New Semantical Insights Into Call-by-Value λ-Calculus

Giulio Manzonetto, Michele Pagani, Simona Ronchi Della Rocca

Fundamenta Informaticae 167 (2019) 1–25
DOI 10.3233/FI-2019-1800
IOS Press

New Semantical Insights Into Call-by-Value λ-Calculus

Giulio Manzonetto*
CNRS, LIPN
Université Paris–Nord, Villetaneuse, France
giulio.manzonetto@lipn.univ-paris13.fr

Michele Pagani
IRIF
Université Paris–Diderot, Paris, France
pagani@irif.fr

Simona Ronchi Della Rocca
Dipartimento di Informatica
Università di Torino, Torino, Italy
ronchi@di.unito.it

Abstract. Despite the fact that call-by-value λ-calculus was defined by Plotkin in 1977, we believe that its theory of program approximation is still at the beginning. A problem that is often encountered when studying its operational semantics is that, during the reduction of a λ-term, some redexes remain stuck (waiting for a value). Recently, Carraro and Guerrieri proposed to endow this calculus with permutation rules, naturally arising in the context of linear logic proof-nets, that succeed in unblocking a certain number of such redexes. In the present paper we introduce a new class of models of call-by-value λ-calculus, arising from non-idempotent intersection type systems. Beside satisfying the usual properties as soundness and adequacy, these models validate the permutation rules mentioned above as well as some reductions obtained by contracting suitable λI-redexes. Thanks to these (perhaps unexpected) features, we are able to demonstrate that every model living in this class satisfies an Approximation Theorem with respect to a refined notion of syntactic approximant. While this kind of results often require impredicative techniques like reducibility candidates, the quantitative information carried by type derivations in our system allows us to provide a combinatorial proof.

*Address for correspondence: Université Paris–Nord, Sorbonne Paris Cité, F-93430, Villetaneuse, France

Kewords: lambda calculus, call-by-value, relational semantics, approximation theorem, intersection types.

1. Introduction

The state-of-the-art Call-by-name and call-by-value evaluations constitute the two principal parameter passing mechanisms that are used in real programming languages: in the latter parameters are evaluated before being passed to a program, while in the former they can be passed as they are. The classical λ-calculus defined by Church [1] and the λ_v-calculus designed by Plotkin [2] are two theoretical calculi respectively modeling these parameter passing styles, through two slightly different evaluation rules. In λ-calculus the evaluation is made using the β-rule, formalized as $(\lambda \mathtt{x}.\mathtt{M})\mathtt{N} \to_\beta \mathtt{M}[\mathtt{N}/\mathtt{x}]$, while the λ_v-calculus is endowed with the β_v-rule, which is a restriction of $\to_\beta$ to the case where $\mathtt{N}$ is a value, i.e., an already evaluated term. The λ_v-calculus, however, is not yet *the* paradigmatic calculus for the call-by-value evaluation — in particular the reduction β_v is too weak and does not allow to model important operational properties like potential valuability and solvability. In fact some attempts have been made for enforcing it, the most important being the one in [3], where two commutation rules have been introduced with the result of avoiding some paradoxical situations, like the existence of unsolvable normal forms. Moreover, in [4], it has been proved that the theory of the extended calculus is conservative with respect to that of λ_v-calculus, so it can be considered as the best language for modelling the call-by-value reduction, until now.

The denotational semantics of both λ and λ_v-calculus has been intensively studied in the literature, but the mathematical strength of the results achieved is quite different in the two settings. Indeed, while for the first there are many models that are fully abstract with respect to the more usual operational semantics, for the the latter there is no known fully abstract model. A particularly interesting feature to study in denotational semantics is the Approximation Theorem, saying that the behavior of a term is the least upper bound of the behavior of its approximants, that are normal forms living in an extended language. In λ-calculus a very satisfying notion of approximant for Scott's model $\mathcal{D}_\infty$ has been built in [5]: in fact, the denotational equivalence is reflected in the syntactical shape of the approximants, and syntactically different approximants can be separated [6]. Such a separation property is the key tool to prove that this model is fully abstract with respect to the operational semantics induced by the leftmost-outermost evaluation strategy. For the λ_v-calculus the results are less satisfactory. Its denotational semantics has been studied in various settings: continuous functions domains [7, 8], stable functions domains [9], and relational domains [10, 11], but in each of these models there is a counterexample to the full abstraction property. In particular, in [8], an Approximation Theorem has been proven, but the syntactical shape of these approximants is too weak to be useful in practice.

Our result In this paper we use a semantic tool to further refine the reduction rule of λ_v-calculus. Namely we define in Section 3 a class of relational models of the λ_v-calculus, through a non-idempotent intersection type assignment system, parametrized with respect to an equivalence relation on types: every equivalence satisfying some given constraints induces a different model (Theorem 3.8). Every model is closed not only under the β_v equality and under the commutation rules cited in the previous paragraph, but also under a new rule (Definition 4.12), that we discovered studying the cut elimination of the type assignment system. The resulting notion of reduction, which we call $\to_{\mathtt{v}}$, is in our opinion the best candidate for modeling the call-by-value evaluation: in fact, we prove that the calculus equipped with the $\to_{\mathtt{v}}$-reduction is conservative with respect to the λ_v-calculus and that every model in the class is adequate for Plotkin's operational semantics (Theorem 5.12). The proof is based on an Approximation Theorem (Theorem 5.10), satisfied by all the models living in this class, where the notion of approximant is syntactically much more refined than in [8], being normal forms in an extended syntax, equipped with the $\to_{\mathtt{v}}$-reduction. The Approximation Theorem allows to reason in a finitary way about the denotation of terms: for example we use it to prove that all recursion operators (Definition 5.13) are equated in all models in the class, and that our models characterize the potentially valuable terms (Corollary 5.11), where a term is potentially valuable if there exists a substitution replacing variables by values making such a term reducible to a value. An important technical point needs to be cited: while in continuous and stable models the proof of an Approximation Theorem requires quite complex proof-theoretical tools like computability or reducibility candidates, the fact that the type assignment system we use carries out quantitative information opens the way to a very simple proof, by induction on the type derivation. This is in line with what happens in other relational based models, e.g. see the adequacy proof of [10].

Finally, we prove that the class of models induced by our type assignment system increases the semantic knowledge of the λ_v-calculus: in fact all the induced theories are different from those induced by the continuous models in [7, 8] and the relational models in [10, 11].

Related work Several variants of λ_v, have been introduced in the literature for modeling the call-by-value computation. Some authors proposed the introduction of new constructs to the syntax of λ_v, like Curien and Herbelin [12], Dyckhoff and Lengrand [13], Herbelin and Zimmerman [14], Accattoli and Paolini [15], Accattoli and Sacerdoti Coen [16]. Others proposed to extend the β_v reduction with new reduction rules, in particular [17, 18, 19, 20, 21, 22, 23, 14] present some variants of the commutation rules introduced by Carraro and Guerrieri in [3] and further studied in [4, 24, 25], often in a setting with explicit substitutions. A generalization of the commutation rules is used in [26] for a variant of the λ-calculus subsuming both call-by-name and call-by-value evaluations.

2. The $\lambda_{\mathtt{V}}$-calculus

Syntax We generally follow the syntax and the notation for λ-calculus, as defined in [27]. The syntax of terms of $\Lambda_{\mathtt{V}}$ is the same as the one of ordinary λ-calculus, i.e., *terms* and *term contexts* of $\Lambda_{\mathtt{V}}$ are generated respectively by the grammars:

$$\begin{array}{lcl} \mathtt{M},\mathtt{N},\mathtt{P},\mathtt{Q} & ::= & \mathtt{x} \mid \lambda\mathtt{x}.\mathtt{M} \mid \mathtt{M}\mathtt{M} \\ \mathtt{C} & ::= & \square \mid \mathtt{x} \mid \lambda\mathtt{x}.\mathtt{C} \mid \mathtt{M}\mathtt{C} \mid \mathtt{C}\mathtt{M} \end{array}$$

where $\mathtt{x}$ ranges over a countable set Var of *variables* (denoted by $\mathtt{x},\mathtt{y},\mathtt{z},\dots$), and $\square$ denotes the *hole* of the term context. As usual, we assume that λ-abstraction associates to the right, and has higher priority than application. So, for instance, we may write $\lambda\mathtt{xyz}.\mathtt{xyz}$ instead of $\lambda\mathtt{x}.(\lambda\mathtt{y}.(\lambda\mathtt{z}.((\mathtt{xy})\mathtt{z})))$.

The set $\mathsf{Val} \subset \Lambda_{\mathtt{V}}$ of *values* is defined by:

$$\mathtt{U}, \mathtt{V} \quad ::= \quad \mathtt{x} \mid \lambda\mathtt{x}.\mathtt{M}$$

Note that the set of values is closed under substitution. The set of *free variables* of a term $\mathtt{M}$ is denoted by $\mathtt{FV}(\mathtt{M})$. We say that a term $\mathtt{M}$ is *closed* whenever $\mathtt{FV}(\mathtt{M}) = \emptyset$ and we denote by $\Lambda^0_{\mathtt{V}}$ the set of all closed terms. Both terms and term contexts are considered up to *α-conversion*, i.e., modulo renaming of bound variables.

The symbol $\equiv$ denotes the syntactic identity, modulo α-conversion.

Given a term context $\mathtt{C}$, we denote by $\mathtt{C}(\mathtt{M})$ the term obtained from $\mathtt{C}$ by filling the hole with $\mathtt{M}$, possibly allowing the capture[1] of some free variables.

Definition 2.1. A *head term context* $\mathtt{H}$ is a term context having shape $\mathtt{H} \equiv (\lambda\mathtt{x}_1 \ldots \mathtt{x}_n.\Box)\mathtt{U}_1 \cdots \mathtt{U}_m$, for some $m, n \geq 0$, such that $\mathtt{U}_i \in \mathsf{Val}$, for all i $(1 \leq i \leq m)$.

Reduction The reduction relation $\to_{\beta_v}$ is the contextual closure of the rule:

$$(\lambda\mathtt{x}.\mathtt{M})\mathtt{V} \to \mathtt{M}[\mathtt{V}/\mathtt{x}] \text{ if } \mathtt{V} \in \mathsf{Val}$$

where $\mathtt{M}[\mathtt{V}/\mathtt{x}]$ denotes the capture avoiding simultaneous substitution of $\mathtt{V}$ for all free occurrences of $\mathtt{x}$ in $\mathtt{M}$. As usual, $\to^*_{\beta_v}$ denotes the reflexive and transitive closure of $\to_{\beta_v}$, and $=_{\beta_v}$ its reflexive, transitive and symmetric closure.

Remember that, in the call-by-value setting, the general η-reduction is unsound, but the following version restricted to values is sound:

$$\lambda\mathtt{x}.\mathtt{V}\mathtt{x} \to_{\eta_v} \mathtt{V} \text{ if } \mathtt{V} \in \mathsf{Val} \text{ and } \mathtt{x} \notin \mathtt{FV}(\mathtt{V}).$$

Operational semantics In the pioneering article [2], Plotkin defined an *operational preorder* on the call-by-value λ-calculus in the following way:

$$\mathtt{M} \leq_{\mathsf{op}} \mathtt{N} \quad \Leftrightarrow \quad \forall \mathtt{C}\,.\,\mathtt{C}(\mathtt{M}), \mathtt{C}(\mathtt{N}) \in \Lambda^0_{\mathtt{V}}\, (\exists \mathtt{U}_1 \in \mathsf{Val}\,.\,\mathtt{C}(\mathtt{M}) \to^*_{\beta_v} \mathtt{U}_1 \Rightarrow \exists \mathtt{U}_2 \in \mathsf{Val}\,.\,\mathtt{C}(\mathtt{N}) \to^*_{\beta_v} \mathtt{U}_2)$$

By replacing the double implication $\Leftrightarrow$ for the implication $\Rightarrow$ in the above formula, we obtain the definition of *operational equivalence* $=_{\mathsf{op}}$ between terms.

Semantic classes Terms are classified into (potentially) valuable or non (potentially) valuable depending on their capability of reducing to a value in suitable term contexts.

Definition 2.2. A term $\mathtt{M}$ is *valuable* if it reduces to a value; *potentially valuable* if there is a substitution s, replacing variables by values, such that $s(\mathtt{M})$ is valuable.

The notion of solvability, trying to grasp the notion of meaningful programs, can be defined in a similar way as for λ-calculus.

[1] In other words, the symbol $\Box$ can be thought of as a distinguished algebraic variable.

Definition 2.3. A term $\mathtt{M}$ is *solvable* if there exists a head term context $\mathtt{H}$ such that $\mathtt{H}(\mathtt{M}) \to^*_{\beta_v} \mathtt{I}$, where $\mathtt{I} \equiv \lambda\mathtt{x}.\mathtt{x}$ is the identity.

By definition, every term solvable in the $\lambda_{\mathtt{V}}$-calculus is also solvable in the regular λ-calculus. The converse is false, as shown by the term $\mathtt{M} \equiv (\lambda\mathtt{yx}.\mathtt{x})(\mathtt{DD})$ where $\mathtt{D} \equiv \lambda\mathtt{z}.\mathtt{zz}$. Indeed, $\mathtt{M}$ β-reduces to $\mathtt{I}$ (by contracting its outermost redex) while it can only β_v-reduce to itself as $\mathtt{DD} \to_{\beta_v} \mathtt{DD} \notin \mathsf{Val}$. Moreover, no head term context can interfere with this behaviour.

The following properties are easy to check:

Property 2.4. The class of solvable terms is a strict subclass of the potentially valuables.

In fact, if the term $\mathtt{M}$ is unsolvable then $\lambda\mathtt{x}.\mathtt{M}$ is unsolvable as well, but it is also a value and as such potentially valuable.

Property 2.5. Let $\mathtt{M}, \mathtt{N} \in \Lambda_{\mathtt{V}}$ be such that $\mathtt{M} =_{\beta_v} \mathtt{N}$. Then $\mathtt{M}$ is valuable (potentially valuable, solvable) if and only if $\mathtt{N}$ is valuable (potentially valuable, solvable).

$\lambda_{\mathtt{V}}$-models The general definition of a denotational model for $\Lambda_{\mathtt{V}}$, as first defined in [7] and further simplified in [28], is a modification of that for Λ given in [29].

