Essential and relational models

DOI:10.1017/S0960129515000316

Under consideration for publication in Math. Struct. in Comp. Science

Essential and Relational Models †

LUCA PAOLINI[1] , MAURO PICCOLO[2] and SIMONA RONCHI DELLA ROCCA[1]
[1] *Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185, 10149 Torino, Italy*
[2] *Dipartimento di Informatica Scienze ed Ingegneria, Università di Bologna*

Received 14 February 2015

Dedicated to Corrado Böhm for his 90th birthdays.

Intersection type assignment systems can be used as a general framework for building logical models of λ-calculus that allow to reason about the denotation of terms in a finitary way. We define *essential* models (a new class of logical models) through a parametric type assignment system using non idempotent intersection types. Under an interpretation of terms based on typings instead than the usual one based on types, every suitable instance of the parameters induces a λ-model, whose theory is sensible. We prove that this type assignment system provides a logical description of a family of λ-models arising from a category of sets and relations.

1. Introduction

In the general framework of denotational semantics of λ-calculus, logical models are a particular class of λ-models supplying a finitary description of the interpretation of terms, through type assignment systems. Types are built from a set of constants, via two type-constructors: the arrow ($\rightarrow$) and the intersection ($\wedge$). Terms are interpreted as sets of types, so reasoning about the interpretation of a term in these models can be done via type inference; in fact, in order to prove the equivalence between two terms, it is sufficient to show that they can be assigned the same set of types. Although the type inference is undecidable, logical models are concrete tools for reasoning in finitary way on the interpretation of terms, since a (finite) derivation grasps a finite piece of the semantic-interpretations.

The relationship between logical models and domain-theoretical models has been widely studied, and it has been proved that some interesting classes of such models can be described in logical form. For instance, filter models supply a logical description of a class of Scott models based on continuous functions, since they can be seen as a restriction of the domain theory in logical form, which goes back to Stone duality (see Abramsky (1991)). A first characterization of a logical model where types represent continuous functions is in Coppo et al. (1984), a sketch of the proof of the correspondence between filter

† This work was partly supported by MURST - PRIN 2010FP79LR-007.

models and (a class of) Scott models is in Honsell and Ronchi Della Rocca (1992) among others and a more complete proof is in Ronchi Della Rocca and Paolini (2004), Chapter 13. Furthermore a logical description of models based on stable functions has been first proposed in Honsell and Ronchi della Rocca (1990), where qualitative domains have been considered, while in Paolini et al. (2009) a logical description of stable models based on coherence spaces has been proposed. Intersection types roughly correspond to names for finite elements of a domain, e.g. for compact elements in a Scott's domain and for tokens in a coherence space. In both these cases, intersection is considered modulo idempotency ($\sigma \wedge \sigma = \sigma$), commutativity and associativity. So, an intersection is treated as a n-ary operator, and an intersection of types $\sigma_1 \wedge ... \wedge \sigma_n$ can be naturally considered as a set, so inducing a functional interpretation of types: a type $\sigma_1 \wedge ... \wedge \sigma_n \to \tau$ represents a step-function in the continuous setting and a token of the trace of a stable function in the stable setting respectively. Types come equipped with a *type theory*, reflecting the structure of the underlying domain: in case of the Scott models, the type theory is a preorder on types reflecting the order between the compact elements of the domain, in case of the stable models the type theory is an equivalence relation, inherited by the coherence relation between tokens. Therefore each class of models can be represented by a parametric type assignment system, where the parameters are the set of type constants and the type theory. Every correct instance of them supplies a particular model. In fact, a key notion of the logical description of a class of models is the notion of *legality* of a type theory, i.e., the constraint that a type theory must satisfy in order to represent a λ-model. The main structural difference between these two kinds of logical descriptions is that weakening is necessary in order to describe a Scott model, while it is unsound in the description of stable models. In both cases, a type can be assigned to a term whenever the corresponding element belongs to its interpretation in the model, and so the interpretation of a term is the set of types derivable for it.

In this paper we define a new class of λ-models, using non-idempotent intersection types. In particular, we propose a parametric type assignment system, where intersection is associative and commutative, but not idempotent, and where the parameters are the set of type constants and a type theory, i.e. a congruence on types on which we only ask the constraint of preserving the number of components of an intersection. Weakening is not allowed, since we want to consider the minimal information necessary to give type to terms. Every suitable instance of the parameters gives rise to a λ-model, by using a non standard interpretation of terms. In fact, in this setting interpreting a term as the set of types derivable for it is no more sound (it induces a λ-algebra but not a λ-model). Therefore, we interpret a term as a set of pre-typings, which are pairs $(\mathtt{B}; \sigma)$, where $\mathtt{B}$ is a finite map from variables to types and σ is a type. In case of a closed term $\mathtt{M}$, the interpretation is reminiscent of the standard one, since it is the set $\{(\emptyset; \sigma) \mid \sigma \text{ is derivable for } \mathtt{M}\}$. We call this class of models the class of *essential* models. This class turns out to be quite poor, with respect to the induced λ-theories, since it induces only sensible theories.

It is worth noticing that the loss of idempotency implies that a type of the shape $\sigma_1 \wedge ... \wedge \sigma_n \to \tau$ has no more a pure functional interpretation, since it carries out some intensional information about how many times an argument is used. In fact, the family of essential models we define provides a logical description of a class of models formalized in Bucciarelli et al. (2007) in $M\mathbb{R}el$, a category of sets and relations. The proposed non standard interpretation of terms in the essential λ-models reflects the fact that $M\mathbb{R}el$ has not "enough points". Following Selinger (2002), in order to build lambda-models each term is interpreted as a suitable morphism of $M\mathbb{R}el(U^n, U)$, where U is a reflexive object and $n \in \mathbb{N}$. Indeed, this issue is logically modeled by the use of pre-typings.

Two type assignment systems based on non idempotent intersection types and inducing a λ-algebra have been already defined, namely in Coppo et al. (1981) and de Carvalho (2009). Both can be obtained from the class of essential type assignment systems, by a suitable instance of the parameters, as we will show in the conclusion.

Related Works Type assignment systems with non-idempotent intersection have been already studied in literature, for many different purposes. Kfoury and Wells (2004) used non-idempotent intersection in order to formalize a type inference semi-algorithm whose complexity has been studied in Mairson and Neergaard (2004). Kfoury (2000), connected non-idempotent intersection types and linear β-reduction. Non idempotent intersection types have been used in Pagani and Ronchi Della Rocca (2010b,a) to characterize the solvability in the resource λ-calculus. Non idempotent intersection types have been used as technical tool for a characterization of strong normalization in Bernadet and Lengrand (2011b). Non idempotency has a quantitative flavour, in fact they have been used for proving interesting quantitative properties about the complexity of the β-reduction in de Carvalho (2009), Bernadet and Lengrand (2011a) and De Benedetti and Ronchi Della Rocca (2013). Some observations about the use of non idempotent intersection types in the setting of implicit computational complexity have been made in Terui (2006). In Di Gianantonio et al. (2008) game semantics of a typed λ-calculus has been described in logical form using an intersection type assignment system where the intersection is not idempotent neither commutative nor associative. Ehrhard (2012) has presented a paper, where the logical descriptions of continuous and relational models for the call by value λ-calculus are put in correspondence by an extensional collapse, and both idempotent and not idempotent intersection types are used to prove this relation. Very recently, in Bucciarelli et al. (2014) it has been defined a sound and complete inhabitation algorithm for non-idempotent intersection types, while in Urzyczyn (1999) the problem has been proved undecidable in the idempotent case.

Outline The paper is organized as follows. In Section 2 the crucial definitions of λ-algebra and λ-model are recalled. Section 3 contains the definition of the essential type assignment system, and the proofs of some of its properties. In Section 4 it is proved that every essential type system induces a λ-model, under a suitable notion of semantic interpretation. In Section 5 we discuss the interpretation of terms in essential models. In Section 6 we recall the class of models in the category $M\mathbb{R}el$, formalized by Bucciarelli, Erhard and Manzonetto, and in Section 7 we prove that the essential models are a correct and complete logical description of them. Section 8 contains a proof that all theories induced by essential models are sensible. In Section 9, we give some final considerations and sketch some future developments.

2. λ-algebras and λ-models

We recall the λ-calculus syntax, and the notions of λ-algebra and λ-model, following the definitions provided in Barendregt (1984).

Definition 1.

(1) *Terms* of λ-calculus are defined by the following syntax:

$$\mathtt{M} ::= \mathtt{x} \mid \lambda\mathtt{x}.\mathtt{M} \mid \mathtt{MM}$$

where $\mathtt{x}$ belongs to Var, i.e. a countable set of variables. Variables are ranged over by $\mathtt{x}, \mathtt{y}, \mathtt{z}$ and terms by $\mathtt{M}, \mathtt{N}, \mathtt{P}$. The set of terms is denoted by Λ, the set of closed terms by Λ^0. $\mathtt{FV(M)}$ denotes the set of free variables of $\mathtt{M}$.

(2) *Contexts* are generated by adding a special constant $[.]$ (the *hole*) to the above term-grammar. $\mathbb{C}[\mathtt{M}]$ denotes the term obtained from $\mathbb{C}[.]$ by replacing the hole by the term $\mathtt{M}$. Note that free variables of $\mathtt{M}$ could be captured by the replacement.

(3) The *α-equality* is the contextual closure of the following rule:

$$\lambda\mathtt{x}.\mathtt{M} =_\alpha \lambda\mathtt{y}.\mathtt{M}[\mathtt{y}/\mathtt{x}]$$

where $\mathtt{y}$ is fresh. As usual, terms are consider modulo $=_\alpha$.

(4) *β-reduction* is the contextual closure of the following rule:

$$(\lambda\mathtt{x}.\mathtt{M})\mathtt{N} \to_\beta \mathtt{M}[\mathtt{N}/\mathtt{x}]$$

where $\mathtt{M}[\mathtt{N}/\mathtt{x}]$ denotes the capture free substitution of $\mathtt{N}$ to all occurrences of $\mathtt{x}$ in $\mathtt{M}$. We denote by $=_\beta$ the minimal congruence induced by $\to_\beta$. We denote by $=_\eta$ the least congruence satisfying also $\mathtt{M} =_\eta \lambda\mathtt{x}.\mathtt{Mx}$ if $\mathtt{x} \notin \mathtt{FV(M)}$.

$\equiv$ is an alternative notation for $=_\alpha$. $\mathtt{FV(M)}$ denotes the set of variables occurring free in $\mathtt{M}$. Many equivalent notions of λ-algebra and λ-model are present in the literature. We will use the syntactical characterization given by Barendregt (1984), Ch. 5.3, based on definition of syntactical λ-model given by Hindley and Longo (1980).

Let $\mathbb{D}$ be a set. An *environment* is a function from Var to $\mathbb{D}$. Environments are ranged over by ρ, ρ'. We denote by $\mathbb{E}$ the collection of all environments. If ρ is an environment $\mathtt{x} \in$ Var and $d \in \mathbb{D}$, then we denote with $\rho[d/\mathtt{x}]$ the environment ρ' such that $\rho'(\mathtt{x}) = d$ while $\rho'(\mathtt{y}) = \rho(\mathtt{y})$ for all $\mathtt{y} \neq \mathtt{x}$.

Definition 2. (See Barendregt (1984))

(1) A *syntactical λ-algebra* is a triple $\mathcal{A} = \langle \mathbb{D}, \circ, [\![.]\!]^{\mathcal{A}} \rangle$, such that $\mathbb{D}$ is a set (the *carrier set*) and $\circ$ is a total map from $\mathbb{D}^2$ to $\mathbb{D}$. The *interpretation function* $[\![.]\!]^{\mathcal{A}} : \Lambda \times \mathbb{E} \to \mathbb{D}$ satisfies the following conditions:

   (a) $[\![\mathtt{x}]\!]^{\mathcal{A}}_\rho = \rho(\mathtt{x})$;

   (b) $[\![\mathtt{MN}]\!]^{\mathcal{A}}_\rho = [\![\mathtt{M}]\!]^{\mathcal{A}}_\rho \circ [\![\mathtt{N}]\!]^{\mathcal{A}}_\rho$;

   (c) $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathcal{A}}_\rho \circ d = [\![\mathtt{M}]\!]^{\mathcal{A}}_{\rho[d/\mathtt{x}]}$;

   (d) $\forall \mathtt{x} \in \mathtt{FV(M)}.\rho(\mathtt{x}) = \rho'(\mathtt{x})$ implies $[\![\mathtt{M}]\!]^{\mathcal{A}}_\rho = [\![\mathtt{M}]\!]^{\mathcal{A}}_{\rho'}$;

   (e) $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathcal{A}}_\rho = [\![\lambda\mathtt{y}.\mathtt{M}[\mathtt{y}/\mathtt{x}]]\!]^{\mathcal{A}}_\rho$, if $\mathtt{y} \notin \mathtt{FV(M)}$;

(f) if $\mathtt{M} =_\beta \mathtt{N}$ then $\forall\rho.[\![\mathtt{M}]\!]^{\mathcal{A}}_\rho = [\![\mathtt{N}]\!]^{\mathcal{A}}_\rho$.

(2) A syntactical λ-algebra $\mathcal{A} = \langle \mathbb{D}, \circ, [\![.]\!]^{\mathcal{A}} \rangle$ is a *syntactical λ-model* if the interpretation function $[\![.]\!]^{\mathcal{A}}$ satisfies the further requirement:

(g) if $[\![\mathtt{M}]\!]^{\mathcal{A}}_{\rho[d/\mathtt{x}]} = [\![\mathtt{M}']\!]^{\mathcal{A}}_{\rho[d/\mathtt{x}]}$ for each $d \in \mathbb{D}$, then $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathcal{A}}_\rho = [\![\lambda\mathtt{x}.\mathtt{M}']\!]^{\mathcal{A}}_\rho$.

Some comments are in order. The definition ensures that the interpretation function respects some elementary key properties, namely that the syntactical substitution is modeled by the environment (conditions (a) and (c)), that the interpretation of a term depends only on the behavior of the environment on its free variables (condition (d)), that α and β-equalities are respected (conditions (d) and (f)), that the interpretation is closed under applicative contexts (condition (b)). The further request (g) makes the interpretation closed also under abstraction contexts, thus the induced semantic equivalence:

$$\mathtt{M} =_{\mathcal{A}} \mathtt{N} \text{ if and only if } \forall\rho.[\![\mathtt{M}]\!]^{\mathcal{A}}_\rho = [\![\mathtt{N}]\!]^{\mathcal{A}}_\rho$$

is a lambda-theory, i.e., a congruence on terms closed under β-equality. As usual, the lambda-theory $=_{\mathcal{A}}$ will be called simply the theory of the model $\mathcal{A}$. A lambda theory is *extensional* if and only if it is closed under $=_\eta$.

