrow [\![\lambda x.M]\!]_\rho = [\![\lambda x.N]\!]_\rho$

6) A *lambda algebra* is a c.s. where the interpretation function satisfies conditions i) - v) above and also the following rule:
(ξ) $\vdash_\lambda M = N \Rightarrow [\![\lambda x.M]\!]_\rho = [\![\lambda x.N]\!]_\rho$
where $\vdash_\lambda$ denotes derivability in the theory of λ-calculus.
7) A *combinatory algebra* is a c.s. where the interpretation function satisfies only conditions i)-v) above. Δ

It is well known that lambda models are lambda algebras and lambda algebras are combinatory algebras. The following proposition illustrates the importance of combinatory algebras and establishes the equivalence of our definition with the usual ones.

Proposition 1. Let $\langle A, [\![ ]\!]\rangle$ be a combinatory structure. The following properties are equivalent:
a) A can be extended to a combinatory algebra $\langle A, [\![ ]\!]'\rangle$;
b) There exist distinguished constants K and S in A such that for all constants a,b,c$\in$A :
$((K \circ a) \circ b) = a$ and $(((S \circ a) \circ b) \circ c) = ((a \circ c) \circ (b \circ c))$;
c) The interpretation function $[\![ ]\!]: Env \to (\Lambda \to A)$ satisfies conditions i) and ii) in Definition 1, and moreover for all applicative term $M \in \Lambda$ such that $FV(M) \subseteq \{x_1,...,x_n\}$, there is a constant c, such that $[\![M]\!]_\xi = [\![cx1...xn]\!]_\xi$, for $x \in Env$. This property is usually called *functional completeness*.

Proof. Standard Δ

3 Concrete Models of Application

In this section we present some classical constructions of concrete models of the untyped λ-calculus language, some alternative presentations of these and some entirely new models. As remarked in the introduction any of these could be turned into a full-fledged domain structure for denotational semantics, but for lack of space we shall not do it here.

Perhaps the best known example of a natural concrete model of lambda calculus is the Plotkin-Engeler set theoretical model see [13, 7]. This model construction is closely related to the Filter Model constructions in [3, 5, 4], where arbitrary sets are replaced by particular ones called filters. Interestingly enough, Plotkin-Engeler Model and the Filter Model are indeed lambda models, the model introduced in [5], on the other hand, is only a lambda algebra. Nonetheless this latter structure, which we call Intersection Algebra, is quite remarkable since it is the first example of a lambda algebra which has not been defined by purely syntactical means.

In the literature the applicative structure underlying Plotkin-Engeler set theoretical model is defined as follows:

Definition 2. The applicative structure $\langle \mathbb{B}, \circ \rangle$ is defined inductively by:
B_0 is an arbitrary set of atoms,
$B_{n+1} \equiv B_0 \cup \{ (\beta, b) \mid \beta \subseteq B_n, \beta \text{ finite}, b \in B_n \}$
$\mathbb{B} \equiv \wp(\bigcup_n B_n)$

given $U,V \in \mathcal{B}$ we put $U \circ V \equiv \{ b \mid (\beta,b) \in U, \beta \subseteq V\}$. Δ

We now give an alternative presentation of the above structure. This will be the first example of the standard format which will be used throughout the paper for presenting concrete applicative structures. The definition of the general notion of applicative exponential structure in Section 4 will build upon the shape of this format. The original presentation of Engeler's model, i.e.Definition 2, was given just for introductory purposes.

We need first some notation. Given a set A and a set B we denote by $[A \to B]$ the set of functions from A to B. If B contains a distinguished point 0, we denote by $[A \xrightarrow{\circ} B]$ the set of functions from A to B with value almost everywhere 0. As will become clear in the sequel these particular functions are introduced essentially as a useful "trick" for encoding functions with a finite domain, without having to bother about issues of definedness. According to this intended meaning, given $r \in [A \xrightarrow{\circ} \mathcal{P}B]$, dom(r) will denote the set $\{a \mid a \in A, r(a) \neq 0\}$; and the term $\{a_1:x_1,...,a_n:x_n\}$ will denote the function $s:A \to B$ defined by

$$s(a) \equiv \begin{cases} x_i & \text{if } a=a_i \text{ for some } i\ 1 \le i \le n; \\ 0 & \text{otherwise} \end{cases}$$

Finally $\mathbf{2}$ will denote the boolean algebra of truth-values where "false" is taken to be 0 and "true" to be 1.

Definition 3. Let $J: \mathbb{N} \to \mathrm{Set}$ be inductively defined by:

$J(0) \equiv J_0$, an arbitrary set of atoms, $J(n+1) \equiv J(0) \cup ([J(n) \xrightarrow{\circ} \mathbf{2}] \times J(n))$.

Now put $\mathbf{J} \equiv \bigcup_n J(n)$ and $\mathcal{J} \equiv [\mathbf{J} \to \mathbf{2}]$.

Let $\langle \mathcal{J}, \circ \rangle$ be the applicative structure where application is defined by:

$$(f \circ g)(j) \equiv \sum_{r \in [\mathbf{J} \xrightarrow{\circ} \mathbf{2}]} f((r,j)) \times \prod_{k \in dom(r)} (r(k) \Rightarrow g(k))$$

where $f,g \in \mathcal{J}$, $j \in \mathbf{J}$, $\Rightarrow$ is logical implication, + is disjunction and $\times$ is conjunction . Δ

Notice that any function $r \in [A \xrightarrow{\circ} \mathbf{2}]$ is indeed the characteristic function of a finite subset of A, just as any function $f : A \to \mathbf{2}$ is the characteristic function of an arbitrary subset of A.

The expression $\prod_{k \in dom(r)} (r(k) \Rightarrow g(k))$, used above, is therefore true, i.e. equal to 1 if and only if the finite set represented by r is a subset of the set represented by g.

What we have done in this new presentation amounts to substituting subsets with their characteristic functions. It is now easy to show that $\langle \mathcal{B}, \circ \rangle$ and $\langle \mathcal{J}, \circ \rangle$ are isomorphic.