Definition 2.6. ([28], Def. 10.0.1)
A *λ_v-model* is a quadruple $\mathcal{M} = \langle \mathbb{D}, \mathbb{V}, \circ, [\![\cdot]\!]^{\mathcal{M}} \rangle$, such that $\mathbb{D}$ is a set (the *carrier set*), $\mathbb{V} \subseteq \mathbb{D}$ is the set of *semantic values*, $\circ$ is a total map from $\mathbb{D}^2$ to $\mathbb{D}$. Moreover, if $\mathbb{E}$ is the class of functions from Var to $\mathbb{V}$, called *semantic environments* and ranged over by $\rho, \rho', \ldots$, the *interpretation function* $[\![\cdot]\!]^{\mathcal{M}}_{(\cdot)} : \Lambda_{\mathtt{V}} \times \mathbb{E} \to \mathbb{D}$ satisfies the following conditions:

1. $[\![\mathtt{x}]\!]^{\mathcal{M}}_\rho = \rho(\mathtt{x})$;
2. $[\![\mathtt{MN}]\!]^{\mathcal{M}}_\rho = [\![\mathtt{M}]\!]^{\mathcal{M}}_\rho \circ [\![\mathtt{N}]\!]^{\mathcal{M}}_\rho$;
3. $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathcal{M}}_\rho \circ d = [\![\mathtt{M}]\!]^{\mathcal{M}}_{\rho[d/\mathtt{x}]}$ if $d \in \mathbb{V}$;
4. if $[\![\mathtt{M}]\!]^{\mathcal{M}}_{\rho[d/\mathtt{x}]} = [\![\mathtt{M}']\!]^{\mathcal{M}}_{\rho'[d/\mathtt{x}]}$ for each $d \in \mathbb{V}$, then $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathcal{M}}_\rho = [\![\lambda\mathtt{x}.\mathtt{M}']\!]^{\mathcal{M}}_{\rho'}$;
5. $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathcal{M}}_\rho \in \mathbb{V}$ for all $\rho \in \mathbb{E}$;

where $\rho[d/\mathtt{x}]$ denotes the environment ρ' which coincides with ρ, except on $\mathtt{x}$, where ρ' takes the value d. When $\mathtt{M} \in \Lambda^0_{\mathtt{V}}$ we simply write $[\![\mathtt{M}]\!]^{\mathcal{M}}$ since the interpretation does not depend on ρ. We write $\mathcal{M} \models \mathtt{M} = \mathtt{N}$ whenever $\forall\rho \,.\, [\![\mathtt{M}]\!]^{\mathcal{M}}_\rho = [\![\mathtt{N}]\!]^{\mathcal{M}}_\rho$.

The principal difference with respect to the general definition of a λ-model [27, Def. 5.3.2] is the introduction of a subset $\mathbb{V}$ of the carrier set, which represents the semantical counterpart of the set Val of values. The fact that environments send variables to elements of $\mathbb{V}$ is a natural consequence of the fact that variables are values. Let us discuss now the five conditions on the interpretation function. Point 1 is rather standard. Points 2 and 4 ensure that the interpretation is context closed. In point 3 the equality is required to hold just in case the argument is a semantic value (thus reflecting the behaviour of $\to_{\beta_v}$). Finally, the condition in point 5 requires that all abstraction terms are interpreted as semantic values, thus reflecting the syntactical definition of Val. An immediate consequence of this definition is that any λ_v-model $\mathcal{M}$ is correct with respect to β_v- conversion, namely $\mathtt{M} =_{\beta_v} \mathtt{N}$ entails $\mathcal{M} \models \mathtt{M} = \mathtt{N}$.

Definition 2.7. A λ_v-model $\mathcal{M}$ is called *extensional* if $\mathtt{M} =_{\eta_v} \mathtt{N}$ implies $\mathcal{M} \models \mathtt{M} = \mathtt{N}$.

An extension of the $\lambda_{\mathtt{V}}$-calculus In [28, 8] it has been shown that the reduction $\to_{\beta_v}$ is too weak to obtain a syntactic characterization of the semantic properties defined above, like potential valuability and solvability. To solve this problem, the following two commutation rules have been introduced in [3]:

$$\begin{array}{lll} (\lambda\mathtt{x}.\mathtt{M})\mathtt{N}\mathtt{P} & \to_{\sigma_1} (\lambda\mathtt{x}.\mathtt{M}\mathtt{P})\mathtt{N} & \text{with } \mathtt{x} \notin \mathtt{FV}(\mathtt{P}),\\ \mathtt{V}((\lambda\mathtt{x}.\mathtt{M})\mathtt{N}) & \to_{\sigma_3} (\lambda\mathtt{x}.\mathtt{V}\mathtt{M})\mathtt{N} & \text{with } \mathtt{x} \notin \mathtt{FV}(\mathtt{V}). \end{array}$$

Notice that the rule σ_1 is the standard one, originally introduced by Regnier in call-by-name [30], while σ_3 is more recent and proper to the call-by-value setting.

Let $\to_{\beta_v,\sigma}$ be the union of the β_v-reduction and the two commutation reductions, and let the $\lambda_{\mathtt{V}}^{\sigma}$-*calculus* be the calculus equipped with this reduction rule: in [4] it has been proved that it is conservative with respect to $\lambda_{\mathtt{V}}$-calculus, in the following sense.

Property 2.8. A term of the $\lambda_{\mathtt{V}}^{\sigma}$-calculus is valuable (potentially valuable, solvable) if and only if it is valuable (potentially valuable, solvable) in the $\lambda_{\mathtt{V}}$-calculus.

In the same article the authors show that the operational semantics of $\lambda_{\mathtt{V}}$ is also preserved in $\lambda_{\mathtt{V}}^{\sigma}$.

The advantage of adding these permutation rules is that they open the way for an internal characterization of the semantical properties listed above. In particular, a nice syntactical characterization of solvability has been supplied.

Theorem 2.9. (Carraro and Guerrieri, [3])
A term $\mathtt{M}$ is solvable if and only if it β_v,σ-reduces to an $\mathtt{S}$-term of the following grammar:

$$\begin{array}{lll} \mathtt{S} & ::= & \mathtt{x}\mathtt{W}_1\cdots\mathtt{W}_n \mid \lambda\mathtt{x}.\mathtt{S} \mid (\lambda\mathtt{x}.\mathtt{S})(\mathtt{y}\mathtt{W}_1\cdots\mathtt{W}_m) \quad (n \geq 0, m > 0)\\ \mathtt{W} & ::= & \mathtt{V} \mid \mathtt{S} \qquad\qquad\qquad\qquad\qquad\qquad \text{where } \mathtt{V} \in \mathsf{Val} \end{array}$$

So the $\lambda_{\mathtt{V}}^{\sigma}$-calculus can be used as a syntactical tool to study the semantics of the $\lambda_{\mathtt{V}}$-calculus. We will show in the following sections that this feature is preserved by the denotational semantics.

3. A class of relational models for $\Lambda_{\mathtt{V}}$

The relational models of (call-by-name) λ-calculus can be easily described through non-idempotent intersection types (first shown in [31], then generalized in [32, 33]). Following the same idea, we define here a class of relational models of the $\lambda_{\mathtt{V}}$-calculus, described by an intersection type assignment system, parametric with respect to a set of type constants and a congruence relation between types.

$$\frac{}{\mathtt{x} : [\mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{x} : \mathtt{A}}\ \mathrm{ax} \qquad \frac{\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A} \quad \mathtt{A} \simeq \mathtt{B}}{\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{B}}\ \simeq$$

$$\frac{\Gamma, \mathtt{x} : \sigma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}}{\Gamma \vdash_{\mathbb{C},\simeq} \lambda \mathtt{x}.\mathtt{M} : \sigma \to \mathtt{A}}\ \to_I \qquad \frac{\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \sigma \to \mathtt{A} \quad \Delta \vdash_{\mathbb{C},\simeq} \mathtt{N} : \sigma}{\Gamma \uplus \Delta \vdash_{\mathbb{C},\simeq} \mathtt{MN} : \mathtt{A}}\ \to_E$$

$$\frac{}{\vdash_{\mathbb{C},\simeq} \mathtt{V} : [\,]}\ !_0 \qquad \frac{n > 0 \quad \forall i, 1 \leq i \leq n,\ \Gamma_i \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}_i}{\biguplus_{i=1}^{n} \Gamma_i \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\mathtt{A}_1, \ldots, \mathtt{A}_n]}\ !_{>0}$$

Figure 1. The relational parametric type assignment system $\mathsf{S}^{\mathbb{C},\simeq}$.

Definition 3.1.

- Let $\mathbb{C}$ be a countable (possibly empty) set of constants called *atomic types*. We define the set $\mathcal{T}_{\mathbb{C}}$ of *types over* $\mathbb{C}$ and the set $\mathcal{T}_{\mathbb{C}}^{!}$ of *finite multisets of types over* $\mathbb{C}$, by mutual induction as follows:

$$\begin{array}{rlr} \mathtt{A}, \mathtt{B}, \mathtt{C} & ::= \mathtt{a} \mid [\,] \mid \sigma \to \mathtt{A} & \qquad \mathcal{T}_{\mathbb{C}} \\ \sigma, \tau & \in \mathsf{M_f}(\mathcal{T}_{\mathbb{C}} \setminus \{[\,]\}) & \qquad \mathcal{T}_{\mathbb{C}}^{!} \end{array}$$

  where $\mathtt{a} \in \mathbb{C}$ and $\mathsf{M_f}(\mathcal{T}_{\mathbb{C}} \setminus \{[\,]\})$ denotes the set of finite multisets containing types different from $[\,]$. Multisets are represented as unordered lists, possibly with repetitions, and delimited by square parentheses. Given two multisets σ, τ we write $\sigma \uplus \tau$ for their union. Notice that the empty multiset $[\,]$ belongs to both $\mathcal{T}_{\mathbb{C}}$ and $\mathcal{T}_{\mathbb{C}}^{!}$. We denote by $\mathcal{T}_{\mathbb{C}}^{(!)}$ the set-theoretical union of $\mathcal{T}_{\mathbb{C}}$ and $\mathcal{T}_{\mathbb{C}}^{!}$, the meta-variables α, β varying over $\mathcal{T}_{\mathbb{C}}^{(!)}$.

- A *relational type theory* $\simeq$ is any congruence over $\mathcal{T}_{\mathbb{C}}$ such that:

$$\begin{array}{c} [\,] \simeq \mathtt{A} \text{ iff } [\,] = \mathtt{A} \\ \sigma \to \mathtt{A} \simeq \tau \to \mathtt{B} \text{ iff } \mathtt{A} \simeq \mathtt{B} \text{ and } \exists n \geq 0, \sigma = [\mathtt{A}_1, \ldots, \mathtt{A}_n], \tau = [\mathtt{B}_1, \ldots, \mathtt{B}_n] \\ \text{and } \forall i \leq n, \mathtt{A}_i = \mathtt{B}_i. \end{array}$$

Remark 3.2. Intuitively, the multiset of types $[\mathtt{A}_1, \ldots, \mathtt{A}_n]$ is an alternative notation for $\mathtt{A}_1 \wedge \cdots \wedge \mathtt{A}_n$, where the intersection connective enjoys associativity and commutativity, but not idempotency. Therefore the congruence on multisets needs to take into account the multiplicity of the elements.

A *context* Γ is a function from Var to $\mathcal{T}_{\mathbb{C}}^{!}$ having finite domain, this latter being defined as the set $\mathtt{dom}(\Gamma) = \{\mathtt{x} \in \mathrm{Var} \mid \Gamma(\mathtt{x}) \neq [\,]\}$. We write $\mathtt{x}_1 : \sigma_1, \ldots, \mathtt{x}_n : \sigma_n$ for the context Γ such that $\mathtt{dom}(\Gamma) \subseteq \{\mathtt{x}_1, \ldots, \mathtt{x}_n\}$ and $\Gamma(\mathtt{x}_i) = \sigma_i$ (for $1 \leq i \leq n$). Given two contexts Γ and Δ their union $\Gamma \uplus \Delta$ is defined component-wise.

Definition 3.3. The *relational parametric type assignment system* $\mathsf{S}^{\mathbb{C},\simeq}$, parametric with respect to $\mathbb{C}$ and $\simeq$ is defined in Figure 1, where Γ and Δ are contexts.

The system proves statements of the shape $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$, where Γ is a context, and α is either a type or a finite multiset of types over $\mathbb{C}$. When writing $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$ we intend that there is a derivation proving such a statement, and by $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$ that there exists a particular derivation, called Π, proving this statement.

We write $\vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$ when the context has empty domain.

A comment on the typing rules is in order. Rules (ax), $(\to_I)$, $(\to_E)$ are the usual rules for λ-calculus. Rule $(!_0)$ allows to characterize values through the empty multiset type, and it is essential to type terms like $\lambda\mathtt{x}.\mathtt{DD}$, which is a value despite the fact that $\mathtt{DD}$ is not typable in this system. Rule $(!_{>0})$ collects in a multiset a finite amount of types for the same term, so it is an auxiliary rule for typing the argument of an application; note that it cannot be iterated. The parametric rule $(\simeq)$ is self-explanatory.

In the denotational models for λ or $\lambda_{\mathtt{V}}$-calculus described through intersection types, two cases are possible. If the intersection is idempotent, the interpretation of a term is simply the set of types derivable for it. When the intersection is non-idempotent, this interpretation is not correct anymore, as shown in [31]: as proved in [32], a term needs to be interpreted as a set of pre-typings, where a pre-typing is a pair of a context and a type.

Definition 3.4. Let $\mathsf{S}^{\mathbb{C},\simeq}$ be a relational parametric type assignment system.

- A *pre-typing* is a pair $(\Gamma; \mathtt{A})$, where Γ is a context and $\mathtt{A}$ is a type.
- The equivalence $\simeq$ is extended to pre-typings component-wise:

$$\begin{array}{lcl} \Gamma \simeq \Delta & \iff & \mathtt{dom}(\Gamma) = \mathtt{dom}(\Delta) \text{ and } \forall \mathtt{x} \in \mathtt{dom}(\Gamma), \Gamma(\mathtt{x}) \simeq \Delta(\mathtt{x}), \\ (\Gamma; \mathtt{A}) \simeq (\Gamma'; \mathtt{A}') & \iff & \mathtt{A} \simeq \mathtt{A}' \text{ and } \Gamma \simeq \Gamma'. \end{array}$$

Lemma 3.5. Let $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$. If $\Gamma \simeq \Delta$ and $\alpha \simeq \beta$ then $\Delta \vdash_{\mathbb{C},\simeq} \mathtt{M} : \beta$.