3. The class of Essential Type Assignment Systems

We introduce a class of type assignment systems for the λ-calculus, that we call *essential.* We prove that, under a given condition, subject reduction and expansion lemmas hold. Types are built through two connectives: *arrow* ($\to$) and *intersection* ($\wedge$). The key characteristics of these systems are the non-idempotency of intersection and the lack of weakening rule.

Definition 3.

(1) Let C be a non empty set of *type constants*, ranged over by a, b, c. The set of *types* $\mathsf{T}(C)$ and the set of *intersections* $\mathsf{M}(C)$ are mutually defined by the following grammar:

$$\begin{array}{lcll} \sigma, \tau, \pi & ::= & a \mid \omega \to \sigma \mid \mu \to \sigma & \text{(types)} \\ \mu, \nu & ::= & \sigma \mid \mu \wedge \nu & \text{(intersections)} \end{array}$$

where a belongs to the set C. Note that ω is not a type constant. In order to avoid a redundant use of parentheses, we assume an order relation between connectives, imposing $\wedge$ takes precedence over $\to$: so $\sigma_1 \wedge \sigma_2 \to \tau$ stands for $(\sigma_1 \wedge \sigma_2) \to \tau$.

(2) Let $=$ denote the syntactical equality on types and intersections, modulo commutativity and associativity of $\wedge$, i.e., $\mu \wedge \nu = \nu \wedge \mu$, and $(\sigma \wedge \tau) \wedge \pi = \sigma \wedge (\tau \wedge \pi)$. Henceforth, types and intersections will be always considered modulo $=$. To consider intersection modulo associativity allows us to use it as an n-ary connective ($n \geq 0$), so $(\sigma \wedge \tau) \wedge \pi$ and $\sigma \wedge (\tau \wedge \pi)$ are both denoted by $\sigma \wedge \tau \wedge \pi$. To consider intersection modulo commutativity allows us to freely permute types in a n-ary intersection.

(3) An *essential type theory* $\simeq$ is a congruence relation on types and intersections (including the syntactical equality).

We will show that, in order to build a λ-model, we need to restrict ourselves to consider essential types theories which are legal. Roughly speaking, an essential type theory is legal if it respects the non-idempotency of the intersection, so forcing intersections to model multisets (of types). Then, morally, an intersection joins non empty multisets of types and ω represents the empty multiset.

Definition 4 (Legality). A *legal essential type theory* is an essential type theory $\simeq$ satisfying the following conditions:

- $\mu \simeq \nu$ implies $\mu = \sigma_1 \wedge \ldots \wedge \sigma_n$, $\nu = \tau_1 \wedge \ldots \wedge \tau_m$, $n = m$, and $\sigma_i \simeq \tau_i$ $(1 \leq i \leq n)$;
- $\mu \to \tau \simeq \nu \to \tau'$ implies $\mu \simeq \nu$ and $\tau \simeq \tau'$;
- $\omega \to \sigma \simeq \omega \to \tau$ implies $\sigma \simeq \tau$.

Proposition 1. Let $\simeq$ be a legal essential type theory. If $\sigma \simeq \sigma'$ then one of the following cases holds:

(1) there is $a \in C$ such that either $\sigma = a$ or $\sigma' = a$;
(2) $\sigma = \omega \to \tau$ and $\sigma' = \omega \to \tau'$, with $\tau \simeq \tau'$;
(3) $\sigma = \sigma_1 \wedge ... \wedge \sigma_n \to \sigma_0$ and $\sigma' = \sigma'_1 \wedge ... \wedge \sigma'_n \to \sigma'_0$ such that $\sigma_i \simeq \sigma'_i$, for all $0 \leq i \leq n$, $n \geq 0$.

Proof. The proof follows easily from the definition of $\simeq$ and by the fact that it is a congruence. □

Now we can introduce a new class of intersection type assignment systems. Up to details, the type systems in Coppo et al. (1981) and de Carvalho (2009) can be considered instances of this class.

Definition 5.

(1) An *essential type system* ∇ is a pair $\langle C, \simeq \rangle$, where C is a set of type constants and $\simeq$ is a legal essential type theory on $\mathsf{T}(C)$ and $\mathsf{M}(C)$.
(2) A *basis* $\mathtt{B}$ is a partial function from variables to $\mathsf{M}(C)$, such that $\mathtt{B}(\mathtt{x})$ is defined only for finitely many variables. We denote with $\mathtt{dom(B)}$ the domain of $\mathtt{B}$. The relation $\simeq$ can be easily extended to basis in the following way:

$$\mathtt{B} \simeq \mathtt{B}' \text{ if and only if } \mathtt{dom(B)} = \mathtt{dom(B')} \text{ and, for all } \mathtt{x} \in \mathtt{dom(B)},\ \mathtt{B(x)} \simeq \mathtt{B'(x)}.$$

(3) Let ∇ be a type system. The ∇-*essential type assignment system* is a formal system proving statements of the shape:

$$\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$$

where $\mathtt{M}$ is a term, $\sigma \in \mathsf{T}(C)$ and $\mathtt{B}$ is a basis. The rules of the system are given in Table 1 which uses notations defined in Notation 1.

Notation 1. If $\mu = \sigma_1 \wedge \ldots \wedge \sigma_n$, we can write alternatively μ as $\wedge_{1 \leq i \leq n} \sigma_i$. If ∇ is an essential type system then C_∇, T_∇ and M_∇ denote respectively its set of constants, types and intersections and $\simeq_\nabla$ denotes its essential type theory. Let $\mathtt{B}$ be a basis, we denote by $\mathtt{B}, \mathtt{x} : \mu$ the basis $\mathtt{B}'$ such that $\mathtt{x} \notin \mathtt{dom(B)}$, $\mathtt{B'(x)} = \mu$ and $\mathtt{B'(y)} = \mathtt{B(y)}$ for all $\mathtt{y} \neq \mathtt{x}$.

Table 1. The Type Assignment Systems $\vdash_\nabla$.

$$\begin{array}{cc}
\dfrac{}{\mathtt{x}:\sigma \vdash_\nabla \mathtt{x}:\sigma}\ (\mathrm{var}) & \dfrac{\mathtt{B} \vdash_\nabla \mathtt{M}:\tau \quad \sigma \simeq_\nabla \tau}{\mathtt{B} \vdash_\nabla \mathtt{M}:\sigma}\ (\simeq_\nabla) \\
\dfrac{\mathtt{B}_0 \vdash_\nabla \mathtt{M}:\sigma_1 \wedge \ldots \wedge \sigma_n \to \sigma \quad \left(\mathtt{B}_i \vdash_\nabla \mathtt{N}:\sigma_i\right)_{1\leq i\leq n}}{\left(\bigwedge_{0\leq i\leq n} \mathtt{B}_i\right) \vdash_\nabla \mathtt{MN}:\sigma}\ (\to E) & \dfrac{\mathtt{B},\mathtt{x}:\sigma_1 \wedge \ldots \wedge \sigma_n \vdash_\nabla \mathtt{M}:\tau}{\mathtt{B} \vdash_\nabla \lambda\mathtt{x}.\mathtt{M}:\sigma_1 \wedge \ldots \wedge \sigma_n \to \tau}\ (\to I) \\
\dfrac{\mathtt{B} \vdash_\nabla \mathtt{M}:\omega \to \sigma}{\mathtt{B} \vdash_\nabla \mathtt{MN}:\sigma}\ (\to E_\omega) & \dfrac{\mathtt{B} \vdash_\nabla \mathtt{M}:\tau \quad \mathtt{x} \notin \mathtt{dom(B)}}{\mathtt{B} \vdash_\nabla \lambda\mathtt{x}.\mathtt{M}:\omega \to \tau}\ (\to I_\omega)
\end{array}$$

$\emptyset$ denotes the basis mapping every variable to the undefined element. We write $\mathtt{x}:\mu$ in place of $\emptyset, \mathtt{x}:\mu$. If $\mathtt{B}_1, \mathtt{B}_2$ are two bases then $\mathtt{B}_1 \bigwedge \mathtt{B}_2$ is the basis such that

$$(\mathtt{B}_1 \textstyle\bigwedge \mathtt{B}_2)(\mathtt{x}) = \begin{cases} \mathtt{B}_1(\mathtt{x}) & \text{if } \mathtt{x} \in \mathtt{dom}(\mathtt{B}_1) \quad \text{and } \mathtt{x} \notin \mathtt{dom}(\mathtt{B}_2) \\ \mathtt{B}_2(\mathtt{x}) & \text{if } \mathtt{x} \in \mathtt{dom}(\mathtt{B}_2) \quad \text{and } \mathtt{x} \notin \mathtt{dom}(\mathtt{B}_1) \\ \mathtt{B}_1(\mathtt{x}) \wedge \mathtt{B}_2(\mathtt{x}) & \text{if } \mathtt{x} \in \mathtt{dom}(\mathtt{B}_1) \quad \text{and } \mathtt{x} \in \mathtt{dom}(\mathtt{B}_2) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Thanks to associativity, $\bigwedge$ can be extended to an n-ary operator in a straightforward way, for $n \geq 0$. Sometimes we will write $\mathtt{B} \bigwedge_i \mathtt{B}_i$ as an abbreviation for $\mathtt{B} \bigwedge (\bigwedge_i \mathtt{B}_i)$. Derivations are ranged over by Π, Σ, and $\Pi \rhd \mathtt{B} \vdash_\nabla \mathtt{M}:\sigma$ denotes a derivation whose conclusion is $\mathtt{B} \vdash_\nabla \mathtt{M}:\sigma$. The term $\mathtt{M}$ is commonly called *the subject of* Π.

Table 1 defines a parametric type assignment system, where the parameter is an essential type system ∇. Every choice of ∇, i.e., of a set of type constants and of a legal essential type theory, provides a particular instance of essential systems. Note that the system $\vdash_\nabla$ does not allow weakening, i.e. it is *relevant*. The rules are quite standard. We remark that $(\to E)$ is quantitative, in the sense that the number of derivations for the arguments is exactly the number of the components of the intersection in the left part of the type in functional position. Moreover, in rule $(\to E_\omega)$, the argument of the application in the subject is not asked to have a type, so also terms with unsolvable subterms can be typed, e.g. $\mathtt{y}((\lambda\mathtt{x}.\mathtt{xx})(\lambda\mathtt{x}.\mathtt{xx}))$.

Lemma 1 (Generation). Let ∇ be an essential type system.

(1) $\mathtt{B} \vdash_\nabla \mathtt{x}:\sigma$ implies $\mathtt{dom}(\mathtt{B}) = \{\mathtt{x}\}$ and $\sigma \simeq_\nabla \mathtt{B}(\mathtt{x})$;

(2) $\mathtt{B} \vdash_\nabla \mathtt{M}:\sigma$ implies $\mathtt{dom}(\mathtt{B}) \subseteq \mathtt{FV}(\mathtt{M})$;

(3) $\mathtt{B} \vdash_\nabla \lambda\mathtt{x}.\mathtt{M}:\sigma$ implies either $\sigma \simeq_\nabla \omega \to \tau$ or $\sigma \simeq_\nabla \mu \to \tau$ for some μ, τ;

(4) $\mathtt{B} \vdash_\nabla \mathtt{MN}:\sigma$ implies either $\mathtt{B} \vdash_\nabla \mathtt{M}:\omega \to \sigma$ or there are $\mathtt{B}_0, \ldots, \mathtt{B}_n$ $(n \geq 0)$ such that $\mathtt{B}_0 \vdash_\nabla \mathtt{M}:\sigma_1 \wedge \ldots \wedge \sigma_n \to \sigma$, $\mathtt{B}_i \vdash_\nabla \mathtt{N}:\sigma_i$ $(1 \leq i \leq n)$ and $\mathtt{B} = (\bigwedge_{0\leq i\leq n} \mathtt{B}_i)$;

(5) $\mathtt{B} \vdash_\nabla \mathtt{M}:\sigma$ and $\mathtt{B} \simeq_\nabla \mathtt{B}'$ imply $\mathtt{B}' \vdash_\nabla \mathtt{M}:\sigma$;

(6) If $\simeq_\nabla$ is a legal essential type theory, then $\mathtt{B} \vdash_\nabla \lambda\mathtt{x}.\mathtt{M}:\sigma_1 \wedge \ldots \wedge \sigma_n \to \tau$ if and only if $\mathtt{B}, \mathtt{x}:\sigma_1 \wedge \ldots \wedge \sigma_n \vdash_\nabla \mathtt{M}:\tau$ $(n \geq 1)$.

(7) If $\simeq_\nabla$ is a legal essential type theory, then $\mathtt{B} \vdash_\nabla \lambda\mathtt{x}.\mathtt{M}:\omega \to \tau$ if and only if $\mathtt{B} \vdash_\nabla \mathtt{M}:\tau$ with $\mathtt{x} \notin \mathtt{dom}(\mathtt{B})$.

Proof. We just discuss some points.

- Proof of point (2). By induction on the derivation. Let the last applied rule be $(\rightarrow E_\omega)$:

$$\dfrac{\mathtt{B} \vdash_\nabla \mathtt{M} : \omega \rightarrow \sigma}{\mathtt{B} \vdash_\nabla \mathtt{MN} : \sigma}\ (\rightarrow E_\omega)$$

Remember that $\mathtt{FV(MN)} = \mathtt{FV(M)} \cup \mathtt{FV(N)}$. By induction, $\mathtt{dom(B)} \subseteq \mathtt{FV(M)}$. Since the final judgment is $\mathtt{B} \vdash_\nabla \mathtt{MN} : \sigma$, in case there is $\mathtt{x} \in \mathtt{FV(N)} - \mathtt{FV(M)}$, $\mathtt{x} \notin \mathtt{dom(B)}$, so the inclusion can be strict. The other cases follow by induction.