Filter Models and Intersection Algebras can be accounted for similarly as follows. The structure $\langle \mathcal{J}, \circ \rangle$ is precisely the applicative structure underlying the Intersection Algebra in [5]. The applicative structure underlying the filter model in [3], instead, is obtained by taking $\mathcal{J}$ in Definition 3 to be the set of only those subsets of $\mathbf{J}$ which are *filters*. Elements of $\mathbf{J}$ behave in fact like intersection types. A filter f is a subset of $\mathbf{J}$ which is upwards closed under the order relation $\le$ induced by the following rule:

$$\frac{j \le j' \quad \forall k \in dom(r).\ \exists k' \in dom(r').\ k' \le k}{(r,j) \le (r',j')}$$

We are now ready to turn these applicative structures into combinatory structures. As remarked in the introduction, we will utilize throughout the paper,*type assignment systems* in the sense of [3,4] to define the interpretation function. This is made possible because elements of $\mathbf{J}$ behave as a generalized "intersection types" [3,4]. In general we construe type assignment systems as formal systems for establishing assignment judgements of the form $\beta \vdash M:j$ where $M \in Term$, $j \in \mathbf{J}$ and β is a (multi)set of assumptions of the shape $x:j'$.

Apart from the particular choice of the set **J**, the type assignment systems in the paper will vary greatly in the structural rules assumed in the formal system. The intended meaning of the judgement $\beta \vdash M:j$ will be made formally precise case by case, but it will always mean something of the kind "under the assumptions recorded in β the interpretation of M depends on the type j". This is both an "endogenous logic" and a "logical" presentation or "finitary" presentation of the structure, in the sense of Abramsky, [1].

In order to define the interpretation function in the Plotkin-Engeler set theoretical model, in the Intersection Algebra and in the Filter Model we shall utilize judgements of the form $\beta \vdash M:j$ where the bases β are sets i.e. functions $\beta \in [(Var \times \mathbf{J}) \stackrel{\circ}{\to} \mathbf{2}]$.

Definition 4.
1) Consider the following set of rules:

(axiom) $$\frac{}{\{(x,j):1\} \vdash x:j}$$ provided $j \in \mathbf{J}$

(abstraction) $$\frac{\beta + \{(x,j_1):1,\ldots,(x,j_k):1\} \vdash M:j}{\beta \vdash \lambda x.M:(\{j_1:1,\ldots,j_k:1\},j)}$$ provided $\forall j\ \beta((x,j)) = 0$

(application) $$\frac{\beta \vdash M:(\{j_1:1,\ldots,j_k:1\},j) \quad \beta_1 \vdash N:j_1 \quad \ldots \quad \beta_k \vdash N:j_k}{\beta + \sum_{1 \le i \le k} \beta_i \vdash MN:j}$$

(weakening) $$\frac{\beta \vdash M:j}{\beta + \beta' \vdash M:j}$$

(η) $$\frac{\beta \vdash \lambda x.Mx:j}{\beta \vdash M:j}$$ provided $x \notin FV(M)$

boolean operations are extended pointwise to bases.
2) Let S_1 be the type assignment system consisting of the rules{(axiom), (abstraction), (application)}, S_2 be the type assignment system consisting of the rules {(axiom), (abstraction), (application), (weakening)} and S_3 be the type assignment system consisting of the rules{(axiom), (abstraction), (application), (weakening), (η)}.
3) Let the interpretation functions $[\![\ ]\!]^i$: Env $\to (\Lambda \to \mathbf{J}^i)$, ($i \in \{1,2,3\}$), be defined as

$$[\![M]\!]^i_\rho(j) \equiv \sum_{\beta \vdash_i M:j} \left(\prod_{k \in dom(\beta)} (\beta(k) \Rightarrow \rho(k)) \right)$$

where we write $\beta \vdash_i M:j$ to indicate that the judgment $\beta \vdash M:j$ can be derived in the system S_i ($i \in \{1,2,3\}$), boolean operations are extended pointwise to environments which are taken to be functions $\rho : (Var \times \mathbf{J}) \to 2$; the expression $\sum_{\beta \vdash_i M:j}$ denotes the disjunction over provable judgments in S_i; and finally $\mathbf{J}^1$ and $\mathbf{J}^2$ are $\mathbf{J}$ of Definition 3 above, while $\mathbf{J}^3$ is the set of filters on $\mathbf{J}$. Δ

The definition above illustrates how type assignment systems can be used to define in a finitary way interpretation functions. The proposition below illustrates what we have achieved so far.

Proposition 2. The combinatory structure $\langle \mathbf{J}^1, [\![\ ]\!]^1 \rangle$ is the intersection algebra of [5], the combinatory structure $\langle \mathbf{J}^2, [\![\ ]\!]^2 \rangle$ is Plotkin-Engeler's lambda model while $\langle \mathbf{J}^3, [\![\ ]\!]^3 \rangle$ is the Filter Model [3].

Proof. A tedious but routine verification that conditions in Definition 1 are satisfied. Δ

Once we have presented Plotkin-Engeler's model in the format $\langle \mathcal{J}^2, [\![\ ]\!]^2\rangle$ it is natural to generalize the role of the Boolean Algebra **2** in the construction of Definiton 3 to an arbitrary boolean algebra B. What we obtain is then an entirely new class of combinatory structures. This kind of construction is very closely related to that which would arise if we carried out the construction of $\langle \mathcal{J}, \circ\rangle$ in a Boolean-valued universe of Set Theory: i.e. using boolean sets instead of ordinary sets. We cannot follow up this here. We give just appropriate generalizations of Definitions 3,4 and Proposition 2 to the case of an arbitrary complete boolean algebra.

Definition 5. Let B be a complete boolean algebra and A_o an arbitrary sets of constants
1) the combinatory structure $\langle \mathcal{J}_B, \circ\rangle$ is defined as follows:
let $J_B: \mathbb{N} \to Set$ be inductively defined by

$J_B(0) \equiv A_o$ and $J_B(n+1) \equiv J_B(0) \cup ([J_B(n) \stackrel{\circ}{\to} B] \times J_B(n))$

now put $\mathbf{J}_B \equiv \bigcup_n J_B(n)$ and let $\mathcal{J}_B \equiv [\mathbf{J}_B \to B]$, application over $\mathcal{J}_B$ is defined by

$$(f \circ g)(j) \equiv \sum_{r \in [\mathbf{J}_B \stackrel{\circ}{\to} B]} f((r, j)) \times \prod_{k \in dom(r)} (r(k) \Rightarrow g(k))$$

where $f,g \in \mathcal{J}_B$ and $j \in \mathbf{J}_B$.
2) Let S_{B1} be the type assignment system consisting of the following rules and:

(axiom) $$\frac{}{\{(x,j):1\} \vdash x:j}$$

(abstraction) $$\frac{\beta + \{(x,j_1):b_1,\dots,(x,j_k):b_k\} \vdash M:j}{\beta \vdash \lambda x.M:(\{j_1:b_1,\dots,j_k:b_k\},j)} \text{ provided } \forall j\ \beta((x,j)) = 0$$