Proof:
By induction on a derivation of $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$. □

Now we prove that every parametric relational type assignment system induces a model of $\Lambda_{\mathtt{V}}$. When writing "$\forall \mathtt{x} \in \mathtt{dom}(\Gamma), \forall \mathtt{B}^{\mathtt{x}} \in \Gamma(\mathtt{x}).\mathsf{P}(\mathtt{B}^{x})$" for some property P, we mean that if the context Γ has shape $\mathtt{x}_1 : [\mathtt{A}^1_1, \ldots, \mathtt{A}^1_{k_1}], \ldots, \mathtt{x}_n : [\mathtt{A}^n_1, \ldots, \mathtt{A}^n_{k_n}]$ then $\mathsf{P}(\mathtt{A}^i_j)$ holds for all i $(1 \le i \le n)$ and j $(1 \le j \le k_n)$.

Definition 3.6. Let $\mathsf{S}^{\mathbb{C},\simeq}$ be a parametric relational type assignment system. We define a quadruple $\mathcal{M}^{\mathbb{C},\simeq} = \langle \mathcal{S}^{\mathbb{C},\simeq}, \mathcal{V}^{\mathbb{C},\simeq}, \circ^{\mathbb{C},\simeq}, [\![\cdot]\!]^{\mathbb{C},\simeq} \rangle$ as follows.

- The carrier set $\mathcal{S}^{\mathbb{C},\simeq}$ of $\mathcal{M}^{\mathbb{C},\simeq}$ and its subset $\mathcal{V}^{\mathbb{C},\simeq}$ of semantic values are defined as follows (where s denotes a set of pre-typings):

$$\begin{array}{rl} \mathcal{S}^{\mathbb{C},\simeq} & := \{ s \mid \cup_{(\Gamma;\mathtt{A}) \in s} \mathtt{dom}(\Gamma) \text{ is finite and } s \text{ is closed under } \simeq \}, \\ \mathcal{V}^{\mathbb{C},\simeq} & := \left\{ s \in \mathcal{S}^{\mathbb{C},\simeq} \mid (\emptyset; [\,]) \in s \text{ and if } (\Gamma; [\,]) \in s \text{ then } \Gamma = \emptyset \right\}. \end{array}$$

  The first condition in $\mathcal{V}^{\mathbb{C},\simeq}$ is needed for having $[\![\mathtt{x}]\!]^{\mathbb{C},\simeq}_{\rho} = \rho(\mathtt{x})$. The second condition is necessary for having $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho} \circ d = [\![\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho[d/\mathtt{x}]}$. These conditions are non-standard but unproblematic because they are enjoyed by the sets of pre-typings interpreting values.

- The binary operation $\circ^{\mathbb{C},\simeq}$ is defined on $\mathcal{S}^{\mathbb{C},\simeq}$ in the following way:

$$\begin{aligned} s_1 \circ^{\mathbb{C},\simeq} s_2 = &\{(\Gamma \uplus \Delta; \mathtt{A}) \mid (\Gamma; [\,] \to \mathtt{A}) \in s_1, (\Delta; [\,]) \in s_2\} \\ &\cup \left\{(\Gamma \uplus (\uplus_{i=1}^{n} \Delta_i); \mathtt{A}) \mid n > 0, (\Gamma; [\mathtt{B}_1, \ldots, \mathtt{B}_n] \to \mathtt{A}) \in s_1, \forall i, 1 \leq i \leq n, (\Delta_i; \mathtt{B}_i) \in s_2\right\}. \end{aligned}$$

- Given an environment ρ, i.e. a map from Var to $\mathcal{V}^{\mathbb{C},\simeq}$, and a term $\mathtt{M}$, we set:

$$[\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq} = \left\{ (\Gamma; \mathtt{A}) \left| \begin{array}{l} \exists\Delta,\ \Delta \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A} \text{ and } \forall \mathtt{x} \in \mathsf{dom}(\Delta),\ \forall \mathtt{B}^{\mathtt{x}} \in \Delta(\mathtt{x}), \\ \exists (\Sigma_{\mathtt{B}^{\mathtt{x}}}; \mathtt{B}^{\mathtt{x}}) \in \rho(\mathtt{x}),\ \text{s.t. } \Gamma = \uplus_{\mathtt{x} \in \mathsf{dom}(\Delta)} \uplus_{\mathtt{B}^{\mathtt{x}} \in \Delta(\mathtt{x})} \Sigma_{\mathtt{B}^{\mathtt{x}}} \end{array} \right. \right\}$$

When $\mathtt{M} \in \Lambda_{\mathtt{V}}^{0}$ we have, up to isomorphism, that $[\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq} = \{\mathtt{A} \mid \ \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}\} \subseteq \mathcal{T}_{\mathbb{C}}$. We write $\mathcal{M}^{\mathbb{C},\simeq} \models \mathtt{M} \leq \mathtt{N}$ if and only if $\forall \rho\ .\ [\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq} \subseteq [\![\mathtt{N}]\!]_{\rho}^{\mathbb{C},\simeq}$ holds.

The definition of the interpretation $[\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq}$ given above might seem intricate at first, but the idea is to consider a derivation

$$\mathtt{x}_1 : [\mathtt{B}_1^1, \ldots, \mathtt{B}_{k_1}^1], \ldots, \mathtt{x}_n : [\mathtt{B}_1^n, \ldots, \mathtt{B}_{k_n}^n] \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$$

to decompose the context Γ into $k_1 + \cdots + k_n$ many pieces Σ_j^i (where $1 \leq i \leq n$ and $1 \leq j \leq k_i$) and to fetch the pairs $(\Sigma_j^i; \mathtt{B}_j^i)$ in the corresponding set $\rho(\mathtt{x}_i)$ as described in the following scheme:

$$\begin{array}{l} \qquad\qquad \rho(\mathtt{x}_1) \qquad\qquad \cdots \qquad\qquad \rho(\mathtt{x}_n) \\ \qquad\quad \in \qquad\quad \ni \qquad\qquad\quad \in \qquad\quad \ni \\ \Gamma = \widehat{\Sigma_1^1} \uplus \cdots \uplus \widehat{\Sigma_{k_1}^1} \uplus \ \cdots \ \uplus \ \widehat{\Sigma_1^n} \uplus \cdots \uplus \widehat{\Sigma_{k_n}^n} \\ \exists\Delta = \mathtt{x}_1 : [\,\underbrace{\mathtt{B}_1^1}\,, \ \ldots\ , \underbrace{\mathtt{B}_{k_1}^1}\,], \ldots, \mathtt{x}_n : [\,\underbrace{\mathtt{B}_1^n}\,, \ \ldots\ , \underbrace{\mathtt{B}_{k_n}^n}\,] \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A} \quad \Longleftrightarrow \quad (\Gamma; \mathtt{A}) \in [\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq}. \end{array}$$

Clearly the definition of interpretation is simpler in case of a closed term. In fact, if $\mathtt{M}$ is closed, then $\Delta \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$ implies $\mathsf{dom}(\Delta) = \emptyset$, which in its turn entails $[\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq} = \{(\emptyset; \mathtt{A}) \mid \emptyset \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}\}$.

Example 3.7. Consider, for the sake of simplicity, the relational type assignment system $\mathsf{S}^{\emptyset,=}$ having no atomic types and the syntactic equality as relational type theory. In the associated $\lambda_{\mathtt{V}}$-model $\mathcal{M}^{\emptyset,=}$, we have:

1. Let $\rho(\mathtt{z}) = \{(\mathtt{y} : \tau; \sigma \to \mathtt{A}), (\mathtt{y} : \sigma, \mathtt{z} : \tau; \sigma \to \mathtt{A})\}$. Since $\mathtt{z} : [\sigma \to \mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{z} : \sigma \to \mathtt{A}$, then $[\![\mathtt{z}]\!]_{\rho}^{\emptyset,=} = \{(\mathtt{y} : \tau; \sigma \to \mathtt{A}) \mid \mathtt{z} : [\sigma \to \mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{z} : \sigma \to \mathtt{A}\} \cup \{(\mathtt{y} : \sigma, \mathtt{z} : \tau; \sigma \to \mathtt{A}) \mid \mathtt{z} : [\sigma \to \mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{z} : \sigma \to \mathtt{A}\} = \rho(\mathtt{z})$.

2. $[\![\lambda\mathtt{x}.\mathtt{x}]\!]_{\rho}^{\emptyset,=} = [\![\lambda\mathtt{xy}.\mathtt{xy}]\!]_{\rho}^{\emptyset,=} = \{(\emptyset; [\mathtt{A}] \to \mathtt{A}) \mid \mathtt{A} \in \mathcal{T}_{\emptyset}\} \cup \{(\emptyset; [\,]), (\emptyset; [\,] \to [\,])\}$, hence the model is extensional.

3. $[\![\mathtt{D}]\!]_{\rho}^{\emptyset,=} = \{(\emptyset; [\sigma \to \mathtt{A}, \sigma] \to \mathtt{A}) \mid \sigma \in \mathcal{T}_{\emptyset}^{!}, \mathtt{A} \in \mathcal{T}_{\emptyset}\} \cup \{(\emptyset; [\,])\}$ for $\mathtt{D} \equiv \lambda\mathtt{z}.\mathtt{zz}$. From this it follows:

4. $[\![\mathtt{DD}]\!]_{\rho}^{\emptyset,=} = \emptyset$ and $[\![\lambda\mathtt{x}.\mathtt{DD}]\!]_{\rho}^{\emptyset,=} = \{(\emptyset; [\,])\}$.

We prove that for each choice of $\mathbb{C}$ and $\simeq$ respecting the conditions of Definition 3.1, the quadruple $\mathcal{M}^{\mathbb{C},\simeq}$ is a λ_v-model, according to Definition 2.6.

Theorem 3.8. $\mathcal{M}^{\mathbb{C},\simeq}$ is a λ_{v}-model.

Proof:
By checking that all conditions on the interpretation function stated in Definition 2.6 are satisfied.

(1) By the fact that only the axiom, the $\simeq$ and the $!_0$ rules can type a variable, the definition of $[\![\mathtt{x}]\!]_\rho^{\mathbb{C},\simeq}$ is equal to the following union of two sets:

$$\begin{aligned}[\![\mathtt{x}]\!]_\rho^{\mathbb{C},\simeq} &= \{(\Gamma;\mathtt{A}) \mid \exists(\Sigma_{\mathtt{A}'};\mathtt{A}') \in \rho(\mathtt{x}) \text{ s.t. } (\Sigma_{\mathtt{A}'};\mathtt{A}') \simeq (\Gamma;\mathtt{A})\} \\ &\quad \cup \{(\Gamma;\mathtt{A}) \mid \mathtt{A} = [\,], \Gamma = \emptyset\} \\ &= \rho(\mathtt{x}) \cup \{(\emptyset;[\,])\} = \rho(\mathtt{x})\end{aligned}$$

The first equality holds because $\rho(\mathtt{x})$ is closed under $\simeq$. The last equality holds because the semantic value $\rho(\mathtt{x})$ contains $\{(\emptyset;[\,])\}$ by Definition 3.6.

(2) Notice that if $\Delta \vdash_{\mathbb{C},\simeq} \mathtt{MN} : \mathtt{A}$ then either there exists $[\mathtt{B}_1,\ldots,\mathtt{B}_n]$ with $n > 0$, such that there exist $\Delta_0 \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\mathtt{B}_1,\ldots,\mathtt{B}_n] \to \mathtt{A}$ and $\Delta_1 \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{B}_1, \ldots, \Delta_n \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{B}_n$ for $\Delta = \uplus_i\Delta_i$, or there exist $\Delta_0 \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\,] \to \mathtt{A}$ and $\Delta_1 \vdash_{\mathbb{C},\simeq} \mathtt{N} : [\,]$ for $\Delta = \Delta_0 \uplus \Delta_1$.

This means that the set $[\![\mathtt{MN}]\!]_\rho^{\mathbb{C},\simeq}$ can be given by the following union:

$$[\![\mathtt{MN}]\!]_\rho^{\mathbb{C},\simeq} = \left\{(\Gamma;\mathtt{A}) \left| \begin{array}{l}\exists n > 0, \exists\Delta_0 \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\mathtt{B}_1,\ldots,\mathtt{B}_n] \to \mathtt{A} \text{ and } \forall i, 0 < i \leq n, \exists\Delta_i \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{B}_i \\ \text{and } \forall \mathtt{x} \in \mathsf{dom}(\Delta_i), \forall \mathtt{C}^\mathtt{x} \in \Delta_i(\mathtt{x}), \exists(\Sigma_{\mathtt{C}^\mathtt{x}};\mathtt{C}^\mathtt{x}) \in \rho(\mathtt{x}), \\ \text{s.t. } \Gamma = \uplus_{i=0}^n \uplus_{\mathtt{x}\in\mathsf{dom}(\Delta_i)} \uplus_{\mathtt{C}^\mathtt{x}\in\Delta_i(\mathtt{x})}\Sigma_{\mathtt{C}^\mathtt{x}}\end{array}\right.\right\}$$
$$\cup \left\{(\Gamma;\mathtt{A}) \left| \begin{array}{l}\exists\Delta_0 \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\,] \to \mathtt{A} \text{ and } \exists\Delta_1 \vdash_{\mathbb{C},\simeq} \mathtt{N} : [\,] \text{ and} \\ \forall i \in \{0,1\}, \forall \mathtt{x} \in \mathsf{dom}(\Delta_i), \forall \mathtt{C}^\mathtt{x} \in \Delta_i(\mathtt{x}), \exists(\Sigma_{\mathtt{C}^\mathtt{x}};\mathtt{C}^\mathtt{x}) \in \rho(\mathtt{x}), \\ \text{s.t. } \Gamma = \uplus_{i=0}^1 \uplus_{\mathtt{x}\in\mathsf{dom}(\Delta_i)} \uplus_{\mathtt{C}^\mathtt{x}\in\Delta_i(\mathtt{x})}\Sigma_{\mathtt{C}^\mathtt{x}}\end{array}\right.\right\}.$$