- Proof of point (6). We just prove the left-to-right implication, by induction on the derivation. If the last applied rule is either $(\rightarrow I)$ or $(\rightarrow I_\omega)$, then the proof is immediate. Otherwise, the derivation proving $\mathtt{B} \vdash_\nabla \lambda\mathtt{x.M} : \sigma_1 \wedge ... \wedge \sigma_n \rightarrow \tau$ ends in the following way:

$$\dfrac{\dfrac{\mathtt{B}, \mathtt{x} : \pi_1 \wedge ... \wedge \pi_m \vdash_\nabla \mathtt{M} : \pi}{\mathtt{B} \vdash_\nabla \lambda\mathtt{x.M} : \pi_1 \wedge ... \wedge \pi_m \rightarrow \pi}\ (\rightarrow\mathrm{I})}{\mathtt{B} \vdash_\nabla \lambda\mathtt{x.M} : \sigma_1 \wedge ... \wedge \sigma_n \rightarrow \tau}\ (\simeq)$$

where $\pi_1 \wedge ... \wedge \pi_m \rightarrow \pi \simeq_\nabla \sigma_1 \wedge ... \wedge \sigma_n \rightarrow \tau$ by transitivity of $\simeq_\nabla$ (as usual, the double line in the derivation means many applications of the considered rule). Since $\simeq_\nabla$ is a legal essential type theory, $\pi_1 \wedge ... \wedge \pi_m \simeq_\nabla \sigma_1 \wedge ... \wedge \sigma_n$ and $\pi \simeq_\nabla \tau$, so by definition $m = n$ and $\pi_i \simeq \sigma_i$ $(1 \leq i \leq n)$. Then the proof follows by point (5) of this Lemma.

- Point (7) is similar to the Point (6) but simpler.

The proof of all other points is immediate, by induction on derivation. □

Note that the point (5) of the previous lemma implies that the following is a derived rule:

$$\dfrac{\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma \qquad \mathtt{B} \simeq_\nabla \mathtt{B}'}{\mathtt{B}' \vdash_\nabla \mathtt{M} : \sigma}\ (\simeq^{\mathtt{B}}_\nabla)$$

From now on, where not specified otherwise, we will always consider only legal essential type theories. Every type assignment system $\vdash_\nabla$ satisfies the β-equality on terms. For proving it, first we need to prove that it enjoys the substitution property.

Lemma 2 (Substitution). $\mathtt{B}, \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash_\nabla \mathtt{M} : \tau$ and $\mathtt{B}_i \vdash_\nabla \mathtt{N} : \sigma_i$ $(1 \leq i \leq n)$ imply $\mathtt{B} \bigwedge_{1 \leq i \leq n} \mathtt{B}_i \vdash_\nabla \mathtt{M[N/x]} : \tau$ $(n \geq 0)$.

Proof. By induction on the derivation proving $\mathtt{B}, \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash_\nabla \mathtt{M} : \tau$. The only non obvious case is when the last applied rule is $(\rightarrow E)$. Then $\mathtt{M} \equiv \mathtt{PQ}$, and the derivation ends by:

$$\dfrac{\mathtt{B}_0, \mathtt{x} : \tau_1 \wedge ... \wedge \tau_p \vdash_\nabla \mathtt{P} : \pi_1 \wedge ... \wedge \pi_m \rightarrow \sigma \quad \left(\mathtt{B}^j, \mathtt{x} : \tau^j_1 \wedge ... \wedge \tau^j_{q_j} \vdash_\nabla \mathtt{Q} : \pi_j\right)_{1 \leq j \leq m}}{\left(\mathtt{B}_0 \bigwedge_{1 \leq j \leq m} \mathtt{B}^j\right), \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash_\nabla \mathtt{PQ} : \sigma}\ (\rightarrow E)$$

where $\sigma_1 \wedge ... \wedge \sigma_n = \tau_1 \wedge ... \wedge \tau_p \wedge \tau^1_1 \wedge ... \wedge \tau^1_{q_1} \wedge ... \wedge \tau^m_1 \wedge ... \wedge \tau^m_{q_m}$. So $n = p + \Sigma_{1 \leq j \leq m} q_j$, and σ_i coincides either with τ_r or with τ^h_k, for some r, h, k $(1 \leq r \leq p, 1 \leq h \leq m, 1 \leq k \leq q_h)$. Let $\mathtt{B}_h \vdash_\nabla \mathtt{N} : \tau_h$ $(1 \leq h \leq p)$ and $\mathtt{B}^j_{k_j} \vdash_\nabla \mathtt{N} : \tau^j_{k_j}$ $(1 \leq j \leq m, 1 \leq k_j \leq q_j)$. By induction we have that

(1)$\mathtt{B}_0 \bigwedge_{1 \leq h \leq p} \mathtt{B}_h \vdash_\nabla \mathtt{P[N/x]} : \pi_1 \wedge ... \wedge \pi_\mathtt{m} \rightarrow \sigma$, (2)$\mathtt{B}^j \bigwedge_{1\leq k_j\leq q_j} \mathtt{B}^j_{k_j} \vdash_\nabla \mathtt{Q[N/x]} : \pi_j$ $(1 \leq j \leq m)$.

Then it follows that $\mathtt{B}_0 \bigwedge_{1\leq h\leq p} \mathtt{B}_h \bigwedge_{1\leq j\leq m}(\mathtt{B}^j \bigwedge_{1\leq k_j\leq q_j} \mathtt{B}^j_{k_j}) \vdash_\nabla \mathtt{P[N/x]Q[N/x]} : \sigma$ by rule $(\to{}_E)$ and the proof is given, since $\mathtt{(PQ)[N/x]} =_\alpha \mathtt{P[N/x]Q[N/x]}$. □

Lemma 3 (Subject Reduction). $\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$ and $\mathtt{M} \to_\beta \mathtt{M}'$ imply $\mathtt{B} \vdash_\nabla \mathtt{M}' : \sigma$.

Proof. $\mathtt{M} \to_\beta \mathtt{M}'$ means $\mathtt{M} \equiv \mathbb{C}[(\lambda\mathtt{x.Q})\mathtt{N}]$ and $\mathtt{M}' \equiv \mathbb{C}[\mathtt{Q[N/x]}]$. The proof is by induction on $\mathbb{C}[.]$. The case $\mathbb{C}[.] \equiv [.]$ follows from Lemmas 1.(4) and Substitution. Inductive cases are straightforward. □

Lemma 4 (Subject Expansion). $\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$ and $\mathtt{M}' \to_\beta \mathtt{M}$ imply $\mathtt{B} \vdash_\nabla \mathtt{M}' : \sigma$.

Proof. $\mathtt{M}' \to_\beta \mathtt{M}$ means $\mathtt{M}' \equiv \mathbb{C}[(\lambda\mathtt{x.Q})\mathtt{N}]$ and $\mathtt{M} \equiv \mathbb{C}[\mathtt{Q[N/x]}]$. The proof is by induction on $\mathbb{C}[.]$.
Let us consider the basic case $\mathbb{C}[.] \equiv [.]$. Let $\Pi \rhd \mathtt{B} \vdash_\nabla \mathtt{Q[N/x]} : \sigma$ and let $\Pi_i \rhd \mathtt{B}_i \vdash_\nabla \mathtt{N} : \tau_i$ be the sub-derivations of Π whose subject is $\mathtt{N}$ for some $0 \leq i \leq n$ (note that it is possible that $n = 0$). W.l.o.g. we can assume that $\mathtt{x}$ does not occur in $\mathtt{N}$, so $\mathtt{x} \notin \mathtt{dom(B}_i)$ by Lemma 1.(2). Thus, Π can be transformed into a derivation $\Pi' \rhd \mathtt{B}'' \vdash_\nabla \mathtt{Q} : \sigma$ just by replacing every Π_i by the derivation $\mathtt{x} : \tau_i \vdash_\nabla \mathtt{x} : \tau_i$ and by defining $\mathtt{B}''$ as follows. If $n = 0$ then $\mathtt{B}'' = \mathtt{B}$ and Π' is the following derivation:

$$\dfrac{\dfrac{\mathtt{B} \vdash_\nabla \mathtt{Q} : \sigma}{\mathtt{B} \vdash_\nabla \lambda\mathtt{x.Q} : \omega \to \sigma}\,(\to I_\omega)}{\mathtt{B} \vdash_\nabla (\lambda\mathtt{x.Q})\mathtt{N} : \sigma}\,(\to E_\omega)$$

Otherwise, let $\{\mathtt{y}_1, ..., \mathtt{y}_h\} = \mathtt{FV(N)}$.
Then we choose $\mathtt{B}''$ such that $\mathtt{x} \notin \mathtt{dom(B}'')$ and $\mathtt{B(y}_k) = (\bigwedge_{0\leq i\leq n} \mathtt{B}_i(\mathtt{y}_k)) \bigwedge \mathtt{B}''(\mathtt{y}_k)$ $(0 \leq k \leq h)$, while $\mathtt{B(z)} = \mathtt{B}''(\mathtt{z})$, for $\mathtt{z} \notin \mathtt{FV(N)}$. So we can build the following derivation:

$$\dfrac{\dfrac{\begin{matrix}\Pi' \\ \vdots \\ \mathtt{B}'', \mathtt{x} : \tau_1 \wedge ... \wedge \tau_n \vdash_\nabla \mathtt{Q} : \sigma\end{matrix}}{\mathtt{B}'' \vdash_\nabla \lambda\mathtt{x.Q} : \tau_1 \wedge ... \wedge \tau_n \to \sigma}\,(\to I) \qquad (\mathtt{B}_i \vdash_\nabla \mathtt{N}_i : \tau_i)_{0\leq i\leq n}}{\mathtt{B}'' \bigwedge_{0\leq i\leq n} \mathtt{B}_i \vdash_\nabla (\lambda\mathtt{x.Q})\mathtt{N} : \sigma}\,(\to E)$$

Since $\mathtt{B} = \mathtt{B}'' \bigwedge_{0\leq i\leq n} \mathtt{B}_i$, the proof is given. The induction case is straightforward. □

Next theorem trivially follows by Lemmas 3 and 4.

Theorem 1. The type assignment system $\vdash_\nabla$ is closed under $=_\beta$.

4. Essential Logical Models

In this section we prove that each legal essential type system induces a λ-model. The main difference with respect to the usual logical model construction is that terms are not interpreted as sets of types, but are interpreted as sets of *typings*. A typing is a pair $(\mathtt{B}; \sigma)$ such that $\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$ for some term $\mathtt{M}$, and a typing for $\mathtt{M}$ is a pair $(\mathtt{B}; \sigma)$ such that

$\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$. Then the interpretation domain is the set of pre-typings, i.e., pairs whose components are a basis and a type, in principle unrelated from each other.

Definition 6. Let ∇ be a legal essential type system.

(1) A *pre-typing* is a pair $(\mathtt{B};\sigma)$, where $\mathtt{B}$ is a basis and σ is a type. The equivalence $\simeq_\nabla$ can be extended to pre-typings in the following way:

$$(\mathtt{B};\sigma) \simeq_\nabla (\mathtt{B}';\tau) \quad \text{if and only if} \quad \sigma \simeq_\nabla \tau \text{ and } \mathtt{B} \simeq_\nabla \mathtt{B}'.$$

A *pre-typing set* s of ∇ is a set of pre-typings closed under $\simeq_\nabla$ such that the set $\bigcup_{(\mathtt{B},\sigma)\in s} \mathtt{dom}(\mathtt{B})$ is finite. Pre-typings sets are ranged over by s, t. Let $\mathfrak{T}(\nabla)$ be the family of all pre-typing sets.

(2) $\circ_\nabla$ is a binary operation defined on $\mathfrak{T}(\nabla)$ in the following way:

$$s_1 \circ_\nabla s_2 = \{(\mathtt{B};\sigma) \mid (\mathtt{B};\omega \to \sigma) \in s_1\} \bigcup \left\{ (\bigwedge_{0\le i\le n} \mathtt{B}_i;\sigma) \;\middle|\; \begin{array}{l} (\mathtt{B}_0;\sigma_1 \wedge ... \wedge \sigma_n \to \sigma) \in s_1, \\ (\mathtt{B}_i;\sigma_i) \in s_2 \quad (1 \le i \le n, n > 0) \end{array} \right\}$$

The set of pre-typings is closed under composition.

Lemma 5. Let ∇ be a legal essential type system. If $s_1, s_2 \in \mathfrak{T}(\nabla)$ then $s_1 \circ_\nabla s_2 \in \mathfrak{T}(\nabla)$.

Proof. Obvious. □

Terms are interpreted as set of *pre-typings*, closed under the relation $\simeq_\nabla$. Two terms $\mathtt{M}$ and $\mathtt{N}$ will be said denotationally equal if and only if they share the same typings.

Let an *environment* be a function from Var to $\mathfrak{T}(\nabla)$, and let $\mathbb{E}_\nabla$ be the set of environments. The interpretation function $[\![.]\!]^\nabla : \Lambda \longrightarrow \mathbb{E}_\nabla \longrightarrow \mathfrak{T}(\nabla)$ is defined as follows.

$$[\![\mathtt{M}]\!]^\nabla_\rho = \left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}' \vdash_\nabla \mathtt{M} : \sigma \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1, ..., \mathtt{x}_n\}, \mathtt{B}'(\mathtt{x}_i) = \sigma^i_1 \wedge \ldots \wedge \sigma^i_{n_i} (n_i \ge 1), \\ \text{and } \exists \mathtt{B}^i_j \text{ s.t. } (\mathtt{B}^i_j;\sigma^i_j) \in \rho(\mathtt{x}_i) \text{ and } \mathtt{B} = \bigwedge_{1\le i\le n}(\bigwedge_{1\le j\le n_i} \mathtt{B}^i_j) \end{array} \right\}$$

Let $\mathcal{M}^\nabla = \langle \mathfrak{T}(\nabla), \circ_\nabla, [\![.]\!]^\nabla \rangle$. We prove that each type system ∇ induces a model $\mathcal{M}^\nabla$ of λ-calculus, according to Definition 2.

Theorem 2. Let ∇ be a type system. Then $\mathcal{M}^\nabla$ is a λ-model.

Proof. We need to prove that all conditions of Definition 2 are satisfied.

$$\text{(a)}\ \ [\![\mathtt{x}]\!]^\nabla_\rho = \left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}' \vdash_\nabla \mathtt{x} : \sigma \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1, ..., \mathtt{x}_n\}, \mathtt{B}'(\mathtt{x}_i) = \sigma^i_1 \wedge \ldots \wedge \sigma^i_{n_i} \\ \text{and } \mathtt{B} = \bigwedge_{1\le i\le n} \bigwedge_{1\le j\le n_i} \mathtt{B}^i_j \text{ where } (\mathtt{B}^i_j;\sigma^i_j) \in \rho(\mathtt{x}_i) \end{array} \right\}$$

$$= \{(\mathtt{B};\sigma) | \mathtt{x} : \tau \vdash_\nabla \mathtt{x} : \sigma \text{ where } \tau \simeq_\nabla \sigma \text{ and } (\mathtt{B};\tau) \in \rho(\mathtt{x})\} = \rho(\mathtt{x})$$

where the first equality follows by definition, the second follows by Lemma 1.1, and the third follows by the fact that typing sets are closed under $\simeq_\nabla$.