(application) $$\frac{\beta \vdash M:(\{j_1:b_1,\dots,j_k:b_k\},j) \quad \beta_1 \vdash N:j_1 \quad \dots \quad \beta_k \vdash N:j_k}{\beta + \sum_{1 \le i \le k}(b_i \times \beta_i) \vdash MN:j}$$

(weakening) $$\frac{\beta \vdash M:j}{\beta + \beta' \vdash M:j}$$

where $j \in \mathbf{J}_B$ and $b \in B$;
3) let S_{B2} be the subsystem obtained from S_{B1} by omitting (weakening),
4) The interpretation functions $[\![\ ]\!]^{Bi}: Env \to (\Lambda \to \mathcal{J}_B)$ $(i \in \{1,2\})$, are defined by :

$$[\![M\sigma]\!]^{Bi}_\rho(j) \equiv \sum_{\beta \vdash_{Bi} M:j} \left(\prod_{k \in dom(\beta)} (\beta(k) \Rightarrow \rho(k)) \right).$$ Δ

Proposition 3

i) The combinatory structures $\langle\langle \mathcal{J}_B, \circ\rangle, [\![\ ]\!]^i\rangle$ $(i \in \{1,2\})$ are combinatory algebras;

ii) If B is not trivial then $\langle\langle \mathcal{J}_B, \circ\rangle, [\![\ ]\!]^i\rangle$ $(i \in \{1,2\})$ are not lambda algebra

Proof. i) Omitted, since it is a special case of the proof of Proposition 5 below.
ii) Let $\mathbf{B} \equiv \lambda xyz.x(yz)$ denote the usual composition combinator, one has immediately that $\vdash_\lambda \lambda yz.\mathbf{B}x(\mathbf{B}yz) = \lambda yz.\mathbf{B}(\mathbf{B}xy)z)$ i.e. compostion of functions is associative. A tedious computation shows that $[\![\lambda xyz.\mathbf{B}x(\mathbf{B}yz)\ ]\!]^i \neq [\![\lambda xyz.\mathbf{B}(\mathbf{B}xy)z)]\!]^i$. Δ

It is interesting to notice that the role played by **J** in the above Proposition is again akin to an "intersection type" to whom a "boolean weight" has been attached. The corresponding

type assignment system is then built so as to take into account also this non-standard "weight". Boolean coefficients appear in the hypotheses and consequently, the structural rule of*contraction* is replaced by a sort of*boolean contraction*.. This "boolean set" construction however, does not even give rise to a category. We conjecture that in order to turn the structures $\langle\langle \mathcal{J}_B, \circ\rangle, [\![\ ]\!]^i\rangle$ ($i\in\{1,2,\}$) into λ-algebras we need to define a suitable quotient.

So far we have only considered examples where sets, be these ordinary or boolean, were used in defining types, i.e. elements of $\mathbf{J}$. Yet more examples of combinatory structures can be obtained by "repeating" Plotkin-Engeler's construction using multisets in the definition of types. Surprisingly enough this construction amounts to the construction carried out by Ore [12]. This in turn was introduced as a simplified version of the notion due to Girard of *quantitative domain* [8]. Loosely speaking Girard's construction corresponds to Plotkin-Engeler's construction using arbitrary set-valued functions in place of multisets. Here we will analyze in detail only the multiset case.

Ore's notion of domain can be put into our framework by replacing the boolean algebra B in Definition 5 with $\mathbb{N}$, the set of natural numbers. Of course, the boolean operations which appear in the definiton of application and interpretation, which in turn generalized the simple set theoretic concepts of Plotkin-Engeler's algebra, have to be replaced here with arithmetic operation on natural number, i.e. disjunction with addition, conjunction with multiplication and logical implication with exponentiation. It comes almost as a surprise that under this twist of perspective, Girard-Ore's construction can be naturally related to Plotkin-Engeler's. Notice the close similarity between the following definition and Definition 5.

Definition 6. Let A_o an arbitrary sets of atoms, the combinatory structure $\langle \mathcal{J}_N, \circ\rangle$ is defined as follows: let $J_N:\mathbb{N} \to$ Set be inductively defined by

$J_N(0) \equiv A_o$ and $J_N(n+1) \equiv J_N(0)\cup([J_N(n)\overset{\circ}{\to}\mathbb{N}] \times J_N(n))$

put $\mathbf{J}_N \equiv \cup_n J_N(n)$ and let $\mathcal{J}_N \equiv [\mathbf{J}_N \to (\mathbb{N}\cup\{\infty\})]$, application over $\mathcal{J}_N$ is defined by

$$(f\circ g)(j) \equiv \sum_{r\in[\mathbf{J}_N\overset{\circ}{\to}\Box\mathbb{N}]} \Box f((r,\Box j)) \times \prod_{k\in dom(r)}^{\Box} \Box(r(k)\Box\Rightarrow\Box g(k))$$

where $f,g\in\mathcal{J}_N$, $j\in\mathbf{J}_N$, arithmetic operations are extended to $\mathbb{N}\cup\{\infty\}$ in the obvious way and $\Rightarrow$ is the usual operation of exponentiation. Δ

The above definition can be easily modified to encompass the case of a notion of domain intermediate between that of Girard's quantitative domain and Ore's domain. This notion of domain is obtained using the set *Card* of cardinals in place of the set ($\mathbb{N}\cup\{\infty\}$). In order to avoid the use of proper classes we can always think of Card as the set of cardinals smaller than a given inaccessible cardinal. The structure $\langle \mathcal{J}^C_{\mathbb{N}}, \circ\rangle$ is then obtained taking $\mathcal{J}_{C;N}$ to be $[\mathbf{J}_N \to$ *Card*], the definition of application remaining unchanged. As remarked in the introduction, presentations in [12],[8] and[11] rely heavily on the notion of analytic function. In fact both in $\langle \mathcal{J}^C_N, \circ\rangle$ and $\langle \mathcal{J}_N, \circ\rangle$ we can interpret the formula defining the application as the evaluation of an analytic function. See Appendix A for a brief illustration of this alternative viewpoint.

Going back to the structure in Definition 6, the interpretation of a λ-term M with respect to $\langle \mathcal{J}_{\mathbb{N}}, \circ\rangle$ can be given again following the familiar pattern using a type assignment system. Again, in fact , elements of $\mathbf{J}_N$ can play the role of generalized "intersection types". The coefficients being now integers. In this case however it is slightly more complex. In order to define the interpretation function it is necessary to introduce an equivalence relation on proofs of typing judgements to take care of multiplicities.