Introducing the definition $\Gamma_i = \uplus_{x\in\mathsf{dom}(\Delta_i)} \uplus_{\mathtt{C}^\mathtt{x}\in\Delta_i(\mathtt{x})} \Sigma_{\mathtt{C}^\mathtt{x}}$ into the expression, we can rewrite the union above as follows:

$$\left\{(\uplus_{i=0}^n\Gamma_i;\mathtt{A}) \left| \begin{array}{l}\exists n > 0, \exists\Delta_0 \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\mathtt{B}_1,\ldots,\mathtt{B}_n] \to \mathtt{A} \text{ and } \forall i, 0 < i \leq n, \exists\Delta_i \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{B}_i \\ \text{and } \forall \mathtt{x} \in \mathsf{dom}(\Delta_i), \forall \mathtt{C}^\mathtt{x} \in \Delta_i(\mathtt{x}), \exists(\Sigma_{\mathtt{C}^\mathtt{x}};\mathtt{C}^\mathtt{x}) \in \rho(\mathtt{x}), \\ \text{s.t. } \Gamma_i = \uplus_{\mathtt{x}\in\mathsf{dom}(\Delta_i)} \uplus_{\mathtt{C}^\mathtt{x}\in\Delta_i(\mathtt{x})} \Sigma_{\mathtt{C}^\mathtt{x}}\end{array}\right.\right\}$$
$$\cup \left\{(\Gamma_0 \uplus \Gamma_1;\mathtt{A}) \left| \begin{array}{l}\exists\Delta_0 \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\,] \to \mathtt{A} \text{ and } \exists\Delta_1 \vdash_{\mathbb{C},\simeq} \mathtt{N} : [\,] \text{ and} \\ \forall i \in \{0,1\}, \forall \mathtt{x} \in \mathsf{dom}(\Delta_i), \forall \mathtt{C}^\mathtt{x} \in \Delta_i(\mathtt{x}), \exists(\Sigma_{\mathtt{C}^\mathtt{x}};\mathtt{C}^\mathtt{x}) \in \rho(\mathtt{x}), \\ \text{s.t. } \Gamma_i = \uplus_{\mathtt{x}\in\mathsf{dom}(\Delta_i)} \uplus_{\mathtt{C}^\mathtt{x}\in\Delta_i(\mathtt{x})} \Sigma_{\mathtt{C}^\mathtt{x}}\end{array}\right.\right\}$$

which is then equal by definition to:

$$\{(\uplus_{i=0}^{n}\Gamma_i;\mathtt{A}) \mid \exists n>0, (\Gamma_0;[\mathtt{B}_1,\dots,\mathtt{B}_n]\to\mathtt{A})\in[\![\mathtt{M}]\!]_\rho^{\mathbb{C},\simeq}, \forall i, 1\le i\le n, (\Gamma_i;\mathtt{B}_i)\in[\![\mathtt{N}]\!]_\rho^{\mathbb{C},\simeq}\}$$
$$\cup\,\{(\Gamma_0\uplus\Gamma_1;\mathtt{A}) \mid (\Gamma_0;[\,]\to\mathtt{A})\in[\![\mathtt{M}]\!]_\rho^{\mathbb{C},\simeq}, (\Gamma_1;[\,])\in[\![\mathtt{N}]\!]_\rho^{\mathbb{C},\simeq}\}$$
$$= [\![\mathtt{M}]\!]_\rho^{\mathbb{C},\simeq}\circ[\![\mathtt{N}]\!]_\rho^{\mathbb{C},\simeq}.$$

(3) Given $d\in\mathcal{V}^{\mathbb{C},\simeq}$, we have:

$$[\![\lambda\mathtt{x}.\mathtt{M}]\!]_\rho^{\mathbb{C},\simeq}\circ d =$$
$$\left\{(\uplus_{i=0}^{n}\Gamma_i;\mathtt{A})\left|\begin{array}{l}\exists n>0,\ \Delta,\mathtt{x}:[\mathtt{B}_1,\dots,\mathtt{B}_n]\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A},\\ \text{and } \forall\mathtt{y}\in\mathtt{dom}(\Delta),\forall\mathtt{C}^{\mathtt{y}}\in\Delta(\mathtt{y}),\\ \exists(\Sigma_{\mathtt{C}^{\mathtt{y}}};\mathtt{C}^{\mathtt{y}})\in\rho(\mathtt{y}) \text{ s.t. } \Gamma_0=\uplus_{\mathtt{y}\in\mathtt{dom}(\Delta)}\uplus_{\mathtt{C}^{\mathtt{y}}\in\Delta_i(\mathtt{y})}\Sigma_{\mathtt{C}^{\mathtt{y}}},\\ \text{and } (\Gamma_i,\mathtt{B}_i)\in d \text{ for all } 1\le i\le n\end{array}\right.\right\}$$
$$\cup\left\{(\Gamma_0\uplus\Gamma_1;\mathtt{A})\left|\begin{array}{l}\exists\Delta,\mathtt{x}:[\,]\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A}, \text{and } \forall\mathtt{y}\in\mathtt{dom}(\Delta),\forall\mathtt{C}^{\mathtt{y}}\in\Delta(\mathtt{y}),\\ \exists(\Sigma_{\mathtt{C}^{\mathtt{y}}};\mathtt{C}^{\mathtt{y}})\in\rho(\mathtt{y}) \text{ s.t. } \Gamma_0=\uplus_{\mathtt{y}\in\mathtt{dom}(\Delta)}\uplus_{\mathtt{C}^{\mathtt{y}}\in\Delta_i(\mathtt{y})}\Sigma_{\mathtt{C}^{\mathtt{y}}},\\ \text{and } (\Gamma_1,[\,])\in d\end{array}\right.\right\}.$$

By hypothesis $d\in\mathcal{V}^{\mathbb{C},\simeq}$, so we have that $(\Gamma_1;[\,])\in d$ implies $\Gamma_1=\emptyset$ and that $(\emptyset,[\,])\in d$ is always true, so that we can rewrite the second component of the union above as:

$$\left\{(\Gamma_0;\mathtt{A})\left|\begin{array}{l}\exists\Delta\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A}, \text{and } \forall\mathtt{y}\in\mathtt{dom}(\Delta),\forall\mathtt{C}^{\mathtt{y}}\in\Delta(\mathtt{y}),\\ \exists(\Sigma_{\mathtt{C}^{\mathtt{y}}};\mathtt{C}^{\mathtt{y}})\in\rho(\mathtt{y}) \text{ s.t. } \Gamma_0=\uplus_{\mathtt{y}\in\mathtt{dom}(\Delta)}\uplus_{\mathtt{C}^{\mathtt{y}}\in\Delta_i(\mathtt{y})}\Sigma_{\mathtt{C}^{\mathtt{y}}}\end{array}\right.\right\}.$$

We conclude that the union above is equal to (where $\overline{\Delta}\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A}$ gathers the cases $\Delta,\mathtt{x}:[\mathtt{B}_1,\dots,\mathtt{B}_n]\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A}$ for $n>0$ and $\Delta,\mathtt{x}:[\,]\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A}$):

$$\left\{(\Gamma;\mathtt{A})\left|\begin{array}{l}\exists\overline{\Delta}\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{A}, \text{and } \forall\mathtt{y}\in\mathtt{dom}(\overline{\Delta}),\forall\mathtt{C}^{\mathtt{y}}\in\Delta(\mathtt{y}),\\ \exists(\Sigma_{\mathtt{C}^{\mathtt{y}}};\mathtt{C}^{\mathtt{y}})\in\rho[d/\mathtt{x}](\mathtt{y}) \text{ s.t. } \Gamma=\uplus_{\mathtt{y}\in\mathtt{dom}(\overline{\Delta})}\uplus_{\mathtt{C}^{\mathtt{y}}\in\overline{\Delta}(\mathtt{y})}\Sigma_{\mathtt{C}^{\mathtt{y}}}\end{array}\right.\right\} = [\![\mathtt{M}]\!]_{\rho[d/\mathtt{x}]}^{\mathbb{C},\simeq}$$

4) We show that there exists an element $d\in\mathcal{V}^{\mathbb{C},\simeq}$ such that $[\![\mathtt{M}]\!]_{\rho[d/\mathtt{x}]}^{\mathbb{C},\simeq}=[\![\mathtt{M}']\!]_{\rho'[d/\mathtt{x}]}^{\mathbb{C},\simeq}$ entails $[\![\lambda\mathtt{x}.\mathtt{M}]\!]_\rho^{\mathbb{C},\simeq}=[\![\lambda\mathtt{x}.\mathtt{M}']\!]_{\rho'}^{\mathbb{C},\simeq}$.

Let us assume $(\Gamma;\mathtt{A})\in[\![\lambda\mathtt{x}.\mathtt{M}]\!]_\rho^{\mathbb{C},\simeq}$ and show that $(\Gamma;\mathtt{A})\in[\![\lambda\mathtt{x}.\mathtt{M}']\!]_{\rho'}^{\mathbb{C},\simeq}$, the other inclusion being symmetrical. By definition, we have $\Gamma'\vdash_{\mathbb{C},\simeq}\lambda\mathtt{x}.\mathtt{M}:\mathtt{A}$ for some Γ'. Moreover, if $\mathtt{dom}(\Gamma')=\{\mathtt{x}_1,\dots,\mathtt{x}_m\}$ then $\Gamma'(\mathtt{x}_j)=[\mathtt{B}_1^j,\dots,\mathtt{B}_{n_j}^j]$ and there is a decomposition $\Gamma=\uplus_{1\le j\le m}\uplus_{1\le h\le n_j}\Gamma_h^j$ such that $(\Gamma_h^j;\mathtt{B}_h^j)\in\rho(\mathtt{x}_j)$. W.l.o.g., we may assume $\mathtt{x}\not\equiv\mathtt{x}_i$, for all $i\le m$, since $\mathtt{x}$ can be freely renamed. So either $\mathtt{A}\simeq[\mathtt{C}_1,\dots,\mathtt{C}_n]\to\mathtt{B}$, for some $n\ge1$, and $\Gamma',\mathtt{x}:[\mathtt{C}_1,\dots,\mathtt{C}_n]\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{B}$ or $\mathtt{A}\simeq[\,]\to\mathtt{B}$ and $\Gamma'\vdash_{\mathbb{C},\simeq}\mathtt{M}:\mathtt{B}$.

In the first case, let $d = \{(\mathtt{x} : [\mathtt{C}'_h]; \mathtt{C}''_h) \mid \mathtt{C}'_h \simeq \mathtt{C}''_h \simeq \mathtt{C}_h, 1 \leq h \leq n\} \cup \{(\emptyset, [\,])\}$. From this, it follows $(\Gamma, \mathtt{x} : [\mathtt{C}_1, \ldots, \mathtt{C}_n]; \mathtt{B}) \in [\![\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho[d/\mathtt{x}]}$ and consequently $(\Gamma, \mathtt{x} : [\mathtt{C}_1, \ldots, \mathtt{C}_n]; \mathtt{B}) \in [\![\mathtt{M}']\!]^{\mathbb{C},\simeq}_{\rho'[d/\mathtt{x}]}$ since $[\![\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho[d/\mathtt{x}]} = [\![\mathtt{M}']\!]^{\mathbb{C},\simeq}_{\rho'[d/\mathtt{x}]}$ by hypothesis.

Therefore there exist Δ, with $\mathtt{x} \notin \mathtt{dom}(\Delta)$, and $\Delta'_1, \ldots, \Delta'_k$ satisfying:

- $\Delta, \mathtt{x} : [\mathtt{C}'_1, \ldots, \mathtt{C}'_k] \vdash_{\mathbb{C},\simeq} \mathtt{M}' : \mathtt{B}$,
- $\forall \mathtt{y} \in \mathtt{dom}(\Delta)$, if $\Delta(\mathtt{y}) = [\mathtt{B}^{\mathtt{y}}_1, \ldots, \mathtt{B}^{\mathtt{y}}_{n_{\mathtt{y}}}]$ then there are $(\Delta^{\mathtt{y}}_h; \mathtt{B}^{\mathtt{y}}_h) \in \rho(\mathtt{y})$ for all h $(1 \leq h \leq n_{\mathtt{y}})$,
- $(\Delta'_j; \mathtt{C}'_j) \in d$ for all j $(1 \leq j \leq k)$,
- $\Gamma, \mathtt{x} : [\mathtt{C}_1, \ldots, \mathtt{C}_n] = (\uplus_{\mathtt{y} \in \mathtt{dom}(\Delta)} \uplus_{1 \leq h \leq n_{\mathtt{y}}} \Delta^{\mathtt{y}}_h) \uplus \Delta'_1 \uplus \cdots \uplus \Delta'_k$.

From the definition of d, we get $k = n$, $\Delta'_j \simeq \mathtt{x} : [\mathtt{C}_j]$ and $\mathtt{C}'_j \simeq \mathtt{C}_j$ for all j $(1 \leq j \leq n)$. By Lemma 3.5 and rules $(\to_I)$ and $(\simeq)$, we have $\Delta \vdash_{\mathbb{C},\simeq} \lambda\mathtt{x}.\mathtt{M}' : [\mathtt{C}_1, \ldots, \mathtt{C}_n] \to \mathtt{B}$, which gives by the conditions above $(\Gamma; \mathtt{A}) \in [\![\lambda\mathtt{x}.\mathtt{M}']\!]^{\mathbb{C},\simeq}_{\rho'}$.

In the second case, just take $d = \{(\emptyset; [\,])\}$. Then we have $(\Gamma; \mathtt{B}) \in [\![\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho[d/\mathtt{x}]}$ and consequently $(\Gamma; \mathtt{B}) \in [\![\mathtt{M}']\!]^{\mathbb{C},\simeq}_{\rho'[d/\mathtt{x}]}$ by hypothesis. By a similar reasoning as before we can conclude $(\Gamma; \mathtt{A}) \in [\![\lambda\mathtt{x}.\mathtt{M}']\!]^{\mathbb{C},\simeq}_{\rho'}$.