(b) $[\![\mathtt{MN}]\!]^{\nabla}_{\rho} =$

$$\left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}' \vdash_{\nabla} \mathtt{MN} : \sigma \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1,...,\mathtt{x}_n\}, \mathtt{B}'(\mathtt{x}_i) = \sigma^i_1 \wedge \ldots \wedge \sigma^i_{n_i} \text{ and} \\ \mathtt{B} = \bigwedge_{1\le i\le n}\bigwedge_{1\le j\le n_i} \mathtt{B}^i_j \text{ where } (\mathtt{B}^i_j;\sigma^i_j) \in \rho(\mathtt{x}_i) \end{array} \right\} =$$

$$\left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}' \vdash_{\nabla} \mathtt{M} : \omega \to \sigma \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1,...,\mathtt{x}_n\}, \mathtt{B}'(\mathtt{x}_i) = \sigma^i_1 \wedge \ldots \wedge \sigma^i_{n_i} \text{ and} \\ \mathtt{B} = \bigwedge_{1\le i\le n}\bigwedge_{1\le j\le n_i} \mathtt{B}^i_j \text{ where } (\mathtt{B}^i_j;\sigma^i_j) \in \rho(\mathtt{x}_i) \end{array} \right\} \bigcup$$

$$\left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}'_0 \vdash_{\nabla} \mathtt{M} : \sigma_1 \wedge \ldots \wedge \sigma_m \to \sigma, \mathtt{B}_h \vdash_{\nabla} \mathtt{N} : \sigma_h (1 \le h \le m), \\ \mathtt{B}' = \mathtt{B}_0 \bigwedge_{1\le h\le m} \mathtt{B}_h \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1,...,\mathtt{x}_n\}, \mathtt{B}'(\mathtt{x}_i) = \sigma^i_1 \wedge \ldots \wedge \sigma^i_{n_i} \\ \text{ and } \mathtt{B} = \bigwedge_{1\le i\le n}\bigwedge_{1\le j\le n_i} \mathtt{B}^i_j \text{ where } (\mathtt{B}^i_j;\sigma^i_j) \in \rho(\mathtt{x}_i) \end{array} \right\} =$$

$$[\![\mathtt{M}]\!]^{\nabla}_{\rho} \circ [\![\mathtt{N}]\!]^{\nabla}_{\rho}$$

where the first equality follows by definition, the second equality follows by Lemma 1.4. and the last equality follows by definition of $\circ_{\nabla}$.

(c) $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\nabla}_{\rho} \circ_{\nabla} s =$

$$\{(\mathtt{B};\sigma) \mid (\mathtt{B};\omega \to \sigma) \in [\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\nabla}_{\rho}\} \bigcup \left\{ (\bigwedge_{i=0}^{n} \mathtt{B}_i;\sigma) \;\middle|\; \begin{array}{l} (\mathtt{B}_0;\sigma_1 \wedge ... \wedge \sigma_n \to \sigma) \in [\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\nabla}_{\rho}, \\ (\mathtt{B}_i;\sigma_i) \in s \quad (1 \le i \le n, n > 0) \end{array} \right\} =$$

$$\left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}' \vdash_{\nabla} \lambda\mathtt{x}.\mathtt{M} : \omega \to \sigma, \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1,\ldots,\mathtt{x}_q\}, \mathtt{B}'(\mathtt{x}_j) = \sigma^j_1 \wedge \ldots \wedge \sigma^j_{q_j} \\ \text{ and } \mathtt{B} = \bigwedge_{1\le j\le q}\bigwedge_{1\le l\le q_j} \mathtt{B}^j_l \text{ where } (\mathtt{B}^j_l;\sigma^j_l) \in \rho(\mathtt{x}_i) \end{array} \right\} \bigcup$$

$$\left\{ (\bigwedge_{i=0}^{n} B_i;\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}'' \vdash_{\nabla} \lambda\mathtt{x}.\mathtt{M} : \sigma_1 \wedge ... \wedge \sigma_n \to \sigma, \text{ where } \mathtt{dom}(\mathtt{B}'') = \{\mathtt{y}_1,\ldots,\mathtt{y}_p\}, \\ \mathtt{B}''(\mathtt{y}_h) = \sigma^h_1 \wedge \ldots \wedge \sigma^h_{n_h} \text{ and} \\ \mathtt{B}_0 = \bigwedge_{1\le h\le p}\bigwedge_{1\le k\le n_h} \mathtt{B}^h_k \text{ where } (\mathtt{B}^h_k;\sigma^h_k) \in \rho(\mathtt{y}_h), \\ \text{ and } (\mathtt{B}_i;\sigma_i) \in s \quad (1 \le i \le n, n > 0) \end{array} \right\} =$$

$$\left\{ (\mathtt{B};\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}' \vdash_{\nabla} \mathtt{M} : \sigma, \text{ where } \mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1,\ldots,\mathtt{x}_q\}, \mathtt{B}'(\mathtt{x}_j) = \sigma^j_1 \wedge \ldots \wedge \sigma^j_{q_j} \text{ and} \\ \mathtt{B} = \bigwedge_{1\le j\le q}\bigwedge_{1\le l\le q_j} \mathtt{B}^j_l \text{ where } (\mathtt{B}^j_l;\sigma^j_l) \in \rho(\mathtt{x}_i) \end{array} \right\} \bigcup$$

$$\left\{ (\bigwedge_{i=0}^{n} \mathtt{B}_i;\sigma) \;\middle|\; \begin{array}{l} \mathtt{B}'', \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash_{\nabla} \mathtt{M} : \sigma, \text{ where } \mathtt{dom}(\mathtt{B}'') = \{\mathtt{y}_1,\ldots,\mathtt{y}_p\}, \\ \mathtt{B}'(\mathtt{y}_h) = \sigma^h_1 \wedge \ldots \wedge \sigma^h_{n_h} \text{ and} \\ \mathtt{B}_0 = \bigwedge_{1\le h\le p}\bigwedge_{1\le k\le n_h} \mathtt{B}^h_k \text{ where } (\mathtt{B}^h_k;\sigma^h_k) \in \rho(\mathtt{y}_h), \\ \text{ and } (\mathtt{B}_i;\sigma_i) \in s \quad (1 \le i \le n, n > 0) \end{array} \right\} = [\![\mathtt{M}]\!]^{\nabla}_{\rho[s/\mathtt{x}]}$$

where the first equality follows by definition of $\circ_{\nabla}$, the second equality follows by definition of interpretation, the third follows by Lemma 1 (points 3 and 6) and the third equality follows again by definition of $\circ_{\nabla}$.

(d)Obvious.

(e)Obvious.

(f)It follows directly from Theorem 1.

(g)We prove $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\nabla}_{\rho} \subseteq [\![\lambda\mathtt{x}.\mathtt{M}']\!]^{\nabla}_{\rho}$ whenever, $[\![\mathtt{M}]\!]^{\mathcal{A}}_{\rho[s/\mathtt{x}]} = [\![\mathtt{M}']\!]^{\mathcal{A}}_{\rho[s/\mathtt{x}]}$ for each $s \in\mathfrak{T}(\nabla)$, since the other one is symmetric. Let us assume $(\mathtt{B};\sigma) \in [\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\nabla}_{\rho}$. By definition, $\mathtt{B}' \vdash_{\nabla} \lambda\mathtt{x}.\mathtt{M} : \sigma$ for some $\mathtt{B}'$, moreover, if $\mathtt{dom}(\mathtt{B}') = \{\mathtt{x}_1,...,\mathtt{x}_m\}$, then $\mathtt{B}'(\mathtt{x}_j) = \sigma^j_1 \wedge ... \wedge \sigma^j_{n_j}$, such that there is $(\mathtt{B}^j_h;\sigma^j_h) \in \rho(\mathtt{x}_j)$, $\mathtt{B} = \bigwedge_{1\le j\le m}(\bigwedge_{1\le h\le n_j} \mathtt{B}^j_h)$. W.l.o.g. we can assume $\mathtt{x} \neq \mathtt{x}_i$ for all i, since $\mathtt{x}$ can be freely renamed. So either $\sigma \simeq_{\nabla} \omega \to \tau$ and $\mathtt{B}' \vdash_{\nabla} M : \tau$ or

$\sigma \simeq_{\nabla} \sigma_1 \wedge ... \wedge \sigma_n \to \tau$, for some τ and $n \geq 1$, by Lemma 1 (points (3), (6) and (7)). In the first case, let $s = \emptyset$, then $(\mathtt{B};\sigma) \in [\![\mathtt{M}]\!]^{\nabla}_{\rho[s/\mathtt{x}]}$, and consequently $(\mathtt{B};\sigma) \in [\![\mathtt{M}']\!]^{\nabla}_{\rho[s/\mathtt{x}]}$ by hypothesis. So $\mathtt{B}' \vdash_{\nabla} \mathtt{M}' : \tau$, and the proof follows by rules $(\to I_\omega)$ and $(\simeq_\nabla)$.

In the second case, let $s = \{(\mathtt{x} : \sigma_h; \pi_h) \mid \sigma_h \simeq_{\nabla} \pi_h, 1 \leq h \leq n\}$, thus $(\mathtt{B} \wedge \{\mathtt{x} : \sigma_1 \wedge \ldots \wedge \sigma_n\}; \tau) \in [\![\mathtt{M}]\!]^{\nabla}_{\rho[s/\mathtt{x}]}$ and, consequently, $(\mathtt{B} \wedge \{\mathtt{x} : \sigma_1 \wedge \ldots \wedge \sigma_n\}; \tau) \in [\![\mathtt{M}']\!]^{\nabla}_{\rho[s/\mathtt{x}]}$ by hypothesis. Therefore $\mathtt{B}', \mathtt{x} : \sigma_1 \wedge \ldots \wedge \sigma_n \vdash_{\nabla} \mathtt{M}' : \tau$. By applying $(\to I)$ and $(\simeq_\nabla)$, we obtain $\mathtt{B}' \vdash_{\nabla} \lambda \mathtt{x}.\mathtt{M}' : \sigma$. So, the proof follows.

□

So, an essential type assignment system is a model for λ-calculus, which at the same time supply a tool for reasoning about the denotational meaning of terms, and for comparing terms.

Lemma 6.
(1) If $\mathtt{M} \in \Lambda^0$ then $[\![\mathtt{M}]\!]^{\nabla}_{\rho} = \{(\emptyset;\sigma) \mid \emptyset \vdash_{\nabla} \mathtt{M} : \sigma\}$, for all environment ρ.
(2) $\mathtt{M} =_{\mathcal{M}^\nabla} \mathtt{N}$ if and only if, $(\mathtt{B} \vdash_{\nabla} \mathtt{M} : \sigma \Leftrightarrow \mathtt{B} \vdash_{\nabla} \mathtt{N} : \sigma)$ for all pre-typings $(\mathtt{B};\sigma)$.

Proof. Both points follow easily from the definition of interpretation. □

5. Justifying the interpretation choice

We further justify our choice of interpretation by proving that the standard construction of logical models applied to an essential type assignment system gives rise to a λ-algebra, but not to a λ-model. To achieve this goal, we generalize to our parametric class of models a counterexample already presented in de Carvalho (2009) for a particular case.

Let us recall that in the standard approach, the domain of interpretations is a subset of the power set of types: the set of filters, in case of continuous semantics, the power set modulo the coherence equivalence in case of stable function. Then a term is interpreted by the set of types derivable for it. Mimicking this approach, we could define the domain of interpretations as the power set of types, modulo the relation $\simeq$, and interpret a term as the set of its types.

Definition 7. Let ∇ be an essential type system, and let $\wp(\mathsf{T}_\nabla/_{\simeq_\nabla})$ be the power set on $\mathsf{T}_\nabla/_{\simeq_\nabla}$. The composition function $\circ : \wp(\mathsf{T}_\nabla/_{\simeq_\nabla}) \to \wp(\mathsf{T}_\nabla/_{\simeq_\nabla})$ is defined as follows:

$$s \circ s' = \{\sigma \mid \omega \to \sigma \in s\} \bigcup \{\sigma \mid \sigma_1 \wedge ... \wedge \sigma_n \to \sigma \in s, \sigma_i \in s'(1 \leq i \leq n)\}$$

Moreover, if $\mathbb{E}$ is the collection of functions (environments) from Var to $\wp(\mathsf{T}_\nabla/_{\simeq_\nabla})$, ranged over by ρ, then the *interpretation function* $(\!|.|\!)^\nabla$ is defined as:

$$(\!|\mathtt{M}|\!)^{\nabla}_{\rho} = \{\sigma \mid \exists \mathtt{B} \trianglelefteq \rho, \quad \mathtt{B} \vdash_{\nabla} \mathtt{M} : \sigma\} \tag{1}$$

where $\mathtt{B} \trianglelefteq \rho$ means that, for all $\mathtt{x}$, $\mathtt{B}(\mathtt{x}) = \sigma_1 \wedge ... \wedge \sigma_n$ implies $\sigma_i \in \rho(\mathtt{x})$, for all $1 \leq i \leq n$.

We show that the above attempt of interpretation does not provide a lambda-model, but just a lambda-algebra. The same result would be obtained through the correspondence with relational models, given in the next sections, using some observations in

Selinger (2002), but here we would like to supply a proof inside our essential type assignment system.

Theorem 3.

(1) $\langle \wp(\mathsf{T}_{\nabla}/_{\simeq_{\nabla}}), \circ, \llparenthesis . \rrparenthesis^{\nabla} \rangle$ is a syntactical λ-algebra.
(2) $\langle \wp(\mathsf{T}_{\nabla}/_{\simeq_{\nabla}}), \circ, \llparenthesis . \rrparenthesis^{\nabla} \rangle$ is not a syntactical λ-model.

Proof.