Definition 7.
1) The type assignment system N consists of the following rules:

(axiom) $$\frac{}{\Box\{(x,j):1\}\vdash\Box x:j\Box}$$

(abstraction) $$\frac{\Box\beta\Box+\Box\{(x,j_1):n_1,\ldots,(x,j_k):n_k\}\vdash\Box M:j\Box}{\Box\beta\vdash\Box\lambda x.M:(\{j_1:n_1,\ldots,j_k:n_k\},j)}\ \text{provided}\ \forall j\ \beta((x,j))=0$$

(application) $$\frac{\beta\vdash\Box M:(\{j_1:n_1,\ldots,j_k:n_k\},j)\quad \Box\beta_{i,t_i}\vdash\Box N:j_i\Box\Box\Box 1\le i\le k\Box\Box 1\le t_i\le n_i}{\beta+\sum_{\substack{1\le i\le k\\ 1\le t_i\le n_i}}\Box\beta_{i,t_i}\vdash\Box MN:j}$$

Here a basis β can be seen as a finite multiset of hypothesis of the form x:j, accordingly arithmetic operations are extended to bases in the natural way.

2) The equivalence relation $\equiv$ on proofs of typing judgements is the finest equivalence relation satisfying the following two conditions:
a) $\equiv$ is a congruence relation on the structure of the proof, i.e.:

$$\Delta_i\equiv\Delta'_i\ \Rightarrow\ \frac{\Delta_1\ldots\Delta_i\ldots\Delta_n}{\Delta}\equiv\frac{\Delta_1\ldots\Delta'_i\ldots\Delta_n}{\Delta}$$

b) For every permutation σ of the set $\{1,...,k\}$

$$\frac{\beta\vdash\Box M:(\{j_1:n_1,\ldots,j_k:n_k\},j)\quad \Box\beta_{i,t_i}\vdash\Box N:j_i\Box\Box\Box 1\le i\le k\Box\Box 1\le t_i\le n_i}{\beta+\sum_{\substack{1\le i\le k\\ 1\le t_i\le n_i}}\Box\beta_{i,t_i}\vdash\Box MN:j}\equiv\frac{\beta\vdash\Box M:(\{j_1:n_1,\ldots,j_k:n_k\},j)\quad \Box\beta_{\sigma(i),t_{\sigma(i)}}\vdash\Box N:j_{\sigma(i)}\Box\Box\Box 1\le i\le k\Box\Box 1\le t_i\le n_i}{\beta+\sum_{\substack{1\le i\le k\\ 1\le t_i\le n_i}}\Box\beta_{i,t_i}\vdash\Box MN:j}$$

3) The interpretation of a λ-term M in the applicative structure $\langle J_{\mathbf{N}},\circ\rangle$ is defined by:

$$[\![M^{\sigma}]\!]^{\mathbf{N}}\rho(j)\ =\ \sum_{\beta}\Box\rho^{\beta}\times\sum_{\Delta\in\{[\beta\vdash\Box M\Box:\Box j]\}}\Box 1\Box\Box\Box\ =\ \Box\sum_{\beta}\ \sum_{\Delta\in\{[\beta\vdash\Box M\Box:\Box j]\}}\Box\rho^{\beta}$$

where we use the abbreviation g^{β} for $\prod_{k\in dom(\beta)}\Box(\beta(k)\Box\Rightarrow\Box g(k))$, $[\Delta]$ denotes the equivalence class modulo $\equiv$ of Δ and $\{[G]\}$ denotes the set of equivalence classes of proofs having the judgement G as conclusion. Δ

The definition of the equivalence $\equiv$ between proofs can be motivated as follows. When we apply the rule (application) we must fix an order on the domain of the function $\{(j_1:n_1)...(j_k:n_k)\}$. This order is completely arbitrary. The condition b) in Definition 7.2. sets two proofs to be equivalent if they differ just up to the order chosen on the domain of the function $\{(j_1:n_1)...(j_k:n_k)\}$.
We are now ready to establish the properties of $\langle J_{\mathbf{N}},\circ\rangle$.

Proposition 4. The combinatory structure $\langle\langle J_{\mathbf{N}},\circ\rangle,[\![\ ]\!]^{\mathbf{N}}\rangle$ is a lambda algebra but not a lambda model.

Proof. The proof that $\langle\langle J_{\mathbf{N}},\circ\rangle,[\![\ ]\!]^{\mathbf{N}}\rangle$ is a combinatory algebra is given in Appendix B. For lack of space we omit the routine proof that it is a λ-algebra, see[12]. The following counterexample shows that the ξ-rule fails in $\langle\langle J_{\mathbf{N}},\circ\rangle,[\![\ ]\!]^{\mathbf{N}}\rangle$: let $\rho(x)=((\{j_1:1\},j):\infty)$ and $\rho(y)=((\{j_1:2\},j):\infty)$, now $\forall a.[\![xz]\!]_{\rho[z/a]}=[\![yz]\!]_{\rho[z/a]}$ but $[\![\lambda z.xz]\!]_{\rho}=[\![\lambda z.yz]\!]_{\rho}$. Δ

4 The General Case

After having gone through various different examples in the previous section, we are now ready to introduce the general notion of *applicative exponential structure*. This notion subsumes all the concrete notions of application presented up to now. It denotes a general kind of structure where function application can be adequately defined via a type-assignment system. It arises from the abstract characterizations of the structure of the coefficients which are applied either to the "types" or to the "points" of the domains in the previous examples. It turns out infact, that two different kinds of coefficients are actually involved in the construction; a fact this, which was never apparent in the previous examples.

Definition 8. A *preexponential structure* consists of a triple $\langle \mathbf{A}, \mathbf{E}, \Rightarrow \rangle$ where:
1) $\mathbf{A} \equiv \langle A, +_A, \times_A , 0_A, 1_A \rangle$ is an infinitary commutative semiring, i.e. an algebraic structure where $+_A$ is an infinitary commutative, associative operation over A, with identity 0_A, and $\times_A$ is an associative, commutative binary operation over A which distributes over $+_A$, with identity 1_A.
2) $\mathbf{E} \equiv \langle E, +_E, \times_E, 0_E, 1_E \rangle$ is a commutative semiring, i.e. $+_E$ is a binary associative and commutative operation over E with identity 0_E, and $\times_E$ is an associative, commutative binary operation over E which distributes over $+_E$, with identity 1_E.
3) there is a binary operation $\Rightarrow : E \times A \rightarrow A$ satisfying the following axioms:
a) $(e_1 +_E e_2) \Rightarrow a = (e_1 \Rightarrow a) \times_A (e_2 \Rightarrow a)$
b) $e \Rightarrow (a_1 \times_A a_2) = (e \Rightarrow a_1) \times_A (e \Rightarrow a_2)$
c) $(e_1 \Rightarrow (e_2 \Rightarrow a)) = e_1 \times_E e_2 \Rightarrow a$
d) $0_E \Rightarrow a = 1_A$ and $1_E \Rightarrow a = a$ Δ

As will become clear in the following preexponential structures will be the abstract coefficients of our genral notion of applicative structure. Elements of **E** will be the "weights" of the "types" while elements of **A** will be the coefficients of the points.