5) We have to prove that $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho}$ meets the two conditions defining $\mathcal{V}^{\mathbb{C},\simeq}$ in Definition 3.6. First, $(\emptyset, [\,]) \in [\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho}$, because $\vdash_{\mathbb{C},\simeq} \lambda\mathtt{x}.\mathtt{M} : [\,]$ by the $!_0$ rule. Second, suppose that $(\Gamma, [\,]) \in [\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho}$, then there is $\Delta \vdash_{\mathbb{C},\simeq} \lambda\mathtt{x}.\mathtt{M} : [\,]$ for some Δ. Notice that the only rule in Figure 1 giving type $[\,]$ to an abstraction is the $!_0$. (In particular, the $\simeq$ rule cannot be applied because, by Definition 3.1, $[\,]$ is only equivalent to itself.) This means that $\Delta = \emptyset$. Then by definition of $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho}$ we get $\Gamma = \emptyset$. We conclude $[\![\lambda\mathtt{x}.\mathtt{M}]\!]^{\mathbb{C},\simeq}_{\rho} \in \mathcal{V}^{\mathbb{C},\simeq}$. □

Since $\mathcal{M}^{\mathbb{C},\simeq}$ is a cbv-λ-model, it is correct with respect to β_v-conversion.

Theorem 3.9. (Soundness)
Let $\mathsf{S}^{\mathbb{C},\simeq}$ be a parametric relational type assignment system and $\mathcal{M}^{\mathbb{C},\simeq}$ be the associated cbv-λ-model. For all terms $\mathtt{M}, \mathtt{N}$ we have:

$$\mathtt{M} =_{\beta_v} \mathtt{N} \quad \Rightarrow \quad \mathcal{M}^{\mathbb{C},\simeq} \models \mathtt{M} = \mathtt{N}.$$

4. The semantic theory

Theorem 3.9 implies that the theory of our models is closed under $=_{\beta_v}$. We will now prove that it is actually closed under an extension of it. Namely the equality in the model is not only closed under $=_{\beta_v}$ plus the commutation rules, as expected, but also under a more refined notion of conversion. Roughly speaking, while $(\lambda\mathtt{x}.\mathtt{M})\mathtt{N} =_{\beta_v} \mathtt{M}[\mathtt{N}/\mathtt{x}]$ only in case $\mathtt{N}$ is valuable, we will see that $(\lambda\mathtt{x}.\mathtt{M})\mathtt{N}$ and $\mathtt{M}[\mathtt{N}/\mathtt{x}]$ can share the same typings for an arbitrary $\mathtt{N}$, with the proviso that all the occurrences of $\mathtt{x}$ in $\mathtt{M}$ are typed in such a way that the types for $\mathtt{x}$ and $\mathtt{N}$ match. Since Val is a proper subset of the typable terms, the resulting equality is an extension of $=_{\beta_v}$. While it is impossible to exactly reflect this equality in an untyped setting, being the system undecidable, we will grasp an approximation of it through the reduction rule $\to'_{\mathtt{V}}$.

Definition 4.1.

1. The set $\mathtt{o}(\mathtt{M})$ of *occurrences* in $\mathtt{M}$ is the set of contexts $\mathtt{C}$ such that there exists a term $\mathtt{N}$ verifying $\mathtt{C}(\mathtt{N}) = \mathtt{M}$, $\mathtt{N}$ being the *subterm of* $\mathtt{M}$ *at the occurrence* $\mathtt{C}$.

2. Given $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$, the set $\mathtt{to}(\Pi) \subseteq \mathtt{o}(\mathtt{M})$ of *typed occurrences in* $\mathtt{M}$ *in* Π is defined by induction on the structure of Π as follows:

   - $\mathtt{to}(\Pi) = \{\Box\}$ if Π is an instance of the axiom.
   - $\mathtt{to}(\Pi) = \{\Box\} \cup \{\lambda\mathtt{x}.\mathtt{C} \mid \mathtt{C} \in \mathtt{to}(\Pi')\}$ if the last rule of Π is $(\to_I)$, its subject is $\lambda\mathtt{x}.\mathtt{M}'$ and its premise is Π'.
   - $\mathtt{to}(\Pi) = \{\Box\} \cup \{\mathtt{C}\mathtt{M}_2 \mid \mathtt{C} \in \mathtt{to}(\Pi')\} \cup \{\mathtt{M}_1\mathtt{C} \mid \mathtt{C} \in \mathtt{to}(\Delta)\}$ if the last rule of Π is $(\to_E)$, its subject is $\mathtt{M}_1\mathtt{M}_2$, Π' and Δ are the major and minor premises of Π respectively.
   - $\mathtt{to}(\Pi) = \bigcap_{1\le i\le n}\{\mathtt{C} \mid \mathtt{C} \in \mathtt{to}(\Pi'_i)\}$ if the last rule of Π is $(!_{>0})$, and $(\Pi'_i)_{1\le i\le n}$ are its premises.
   - $\mathtt{to}(\Pi) = \emptyset$ if the last rule of Π is $(!_0)$.
   - $\mathtt{to}(\Pi) = \mathtt{to}(\Pi')$ if the last rule of Π is $(\simeq)$ and its premise is Π'.

Example 4.2. Note that, if the last rule is $(!_{>0})$, a subterm is typed if it is typed in all subjects of the premises. For example, in case of the derivation Π:

$$\frac{\Delta_1 \triangleright \mathtt{x} : [[\,] \to \mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{xy} : \mathtt{A} \quad \Delta_2 \triangleright \mathtt{x} : [[\mathtt{B}] \to \mathtt{A}], \mathtt{y} : [\mathtt{B}] \vdash_{\mathbb{C},\simeq} \mathtt{xy} : \mathtt{A}}{\mathtt{x} : [[\,] \to \mathtt{A}, [\mathtt{B}] \to \mathtt{A}], \mathtt{y} : [\mathtt{B}] \vdash_{\mathbb{C},\simeq} \mathtt{xy} : [\mathtt{A}, \mathtt{A}]}\ !_{>0}$$

we have $\mathtt{to}(\Pi) = \{\Box, \Box\mathtt{y}\}$.

Property 4.3. $\mathtt{V} \in \mathsf{Val}$ implies $\vdash_{\mathbb{C},\simeq} \mathtt{V} : [\,]$.

Proof:
Trivial, by the rule $(!_0)$. □

Definition 4.4. Given a derivation Π, the *measure of* Π, written $\mathtt{m}(\Pi)$, is the number of applications of rules in Π, without counting the rule $(\simeq)$.

Recall that α denotes a generic element of $\mathcal{T}_{\mathbb{C}}^{(!)} = \mathcal{T}_{\mathbb{C}} \cup \mathcal{T}_{\mathbb{C}}^{!}$.

Lemma 4.5. (Substitution)
Let $\Pi \triangleright \Gamma, \mathtt{x} : \sigma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$.

1. If $\sigma \neq [\,]$ and either $\mathtt{N} \in \mathsf{Val}$ or all occurrences of $\mathtt{x}$ in $\mathtt{M}$ are typed in Π, then $\Sigma \triangleright \Delta \vdash_{\mathbb{C},\simeq} \mathtt{N} : \sigma$ implies $\Pi' \triangleright \Gamma \uplus \Delta \vdash_{\mathbb{C},\simeq} \mathtt{M}[\mathtt{N}/\mathtt{x}] : \alpha$.

2. If $\sigma = [\,]$ then $\Sigma \triangleright \emptyset \vdash_{\mathbb{C},\simeq} \mathtt{N} : [\,]$ implies $\Pi' \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M}[\mathtt{N}/\mathtt{x}] : \alpha$.

In both cases $\mathtt{m}(\Pi') < \mathtt{m}(\Pi) + \mathtt{m}(\Sigma)$.

Proof:
We prove both cases by mutual induction on Π.

1. We first treat the case $\mathtt{N} \in \mathsf{Val}$. If Π is an application of rule (ax), possibly followed by some applications of rule $(\simeq)$, then $\mathtt{M} \equiv \mathtt{x}$, $\Gamma = \emptyset$ and $\sigma = [\mathtt{A}]$ for some $\mathtt{A} \simeq \alpha$. Moreover $\mathtt{m}(\Pi) = 1$ and $\Pi' = \Sigma$ so we get $\mathtt{m}(\Sigma) < 1 + \mathtt{m}(\Sigma)$.

The case Π is an application of rule $(!_0)$ does not apply.

The only non-trivial induction case is the rule $(!_{>0})$. Let $\alpha = [\mathtt{B}_1, \dots, \mathtt{B}_n]$ with $n > 0$ and let $\Pi_i \triangleright \Gamma_i, \mathtt{x} : \sigma_i \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{B}_i$ be the premises of the rule (for $1 \leq i \leq n$). In particular, $\Gamma = \uplus_{i=1}^n \Gamma_i$ and $\sigma = \uplus_{i=1}^n \sigma_i$. The derivation $\Sigma \triangleright \Delta \vdash_{\mathbb{C},\simeq} \mathtt{N} : \sigma$ must be the conclusion of a rule $(!_{>0})$, with premises $\Sigma_h \triangleright \Delta_h \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{A}_h$ $(1 \leq h \leq m)$. Let $\sigma_i = [\mathtt{A}^i_1, \dots, \mathtt{A}^i_{n_i}]$ with $m = \sum_{i=1}^n n_i$, then, by rule $(!_{>0})$ there exists Σ_i and Δ_i such that $\Sigma_i \triangleright \Delta_i \vdash_{\mathbb{C},\simeq} \mathtt{N} : \sigma_i$ and $\Delta = \uplus_{i=1}^n \Delta_i$. For every i there are two subcases:

- In case $n_i \neq 0$, by the induction hypothesis there exists $\Pi'_i \triangleright \uplus_{k=1}^{n_i} \Gamma_k \vdash_{\mathbb{C},\simeq} \mathtt{M}[\mathtt{N}/\mathtt{x}] : \mathtt{B}_i$ with $\mathtt{m}(\Pi'_i) < \mathtt{m}(\Pi_i) + \mathtt{m}(\Sigma'_i)$.
- Otherwise, the occurrences of $\mathtt{x}$ are either under the scope of an abstraction or subject of a rule $(!_0)$; since $\mathtt{N} \in \mathsf{Val}$, then by Property 4.3, there exists $\Sigma \triangleright \emptyset \vdash_{\mathbb{C},\simeq} \mathtt{N} : [\,]$ and the induction hypothesis on the second case applies. In this case the result follows by the rule $(!_{>0})$.

The case $\mathtt{N} \notin \mathsf{Val}$ is similar but easier. Note that, if the last rule is $(!_{>0})$, then the application of the induction hypothesis is possible thanks to the definition of typed occurrences.

2. Assume $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$ with $\mathtt{x} \notin \mathtt{dom}(\Gamma)$. If Π ends with an axiom, possibly followed by some applications of rule $(\simeq)$, then the subject is a variable $\mathtt{y} \not\equiv \mathtt{x}$. In this case $\mathtt{M}[\mathtt{N}/\mathtt{x}] \equiv \mathtt{y}$, Π is the desired derivation and the condition on the measure is satisfied. If Π ends with the application of rule $(!_0)$, possibly followed by some applications of rule $(\simeq)$, then remembering that values are closed under substitution, we observe that the desired derivation is actually a copy of Π, just changing the subject to $\mathtt{M}[\mathtt{N}/\mathtt{x}]$, and the proof is trivial.

All the other cases follow straightforwardly by induction. □

Lemma 4.6. (Inverse Substitution)
Let $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M}[\mathtt{N}/\mathtt{x}] : \alpha$.

1. If $\mathtt{N} \in \mathsf{Val}$ and some of the occurrences of $\mathtt{N}$ replacing $\mathtt{x}$ in $\mathtt{M}$ are typed by $\Sigma_i \triangleright \Gamma_i \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{A}_i$ $(1 \leq i \leq n)$, then $\Gamma_0, \mathtt{x} : [\mathtt{A}_1, \dots, \mathtt{A}_n] \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$ where $\Gamma = \uplus_{i=0}^n \Gamma_i$.
2. If $\mathtt{N} \notin \mathsf{Val}$ and all the occurrences of $\mathtt{N}$ replacing $\mathtt{x}$ in $\mathtt{M}$ are typed by $\Sigma_1, \dots, \Sigma_n$ $(n > 0)$, where $\Sigma_i \triangleright \Gamma_i \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{A}_i$ $(1 \leq i \leq n)$, then $\Gamma_0, \mathtt{x} : [\mathtt{A}_1, \dots, \mathtt{A}_n] \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$ and $\Gamma = \uplus_{i=0}^n \Gamma_i$.

Proof:
Both cases we proceed by induction on Π. Note that, in the second case, the base case is no more the axiom (which is impossible), but only the rule $(!_0)$. □

Example 4.7. Note that the substitution by a term which is not a value preserves types only when all the occurrences of the variable to be replaced are typed.

For example, consider the following derivation Π:

$$\dfrac{\dfrac{\mathtt{x} : [\mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{x} : \mathtt{A}}{\mathtt{x} : [\mathtt{A}] \vdash_{\mathbb{C},\simeq} \lambda\mathtt{y}.\mathtt{x} : [\,] \to \mathtt{A}} \to_I \quad \dfrac{}{\vdash_{\mathbb{C},\simeq} \mathtt{x} : [\,]} \,!_0}{\mathtt{x} : [\mathtt{A}] \vdash_{\mathbb{C},\simeq} (\lambda\mathtt{y}.\mathtt{x})\mathtt{x} : \mathtt{A}} \to_E$$

The first occurrence of $\mathtt{x}$ in Π is typed while the second occurrence is untyped. Consider now a derivation $\Sigma \rhd \mathtt{z} : [[\mathtt{B}] \to \mathtt{A}], \mathtt{t} : [\mathtt{B}] \vdash_{\mathbb{C},\simeq} \mathtt{zt} : \mathtt{A}$. In this case we cannot derive $\mathtt{z} : [[\mathtt{B}] \to \mathtt{A}], \mathtt{t} : [\mathtt{B}] \vdash_{\mathbb{C},\simeq} (\lambda\mathtt{y}.\mathtt{zt})(\mathtt{zt}) : \mathtt{A}$. Indeed, in order to assign a type to $(\lambda\mathtt{y}.\mathtt{zt})(\mathtt{zt})$, two premises with subject $\mathtt{z}$ are needed.