(1)We need to prove that $\langle \wp(\mathsf{T}_{\nabla}/_{\simeq_{\nabla}}), \circ, \llparenthesis . \rrparenthesis^{\nabla} \rangle$ satisfies all the conditions of Definition 2. We will prove just point (c), since point (f) has been already proved in Theorem 1, point (a) is obvious, and the other come directly by induction.
So let us prove that $\llparenthesis \lambda \mathtt{x}.\mathtt{M} \rrparenthesis^{\nabla}_{\rho} \circ d = \llparenthesis \mathtt{M} \rrparenthesis^{\nabla}_{\rho[d/\mathtt{x}]}$.

$$\begin{aligned}
\llparenthesis \lambda \mathtt{x}.\mathtt{M} \rrparenthesis^{\nabla}_{\rho} \circ d &= \{\sigma \mid \exists \mathtt{B} \trianglelefteq \rho,\ \mathtt{B} \vdash_{\nabla} \lambda \mathtt{x}.\mathtt{M} : \sigma\} \circ d \\
&= \{\sigma \mid \exists \mathtt{B} \trianglelefteq \rho,\ \mathtt{B} \vdash_{\nabla} \lambda \mathtt{x}.\mathtt{M} : \sigma,\ \sigma \simeq \tau_1 \wedge ... \wedge \tau_n \to \tau \text{ or } \sigma \simeq \omega \to \tau\} \circ d \\
&= (\{\tau_1 \wedge ... \wedge \tau_n \to \tau \mid \exists \mathtt{B} \trianglelefteq \rho,\ \mathtt{B} \vdash_{\nabla} \lambda \mathtt{x}.\mathtt{M} : \tau_1 \wedge ... \wedge \tau_n \to \tau\} \circ d) \\
&\quad \bigcup (\{\omega \to \tau \mid \exists \mathtt{B} \trianglelefteq \rho,\ \mathtt{B} \vdash_{\nabla} \lambda \mathtt{x}.\mathtt{M} : \omega \to \tau\} \circ d) \\
&= \{\tau \mid \exists \mathtt{B} \trianglelefteq \rho,\ \mathtt{B}, \mathtt{x} : \tau_1 \wedge ... \wedge \tau_n \vdash_{\nabla} \mathtt{M} : \tau,\ \tau_i \in d\ (1 \leq i \leq n)\} \\
&\quad \bigcup \{\tau \mid \exists \mathtt{B} \trianglelefteq \rho,\ \mathtt{B} \vdash_{\nabla} \mathtt{M} : \tau\} \\
&= \{\tau \mid \exists \mathtt{B}' \trianglelefteq \rho[d/\mathtt{x}],\ \mathtt{B}' \vdash_{\nabla} \mathtt{M} : \tau\}
\end{aligned}$$

where the first equality follows by definition, the second equality follows by Lemma 1.(3), the third equality follows by rule $\simeq_{\nabla}$, the fourth equality follows by Lemma 1.(6), the fifth equality follows by definition of $\circ$ and the last equality follows by definition of $\trianglelefteq$.

(2)The following counterexample to point (g) of Definition 2, given in de Carvalho (2009), applies to all the essential type assignment systems. Let $a \in C_{\nabla}$, and let us consider the environment ρ such that $\rho(\mathtt{y}) = \{a \to a\}$ and $\rho(\mathtt{z}) = \{a \wedge a \to a\}$. Note that, for every essential type theory ∇, $a \not\simeq_{\nabla} a \wedge a$. It is easy to check that $\llparenthesis \mathtt{yx} \rrparenthesis^{\nabla}_{\rho[d/x]} = \llparenthesis \mathtt{zx} \rrparenthesis^{\nabla}_{\rho[d/x]}$, for all d. In fact

$$\llparenthesis \mathtt{yx} \rrparenthesis^{\nabla}_{\rho[d/x]} = \llparenthesis \mathtt{zx} \rrparenthesis^{\nabla}_{\rho[d/x]} = \begin{cases} \{a\} \text{ if } a \in d \\ \emptyset \text{ otherwise} \end{cases}$$

But $\llparenthesis \lambda \mathtt{x}.\mathtt{yx} \rrparenthesis^{\nabla}_{\rho} \neq \llparenthesis \lambda \mathtt{x}.\mathtt{zx} \rrparenthesis^{\nabla}_{\rho}$, since, by rule $(\to I)$, $a \to a \in \llparenthesis \lambda \mathtt{x}.\mathtt{yx} \rrparenthesis^{\nabla}_{\rho}$ and it does not belong to $\llparenthesis \lambda \mathtt{x}.\mathtt{zx} \rrparenthesis^{\nabla}_{\rho}$, while $a \wedge a \to a \in \llparenthesis \lambda \mathtt{x}.\mathtt{zx} \rrparenthesis^{\nabla}_{\rho}$ and it does not belong to $\llparenthesis \lambda \mathtt{x}.\mathtt{yx} \rrparenthesis^{\nabla}_{\rho}$.

□

Remark. A notion of filter model with non idempotent intersection has been defined Bernadet and Lengrand (2011b), where terms are interpreted as filters of types, and this could be considered in contradiction with the previous statement. But in Bernadet and Lengrand (2011b) filters are closed under a preorder relation based on the rule $\sigma \wedge \tau \preceq \sigma$, which in some sense restores the idempotency, as already observed by the authors.

6. $M\mathbb{R}el$ and its models

In this section, we describe a class of λ-models based on the category of sets and relations, defined in Bucciarelli et al. (2007). The reader already acquainted with this class of models could skip this section, which we introduced to make the paper self-contained. All the proofs of the cited results are in Bucciarelli et al. (2007), so for sake of facility we will use the same notations.

A category with terminal object $\mathbf{1}$ has enough points when for all $f, g : A \to B$, if $f \neq g$ then there is $x : \mathbf{1} \to A$ such that $f \cdot x \neq g \cdot x$. The classical way (described in text-books as Barendregt (1984) Sect. 5, Asperti and Longo (1991) Sect. 9, Amadio and Curien (1998) Sect. 4, mainly based on Koymans (1982)) of constructing a combinatory algebra from a reflexive object in a cartesian closed category (ccc in what follows) does not ensure one would obtain a λ-model in absence of enough points. Therefore authors of Bucciarelli et al. (2007) used an alternative technique presented in Selinger (2002) in order to build a λ-model starting from a λ-algebra.

Let S be a set. We denote by $\wp(S)$ (resp. $\wp_f(S)$) the collection of all subsets (resp. finite subsets) of S and we write $A \subseteq_f S$ to express that A is a finite subset of S. A *multiset* can be represented as an unordered list with repetitions. For each element a of a multiset m, the multiplicity of a (relative to m) is the number of its occurrences in m. The support of a multiset is the set of its elements. A multiset m is finite if it is a finite list, $[]$ denotes the empty multiset. Given two finite multisets $m_1 = [a_1, a_2, \ldots, a_n]$ and $m_2 = [b_1, b_2, \ldots, b_m]$ $(n, m \in \mathbb{N})$ the multiset union of m_1, m_2 is defined by $m_1 \sqcup m_2 = [a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m]$. We will write $M_f(S)$ for the set of all finite multisets with support S.

Definition 8. $M\mathbb{R}el$ is the category defined as follows:

- the objects of $M\mathbb{R}el$ are all sets;
- given two sets S and T, a morphism from S to T is a relation between $M_f(S)$ and T, i.e. $M\mathbb{R}el(S, T) = \wp(M_f(S) \times T)$;
- the identity morphism of S is $Id_S = \{([a], a) \mid a \in S\} \in M\mathbb{R}el(S, S)$;
- given two morphisms $s \in M\mathbb{R}el(S, T)$ and $t \in M\mathbb{R}el(T, U)$, we define their composition $t \cdot s$ as follows

$$\{(m, c) \mid \exists (m_1, b_1), \ldots, (m_k, b_k) \in s \text{ such that } m = m_1 \sqcup \ldots \sqcup m_k \text{ and } ([b_1, \ldots, b_k], c) \in t\}.$$

$M\mathbb{R}el$ is cartesian closed. The terminal object $\mathbf{1}$ is the empty set $\emptyset$, and the unique element of $M\mathbb{R}el(S, \emptyset)$ is the empty relation. Given two sets S_1 and S_2, their categorical product $S_1 \& S_2$ in $M\mathbb{R}el$ is their disjoint union: $S_1 \& S_2 = (\{1\} \times S_1) \cup (\{2\} \times S_2)$ and the projections π_1, π_2 are given by: $\pi_i = \{([(i, a)], a) \mid a \in S_i\} \in M\mathbb{R}el(S_1 \& S_2, S_i)$, for $i = 1, 2$. Given $s \in M\mathbb{R}el(U, S_1)$ and $t \in M\mathbb{R}el(U, S_2)$, it is easy to see that the corresponding morphism $\langle s, t \rangle \in M\mathbb{R}el(U, S_1 \& S_2)$ is given by:

$$\langle s, t \rangle = \{(m, (1, a)) \mid (m, a) \in s\} \cup \{(m, (2, b)) \mid (m, b) \in t\}.$$

We treat the canonical bijection between $M_f(S_1) \times M_f(S_2)$ and $M_f(S_1 \& S_2)$ as an equality, hence we denote by (m_1, m_2) the corresponding element of $M_f(S_1 \& S_2)$. If S and T are two objects then the exponential object $S \Rightarrow T$ is $M_f(S) \times T$ and the evaluation morphism is given by:

$$ev_{ST} = \{((([(m,b)],m),b) \mid m \in M_f(S) \text{ and } b \in T\} \in M\mathbb{R}el((S \Rightarrow T)\&S, T).$$

Given any set U and any morphism $s \in M\mathbb{R}el(U\&S, T)$, there is exactly one morphism $\Lambda(s) \in M\mathbb{R}el(U, S \Rightarrow T)$ such that: $ev_{ST} \cdot (\Lambda(s) \times Id_S) = s$ where $\Lambda(s) = \{(p,(m,b)) \mid ((p,m),b) \in s\}$.

As already said, a reflexive object U of a ccc $\mathbb{C}$ induces a λ-algebra, whose carrier set is $\mathbb{C}(\mathbf{1}, U)$, which is a λ-model too in case $\mathbb{C}$ has enough points. Unfortunately $M\mathbb{R}el$ has not enough points. In fact, the points of an object S are the elements of $M\mathbb{R}el(\mathbf{1}, S)$, i.e. the relations between $M_f(\emptyset)$ and S. These are, up to isomorphism, the subsets of S. To see that $M\mathbb{R}el$ has not enough point, suppose $A = \{a\}$ and $B = \{b\}$ and take $r_1 = \{([a,a],b)\}$ and $r_2 = \{([a],b)\}$: these morphisms cannot be distinguished by pre-composition with the points of A. Therefore, the standard construction does not work for $M\mathbb{R}el$. However, if the category is endowed with countable products then a lambda-model can be always built. Since $M\mathbb{R}el$ is endowed with countable products thus, it is possible to circumvent the "enough points" lack.

Let U be a reflexive object of $M\mathbb{R}el$, we denote by U^{Var} the countable product of objects U (indexed by variables). A morphism f of $\mathbb{C}(U^{\mathrm{Var}}, U)$ is *finitary* whenever there exists a finite set J of variables and a morphism $f_J \in \mathbb{C}(U^J, U)$ such that $f = f_J \cdot \pi_J$ where π_J denotes the canonical projection from U^{Var} onto U^J. The "finitary" morphisms of $\mathbb{C}(U^{\mathrm{Var}}, U)$ provides the carrier set of a λ-model.

We denote by $\vec{\mathtt{x}} = (\mathtt{x}_1, \ldots, \mathtt{x}_n)$ a finite ordered sequence of variables without repetitions of given length. Given an arbitrary λ-term $\mathtt{M}$ and a sequence $\vec{\mathtt{x}}$, we say that $\vec{\mathtt{x}}$ is adequate for $\mathtt{M}$ if it contains at least all the free variables of $\mathtt{M}$. We simply say that $\vec{\mathtt{x}}$ is adequate whenever $\mathtt{M}$ is clear from the context. As implicitly done in Bucciarelli et al. (2007), we do not provide a particular enumeration of variables and, essentially, we use adequate lists to provide the initial part of the implicit variable-enumeration that we are considering. In fact, adequacy just guarantees that the considered list of variables covers the non-vacuous finitary part of our countable products.

Definition 9.

(1) An object U of $M\mathbb{R}el$ is a *reflexive object* whenever there are morphisms $\lambda \in M\mathbb{R}el(U \Rightarrow U, U)$ and $Ap \in M\mathbb{R}el(U, U \Rightarrow U)$ such that $Ap \cdot \lambda = Id_{U \Rightarrow U}$.

$$\begin{tikzcd} U \arrow[r, bend left, "Ap"] \arrow[r, phantom, "\triangleright"] & U \Rightarrow U \arrow[l, bend left, "\lambda"] \arrow[loop above, "Id"] \end{tikzcd}$$

(2) Let U be a reflexive object of $M\mathbb{R}el$. For all $\mathtt{M} \in \Lambda$ and for all adequate $\vec{\mathtt{x}}$, we note by $|\mathtt{M}|^U_{\vec{\mathtt{x}}}$ a finitary morphism of $M\mathbb{R}el(U^{\mathrm{Var}}, U)$. Pragmatically, if n is the length of $\vec{\mathtt{x}}$ then we simply write the finitary restriction to the adequate list $\vec{\mathtt{x}}$ of an element of $M\mathbb{R}el(U^{\mathrm{Var}}, U)$, i.e. an element of $M\mathbb{R}el(U^n, U)$ (that unequivocally identify finitary morphism of $M\mathbb{R}el(U^{\mathrm{Var}}, U)$).

- $|\mathtt{y}|_{\vec{\mathtt{x}}} = \pi_{\mathtt{y}}$ projecting the argument indexed $\mathtt{y}$ (belonging to the adequate list $\vec{\mathtt{x}}$);

- $|\mathtt{PQ}|_{\vec{\mathtt{x}}} = ev \cdot \langle Ap \cdot |\mathtt{P}|_{\vec{\mathtt{x}}}, |\mathtt{Q}|_{\vec{\mathtt{x}}}\rangle$;
- $|\lambda \mathtt{z}.\mathtt{P}|_{\vec{\mathtt{x}}} = \lambda \cdot \Lambda(|\mathtt{P}|_{\vec{\mathtt{x}},\mathtt{z}})$, where $\mathtt{z}$ does not occur in $\vec{\mathtt{x}}$.

(3) Relational lambda-models are triple $\langle Fin(U^{\mathrm{Var}}, U), \circ, [\![.]\!]^{\mathcal{U}}\rangle$ such that $Fin(U^{\mathrm{Var}}, U)$ is the set of finitary morphisms in $M\mathbb{R}el(U^{\mathrm{Var}}, U)$, $e_1 \circ e_2$ is defined to be $ev \cdot \langle Ap \cdot e_1, e_2\rangle$ and $[\![\mathtt{M}]\!]^{\mathcal{U}}_{\rho} = |\mathtt{M}|_{\mathtt{x}_1,\ldots,\mathtt{x}_n} \cdot \langle \rho(\mathtt{x}_1), \ldots, \rho(\mathtt{x}_n)\rangle$ where $\rho(\mathtt{x}_i) \in Fin(U^{\mathrm{Var}}, U)$.