Condition 3) above shows that the function $\Rightarrow$ satisfies essentially the properties of an exponential function. In the proof of Proposition 4 an essential property of exponentiation over natural numbers is crucial: Newton's binomial expansion. The corresponding identity over booleans, necessary for proving Proposition 3, on the other hand is trivial. The notion of *exponential structure* below, is the appropriate abstract setting for carrying out the analogue of the "binomial expansion" over a preexponential structure, where elements of **E** play the role of exponents (whence the name) and elements of **A** play the role of bases. Subscripts are omitted.

Definition 9. Let $\langle \mathbf{A}, \mathbf{E}, \Rightarrow \rangle$ be a *preexponential structure* .

1) An element e of E is called *unitary* if $e \Rightarrow \sum_{j \in J} a_j = \sum_{j \in J} (e \Rightarrow a_j)$ holds for all $\sum_{j \in J} a_j$. The set of unitary elements of E is denoted with U_E.

2) Given $e \in E$, a function $f \in \mathbb{N} \xrightarrow{\circ} U_E \cup \{0\}$ is a *unitary decomposition* of e if $e = \sum_{j \in dom(f)} f(j)$. Given $e \in E$ the set of all unitary decompositions of e is denoted with U(e)

3) An *exponential structure* is a preexponential structure $\langle \mathbf{A}, \mathbf{E}, \Rightarrow \rangle$ together with a function

$H : E \rightarrow ([\mathbb{N} \xrightarrow{\circ} U_E \cup \{0\}])$ which satisfies the following two axioms:
Ax1) For each $e \in E$ there is a unitary decomposition of E;
Ax2) The function H chooses a unitary decomposition for each element of E Δ

This is the least intuitive definition among the ones given so far. But its complexity is rewarding. We can now safely use exponential structures as possible coefficients in the machinery that we put to work in the previous section for defining concrete combinatory algebras.

Definition 10. Given an exponential structure $\langle \mathbf{A}, \mathbf{E}, \Rightarrow, H\rangle$ and an arbitrary set of constants C_o, an *applicative exponential structure*, a.e.s. for short, over $\langle \mathbf{A}, \mathbf{E}, \Rightarrow, H\rangle$ is the c.s $\langle\langle J_{\mathbf{E}}^{\mathbf{A}}, \circ\rangle, [\![\ ]\!]\rangle$ defined as follows:
define inductively $J_{\mathbf{E}}: \mathbb{N}\to Set$ by $J_{\mathbf{E}}(0) \equiv C_o$ and
$J_{\mathbf{E}}(n+1) \equiv J_{\mathbf{E}}(0) \cup (([J_{\mathbf{E}}(n) \stackrel{\circ}{\to} \mathbf{E}] \times J_{\mathbf{E}}(n))$,
put $\mathbf{J_E} \equiv \bigcup_n J_{\mathbf{E}}(n)$ and $J_{\mathbf{E}}^{\mathbf{A}} = [\mathbf{J_E} \to A]$, and,for $f, g \in J_{\mathbf{E}}^{\mathbf{A}}$ and $j \in \mathbf{J_E}$, let application over $J_{\mathbf{E}}^{\mathbf{A}}$ be defined by $(f \circ g)(j) \equiv \sum_{r \in [\mathbf{J_E} \stackrel{\circ}{\to} \mathbf{N}]} f((r,j)) \times g^r)$.
The interpretation function $[\![\ ]\!]: Env \to (\Lambda \to J_{\mathbf{E}}^{\mathbf{A}})$ is defined via the type assignment system $S_{\mathbf{E}}^{\mathbf{A}}$ by: $[\![M]\!]_\rho(j) = \sum_\beta \rho^\beta \times \sum_{\Delta \in \{[\beta \vdash M:j]\}} 1 = \sum_\beta \sum_{\Delta \in \{[\beta \vdash M:j]\}} \rho^\beta$
for $j \in \mathbf{J_E}$ and $M \in \Lambda$.
The system $S_{\mathbf{E}}^{\mathbf{A}}$ consists of the following three rules:

(axiom) $$\frac{}{\{(x,j):1\} \vdash x:j} \text{ provided } j \in \mathbf{J_E}$$

(abstraction) $$\frac{\beta + \{(x,j_1):e_1,\ldots,(x,j_k):e_k\} \vdash M:j}{\beta \vdash \lambda x.M:(\{j_1:e_1,\ldots,j_k:e_k\},j)} \text{ provided } \forall j\ \beta((x,j)) = 0$$

(application) $$\frac{\beta \vdash M:(\{j_1:e_1,\ldots,j_k:e_k\},j) \quad \beta_{i,t_i} \vdash N:j_i \quad 1 \le i \le k \quad t_i \in dom(H(e_i))}{\beta + \sum_{\substack{1 \le i \le k \\ t_i \in dom(H(e_i))}} H(e_i)(t_i) \times \beta_{i,t_i} \vdash MN:j}$$

The equivalence relation on proofs used in the definition of $[\![\ ]\!]$ is that of Definition 7 and the abbreviations g^r, ρ^β are those of Definition 7. Δ

The following Proposition is the main result of the paper.

Proposition 5. Applicative exponential structures are combinatory algebras.

Proof. The proof is very similar to the one given in Appendix B. Δ

One can easily check that all the constructions in Section 3, fall under the above general definition. The only non-trivial issue is the choice of the function H in the definition of the exponential structure. Whenever **E** is instantiated by $\mathbb{N}$ there is only one choice possible. Whenever **E** is a boolean algebra the choice is immaterial. Moreover in the latter case the summation over equivalence classes of proofs, in the definition of the interpretation function, is irrelevant.

5 Applications and Directions for Future Works

The general notion of applicative exponential structure suggests other notions of function application and hence other interesting constructions of concrete combinatory algebras. Here are few examples.