Lemma 4.8. (Commutation)
Let $\mathtt{M} \to_{\sigma_1} \mathtt{N}$ or $\mathtt{M} \to_{\sigma_3} \mathtt{N}$, then:

- for any $\Pi \rhd \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$, there exists $\Pi' \rhd \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{N} : \alpha$,
- and vice versa, for any $\Pi' \rhd \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{N} : \alpha$, there exists $\Pi \rhd \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \alpha$,

such that $\mathtt{m}(\Pi') = \mathtt{m}(\Pi)$.

Proof:
In fact, every commutation reduction corresponds exactly to a commutation of some rules in the type derivation.

For instance, let $\mathtt{V}((\lambda\mathtt{x}.\mathtt{M})\mathtt{N}) \to_{\sigma_3} (\lambda\mathtt{x}.\mathtt{VM})\mathtt{N}$, so that $\mathtt{V} \in \mathsf{Val}$ and $\mathtt{x} \notin \mathtt{FV}(\mathtt{V})$. The last rule of a derivation of $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{V}((\lambda\mathtt{x}.\mathtt{M})\mathtt{N}) : \mathtt{A}$ should be a $(\to_E)$, with premises, say, $\Pi_0 \rhd \Gamma_0 \vdash_{\mathbb{C},\simeq} \mathtt{V} : \sigma \to \mathtt{A}$ and $\Pi_1 \rhd \Gamma_1 \vdash_{\mathbb{C},\simeq} (\lambda\mathtt{x}.\mathtt{M})\mathtt{N} : \sigma$. Now there are two cases to consider, depending on whether σ is empty.

- If $\sigma = [\,]$ then the last rule of Π_1 should be the conclusion of a $(\to_E)$, with premises $\Pi'_1 \rhd \Gamma'_1 \vdash_{\mathbb{C},\simeq} \lambda\mathtt{x}.\mathtt{M} : \sigma' \to [\,]$ and $\Pi''_1 \rhd \Gamma''_1 \vdash_{\mathbb{C},\simeq} \mathtt{N} : \sigma'$. The former derives from a $(\to_I)$ with premise $\overline{\Pi'_1} \rhd \Gamma'_1, \mathtt{x} : \sigma' \vdash_{\mathbb{C},\simeq} \mathtt{M} : [\,]$. We then construct a derivation of $\Gamma \vdash_{\mathbb{C},\simeq} (\lambda\mathtt{x}.\mathtt{VM})\mathtt{N} : \mathtt{A}$ by composing $\Pi_0, \overline{\Pi'_1}$ and Π''_1.
- If $\sigma = [\mathtt{A}_1, \ldots, \mathtt{A}_n]$ for some $n > 0$, then for any $i \leq n$ there is a derivation $\Pi^i_1 \rhd \Gamma^i_1 \vdash_{\mathbb{C},\simeq} (\lambda\mathtt{x}.\mathtt{M})\mathtt{N} : \mathtt{A}_i$ with $\Gamma_1 = \biguplus_i \Gamma^i_1$. The last rule of Π^i_1 is a $(\to_E)$, with premises, say, $\Pi^i_\ell \rhd \Gamma^i_\ell \vdash_{\mathbb{C},\simeq} \lambda\mathtt{x}.\mathtt{M} : \sigma'_i \to \mathtt{A}_i$ and $\Pi^i_r \rhd \Gamma^i_r \vdash_{\mathbb{C},\simeq} \mathtt{N} : \sigma'_i$, so the reasoning is analogous to the previous case.

The converse case, i.e. deriving $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{V}((\lambda\mathtt{x}.\mathtt{M})\mathtt{N}) : \mathtt{A}$ from $\Gamma \vdash_{\mathbb{C},\simeq} (\lambda\mathtt{x}.\mathtt{VM})\mathtt{N} : \mathtt{A}$, is similar. □

Remark that Lemma 4.5 implies that the set of types associated with a term is invariant under a reduction which a slight extension of $\to_{\beta_v,\sigma}$. Definition 4.12 will give such an extension and Theorem 4.15 will prove its type invariance.

Example 4.9. 1. Consider the following derivations:

- $\Pi \rhd \mathtt{x} : [[\mathtt{A}] \to \mathtt{B}], \mathtt{y} : [\mathtt{A}] \vdash_{\mathbb{C},\simeq} \mathtt{xy} : \mathtt{B}$,
- $\Sigma \rhd \mathtt{z} : [[\mathtt{E}] \to \mathtt{A}], \mathtt{t} : [\mathtt{E}] \vdash_{\mathbb{C},\simeq} \mathtt{zt} : \mathtt{A}$.

By Lemma 4.5, there is $\Pi' \triangleright \mathtt{z} : [[\mathtt{E}] \to \mathtt{A}], \mathtt{t} : [\mathtt{E}], \mathtt{x} : [[\mathtt{A}] \to \mathtt{B}] \vdash_{\mathbb{C},\simeq} \mathtt{x}(\mathtt{zt}) : \mathtt{B}$, and this would imply that the derivation is sound for the reduction rule $(\lambda\mathtt{y}.\mathtt{xy})(\mathtt{zt}) \to \mathtt{x}(\mathtt{zt})$, but $\mathtt{zt} \notin \mathsf{Val}$. In fact, consider the potential valuability property, and take any substitution s — it is easy to check that $s((\lambda\mathtt{y}.\mathtt{xy})(\mathtt{zt}))$ is valuable if and only if $s(\mathtt{x}(\mathtt{zt}))$ is valuable. Indeed either $s(\mathtt{zt})$ is valuable, and the property holds trivially, by reducing $s(\mathtt{zt})$ to a value and then applying rule β_v to the outermost application, or $s(\mathtt{zt})$ is not valuable, and in this case $s((\lambda\mathtt{y}.\mathtt{xy})(\mathtt{zt}))$ cannot be reduced further, hence it is not valuable.

2. Replace the derivation Π in the previous point by $\Pi'' \triangleright \mathtt{x} : [[\,] \to \mathtt{B}] \vdash_{\mathbb{C},\simeq} \mathtt{x}(\lambda\mathtt{x}'.\mathtt{y}) : \mathtt{B}$: then the lemma does not ensure anymore that the reduction $(\lambda\mathtt{y}.\mathtt{x}(\lambda\mathtt{x}'.\mathtt{y}))(\mathtt{zt}) \to \mathtt{x}(\lambda\mathtt{x}'.\mathtt{zt})$ is sound. Indeed, let s be the substitution replacing $\mathtt{x}$ by $\lambda\mathtt{y}.\mathtt{y}$ and both $\mathtt{z}$ and $\mathtt{t}$ by $\lambda\mathtt{x}.\mathtt{xx}$, then $s((\lambda\mathtt{y}.\mathtt{x}(\lambda\mathtt{x}'.\mathtt{y}))(\mathtt{zt}))$ is not valuable, while $s(\mathtt{x}(\lambda\mathtt{x}'.\mathtt{zt}))$ is valuable, so the new reduction breaks Property 2.5.

In order to formalize the intuition above, we introduce the following notions.

Definition 4.10.

1. A free occurrence of a variable $\mathtt{x}$ *occurs at level* 0 *in* $\mathtt{M}$ if it does not occur free in $\mathtt{M}$ under the scope of a λ-abstraction.
2. An occurrence $\mathtt{C}$ of a variable $\mathtt{x}$ in $\mathtt{M}$ *is in active head position* if it occurs at level 0 in $\mathtt{M}$ and there exists a context $\mathtt{C}'$ such that $\mathtt{M} \equiv \mathtt{C}(\mathtt{x}) \equiv \mathtt{C}'(\mathtt{x}\mathtt{N}_1 \cdots \mathtt{N}_m)$ for some $m > 0$.
3. A variable $\mathtt{x}$ *is in active head position in* $\mathtt{M}$, written $\mathtt{x} \in_{\mathrm{ah}} \mathtt{M}$, if all occurrences of $\mathtt{x}$ in $\mathtt{M}$ are in active head position.

Lemma 4.11. Let $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$. If $\mathtt{x} \in_{\mathrm{ah}} \mathtt{M}$ then all occurrences of $\mathtt{x}$ in $\mathtt{M}$ are typed in Π.

Proof:
The proof is by induction on the structure of $\mathtt{M}$. It is well known that the general shape of a term is $\mathtt{M} \equiv \lambda\mathtt{x}_1 \ldots \mathtt{x}_n.\zeta\mathtt{M}_1 \cdots \mathtt{M}_m$ (for some $n, m \geq 0$), where ζ is either a variable or an abstraction $(\lambda\mathtt{y}.\mathtt{P})$. Let us notice that the hypothesis $\mathtt{x} \in_{\mathrm{ah}} \mathtt{M}$ implies that $\mathtt{M}$ is not a value. Hence we have:

(i) $\mathtt{M}$ is not an abstraction, i.e. $n = 0$, nor a variable, i.e. $m > 0$, and

(ii) for every $\mathtt{M}_i$ (with $1 \leq i \leq m$) if $\mathtt{x} \in_{\mathrm{ah}} \mathtt{M}_i$ then $\mathtt{M}_i$ is not a value, hence cannot be the subject of a $(!_0)$ rule.

Item (i) implies that $\mathtt{M}$ should be fired by a $(\to_E)$ rule. In particular, in the case $\zeta \equiv \mathtt{x}$, this occurrence of $\mathtt{x}$ is typed by the major premise of this $(\to_E)$ rule. Moreover, item (ii) implies that there are subderivations of Π typing each $\mathtt{M}_i$ with $\mathtt{x} \in_{\mathrm{ah}} \mathtt{M}_i$. So we conclude by applying the induction hypothesis. □

Definition 4.12. Define $\to'_{\mathtt{V}}$ as the contextual closure of the reduction rule:

$$(\lambda\mathtt{x}.\mathtt{M})\mathtt{N} \to \mathtt{M}[\mathtt{N}/\mathtt{x}] \qquad \text{if either } \mathtt{N} \in \mathsf{Val} \text{ or } \mathtt{x} \in_{\mathrm{ah}} \mathtt{M}$$

and let $\to_{\mathtt{V}}$ be $\to'_{\mathtt{V}} \cup \to_{\sigma_1} \cup \to_{\sigma_3}$. As usual, $\to^*_{\mathtt{V}}$ is the reflexive and transitive closure of $\to_{\mathtt{V}}$ and $=_{\mathtt{V}}$ is its symmetric, reflexive and transitive closure.

Now we will prove a subject reduction property for $\to_{\mathtt{v}}$. In order to prove it in a weighted version, we will use a measure defined in [3], which decreases when a commutation reduction is performed.

Definition 4.13. Let $\mathtt{s}(\mathtt{M})$ and $\#(\mathtt{M})$ be defined as follows:

$$\begin{array}{ll} \mathtt{s}(\mathtt{x}) = 2, & \#(\mathtt{x}) = 1, \\ \mathtt{s}(\lambda\mathtt{x}.\mathtt{M}) = \mathtt{s}(\mathtt{M}) + 1, & \#(\lambda\mathtt{x}.\mathtt{M}) = \#(\mathtt{M}) + \mathtt{s}(\mathtt{M}), \\ \mathtt{s}(\mathtt{MN}) = \mathtt{s}(\mathtt{M}) + \mathtt{s}(\mathtt{N}) + 1, & \#(\mathtt{MN}) = \#(\mathtt{M}) + \#(\mathtt{N}) + 2\mathtt{s}(\mathtt{M})\mathtt{s}(\mathtt{N}) - 1. \end{array}$$

We define a new measure of a derivation $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$, as $\mathtt{md}(\Pi) = \langle \mathtt{m}(\Pi), \#(\mathtt{M})\rangle$, and we let $<$ be the lexicographic order between pairs.

Lemma 4.14. Let $\mathtt{M} \to_{\mathtt{v}} \mathtt{M}'$:

1. (weighted subject reduction) for every $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$, there exists $\Pi' \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M}' : \mathtt{A}$ such that, in the case the contracted redex is a $\to'_{\mathtt{v}}$ typed redex, $\mathtt{md}(\Pi') < \mathtt{md}(\Pi)$.

2. (subject expansion) for every $\Pi' \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M}' : \mathtt{A}$ there exists $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$.

Proof:
If the reduction step $\mathtt{M} \to_{\mathtt{v}} \mathtt{M}'$ is of type $\to_{\sigma_1}$ or $\to_{\sigma_3}$, then the proof is a consequence of Lemma 4.8 and the fact that $\#(\mathtt{M})$ decreases is proven in [3].

Otherwise, suppose $\mathtt{M} \equiv \mathtt{C}((\lambda\mathtt{x}.\mathtt{N})\mathtt{Q})$ and $\mathtt{M}' \equiv \mathtt{C}(\mathtt{N}[\mathtt{Q}/\mathtt{x}])$ with either $\mathtt{Q} \in \mathsf{Val}$ or $\mathtt{x} \in_{\mathtt{ah}} \mathtt{N}$. The proof is by structural induction on $\mathtt{C}$.

If $\mathtt{C} \equiv \square$, then the statement (1) follows from Lemma 4.5, while (2) follows from Lemma 4.6. Notice that in the case $\mathtt{Q} \notin \mathsf{Val}$, the hypothesis $\mathtt{x} \in_{\mathtt{ah}} \mathtt{N}$ implies that all occurrences of $\mathtt{x}$ in $\mathtt{N}$ are typed in Π (resp. all occurrences of $\mathtt{Q}$ replacing $\mathtt{x}$ in $\mathtt{N}[\mathtt{Q}/\mathtt{x}]$ are typed in Π') by Lemma 4.11.

If $\mathtt{C} \equiv \lambda\mathtt{y}.\mathtt{C}'$, then there are two cases. Either $\mathtt{A} = [\,]$ and the derivation (Π or Π', depending on which statement we are proving) consists in a $(!_0)$ rule, or $\mathtt{A} \simeq \sigma \to \mathtt{B}$ and the derivation contains a sub-derivation typing the context $\mathtt{C}'$. The first case is trivial (in particular notice that in this case the reduced redex is not typed, so we prove nothing about the measure $\mathtt{md}$), while the second case is a consequence of the induction hypothesis.

The cases $\mathtt{C} \equiv \mathtt{PC}'$ and $\mathtt{C} \equiv \mathtt{C}'\mathtt{P}$ follow from the induction hypothesis. □

As a corollary, we obtain the following theorem.