7. Logical characterization of a family of $M\mathbb{R}el$ models

The class of essential models defined in Section 4 is equivalent to a particular class of $M\mathbb{R}el$ models, so essential type assignment system can be used as tool for reasoning in a completely syntactical way about the denotation of terms in these particular relational models. In order to prove this relationship between essential and relational models, we adapt the pattern followed in Paolini et al. (2009), where a logical description of models based on coherence spaces is given.

First, let us define a class of $M\mathbb{R}el$ models that rises from retraction pairs (i.e. the morphisms λ, Ap of Definition 9.(1)) satisfying some additional requirements. A morphism $f \in M\mathbb{R}el(A, B)$ is *linear* whenever $(m, b) \in f$ implies that m is a singleton. Linear morphisms have been introduced to define linear reflexive objects (relying on a pair of linear morphisms for the retraction) and to define models of the differential λ-calculus (see Definitions 5.1 and 5.2 in Manzonetto (2012)). This notion of linearity is quite standard, since $M\mathbb{R}el$ is the Kleisli category of $\mathbb{R}el$ over the finite-multiset comonad and $\mathbb{R}el$ is indeed a model of Linear Logic.

For us, a morphism s in $M\mathbb{R}el(A, B)$ is *strongly linear* whenever it is linear and injective, i.e. if $(m', b), (m'', b) \in f$ then $m' = m''$. It is easy to see that the composition of (strongly) linear morphisms is (strongly) linear.

Definition 10. A reflexive object (U, Ap, λ) in $M\mathbb{R}el$ is *strongly linear* whenever Ap, λ are strongly linear.

It is easy to check that the morphism λ of a reflexive object (see Definition 9.(1)) is always monic, but it can be not injective (i.e. it is not necessarily split monic). We remark that if the reflexive object involves an iso (so, the induced model is extensional) then the retraction pair is formed by strongly linear morphisms. It is not hard to build a strongly linear reflexive object starting form any reflexive object, but the induced lambda-theory can be different. Still, in Manzonetto (2012) many properties are proved about linear reflexive objects.

Now we prove that the essential type assignment system provides a logical description of strongly linear categorical models in $M\mathbb{R}el$. Roughly, we use types as names for elements of the reflexive object of $M\mathbb{R}el$ and we use arrow-types to record their functional behavior. An essential type theory formalizes names in order to can speak of the elements of a domain.

Definition 11. Let $\nabla = \langle C, \simeq\rangle$ be a legal essential type system, and let $(\sigma)_{\simeq}$ denote the equivalence class of σ with respect to $\simeq$. The essential domain induced by ∇ is a triple $(U_\nabla, \lambda_\nabla, Ap_\nabla)$ where U_∇ is an object of $M\mathbb{R}el$, $\lambda_\nabla : (U_\nabla \Rightarrow U_\nabla) \to U_\nabla$ and

$Ap_\nabla : U_\nabla \to (U_\nabla \Rightarrow U_\nabla)$ such that

- $U_\nabla = \mathsf{T}(C)/_{\simeq}$;
- $$\lambda_\nabla = \left\{ ([([(\sigma_1)_{\simeq}, \ldots, (\sigma_n)_{\simeq}], (\sigma)_{\simeq})], (\bigwedge_{i=1}^{n} \sigma_i \to \sigma)_{\simeq}) \;\middle|\; \begin{array}{l} n \geq 0, \\ \sigma_i \in \mathsf{T}(C)(1 \leq i \leq n), \\ \sigma \in \mathsf{T}(C) \end{array} \right\};$$
- $$Ap_\nabla = \left\{ ([(\bigwedge_{i=1}^{n} \sigma_i \to \sigma)_{\simeq}], ([(\sigma_1)_{\simeq}, \ldots, (\sigma_n)_{\simeq}], (\sigma)_{\simeq})) \;\middle|\; \begin{array}{l} n \geq 0, \\ \sigma_i \in \mathsf{T}(C)(1 \leq i \leq n), \\ \sigma \in \mathsf{T}(C) \end{array} \right\}.$$

Notation 2. Since this section tackles various semantics facets, for sake of simplicity, we will use a straightforward shortening notations: $\bigwedge_{i=1}^{n} \sigma_i \to \sigma$ when $n = 0$ is just a notation for $\omega \to \sigma$. Therefore, for instance $\big([([], (\tau)_{\simeq})], (\omega \to \tau)_{\simeq}\big)$ belongs to λ_∇ in the above definition. Moreover, we use $\theta, \theta_1, \ldots$ as metavariables to denote either the empty intersection ω or the non-empty one μ. We will use for these metavariables the same notation we introduce for intersections in Notation 1. If $\theta = \omega$ then $\mathtt{B}, \mathtt{x} : \theta$ is an abuse of notation for the basis $\mathtt{B}$ (where implicitly we assume $\mathtt{x} \notin \mathtt{dom}(\mathtt{B})$, see Notation 1).

Any essential domain is a strongly linear reflexive object in $M\mathbb{R}el$ indeed.

Lemma 7. Let $\nabla = \langle C, \simeq \rangle$ be a legal essential type system. Then the essential domain $(U_\nabla, \lambda_\nabla, Ap_\nabla)$ induced by ∇ is a strongly linear reflexive object in $M\mathbb{R}el$.

Proof. It is not difficult to see that $\lambda_\nabla, Ap_\nabla$ are strongly linear by construction. We prove that $Ap_\nabla \cdot \lambda_\nabla = id_{U_\nabla \Rightarrow U_\nabla}$. If $(m, c) \in Ap_\nabla \cdot \lambda_\nabla$ then there is a multiset $[d_1, \ldots, d_k]$ such that $([d_1, \ldots, d_k], c) \in Ap_\nabla$, $(m_1, d_1) \ldots (m_k, d_k) \in \lambda_\nabla$ and $m = m_1 \sqcup \ldots \sqcup m_k$. Linearity implies that $k = 1$ and m_1 is a singleton, so $m = m_1$ is a singleton too.
Because $U_\nabla \Rightarrow U_\nabla$ is $M_f(U_\nabla) \times U_\nabla$, the shape of c is $\big([(\tau_1)_{\simeq}, \ldots, (\tau_m)_{\simeq}], (\tau)_{\simeq}\big)$ for some $\tau_1, \ldots, \tau_m, \tau \in \mathsf{T}(C)$. By definition $\big([([(\tau_1)_{\simeq}, \ldots, (\tau_m)_{\simeq}], (\tau)_{\simeq})], (\bigwedge_{i=1}^{n} \tau_i \to \tau)_{\simeq}\big) \in \lambda_\nabla$ and, moreover, $\big([(\bigwedge_{i=1}^{n} \tau_i \to \tau)_{\simeq}, [([(\tau_1)_{\simeq}, \ldots, (\tau_m)_{\simeq}], (\tau)_{\simeq})]\big)$ is the unique element of Ap_λ that can be composed with it, so it adds an element to $Ap_\nabla \cdot \lambda_\nabla$. Thus, elements of $Ap_\nabla \cdot \lambda_\nabla$ are all and only elements of the shape $\big([([(\tau_1)_{\simeq}, \ldots, (\tau_m)_{\simeq}], (\tau)_{\simeq})], ([(\tau_1)_{\simeq}, \ldots, (\tau_m)_{\simeq}], (\tau)_{\simeq})\big)$. □

Let ∇ be an essential type system and $(U_\nabla, \lambda_\nabla, Ap_\nabla)$ be the essential domain induced by ∇. We define the map $(\cdot)^*$ from types to finite multisets of U_∇ as follows: $\omega^* = []$, $\sigma^* = [(\sigma)_{\simeq}]$ and $(\mu_1 \wedge \mu_2)^* = \mu_1^* \sqcup \mu_2^*$.

Next lemma shows that our typing system provides a logical description of the categorical interpretation of terms in relational models.

Lemma 8. Let ∇ be an essential type system and let $(U_\nabla, \lambda_\nabla, Ap_\nabla)$ be the essential domain induced by ∇. Then

(1) $\mathtt{x}_1 : \theta_1, \dots, \mathtt{x}_n : \theta_n \vdash \mathtt{M} : \sigma$ implies $(\theta_1^*, \dots, \theta_n^*, (\sigma)_{\simeq}) \in |\mathtt{M}|_{\vec{\mathtt{x}}}$;
(2) if $(\theta_1^*, \dots, \theta_n^*, (\sigma)_{\simeq}) \in |\mathtt{M}|_{\vec{\mathtt{x}}}$ for some $\theta_1, \dots, \theta_n, \sigma$ then $\mathtt{x}_1 : \theta_1, \dots, \mathtt{x}_n : \theta_n \vdash \mathtt{M} : \sigma$.

Proof.

(1)Let $\vec{\mathtt{x}}$ be an adequate list of variable of length n. The proof is by induction on the derivation of $\mathtt{x}_1 : \theta_1, \dots, \mathtt{x}_n : \theta_n \vdash \mathtt{M} : \sigma$.

- Case (var). Let $\mathtt{M}$ be $\mathtt{x}_i$ with $1 \leq i \leq n$, so $|\mathtt{x_i}|_{\vec{\mathtt{x}}} = \pi_i$. The conclusion of (var) must have the shape $\mathtt{x}_i : \sigma \vdash \mathtt{x}_i : \sigma$. Since we treat the canonical bijection between $M_f(S_1) \times M_f(S_2)$ and $M_f(S_1 \& S_2)$ as an equality (as done in Bucciarelli et al. (2007)), we have that $\pi_i = \{(\underbrace{[], \dots, []}_{i-1}, [(\sigma)_{\simeq}], \underbrace{[], \dots, []}_{n-i}), (\sigma)_{\simeq}) \mid \sigma \in \mathsf{T}(C)\}$ is the relation projecting the argument indexed $\mathtt{x}_i$ when applied to a suitable list of arguments. Thus the proof is immediate.
- The case $(\simeq_{\nabla})$ is trivial.
- Case $(\to E)$.Let $\mathtt{M} \equiv \mathtt{PQ}$, and suppose $\mathtt{x}_1 : \theta_1, \dots, \mathtt{x}_n : \theta_n \vdash \mathtt{PQ} : \sigma$. By the shape of rule $(\to E)$ we have that $\mathtt{x}_1 : \theta_1^0, \dots, \mathtt{x}_n : \theta_n^0 \vdash \mathtt{P} : \sigma_1 \wedge \dots \wedge \sigma_k \to \sigma$ and $\mathtt{x}_1 : \theta_1^j, \dots, \mathtt{x}_n : \theta_n^j \vdash \mathtt{Q} : \sigma_j$ for all $1 \leq j \leq k$ $(k \geq 1)$ such that $\theta_i = \bigwedge_{j=0}^{k} \theta_i^j$. So by inductive hypothesis, we have both $((\theta_1^0)^*, \dots, (\theta_n^0)^*, (\bigwedge_{j=1}^{k} \sigma_j \to \sigma)_{\simeq}) \in |\mathtt{P}|_{\vec{\mathtt{x}}}$ and $((\theta_1^j)^*, \dots, (\theta_n^j)^*, (\sigma_j)_{\simeq}) \in |\mathtt{Q}|_{\vec{\mathtt{x}}}$. Note that $\vec{\mathtt{x}}$ adequate for $\mathtt{PQ}$ implies $\vec{\mathtt{x}}$ is adequate both for $\mathtt{P}$ and $\mathtt{Q}$. Since $|\mathtt{PQ}|_{\vec{\mathtt{x}}} = ev \cdot \langle Ap \cdot |\mathtt{P}|_{\vec{\mathtt{x}}}, |\mathtt{Q}|_{\vec{\mathtt{x}}} \rangle$ and $((\theta_1^0)^*, \dots, (\theta_n^0)^*, (\bigwedge_{j=1}^{k} \sigma_j)^*, (\sigma)_{\simeq}) \in Ap \cdot |\mathtt{P}|_{\vec{\mathtt{x}}}$ by definition of Ap, then $((\bigwedge_{j=0}^{k} \theta_1^j)^*, \dots, (\bigwedge_{j=0}^{k} \theta_n^j)^*, (\sigma)_{\simeq}) \in |\mathtt{PQ}|_{\vec{\mathtt{x}}}$.
- Case $(\to I)$. Let $\mathtt{M} \equiv \lambda \mathtt{x}_{n+1}.\mathtt{P}$ and suppose $\mathtt{x}_1 : \theta_1, \dots, \mathtt{x}_n : \theta_n \vdash \lambda \mathtt{x}_{n+1}.\mathtt{P} : \sigma_1 \wedge \dots \wedge \sigma_k \to \sigma$ $(k \geq 1)$. Therefore, $\mathtt{x}_1 : \theta_1, \dots, \mathtt{x}_n : \theta_n, \mathtt{x}_{n+1} : \sigma_1 \wedge \dots \wedge \sigma_k \vdash \mathtt{P} : \sigma$. By inductive hypothesis we have $(\theta_1^*, \dots, \theta_n^*, \sigma_1^* \sqcup \dots \sqcup \sigma_k^*, (\sigma)_{\simeq}) \in |\mathtt{P}|_{\vec{\mathtt{x}}, \mathtt{x}_{k+1}}$. So we can easily conclude by definition of λ_{∇}.
- The case $(\to I_{\omega})$ and $(\to E_{\omega})$ are similar to the previous two sub-points, just considering $k = 0$, i.e. reading ω in place of empty intersections.

(2)The proof is by induction on $\mathtt{M}$.