1) Lamarche [11] discusses at length the case of an a.e.s. $\langle\langle \mathbf{J}^{\mathbf{A}}_{\mathbf{N}}, \circ\rangle, [\![\ ]\!]\rangle$ where **A** is a complete Heyting algebra. This is a slight variation of Ore's construction. In this case we have to define the exponential $(\Rightarrow) : \mathbb{N}\times\mathbf{A}\to\mathbf{A}$ by:

$$n \Rightarrow a := \begin{cases} a\times\ldots\times a \text{ (n times)} & \text{if } n \geq 1 \\ 1 & \text{if } n = 0 \end{cases}$$

2) Using the set of positive rational numbers $\mathbf{Q}^+$, or the set of positive real numbers $\mathbb{R}^+$, we can form the exponential algebras $\langle \mathbf{Q}^+\cup\{\infty\}, \mathbb{N}, \Rightarrow, U\rangle$ and $\langle \mathbb{R}^+\cup\{\infty\}, \mathbb{N}, \Rightarrow, U\rangle$ where $\mathbb{N}$ is the set of natural numbers, addition and multiplication are the standard ones and $\Rightarrow$ is the usual exponential, the function U decomposes each natural number n in a sum of 1's. In the corresponding λ-algebras $\langle\langle \mathbf{J}^{\mathbf{Q}^+\cup\{\infty\}}_{\Box\mathbf{N}}, \circ\rangle, [\![\ ]\!]\rangle$ and $\langle\langle \mathbf{J}^{\mathbf{R}^+\cup\{\infty\}}_{\Box\mathbf{N}}, \circ\rangle, [\![\ ]\!]\rangle$ the points can be interpreted as formal power series, i.e. the description of analytic functions either on positive rational or on positive real numbers. More examples of this kind can be defined by restricting the range of the relevant operators to suitable intervals.

3) Given any complete Heyting algebra G and indicating with In_G the set of invertible elements in G, $\mathrm{In}_G = \{a\in G \mid \exists \bar{a}.\ a+\bar{a}=1,\ a\times\bar{a}=0\}$, $\langle G, \mathrm{In}_G, \Rightarrow\rangle$ is an exponential algebra. In this case the function I decomposes every element *a* of In_G in the a sum containing the single element *a*. Combinatory algebras $\langle\langle \mathbf{J}^{\mathbf{G}}_{\mathbf{In_G}}, \circ\rangle, [\![\ ]\!]\rangle$ generated as in Definition 10 by this kind of exponential algebras generalize the boolean c.s. of Definition 6.

4) Let G be a finite Heyting algebra then $\langle G, G, \Rightarrow, H\rangle$ is an exponential algebra for an appropriate choice of H, which always exists. The c.s. $\langle\langle \mathbf{J}^{\mathbf{G}}_{\mathbf{G}}, \circ\rangle, [\![]\!]\rangle$ generated as in Definition 10 is yet another example of a combinatory algebra.

The ideas outlined in this paper need to be investigated further. First of all one can try to strengthen the conditions in the definition of a.e.s. so as to obtain always lambda algebras. In another direction one can try to define a coherence predicate on the elements of **J** so as to be able to subsume notions of domain which involve stable functions. Finally one should explore the relation between the notions of domain arising in this setting, which for instance, are not necessarily ω-algebraic, and those which are normally used in connection with Scott Domains. An abstract notion of implication between elements of **J** can be possibly introduced, which could be used to introduce a general notion of filter.

The structures introduced in this paper can turn out to be quite useful from the point of view of programming language semantics. For example, one can easily get a plethora of different denotational semantics for a simple functional language featuring a non-detrministic **or** operator. For any particular applicative exponential structure based on the exponential structure $\langle \mathbf{A}, \mathbf{E}, \Rightarrow\rangle$, one can give a denotation to the non-deterministic **or** in terms of the operators $+_A$ and $\times_A$. One can take it to be, for instance, an operation $[\![\mathbf{or}]\!]:[\mathbf{J}\to\mathbf{A}]^2\to[\mathbf{J}\to\mathbf{A}]$ defined by applying pointwise on **J** a suitable weighted average. The intuition behind this is that the meaning of **or** is that of evaluating either the left hand with a suitable weight or the right hand with another weight. According to the particular choice made one gets different flavours of non-determinstic operators. Some of these are interesting in themselves and can illuminate our intuition of non-determinism. For example, applicative exponential structures based on boolean sets, where weights are thought of as sets of favorable events, yield a kind of non-determinsm which is settled once and forall before the computation is started; the resulting coefficient being the set of favorable events for a given result of a computation. Semantics based on multisets are more directly related to the frequency with which a given result is obtained following different computations, see [8,12]. Semantics based on real valued sets are finally closer to real probabilities.

References

1. S.Abramsky: Domain Theory in Logical Form. Annals of Pure and Applied Logic, (1991)
2. H.Barendregt: Lambda Calculus: its Syntax and Semantics revised version. Studies in Logic. Amsterdam: North Holland 1984
3. H.Barendregt, M.Coppo, M.Dezani Ciancaglini: A Filter Lambda Model and the Completeness of Type Assignment. Journal fo Symbolic Logic, 48, 4 (1983)
4. M.Coppo, M.Dezani Ciancaglini, F.Honsell, G.Longo: Extended Type Structures and Filter Lambda Models. In: G.Longo et al. (eds.): Logic Colloquium '82. Amsterdam: North Holland 1983
5. M.Coppo, M.Dezani Ciancaglini, B.Venneri: Principal Type Schemes and Lambda Calculus Semantics. In: J.Seldin et al. (eds): To H.B.Curry: Essays. Academic Press 1980
6. P.Di Gianantonio, F.Honsell: A General Type Assignment System for an Abstract Notion of Domain. Talks given at the 4th and 6th Meetings of the Jumelage Typed Lambda Calculus. Edinburgh, October 1989 and Paris, January 1991
7. E.Engeler: Algebras and Combinators. Berichte des Instituts f. Informatik 32, ETH, Zurich 1979
8. J.Y.Girard: Normal Functors Power Series and Lambda Calculus. Annals of Pure and Applied Logic, 37, 2 (1988)
9. R.Hindley, G.Longo: Lambda Calculus Models and Extensionality. Zeit. f. Math. Logik u. Grund. d. Math., 26 (1980)
10. F.Honsell, S.Ronchi della Rocca: Reasoning about interpretations in qualitative Lambda Models. In: M.Broy et al. (eds.) Programming Concepts and Methods. 1990
11. F.Lamarche: Quantitative Domains and Infinitary Algebras. Unpublished manuscript, 1990
12. Ch.-E.Ore: Introducing Girard's quantitative domains. PhD Thesis, Research Report 113. University of Oslo 1988
13. G.Plotkin: A set-theoretical definition of application. Memorandum MIP-R-95, School of Artificial Intelligence, University of Edinburgh, 1972
14. G.Plotkin: Domains for Denotational Semantics, course notes, Stanford 1985
15. D.Scott: Some philosophical issues concerning theories of combinators, lambda calculus and computer science theory. In LNCS 37, Springer Verlag, 1975
16. D.Scott: Data Types as Lattices. SIAM Journal of computing, 5 (1976)

Appendix A

Using the notation introduced in Section 3, we outline, in the 7 points below, the alternative viewpoint under which applicative exponential structure can be considered. This viewpoint focuses on the notion of formal power series and analytic function as a justification for the definition of application in $\langle J^{C}_{\mathbb{N}}, \circ \rangle$ and $\langle J_{\mathbb{N}}, \circ \rangle$.