Theorem 4.15. Every model $\mathcal{M}^{\mathbb{C},\simeq}$ is correct with respect to $=_{\mathtt{v}}$, i.e.,

$$\mathtt{M} =_{\mathtt{v}} \mathtt{N} \quad \Rightarrow \quad \mathcal{M}^{\mathbb{C},\simeq} \models \mathtt{M} = \mathtt{N}.$$

Proof:
By Lemma 4.14, the hypothesis $\mathtt{M} =_{\mathtt{v}} \mathtt{N}$ implies that $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$ if and only if $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{N} : \mathtt{A}$ holds. By the definition of the interpretation function $[\![\cdot]\!]_{\cdot}^{\mathbb{C},\simeq}$, we conclude that $[\![\mathtt{M}]\!]_{\rho}^{\mathbb{C},\simeq} = [\![\mathtt{N}]\!]_{\rho}^{\mathbb{C},\simeq}$ holds, for all environments ρ. □

The next quite technical corollary will be useful in the next section.

Corollary 4.16. Let $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$. Then there is $\mathtt{M}'$ such that $\mathtt{M} \to_{\mathtt{V}}^{*} \mathtt{M}'$, and $\Pi' \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M}' : \mathtt{A}$, where all the $\to_{\mathtt{V}}$ redexes are untyped in Π'.

Proof:
We can limit ourself to reduce the typed redexes, so $\mathtt{md}(\Pi)$ at every step decreases. □

5. An approximation theorem

In this section we prove that the semantical properties of potential valuability and solvability are preserved by the reduction rule $\to_{\mathtt{V}}$ (Theorem 5.5). This suggests a definition of approximants (Definition 5.6) and an approximantion theorem (Theorem 5.10) which gives as a corollary the adequacy of our relational typed models with respect to Plotkin's operational preorder (Theorem 5.12).

Let us recall that the approximation theorem is a standard tool to study the theory induced by a model. Namely approximants are normal forms in an extended language, and the theorem states that the interpretation of term is the collection of the interpretation of its approximants, thus allowing to reason about its interpretation via structural induction.

First, let us introduce the following notation:

Notation 5.1. Let $\mathtt{Q} \subseteq_0 \mathtt{M}$ denote that $\mathtt{Q}$ is a subterm of $\mathtt{M}$, which does not occur under a λ-abstraction.

Lemma 5.2. Let $\mathtt{Q} \subseteq_0 \mathtt{M}$ and assume that $\mathtt{Q}$ is not potentially valuable. Then $\mathtt{M}$ also is not potentially valuable, and so also unsolvable.

Proof:
Assume that $\mathtt{Q}$ is not potentially valuable, i.e., for all substitutions s, we have $s(\mathtt{Q}) \not\to_{\beta_v}^{*} \mathtt{V} \in \mathsf{Val}$. Note that the case $\mathtt{Q} \equiv \mathtt{M}$ is trivial, so we will consider only the case $\mathtt{Q}$ is a proper subterm of $\mathtt{M}$. We proceed by structural induction on $\mathtt{M}$. Note that the hypothesis $\mathtt{Q} \subseteq_0 \mathtt{M}$ implies that $\mathtt{M}$ cannot be a value, therefore the general shape of $\mathtt{M}$ is $\mathtt{P}\mathtt{M}_1 \cdots \mathtt{M}_n$, with $\mathtt{P}$ a value, $n > 0$ and at least for one $i \leq n$, $\mathtt{Q} \subseteq_0 \mathtt{M}_i$. By induction hypothesis we have that $s(\mathtt{M}_i)$ is not valuable. Suppose by contradiction that there exists a substitution s such that $s(\mathtt{M}) \equiv s(\mathtt{P})s(\mathtt{M}_1) \cdots s(\mathtt{M}_n)$ is valuable: this means that all arguments of $s(\mathtt{P})$ should be eventually consumed during the reduction of $s(\mathtt{M})$, in particular $s(\mathtt{M}_i)$ should reduce to a value in order to be fired by a β_v redex. This contradicts the hypothesis that $s(\mathtt{M}_i)$ is not valuable. □

Lemma 5.3.

1. $\mathtt{x} \in_{\mathsf{ah}} \mathtt{M}$ and $\mathtt{M} \to_{\beta_v}^{*} \mathtt{N}$ entail $\mathtt{x} \in_{\mathsf{ah}} \mathtt{N}$;

2. $(\lambda\mathtt{x}.\mathtt{M})\mathtt{N} \to_{\mathtt{V}} \mathtt{M}[\mathtt{N}/\mathtt{x}]$ implies that either $\mathtt{N}$ is valuable, or for all $\mathtt{M}'$ satisfying $\mathtt{M}[\mathtt{N}/\mathtt{x}] \to_{\beta_v}^{*} \mathtt{M}'$, there exists an $\mathtt{N}'$ such that $\mathtt{N} \to_{\beta_v}^{*} \mathtt{N}'$ and $\mathtt{N}' \subseteq_0 \mathtt{M}'$.

Proof:
1. Indeed, $\mathtt{x} \in_{\mathtt{ah}} \mathtt{M}$ implies that every occurrence of $\mathtt{x}$ in $\mathtt{M}$ is of shape $\mathtt{x}\mathtt{M}_1 \cdots \mathtt{M}_m$ for some $m \geq 1$. Such subterms cannot be erased nor substituted for a variable occurring under an abstraction by a $\to_{\beta_v}$ reduction.
2. Assume that $(\lambda\mathtt{x.M})\mathtt{N} \to_{\mathtt{v}} \mathtt{M}[\mathtt{N}/\mathtt{x}]$ and $\mathtt{N}$ is not valuable. Then every occurrence of $\mathtt{N}$ in $\mathtt{M}[\mathtt{N}/\mathtt{x}]$ is of the shape $\mathtt{N}\mathtt{M}_1 \cdots \mathtt{M}_m$, with $m \geq 1$. As $\mathtt{N}$ is not valuable, the same reasoning as in the first point of this lemma applies. □

Lemma 5.4. Let $\mathtt{M} \to_{\mathtt{v}} \mathtt{N}$. $\mathtt{M}$ is solvable if and only if $\mathtt{N}$ is solvable.

Proof:
If $\mathtt{M} \to_{\mathtt{v}} \mathtt{N}$ because either $\mathtt{M} \to_{\sigma_1} \mathtt{N}$ or $\mathtt{M} \to_{\sigma_3} \mathtt{N}$, then this conservativity result has been already proved in [3].

If $\mathtt{M} \to_{\mathtt{v}} \mathtt{N}$ is actually a $\to_{\beta_v}$ reduction, then it is a consequence of the confluence theorem for $\to_{\beta_v}$ [28].

The only case we need to consider is the case where $\mathtt{M} \equiv \mathtt{C}((\lambda\mathtt{x.P})\mathtt{Q})$ and $\mathtt{N} \equiv \mathtt{C}(\mathtt{P}[\mathtt{Q}/\mathtt{x}])$, with $\mathtt{Q} \notin \mathsf{Val}$ and $\mathtt{x} \in_{\mathtt{ah}} \mathtt{P}$. The proof is by structural induction on the context $\mathtt{C}$.

Case $\mathtt{C} \equiv \Box$. Let consider first the only if case: there is a head context $\mathtt{H} \equiv (\lambda\mathtt{y}_1 \ldots \mathtt{y}_n.\Box)\mathtt{M}_1 \cdots \mathtt{M}_m$ such that $\mathtt{H}((\lambda\mathtt{x.P})\mathtt{Q}) \to^*_{\beta_v} \mathtt{I}$, we will actually prove that also $\mathtt{H}(\mathtt{P}[\mathtt{Q}/\mathtt{x}]) \to^*_{\beta_v} \mathtt{I}$. By hypothesis we have: $(\lambda\mathtt{y}_1 \ldots \mathtt{y}_n.(\lambda\mathtt{x.P})\mathtt{Q})\mathtt{M}_1 \cdots \mathtt{M}_m \to^*_{\beta_v} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.(\lambda\mathtt{x.P}')\mathtt{Q}'$, where $\mathtt{P}' \equiv \mathtt{P}[\mathtt{M}_i/\mathtt{y}_i]_{1\leq i\leq m}$ and $\mathtt{Q}' \equiv \mathtt{Q}[\mathtt{M}_i/\mathtt{y}_i]_{1\leq i\leq m}$. So $\mathtt{Q}'$ must be valuable, i.e., $\mathtt{Q}' \to^*_{\beta_v} \mathtt{Q}'' \in \mathsf{Val}$ and the reduction is of the shape

$$\begin{aligned} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.(\lambda\mathtt{x.P}')\mathtt{Q}' &\to^*_{\beta_v} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.(\lambda\mathtt{x.P}'')\mathtt{Q}'' \\ &\to^*_{\beta_v} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.\mathtt{P}''[\mathtt{Q}''/\mathtt{x}] \\ &\to^*_{\beta_v} \mathtt{I} \end{aligned}$$

Consider filling the same context $\mathtt{H}$ with $\mathtt{P}[\mathtt{Q}/\mathtt{x}]$. Then we get

$$(\lambda\mathtt{y}_1 \ldots \mathtt{y}_n.\mathtt{P}[\mathtt{Q}/\mathtt{x}])\mathtt{M}_1 \cdots \mathtt{M}_m \to^*_{\beta_v} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.\mathtt{P}''[\mathtt{Q}''/\mathtt{x}]$$

and the proof is given.

On the other side, assume $\mathtt{H}(\mathtt{P}[\mathtt{Q}/\mathtt{x}]) \to^*_{\beta_v} \mathtt{I}$ and let us prove that we have also $\mathtt{H}((\lambda\mathtt{x.P})\mathtt{Q}) \to^*_{\beta_v} \mathtt{I}$.

$$(\lambda\mathtt{y}_1 \ldots \mathtt{y}_n.\mathtt{P}[\mathtt{Q}/\mathtt{x}])\mathtt{M}_1 \cdots \mathtt{M}_m \to^*_{\beta_v} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.\mathtt{P}'[\mathtt{Q}'/\mathtt{x}] \to^*_{\beta_v} \mathtt{I}.$$

By Lemma 5.3(2), either $\mathtt{Q}'$ is valuable or $\mathtt{Q}' \subseteq_0 \mathtt{P}'[\mathtt{Q}'/\mathtt{x}]$, then by Lemma 5.2, $\mathtt{Q}'$ must be potentially valuable. Always by Lemma 5.3(2), since $\mathtt{Q}'$ occurs in subterms of $\mathtt{P}'[\mathtt{Q}'/\mathtt{x}]$ having shape $\mathtt{Q}'\mathtt{P}_1 \cdots \mathtt{P}_k$, such subterms need to be transformed into values in order to reach the identity, so by definition of $\subseteq_0$, $\mathtt{Q}'$ must reduce to an abstraction $\mathtt{Q}''$. So the reduction is of the shape $\lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.\mathtt{P}'[\mathtt{Q}'/\mathtt{x}] \to^*_{\beta_v} \lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.\mathtt{P}''[\mathtt{Q}''/\mathtt{x}]$, where $\mathtt{Q}''$ is a value. Since the notion of solvability is closed under $=_{\beta_v}$, $\lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.(\lambda\mathtt{x.P}'')\mathtt{Q}'' \to^*_{\beta_v} \mathtt{I}$, and finally $\lambda\mathtt{y}_{m+1} \ldots \mathtt{y}_n.(\lambda\mathtt{x.P})\mathtt{Q} \to^*_{\beta_v} \mathtt{I}$.

The other cases follow by induction, taking into account that unsolvability is preserved by abstraction and application as shown in [28]. □

Theorem 5.5. The notions of potential valuability and solvability are preserved by the $\to_{\mathtt{V}}$ reduction.

Proof:
The case of potential valuability derives directly from Lemma 5.2, just considering that $(\lambda\mathtt{x}.\mathtt{R})\mathtt{Q} \to'_{\mathtt{V}} \mathtt{R}[\mathtt{Q}/\mathtt{x}]$ implies $\mathtt{Q} \subseteq_0 \mathtt{R}[\mathtt{Q}/\mathtt{x}]$. The case of solvability follows from Lemma 5.4. □

The new reduction rule $\to_{\mathtt{V}}$ opens the way to define a notion of approximants of a term which is more precise than the one given in [28].

Definition 5.6. Let us consider $\Lambda_{\mathtt{V}}\Omega$ be the set of terms generated by the rules of λ-calculus plus a constant Ω. We define the following subset $\mathcal{A} \subset \Lambda_{\mathtt{V}}\Omega$ of *approximants*:

$$\begin{array}{rcll} \mathbb{A} & ::= & \lambda\mathtt{x}.\Omega & \\ & \mid & \lambda\mathtt{x}.\mathbb{A} & \\ & \mid & \mathtt{x}\mathbb{A}_1\cdots\mathbb{A}_m & m \geq 0, \\ & \mid & (\lambda\mathtt{x}.\mathbb{A})(\mathtt{y}\mathbb{A}_1\cdots\mathbb{A}_m) & m > 0, \mathtt{x} \notin_{\mathtt{ah}} \mathbb{A}. \end{array}$$

We consider $\Lambda_{\mathtt{V}}\Omega$ partially ordered by the smallest partial order $\preceq$ which is context closed and contains the rule $\Omega \preceq \mathtt{M}$, for any $\mathtt{M} \in \Lambda_{\mathtt{V}}\Omega$.

We define the *set of approximants of a term* $\mathtt{M}$ in the following way:

$$\mathcal{A}(\mathtt{M}) = \{\mathbb{A} \mid \mathtt{M} \to^*_{\mathtt{V}} \mathtt{N} \text{ and } \mathbb{A} \in \mathcal{A} \text{ and } \mathbb{A} \preceq \mathtt{N}\}.$$

Lemma 5.7. If $\mathtt{M}$ is a $\to_{\mathtt{V}}$-normal form, then $\mathtt{M} \in \mathcal{A}$.

Proof:
Assume that $\mathtt{M}$ is normal and proceed by structural induction.

The cases $\mathtt{M} \equiv \lambda\mathtt{x}.\mathtt{M}'$ and $\mathtt{M} \equiv \mathtt{x}\mathtt{M}_1\cdots\mathtt{M}_m$ ($m \geq 0$) follow straightforwardly from the induction hypothesis.