- $\mathtt{M} \equiv \mathtt{x}$. Suppose $|\mathtt{x}|_{\vec{\mathtt{x}}} = \pi_i = \{(\underbrace{[], \dots, []}_{i-1}, [(\tau)_{\simeq}], \underbrace{[], \dots, []}_{n-i}), (\sigma)_{\simeq}) \mid \sigma \simeq_{\nabla} \tau\}$, So we can conclude by applying rule (var) immediately followed by rule $(\simeq)$.
- $\mathtt{M} \equiv \mathtt{PQ}$. Suppose $(\theta_1^*, \dots, \theta_n^*, (\sigma)_{\simeq}) \in |\mathtt{PQ}|_{\mathtt{x}_1, \dots, \mathtt{x}_n} = ev \cdot \langle Ap \cdot |\mathtt{P}|_{\vec{\mathtt{x}}}, |\mathtt{Q}|_{\vec{\mathtt{x}}} \rangle$. This means that there are $\theta_1', \dots, \theta_n', \tau, \sigma_1, \dots, \sigma_k, \theta_i^j$ $(0 \leq i \leq n,\ 0 \leq j \leq k)$ such that $((\theta_1')^*, \dots, (\theta_n')^*, (\tau)_{\simeq}) \in |\mathtt{P}|_{\vec{\mathtt{x}}}$, $((\tau)_{\simeq}, ([(\sigma_1)_{\simeq}, \dots, (\sigma_k)_{\simeq}], (\sigma)_{\simeq})) \in Ap_{\nabla}$ such that $\tau \simeq_{\nabla} \sigma_1 \wedge \dots \wedge \sigma_k \to \sigma$, $((\theta_1^j)^*, \dots, (\theta_n^j)^*, (\sigma_j)_{\simeq}) \in |\mathtt{Q}|_{\vec{\mathtt{x}}}$ $(0 \leq j \leq k)$ with $\theta_i = \theta_i' \wedge \theta_i^1 \wedge \dots \wedge \theta_i^k$ $(0 \leq i \leq n)$. By induction we have $\mathtt{x}_1 : \theta_1', \dots, \mathtt{x}_n : \theta_n' \vdash \mathtt{P} : \tau$ and $\mathtt{x}_1 : \theta_1^j, \dots, \mathtt{x}_n : \theta_n^j : \mathtt{Q} : \sigma_j$ $(0 \leq j \leq k)$. So by applying rule $(\simeq)$ we have $\mathtt{x}_1 : \theta_1', \dots, \mathtt{x}_n : \theta_n' \vdash \mathtt{P} : \sigma_1 \wedge \dots \wedge \sigma_k \to \sigma$. If $k \geq 1$ then the proof follows by applying the rule $(\to E)$, otherwise by applying the rule $(\to E_{\omega})$.
- $\mathtt{M} \equiv \lambda \mathtt{y}.\mathtt{P}$. Suppose $(\theta_1^*, \dots, \theta_n^*, (\tau)_{\simeq}) \in |\lambda \mathtt{y}.\mathtt{P}|_{\vec{\mathtt{x}}}$. This means that $([([(\sigma_1)_{\simeq}, \dots, (\sigma_k)_{\simeq}], (\sigma)_{\simeq})], (\tau)_{\simeq}) \in \lambda_{\nabla}$ such that $\tau \simeq \sigma_1 \wedge \dots \wedge \sigma_k \to \sigma$

and $(\theta_1^*, \ldots, \theta_n^*, (\sigma_1 \wedge \ldots \wedge \sigma_k)^*, \sigma) \in |\mathtt{P}|_{\vec{\mathtt{x}},\mathtt{y}}$ for some $\sigma, \sigma_1, \ldots, \sigma_k$. By induction we have $\mathtt{x}_1 : \theta_1, \ldots, \mathtt{x}_n : \theta_n, \mathtt{y} : \sigma_1 \wedge \ldots \wedge \sigma_k \vdash \mathtt{P} : \sigma$. If $k \geq 1$ then we apply the rule $(\rightarrow I)$, otherwise we apply the rule $(\rightarrow I_\omega)$. We obtain $\mathtt{x}_1 : \theta_1, \ldots, \mathtt{x}_n : \theta_n \vdash \lambda\mathtt{y}.\mathtt{P} : \sigma_1 \wedge \ldots \wedge \sigma_k \rightarrow \sigma$. We conclude by rule $(\simeq)$.

□

Two λ-models $\mathcal{M} = \langle \mathbb{D}, \circ, [\![\cdot]\!]^{\mathcal{M}} \rangle$ and $\mathcal{M}' = \langle \mathbb{D}', \circ', [\![\cdot]\!]^{\mathcal{M}'} \rangle$ are isomorphic if there is a bijective function $h : \mathbb{D}' \rightarrow \mathbb{D}$ such that, for all terms $\mathtt{M}$ the equality $[\![\mathtt{M}]\!]_\rho^{\mathcal{M}} = h([\![\mathtt{M}]\!]_{\rho'}^{\mathcal{M}'})$ holds, where $\rho(\mathtt{x}) = h(\rho'(\mathtt{x}))$.

Theorem 4 (Soundness). If ∇ is a legal essential type system then $\mathcal{M}^\nabla = \langle \mathfrak{T}(\nabla), \circ^\nabla, [\![\cdot]\!]^{\mathcal{M}^\nabla} \rangle$ and $\mathcal{M}^{U_\nabla} = \langle Fin(U_\nabla^{Var}, U_\nabla), \circ^{U_\nabla}, [\![\cdot]\!]^{\mathcal{U}_\nabla} \rangle$ are isomorphic.

Proof. The proof follows quite easily by Lemma 7 and 8. To be precise, our correspondence holds only after few inessential settlements. A pre-typing embeds a finite list of variable (including, at least, all involved variables carrying non-empty information, i.e. different from ω) in our typing-interpretations. On the other hand, the categorical interpretation does not include the adequate list of variables in the interpretation. We overcome this issue, by considering integral part of the the categorical interpretation the involved adequate list of variables. A second minor issue is that the categorical interpretation ranges over sets of type-quotient while the logical interpretation ranges over set of types closed by $\simeq$. This point can be straightforwardly overcome by smashing each set of quotients in its union. □

In order to prove completeness, we show that a strongly linear categorical model in $M\mathbb{R}el$ induces an essential type system whose corresponding λ-model is isomorphic with it.

Lemma 9. Let (U, λ, Ap) be a strongly linear reflexive object. Then there exists an essential type system $\nabla = (C, \simeq)$ and two isomorphisms $(\cdot)^\sharp : U \Rightarrow U \rightarrow U_\nabla \Rightarrow U_\nabla$ and $(\cdot)^\flat : U \rightarrow U_\nabla$ such that the following diagram commutes

$$\begin{tikzcd}
U \Rightarrow U \arrow[r, "\lambda"] \arrow[d, "(\cdot)^\sharp"] & U \arrow[r, "Ap"] \arrow[d, "(\cdot)^\flat"] & U \Rightarrow U \arrow[d, "(\cdot)^\sharp"] \\
U_\nabla \Rightarrow U_\nabla \arrow[r, "\lambda_\nabla"] & U_\nabla \arrow[r, "Ap_\nabla"] & U_\nabla \Rightarrow U_\nabla
\end{tikzcd}$$

Proof. Let (U, λ, Ap) be a strongly linear reflexive object. We define $\nabla = (C, \simeq)$ in the following way:

- let C be a set being in bijection with the set U through the function $(\cdot)^\diamond : U \rightarrow C$;
- $\simeq$ being the least congruence on $\mathsf{T}(C)$ such that, for all $([([a_1, \ldots, a_n], a)], b) \in \lambda$, if $n \geq 1$ then $a_1^\diamond \wedge \ldots \wedge a_n^\diamond \rightarrow a^\diamond \simeq b^\diamond$ otherwise $\omega \rightarrow a^\diamond \simeq b^\diamond$.

First, let us define $(\cdot)^\flat$ from U to $U_\nabla = \mathsf{T}(C)/_\simeq$ element-wise $a^\flat = (a^\diamond)_\simeq$, for each $a \in U$. Injectivity of $(\cdot)^\diamond$ implies injectivity of $(\cdot)^\flat$. Moreover, $(\cdot)^\flat$ is surjective, because $\mathsf{T}(C)/_\simeq$ contains an element of C in each class of equivalence, because $Ap \cdot \lambda = Id_{U \Rightarrow U}$.

Let us now prove that $\nabla = (C, \simeq)$ is an essential type system. Suppose $\sigma_1 \wedge \ldots \wedge$

$\sigma_n \to \sigma$ and $\tau_1 \wedge \ldots \wedge \tau_m \to \tau$ be two types in the same $\simeq$ equivalence class. Since $(\cdot)^\flat : U \to \mathsf{T}(C)/_{\simeq}$ is a bijection, there are (unique) $([a_1, \ldots, a_n], a), ([b_1, \ldots, b_m], b) \in U \Rightarrow U$ such that $\sigma_i \in a_i^\flat$, $\sigma \in a^\flat$, $\tau_j \in b_j^\flat$ and $\tau \in b^\flat$. Since λ is total, we have $([([a_1, \ldots, a_n], a)], a'), ([([b_1, \ldots, b_m], b)], b') \in \lambda$ for some $a', b' \in U$. It is easy to check that $a' = b'$ if and only if $\sigma_1 \wedge \ldots \wedge \sigma_n \to \sigma \simeq \tau_1 \wedge \ldots \wedge \tau_m \to \tau$ (i.e. whenever they belong to the same equivalence class).

Lemma 7 implies that U_∇ is a strongly linear reflexive object in $\mathcal{M}\mathbb{R}el$, since ∇ is an essential type system. We define the map $(\cdot)^\sharp : U \Rightarrow U \to U_\nabla \Rightarrow U_\nabla$ as $([a_1, \ldots, a_n], a)^\sharp = ([a_1^\flat, \ldots, a_n^\flat], a^\flat)$.

Finally we check that the above diagram commutes. Since functions are also relations, we can view the maps $(\cdot)^\flat$ and $(\cdot)^\sharp$ as linear morphisms of $\mathcal{M}\mathbb{R}el$ and that composition of linear morphisms in $\mathcal{M}\mathbb{R}el$ behaves exactly as relational composition. Now let $([([a_1, \ldots, a_n], a)], (\sigma)_{\simeq}) \in (\cdot)^\flat \cdot \lambda$: this means that there is $a' \in U$ such that $([([a_1, \ldots, a_n], a)], a') \in \lambda$ and $(a')^\diamond \in (\sigma)_{\simeq}$. By definition of $\simeq$, we have that $a_1^\diamond \wedge \ldots \wedge a_n^\diamond \to a^\diamond \in (\sigma)_{\simeq}$. By definition of λ_∇, we have $([([a_1, \ldots, a_n], a)^\sharp], (\sigma)_{\simeq}) \in \lambda_\nabla$. So $([([a_1, \ldots, a_n], a)], (\sigma)_{\simeq}) \in \lambda_\nabla \cdot (\cdot)^\sharp$. Thus $(\cdot)^\flat \cdot \lambda \subseteq \lambda_\nabla \cdot (\cdot)^\sharp$. The converse can be proved in a similar way. The right-hand-side of the diagram commutation can be proved in a similar way. □

Essential type assignment systems supply a complete logical characterization of strongly linear relational λ-models .

Theorem 5 (Completeness). Let $\mathcal{M}$ be a λ-model induced by a strongly linear reflexive object of $\mathcal{M}\mathbb{R}el$. Then there is an essential type system ∇ such that the model $\mathcal{M}^\nabla$ is isomorphic to $\mathcal{M}$.

Proof. By Lemma 9 we can build a legal type system inducing a reflexive object in $\mathcal{M}\mathbb{R}el$ whose model is isomorphic to $\mathcal{M}$. □

8. Essential models induce sensible theories

Let us recall the notion of solvability in λ-calculus Barendregt (1984).

Definition 12. A term $\mathtt{M}$ is solvable if and only if there are variables $\mathtt{x}_1, ..., \mathtt{x}_n$ and terms $\mathtt{N}_1, ..., \mathtt{N}_p$ such that $(\lambda\mathtt{x}_1...\mathtt{x}_n.\mathtt{M})\mathtt{N}_1...\mathtt{N}_p =_\beta \mathbf{I}$, where $\mathbf{I} = \lambda\mathtt{x}.\mathtt{x}$ is the identity function.

In a language without ground data types, like the λ-calculus, solvable terms represent meaningful programs (see Barendregt (1984)). Solvable terms are completely characterized from both a syntactical and operational point of view. Let us recall that the general shape of a term is $\lambda\mathtt{x}_1 \ldots \mathtt{x}_n.\zeta\mathtt{M}_1 \ldots \mathtt{M}_m$, for some $m, n \geq 0$ and some terms $\mathtt{M}_i$, $1 \leq i \leq m$, where ζ, the *head*, is either a variable or a redex $(\lambda\mathtt{x}.\mathtt{Q})\mathtt{N}$. If the head is a variable, then the term *is in head normal form.* A term is solvable if and only if either it is in head normal form or the procedure of reducing at every step the head redex eventually stops. A term is *unsolvable* when it is not solvable.

Definition 13. A λ-theory is *sensible* when it equates all unsolvable terms.

We prove that all the λ-theories induced by essential λ-models are sensible, i.e., they equate all unsolvable terms. This proof is an example of the use of types for proving semantical properties.

First, we prove that all solvable terms can be typed.

Lemma 10. Let ∇ be a essential type system and $\mathtt{M}$ be a term. If $\mathtt{M}$ is solvable then it can be typed by the system $\vdash_\nabla$, i.e. there is $\mathtt{B}$ and σ such that $\mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$.

Proof. We prove that all head normal forms can be typed in $\vdash_\nabla$, thus the proof follows by Theorem 1. Let $\mathtt{M} \equiv \lambda \mathtt{x}_1 ... \mathtt{x}_n . \mathtt{x} \mathtt{M}_1 ... \mathtt{M}_m$ and let $\mathtt{B}$ be the basis such that $\mathtt{dom}(\mathtt{B}) = \{\mathtt{x}\}$ and $\mathtt{B}(\mathtt{x}) = \underbrace{\omega \to ... \to \omega}_{m} \to \sigma$, for some σ. Then $\mathtt{B} \vdash_\nabla \mathtt{x}\mathtt{M}_1 ... \mathtt{M}_m : \sigma$ (by rules (var) and $(\to\mathrm{E})$). Then, if $\mathtt{x} \in \mathtt{FV}(\mathtt{M})$, by rule $(\to\mathrm{I})$ we can derive $\mathtt{B} \vdash_\nabla \lambda \mathtt{x}_1 ... \mathtt{x}_n . \mathtt{x} \mathtt{M}_1 ... \mathtt{M}_m : \underbrace{\omega \to ... \to \omega}_{n} \to \sigma$, otherwise, if $\mathtt{x} = \mathtt{x}_p$ for some p $(1 \leq p \leq n)$, we can derive $\vdash_\nabla \lambda \mathtt{x}_1 ... \mathtt{x}_n . \mathtt{x} \mathtt{M}_1 ... \mathtt{M}_m : \underbrace{\omega \to ... \to \omega}_{p-1} \to (\underbrace{\omega \to ... \to \omega}_{m} \to \sigma) \to \underbrace{\omega \to ... \to \omega}_{n-p-1} \to \sigma$. □

For type assignment systems endowed with idempotent intersections, a proof of the converse would need sophisticate arguments, like computability or reducibility candidates. In fact, a proof of solvability has the same logical complexity than a proof of strong normalization. In our system the non-idempotency of intersections carries on quantitative information on derivations, allowing a very simple proof, made by induction on the derivation. The first observation that non idempotent intersection allows for simpler proofs comes from Terui (2006), who applied it to a proof of strong normalization.