1) $\mathbf{J}_{\mathbb{N}}$ is taken to be a set of variables. (In the following $\mathbf{J}_{\mathbb{N}}$ will be denoted simply by $\mathbf{J}$.)

2) any $r \in [\mathbf{J} \xrightarrow{\circ} \mathbb{N}]$ is viewed as a monomial in the variables $\mathbf{J}$:
i.e. r corresponds to the monomial $\prod_{k \in dom(r)} k^{r(k)}$

3) an element $g \in J_{\mathbb{N}} \equiv [\mathbf{J} \to (\mathbb{N} \cup \{\infty\})]$ is viewed as a vector consisting of $\mathbf{J}$ components; the value $\prod_{k \in dom(r)} (r(k) \Rightarrow g(k))$ is then the result of the evaluation of the monomial r with respect to the vector g;

4) a function $h:(\mathbf{J} \xrightarrow{\circ} \mathbb{N}) \to (\mathbb{N} \cup \{\infty\})$ assigns a coefficient to every monomial and therefore

determines a formal power series in the variables $\mathbf{J}$: i.e.

$$\sum_{r\in[\mathbf{J}\overset{\circ}{\to}\mathbb{N}]} \square h(r) \times \prod_{k\in dom(r)} \square k^{r(k)} \ ;$$

5) a function h' : $\mathbf{J}\to([\mathbf{J}\overset{\circ}{\to}\mathbb{N}] \to (\mathbb{N}\cup\{\infty\})$ associates a formal power series to each variable in $\mathbf{J}$ and is therefore a vector valued formal power series. These functions are usually called analytic;

6) the following isomorphisms hold:

$J_{\mathbf{N}} \equiv \mathbf{J}\to(\mathbb{N}\cup\{\infty\}) \cong (J(0)\cup((\mathbf{J}\overset{\circ}{\to}\mathbb{N})\times\mathbf{J}))\to(\mathbb{N}\cup\{\infty\}) \cong$

$\cong (J(0)\to(\mathbb{N}\cup\{\infty\}))\times(((\mathbf{J}\overset{\circ}{\to}\mathbb{N})\times\mathbf{J})\to(\mathbb{N}\cup\{\infty\})) \cong$

$\cong (J(0)\to(\mathbb{N}\cup\{\infty\}))\times(\mathbf{J}\to((\mathbf{J}\overset{\circ}{\to}\mathbb{N})\to(\mathbb{N}\cup\{\infty\}))$

therefore an element $f\in J_{\mathbf{N}}$ is a vector of $\mathbf{J}$ components and containes the representation of a vector valued formal power series;

7) $f\circ g$ is the vector obtained applying the analytic functions described by f to the vector g.

Appendix B

Proof of Proposition 4.
First we need a lemma.

Lemma. The set $\{[\beta\vdash MN : j]\}$ can be decomposed in disjoint singletons in the following way:

$$\{[\beta\vdash MN:j]\} = \bigcup_{\alpha=\{(j_1:n_1)\dots(j_k:n_k)\}} \ \bigcup_{\substack{\beta' \\ \beta_{1,1}\dots\beta_{1,n1} \\ \vdots \\ \beta_{k,1}\dots\beta_{k,nk} \\ \square\beta=\beta'+\beta_{1,1}+\dots+\beta_{k,nk}}} \ \bigcup_{\substack{\Delta'\in\{[\beta\vdash M:j]\} \\ \Delta_{1,1}\dots\Delta_{1,n1} \\ \vdots \\ \Delta_{k,1}\dots\Delta_{k,nk} \\ \square\Delta_{i,h}\in\{[\beta_{i,h}\vdash N:j_i]\}}} \square\left\{\left[\frac{\Delta'\Delta_{1,1}\dots\Delta_{k,nk}}{\beta\vdash\square MN:j}\right]\right\}$$

Proof (lemma). The above formula just enumerates all possible equivalence classes of proofs of judgments $\beta\vdash MN:j$. It is not difficult to see that the formula is correct observing that:
1) in all the proofs of $\beta\vdash MN:j$ the last rule applied is the application rule.
2) in the formula we consider just one of the possible orderings of the domain of the function $\alpha=\{(j_1:n_1)\dots(j_k:n_k)\}$. Δ

Proof (Proposition)

i) We must prove $[\![x]\!]_\rho = \rho(x)$, by definition we have:

$$[\![x]\!]_\rho(j) = \sum_{\beta} \square \sum_{\Delta\in\{[\beta\vdash\square x:j]\}} \square\rho^{\beta}$$

the only proofs of judgments of the form: $\beta\vdash x:j$ are given by an application of the (axiom)-rule $\dfrac{}{\{(x,j):1\}\vdash\square x:j}$, so $[\![x]\!]_\rho(j) = \rho^{\{(x,\,j),\,1\}} = (\rho(x,j))^1 = \rho(x)(j)$.