As we show in Section 2, the subject reduction property is based on the substitution property. We prove that the property of substitution has a quantitative version. Let $|\Pi|$ be the number of rules $(\to\mathrm{E})$ used in the derivation Π, i.e., the number of nodes of the derivation tree.

Lemma 11 (Weighted Substitution). If $\Pi \rhd \mathtt{B}, \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash_\nabla \mathtt{M} : \tau$ and $\Pi_i \rhd \mathtt{B}_i \vdash_\nabla \mathtt{N} : \sigma_i$ $(1 \leq i \leq n)$ then $\Delta \rhd \mathtt{B} \bigwedge_{1 \leq i \leq n} \mathtt{B}_i \vdash_\nabla \mathtt{M}[\mathtt{N}/\mathtt{x}] : \tau$ $(n \geq 0)$, where $|\Delta| < |\Pi| + \sum_{1 \leq i \leq n} |\Pi_i|$.

Proof. The proof is by induction on the derivation Π. All the cases come directly by induction, the only not obvious one is the case the last rule of Π is $(\to\mathrm{E})$. While proving this case, we will use the same notations as in the proof of Lemma 2. Let the last rule be:

$$\frac{\Pi' \rhd \mathtt{B}_0, \mathtt{x} : \tau_1 \wedge ... \wedge \tau_p \vdash_\nabla \mathtt{P} : \pi_1 \wedge ... \wedge \pi_m \to \sigma \quad \left(\Pi'_j \rhd \mathtt{B}^j_1, \mathtt{x} : \tau^j_1 \wedge ... \wedge \tau^j_{q_j} \vdash_\nabla \mathtt{Q} : \pi_j\right)_{1 \leq j \leq m}}{\left(\mathtt{B}_0 \bigwedge_{1 \leq j \leq m} \mathtt{B}^j_1\right), \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash_\nabla \mathtt{PQ} : \sigma} \ (\to\mathrm{E})$$

where $\sigma_1 \wedge ... \wedge \sigma_n = \tau_1 \wedge ... \wedge \tau_p \wedge \tau^1_1 \wedge ... \wedge \tau^1_{q_1} \wedge ... \wedge \tau^m_1 \wedge ... \wedge \tau^m_{q_m}$. So $n = p + \Sigma_{1 \leq j \leq m} q_j$, and σ_i coincides either with τ_r or with τ^h_k, for some r, h, k $(1 \leq r \leq p, 1 \leq h \leq m, 1 \leq k \leq q_h)$. Let $\mathtt{B}_h \vdash_\nabla \mathtt{N} : \tau_h$ $(1 \leq h \leq p)$ and $\mathtt{B}^j_{k_j} \vdash_\nabla \mathtt{N} : \tau^j_{k_j}$ $(1 \leq j \leq m, 1 \leq k_j \leq q_j)$. By induction we have that

(1)$\Delta' \rhd \mathtt{B}_0 \bigwedge_{1 \leq h \leq p} \mathtt{B}_h \vdash_\nabla \mathtt{P}[\mathtt{N}/\mathtt{x}] : \pi_{\mathtt{1}} \wedge ... \wedge \pi_{\mathtt{m}} \to \sigma$, where $|\Delta'| < |\Pi'| + \sum_{1 \leq j \leq p} |\Pi_j|$, (2)$\Delta_j \rhd \mathtt{B}^j_1 \bigwedge_{1\leq k_j\leq q_j} \mathtt{B}^j_{k_j} \vdash_\nabla \mathtt{Q}[\mathtt{N}/\mathtt{x}] : \pi_j$ $(1 \leq j \leq m)$, where $|\Delta_j| < |\Pi'_j| + \sum_{1\leq k_j\leq q_j} |\Pi_{k_j}|$. Then it follows that $\Delta \rhd \mathtt{B}_0 \bigwedge_{1\leq h\leq p} \mathtt{B}_h \bigwedge_{1\leq j\leq m}(\mathtt{B}^j \bigwedge_{1\leq k_j\leq q_j} \mathtt{B}^j_{k_j}) \vdash_\nabla \mathtt{P}[\mathtt{N}/\mathtt{x}]\mathtt{Q}[\mathtt{N}/\mathtt{x}] = \mathtt{PQ}[\mathtt{N}/\mathtt{x}] : \sigma$ by rule $(\rightarrow E)$, where

$$\begin{array}{rcl} |\Delta| & = & |\Delta'| + \sum\limits_{1\leq j\leq m} |\Delta_j| + 1 \\ & < & |\Pi'| + \sum\limits_{1\leq j\leq p} |\Pi_j| + \sum\limits_{1\leq j\leq m} (|\Pi'_j| + \sum\limits_{1\leq k_j\leq q_j} |\Pi_{k_j}|) + 1 \\ & = & |\Pi'| + \sum\limits_{1\leq j\leq n} |\Pi_j| + \sum\limits_{1\leq j\leq m} |\Pi'_j| + 1 \\ & = & |\Pi| + \sum\limits_{1\leq j\leq n} |\Pi_j| \end{array}$$

Thus the proof follows. □

Remember that it is possible that some subterms of the subject of a derivation are not typed in a derivation itself, thanks to rule $(\rightarrow E_\omega)$. But the head has a special status.

Property 1. If $\mathtt{M}$ is such that $\Pi \rhd \mathtt{B} \vdash_\nabla \mathtt{M} : \sigma$, for some $\mathtt{B}$ and σ, then there is a subderivation of Π whose subject is the head of $\mathtt{M}$.

Proof. The proof can be done by induction on $\mathtt{M}$, by using Lemma 1. □

Lemma 12. If $\Pi \rhd \mathtt{B} \vdash_\nabla \lambda \mathtt{x}_1 \ldots \mathtt{x}_n.(\lambda \mathtt{x}.\mathtt{Q})\mathtt{N}\mathtt{M}_1 \ldots \mathtt{M}_m : \sigma$ then there is a derivation $\Sigma \rhd \mathtt{B} \vdash_\nabla \lambda \mathtt{x}_1 \ldots \mathtt{x}_n.\mathtt{Q}[\mathtt{N}/\mathtt{x}]\mathtt{M}_1 \ldots \mathtt{M}_m : \sigma$, such that $|\Sigma| < |\Pi|$.

Proof. The proof comes directly from Lemma 11 and Property 1. □

So we can conclude the desired result.

Theorem 6. For every essential type theory ∇, the λ-theory induced by $\mathcal{M}^\nabla$ is sensible.

Note that, thanks to the isomorphism between essential and relational models, we proved in the same time that all the models of the family of (strongly linear) relational models induce sensible lambda-theories. This result has been already suggested in Carraro et al. (2010), as consequence of the fact that relational models of resource λ-calculus satisfying the Taylor expansion all induce sensible theories. But they do not supply any proof, and moreover the bridge between models of resource calculus and models of λ-calculus depends on the interpretation of the differential operations. So we think it is interesting to see a very simple proof of it. We remark that the sufficient conditions to check if a model in $\mathcal{M}\mathbb{R}el$ induces a (maximal consistent) sensible λ-theory, provided by Manzonetto (2009), does not imply this property for all essential models. In fact, in Manzonetto (2009) only extensional theories are taken into account, while essential models can induce also non extensional theories (just, consider infinite type-constants and the minimal legal type theory).

A similar technique has been used in Pagani and Ronchi Della Rocca (2010b) for charactering solvability in the resource λ-calculus.

9. Conclusions

We defined the class of essential λ-models based on intersection types. The class of essential models is described by a parametric intersection type assignment system, where the intersection is not idempotent and the parameters are the set of type constants and a type theory, i.e., a congruence on types preserving the number of components of an intersection type. Moreover we proved that this class is isomorphic to a class of λ-models based on set and relations, considered by Bucciarelli et al. (2007), characterized by a strongly linear reflexive object.

Instances of essential type assignment systems already present in the literature, namely in Coppo et al. (1981) and de Carvalho (2009). These two systems are almost equivalent, being the latter just a redefinition of the former. Both can be viewed as a particular instance of our parametric type assignment system, obtained by choosing an infinite set of type constants, equipped with the minimal type theory. Both are studied using the classical interpretation of terms by the set of types derivable for them, that provides λ-algebras. Once equipped by the interpretation with pre-typings, both induce a λ-model whose theory is the theory of Böhm trees.

The isomorphic correspondence between essential models and (strongly linear) relational models allows to recover another instance of our parametric system, obtained choosing a single type constant a, equipped with the type theory induced by the equivalence $a \simeq \omega \to a$. The corresponding relational model has been studied in Bucciarelli et al. (2007), and the induced λ-theory has been proved to be H^* by Manzonetto (2009).

It can be interesting to investigate on the existence of essential models inducing further meaningful λ-theories: it will be the topic of a further work.

Moreover, we would like to to design essential models of the resource calculus, starting from the type assignment systems defined in Pagani and Ronchi Della Rocca (2010b,a).

Acknowledgments.

Simona Ronchi Della Rocca would like to thanks Antonio Bucciarelli, Thomas Ehrhard and Giulio Manzonetto for the interesting discussions about the topic of this paper.

References

Abramsky, S. (1991). Domain theory in logical form. *Annals of Pure and Applied Logic*, 51(1-2):1–77.

Amadio, R. and Curien, P.-L. (1998). *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press.

Asperti, A. and Longo, G. (1991). *Categories, Types, and Structures: An Introduction to Category Theory for the Working Computer Scientist.* The MIT Press.

Barendregt, H. (1984). *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in logic and the foundation of mathematics.* North-Holland, revised edition.

Bernadet, A. and Lengrand, S. (2011a). Complexity of strongly normalising λ-terms via non-idempotent intersection types. In *FOSSACS'11*, volume 6604 of *Lecture Notes in Computer Science*, pages 88 – 107. Springer.

Bernadet, A. and Lengrand, S. (2011b). Filter Models: Non-idempotent Intersection Types, Orthogonality and Polymorphism. In *CSL'11*, volume 12 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 51–66.

Bucciarelli, A., Ehrhard, T., and Manzonetto, G. (2007). Not enough points is enough. In *CSL'07*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer.

Bucciarelli, A., Kesner, D., and Ronchi Della Rocca, S. (2014). The inhabitation problem for non-idempotent intersection types. In *IFIP/TCS, Roma,1-3 September 14*, volume 8705 of *Lecture Notes in Computer Science*, pages 341–354. Springer.

Carraro, A., Ehrhard, T., and Salibra, A. (2010). Exponentials with infinite multiplicities. In Dawar, A. and Veith, H., editors, *CSL'10*, volume 6247 of *Lecture Notes in Computer Science*, pages 170–184. Springer.

Coppo, M., Dezani-Ciancaglini, M., Honsell, F., and Longo, G. (1984). Extended type structure and filter lambda models. In *Logic Colloquim'82*, pages 241–262.

Coppo, M., Dezani-Ciancaglini, M., and Venneri, B. (1981). Functional characters of solvable terms. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 27(1):45–58.

De Benedetti, E. and Ronchi Della Rocca, S. (2013). Bounding normalization time through intersection types. In Paolini, L., editor, *Proceedings of Sixth Workshop on Intersection Types and Related Systems (ITRS 2012)*, Electronic Notes in Theoretical Computer Science, pages 48–57. Cornell University Library.

de Carvalho, D. (2009). Execution time of lambda-terms via denotational semantics and intersection types. *CoRR*, abs/0905.4251. Available also as INRIA report RR 6638.

Di Gianantonio, P., Honsell, F., and Lenisa, M. (2008). A type assignment system for game semantics. *Theoretical Computer Science*, 398:150–169.

Ehrhard, T. (2012). Collapsing non-idempotent intersection types. In *CSL'12*, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 259–273.

Hindley, J. R. and Longo, G. (1980). Lambda calculus models and extensionality. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 26:289–310.

Honsell, F. and Ronchi della Rocca, S. (1990). Reasoning about interpretation in qualitative lambda-models. In *IFIP 2.2*, pages 505–521.

Honsell, F. and Ronchi Della Rocca, S. (1992). An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus. *Journal of Computer and System Sciences*, 45(1):49–75.

Kfoury, A. J. (2000). A Linearization of the Lambda-Calculus and Consequences. *Journal of Logic and Computation*, 10(3):411–436.

Kfoury, A. J. and Wells, J. B. (2004). Principality and type inference for intersection types using expansion variables. *Theoretical Computer Science*, 311(1-3):1–70.

Koymans, C. P. J. (1982). Models of the lambda calculus. *Information and Computation*, 52(3):306–323.

Mairson, H. and Neergaard, P. M. (2004). Types, potency, and idempotency: why nonlinearity and amnesia make a type system work. In *ICFP'04*, pages 138–149.

Manzonetto, G. (2009). A general class of models of $\mathcal{H}^\star$. In *MFCS'09*, volume 5734 of *Lecture Notes in Computer Science*, pages 574–586. Springer.

Manzonetto, G. (2012). What is a categorical model of the differential and the resource lambda-calculi? *Mathematical Structures in Computer Science*, 22(03):451–520.

Pagani, M. and Ronchi Della Rocca, S. (2010a). Linearity, non-determinism and solvability. *Fundamenta Informaticae*, 103:358–373.

Pagani, M. and Ronchi Della Rocca, S. (2010b). Solvability in resource lambda-calculus. In *FOSSACS'10*, volume 6014 of *Lecture Notes in Computer Science*, pages 358–373.

Paolini, L., Piccolo, M., and Ronchi Della Rocca, S. (2009). Logical semantics for stability. In *MFPS'09*, volume 249 of *Electronic Notes in Theoretical Computer Science*, pages 429–449.

Ronchi Della Rocca, S. and Paolini, L. (2004). *The Parametric λ-Calculus: a Metamodel for Computation*. Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag.

Selinger, P. (2002). The lambda calculus is algebraic. *Journal of Functional Programming*, 12(6):549–566.

Terui, K. (2006). Intersection types for computational complexity. In *Workshop on Implicit Computational Complexity (as part of GEOCAL'06)*, Marseille, February 2006.

Urzyczyn, P. (1999). The emptiness problem for intersection types. *Journal of Symbolic Logic*, 64(3):1195–1215.