ii) We must prove $[\![MN]\!]_\rho(j) = ([\![M]\!]_\rho\circ[\![N]\!]_\rho)(j)$; expanding the term on the left of this equality we have:

$$[\![MN]\!]_\rho(j) = \sum_\beta \Box \sum_{\Delta\in\{[\beta\vdash\Box MN:j]\}} \Box\rho^\beta = \text{(by the previous lemma)}$$

$$= \sum_\beta \Box \sum_{\alpha=\{(j_1:n_1)\ldots(j_k:n_k)\}} \Box \sum_{\substack{\beta' \\ \beta_{1,1}\ldots\beta_{1,n1} \\ \vdots \\ \beta_{k,1}\ldots\beta_{k,nk} \\ \Box\beta=\beta'+\beta_{1,1}+\ldots+\beta_{k,nk}}} \Box \sum_{\substack{\Delta'\in\{[\beta\vdash M:j]\} \\ \Delta_{1,1}\ldots\Delta_{1,n1} \\ \vdots \\ \Delta_{k,1}\ldots\Delta_{k,nk} \\ \Box\Delta_{i,h}\in\{[\beta_{i,h}\vdash N:j_i]\}}} \Box\rho^{\beta} =$$

$$= \sum_{\alpha=\{(j_1:n_1)\ldots(j_k:n_k)\}} \Box \sum_{\substack{\beta' \\ \beta_{1,1}\ldots\beta_{1,n1} \\ \vdots \\ \beta_{k,1}\ldots\beta_{k,nk}}} \Box \sum_{\substack{\Delta'\in\{[\beta\vdash M:j]\} \\ \Delta_{1,1}\ldots\Delta_{1,n1} \\ \vdots \\ \Delta_{k,1}\ldots\Delta_{k,nk} \\ \Box\Delta_{i,h}\in\{[\beta_{i,h}\vdash N:j_i]\}}} \Box\rho^{\beta'+\beta_{1,1}+\ldots+\beta_{k,nk}}$$

Expanding the term on the right we obtain:

$$([\![M]\!]_\rho\circ[\![N]\!]_\rho)(j) = \sum_{\alpha=\{(j_1:n_1)\ldots(j_k:n_k)\}} [\![M]\!]_\rho(\alpha,j)\times([\![N]\!]_\rho(j_1))^{n_1}\times\ldots\times([\![N]\!]_\rho(j_k))^{n_k} =$$

$$= \sum_{\alpha=\{(j_1:n_1)\ldots(j_k:n_k)\}} \left(\sum_\beta \sum_{\Delta\in\{[\beta\vdash\Box M:(\alpha,j)]\}} \Box\rho^{\beta}\right) \times \left(\sum_\beta \sum_{\Delta\in\{[\beta\vdash\Box N:(j_1)]\}} \Box\rho^{\beta}\right)^{n_1} \times\ldots$$

$$\ldots\times \left(\sum_\beta \sum_{\Delta\in\{[\beta\vdash\Box N:(j_k)]\}} \Box\rho^{\beta}\right)^{n_k} =$$

$$= \sum_{\alpha=\{(j_1:n_1)\ldots(j_k:n_k)\}} \sum_{\substack{\beta' \\ \beta_{1\,1}\ldots\beta_{1\,n_1} \\ \vdots \\ \beta_{k\,1}\ldots\beta_{k\,n_k} \\ \Box\Box\Box\Box\Box}} \sum_{\substack{\Delta'\in\{[\beta\vdash M:j]\} \\ \Delta_{1,1}\ldots\Delta_{1,n1} \\ \vdots \\ \Delta_{k,1}\ldots\Delta_{k,nk} \\ \Box\Delta_{i,h}\in\{[\beta_{i,h}\vdash N:j_i]\}}} \rho^{\beta'}\times\rho^{\beta_{1,1}}\times\cdots\times\rho^{\beta_{k,nk}}$$

and this expression is equal to what we obtained before expanding the other term.

iii) We must prove that $([\![\lambda x.M]\!]_\rho\circ a)(j) = [\![M]\!]_{\rho[a/x]}(j)$, by definition we have the following equalities,

$$([\![\lambda x.M]\!]_\rho\circ a)(j) = \sum_\alpha^{\Box} \Box([\![\lambda x.M]\!]_\rho(\alpha,j))a^\alpha =$$

$$= \sum_\alpha \left(\sum_\beta^{\Box} \sum_{\Delta\in\{[\beta\vdash\Box\lambda x.M:(\alpha,j)]\}} \Box\rho^{\beta}\right) \times a^\alpha =$$

$$= \sum_\alpha \left(\sum_\beta^{\Box} \sum_{\Delta\in\{[\beta+x:\alpha\vdash\Box M:j]\,|\,x\notin\beta\}} \Box\rho^{\beta}\right) \times a^\alpha = \text{(using distributivity)}$$

$$=\sum_{\alpha}\sum_{\beta}\ \sum_{\Delta\in\{[\beta+x:\alpha\vdash\Box M:j]\,|\,x\notin\beta\}} \rho^{\beta}\times a^{\alpha} \quad =$$

$$=\sum_{\alpha}\sum_{\beta}\ \sum_{\Delta\in\{[\beta+x:\alpha\vdash\Box M:j]\,|\,x\notin\beta\}} \rho[a/x]^{\beta+\{x:\alpha\}} \quad =$$

$$=\sum_{\beta'}\ \sum_{\Delta\in\{[\beta'\vdash\Box M:j]\}} \rho[a/x]^{\beta'} = [\![M]\!]_{\rho[a/x]}(j) \quad \text{(by definition).}$$

iv) We must prove that $\forall j.[\![\ \lambda x.M\ ]\!]_{\rho}(j) = [\![\ \lambda y.([x/y]M)]\!]_{\rho}(j)$ provided $y\notin FV(M)$. By definition :

$$[\![\ \lambda x.M]\!]_{\rho}(j) = \sum_{\beta}\Box\ \sum_{\Delta\in\{[\beta\vdash\Box\lambda x.M:j]\}}\Box\rho^{\beta} \quad =$$

if $y^{\sigma}\notin FV(M)$ then there is a bijection between proofs of $\beta\vdash \lambda x.M:j$ and proofs of $\beta\vdash\Box\lambda y.([x/y]M):j$ and these correspondence preserves the equivalence relation on proofs, thus

$$\sum_{\beta}\Box\ \sum_{\Delta\in\{[\beta\vdash\Box\lambda x.M:j]\}}\Box\rho^{\beta} \quad = \quad \sum_{\beta}\Box\ \sum_{\Delta\in\{[\beta\vdash\Box\lambda y.([x/y]M):j]\}}\Box\rho^{\beta} \quad = [[\lambda x([x/y]M)\ ]]_{\rho}(j)$$

v) We must prove that $\forall j.[\![\ M\ ]\!]_{\rho}(j) = [\![\ \lambda x.M\ ]\!]_{\xi}(j)$ provided $\rho(x) = \xi(\ x)$ for $x\in FV(M)$

By definition $[\![\ M\ ]\!]\rho(j) \ = \ \sum_{\beta}\Box\rho^{\beta}\times \sum_{\Delta\in\{[\beta\vdash\Box M:j]\}}\Box\,\Box\,1\,\Box\,\Box$

it is easy to show by induction that in $\beta\vdash M:j$ the domain of β contains only the free variables of M. Moreover ρ^{β} is equal to ξ^{β} if ρ and ξ are equal on the domain of β. Δ