Consider now $\mathtt{M} \equiv (\lambda\mathtt{x}.\mathtt{M}')\mathtt{N}_1\cdots\mathtt{N}_n$ ($n \geq 1$). We must have $\mathtt{x} \notin_{\mathtt{ah}} \mathtt{M}'$ and $n = 1$ otherwise $\mathtt{M}$ would contain a $\to'_{\mathtt{V}}$ or $\to_{\sigma_1}$ redex, respectively. Moreover $\mathtt{N}_1$ cannot have the shape $(\lambda\mathtt{y}.\mathtt{N}')\mathtt{P}_1\cdots\mathtt{P}_k$ otherwise $\mathtt{M}$ would have a $\to_{\beta_v}$ redex (for $k = 0$) or a $\to_{\sigma_3}$ redex for $k > 0$, whence $\mathtt{N}_1 \equiv \mathtt{y}\mathtt{M}_1\cdots\mathtt{M}_m$. Finally, we must have $m > 0$ otherwise $\mathtt{M}$ would have a $\to_{\beta_v}$ redex, so we conclude by induction hypothesis. □

Notice that the converse of Lemma 5.7 does not hold, because approximants may still contain σ_3-redexes, like $\mathtt{x}((\lambda\mathtt{y}.\mathtt{z})(\mathtt{x}\mathtt{z})) \in \mathcal{A}$. A more complicated grammar taking care of such redexes has been obtained by Guerrieri [34] (see also [35]).

Both the type system and the reduction rules generalize straightforwardly to terms in $\Lambda_{\mathtt{V}}\Omega$. To this aim it is sufficient to extend the set of values by defining $\mathsf{Val}_\Omega = \mathrm{Var} \cup \{\lambda\mathtt{x}.\mathtt{M}\}$ for any term $\mathtt{M}$ in $\Lambda_{\mathtt{V}}\Omega$. For every reduction rule $\to_{\mathtt{R}}$, where $\mathtt{R} \in \{\beta_v, \mathtt{V}, \sigma\}$, we denote by $\to_{\mathtt{R}\Omega}$ its extension to terms in $\Lambda_{\mathtt{V}}\Omega$. The type assignment system of Figure 1 is extended to $\Lambda_{\mathtt{V}}\Omega$, taking into account that the constant Ω is not considered as a value and cannot be typed. It is immediate to check that the extended type assignment system is closed under $=_{\mathtt{V}\Omega}$. Clearly, the following holds:

Lemma 5.8. Let $\mathtt{N}$ be a λ-term and $\mathtt{M}, \mathtt{M}'$ be terms in $\Lambda_{\mathtt{V}}\Omega$.

1. If $\mathtt{M} \preceq \mathtt{N}$ and $\mathtt{M} \to_{\beta_v\Omega} \mathtt{M}'$ then there exists a λ-term $\mathtt{N}'$ such that $\mathtt{N} \to_{\beta_v} \mathtt{N}'$ and $\mathtt{M}' \preceq \mathtt{N}'$.
2. If $\mathtt{M} \preceq \mathtt{N}$ and $\mathtt{M} \in \mathsf{Val}_\Omega$ then $\mathtt{N} \in \mathsf{Val}$.

Proof:
1. By structural induction on $\mathtt{M}$, the only interesting case being $\mathtt{M} \equiv (\lambda\mathtt{x}.\mathtt{L})\mathtt{V}$ and $\mathtt{M}' \equiv \mathtt{L}[\mathtt{V}/\mathtt{x}]$. By definition $\mathtt{N} \equiv (\lambda\mathtt{x}.\mathtt{L}')\mathtt{V}'$ for some λ-terms $\mathtt{L}', \mathtt{V}'$ such that $\mathtt{L} \preceq \mathtt{L}'$ and $\mathtt{V} \preceq \mathtt{V}'$. From the definition of $\preceq$, it follows easily that $\mathtt{L}[\mathtt{V}/\mathtt{x}] \preceq \mathtt{L}'[\mathtt{V}'/\mathtt{x}]$.
2. There are two cases. If $\mathtt{M} \equiv \mathtt{x}$ then $\mathtt{N} \equiv \mathtt{x}$, otherwise $\mathtt{M} \equiv \lambda\mathtt{x}.\mathtt{M}'$ for some $\mathtt{M}'$ in $\Lambda_{\mathtt{V}}\Omega$ and in this case $\mathtt{N} = \lambda\mathtt{x}.\mathtt{N}'$ for a λ-term $\mathtt{N}'$ such that $\mathtt{M}' \preceq \mathtt{N}'$. In both cases $\mathtt{N} \in \mathsf{Val}$. □

Mimicking Definition 2.2, we can define a term $\mathtt{M} \in \Lambda_{\mathtt{V}}\Omega$ to be valuable if it $\to_{\beta_v\Omega}$-reduces to a term in Val_Ω, to be potentially valuable if there is a substitution s, replacing variables by terms in Val_Ω, such that $s(\mathtt{M})$ is valuable.

Lemma 5.9. Every $\mathbb{A} \in \mathcal{A}$ is potentially valuable. Moreover, whenever $\mathcal{A}(\mathtt{M}) \neq \emptyset$, $\mathtt{M}$ is potentially valuable.

Proof:
We proceed by induction on the grammar in Definition 5.6 generating $\mathbb{A}$. We perform an induction loading and prove that for every $\mathbb{A}$ there exists K such that $s(\mathbb{A})$ is valuable for all substitutions associating with every variable a projection $\mathtt{P}_k \equiv \lambda\mathtt{x}_1 \dots \mathtt{x}_k.\mathtt{x}_k$ for $k > K$.

- $\mathbb{A} \equiv \lambda\mathtt{x}.\mathtt{M}$ for $\mathtt{M} \equiv \mathbb{A}'$ or $\mathtt{M} \equiv \Omega$. Trivial.
- $\mathbb{A} \equiv \mathtt{x}\mathbb{A}_1 \cdots \mathbb{A}_m$. By induction hypothesis, there exist $K_1, \dots, K_m$ such that $s(\mathbb{A}_i)$, for $1 \leq i \leq m$, is valuable for all s mapping each variable in $\mathtt{FV}(\mathbb{A})$ to some $\mathtt{P}_k$ where $k > \max\{K_1, \dots, K_m, m\}$. Therefore, we conclude $s(\mathbb{A}) \equiv \mathtt{P}_k s(\mathbb{A}_1) \cdots s(\mathbb{A}_m) \to^*_{\beta_v\Omega} \mathtt{P}_{k-m} \in \mathsf{Val}_\Omega$.
- $\mathbb{A} \equiv (\lambda\mathtt{x}.\mathbb{A}')(\mathtt{y}\mathbb{A}_1 \cdots \mathbb{A}_m)$. By induction hypothesis there exist $K_0, \dots, K_m$ respectively associated with $\mathbb{A}', \mathbb{A}_1, \dots, \mathbb{A}_m$ and satisfying the property above. Let s be a substitution sending every variable in $\mathtt{FV}(\mathbb{A})$ to some $\mathtt{P}_k$ for $k > \max\{K_0 + m, K_1, \dots, K_m\}$. Then we have

$$\begin{array}{rcl} s(\mathbb{R}) & \equiv & (\lambda\mathtt{x}.s(\mathbb{A}'))(\mathtt{P}_k s(\mathbb{A}_1) \cdots s(\mathbb{A}_m)) \\ & \to^*_{\beta_v\Omega} & (\lambda\mathtt{x}.s(\mathbb{A}'))\mathtt{P}_{k-m} \\ & \to_{\beta_v\Omega} & s(\mathbb{A}')[\mathtt{P}_{k-m}/\mathtt{x}] \\ & \to^*_{\beta_v\Omega} & \mathtt{V} \in \mathsf{Val}_\Omega \end{array}$$

  where the last step follows from the induction hypothesis.

This concludes the proof of the first statement.

Concerning the second statement, suppose that $\mathbb{A} \in \mathcal{A}(\mathtt{M})$ then there is a λ-term $\mathtt{N}$ such that $\mathtt{M} \to^*_{\mathtt{V}} \mathtt{N}$ and $\mathbb{A} \preceq \mathtt{N}$. By the proof of the first statement there exists a substitution $s : \mathtt{FV}(\mathbb{A}) \to \mathsf{Val}$, since

$\mathtt{P}_k \in \mathsf{Val}$ for all $k > 0$, such that $s(\mathbb{A}) \to^*_{\beta_v\Omega} \mathtt{V} \in \mathsf{Val}_\Omega$. Remark that $\mathbb{A} \preceq \mathtt{N}$ entails $s(\mathbb{A}) \preceq s(\mathtt{N})$, so by Lemma 5.8(1) there exists a λ-term $\mathtt{N}'$ such that $s(\mathtt{N}) \to_{\beta_v} \mathtt{N}'$. By Lemma 5.8(2), $\mathtt{V} \in \mathsf{Val}_\Omega$ and $\mathtt{V} \preceq \mathtt{N}'$ entail $\mathtt{N}' \in \mathsf{Val}$, whence $\mathtt{N}$ is potentially valuable. We conclude that $\mathtt{M}$ is also potentially valuable by Theorem 5.5. ◻

Theorem 5.10. (Approximation Theorem)

$\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$ if and only if $\Gamma \vdash_{\mathbb{C},\simeq} \mathbb{A} : \mathtt{A}$, for some $\mathbb{A} \in \mathcal{A}(\mathtt{M})$.

Proof:

($\Rightarrow$) Let Π be the derivation proving $\Gamma \vdash_{\mathbb{C},\simeq} \mathtt{M} : \mathtt{A}$. By Corollary 4.16, there is a term $\mathtt{N}$, such that $\mathtt{M} \to^*_{\mathtt{V}} \mathtt{N}$, where all redexes (if any) are untyped. So, replacing every subject of a $!_0$ rule by $\lambda\mathtt{x}.\Omega$, we get a term of $\Lambda_{\mathtt{V}}\Omega$ which is a $\to_{\mathtt{V}}$-normal form, so an approximant (Lemma 5.7) belonging to $\mathcal{A}(\mathtt{M})$ by definition.

($\Leftarrow$) If $\mathbb{A} \in \mathcal{A}(\mathtt{M})$, then $\mathbb{A} \preceq \mathtt{N}$, for some $\mathtt{N}$ such that $\mathtt{M} \to^*_{\mathtt{V}} \mathtt{N}$. So, by replacing, in the derivation $\Pi \triangleright \Gamma \vdash_{\mathbb{C},\simeq} \mathbb{A} : \mathtt{A}$, every subterm of the shape $\lambda\mathtt{x}.\Omega$ by a suitable subterm $\lambda\mathtt{x}.\mathtt{P}$ we obtain a derivation for $\mathtt{N}$. Then the result follows, by Lemma 4.14(2). ◻

Corollary 5.11. If $\mathtt{M}$ is not potentially valuable, then $[\![\mathtt{M}]\!]^{\mathbb{C},\simeq}_\rho = \emptyset$.

Proof:
By Lemma 5.9 we have that $\mathcal{A}(\mathtt{M}) = \emptyset$, so we conclude by applying the Approximation Theorem. ◻

As a consequence, we have the following theorem.

Theorem 5.12. Every model $\mathcal{M}^{\mathbb{C},\simeq}$ is adequate for Plotkin's operational preorder $\leq_{\mathtt{op}}$, i.e., $\mathcal{M}^{\mathbb{C},\simeq} \models \mathtt{M} \leq \mathtt{N}$ entails $\mathtt{M} \leq_{\mathtt{op}} \mathtt{N}$.

Proof:
By the way of contradiction, assume that there exists a context $\mathtt{C}$ such that $\mathtt{C}(\mathtt{M}), \mathtt{C}(\mathtt{N}) \in \Lambda^0_{\mathtt{V}}$, $\mathtt{C}(\mathtt{M}) \to^*_{\beta_v} \mathtt{M}' \in \mathsf{Val}$ but $\mathtt{C}(\mathtt{N})$ is not valuable. As the interpretation $[\![-]\!]^{\mathbb{C},\simeq}$ is contextual, we have that $\mathcal{M}^{\mathbb{C},\simeq} \models \mathtt{M} \leq \mathtt{N}$ entails $[\![\mathtt{C}(\mathtt{M})]\!]^{\mathbb{C},\simeq} \subseteq [\![\mathtt{C}(\mathtt{N})]\!]^{\mathbb{C},\simeq}$. By Property 4.3, we get $[\,] \in [\![\mathtt{M}']\!]^{\mathbb{C},\simeq} = [\![\mathtt{C}(\mathtt{M})]\!]^{\mathbb{C},\simeq}$ therefore $[\![\mathtt{C}(\mathtt{N})]\!]^{\mathbb{C},\simeq} \neq \emptyset$. Since $\mathtt{C}(\mathtt{N})$ is closed and not valuable it cannot be potentially valuable either, so we derive a contradiction by Corollary 5.11 ◻

We now show that the converse never holds, hence no model $\mathcal{M}^{\mathbb{C},\simeq}$ can be complete and therefore fully abstract. To prove such a result, we are going to exploit an extensional model $\mathcal{M}^c$ of $\lambda_{\mathtt{V}}$-calculus that has been introduced in [7] in the setting of continuous functions, and further studied in [28, §12.1].

Definition 5.13. A *recursion operator* is a term $\mathtt{Z}$ such that $\mathtt{Zx} =_{\beta_v} \mathtt{x}(\lambda\mathtt{z}.\mathtt{Zxz})$.

For example a recursion operator is $\lambda\mathtt{x}.(\lambda\mathtt{y}.\mathtt{x}(\lambda\mathtt{z}.\mathtt{yyz}))(\lambda\mathtt{y}.\mathtt{x}(\lambda\mathtt{z}.\mathtt{yyz}))$. It is immediate to check, using the approximation theorem, that every term $\mathtt{Z}$ with this behaviour has the same set of approximants, namely

$$\mathcal{A}(\mathtt{Z}) = \{\lambda\mathtt{x}.\mathtt{x}(\lambda\mathtt{z}_0.\mathtt{x}(\lambda\mathtt{z}_1.\mathtt{x}\cdots(\lambda\mathtt{z}_{n-1}.\mathtt{x}(\lambda\mathtt{z}_n.\Omega)\mathtt{z}_{n-1})\cdots\mathtt{z}_1)\mathtt{z}_0) \mid n \geq 1\},$$

so they are all equated in the theory. In [28, Thm. 12.1.22], Paolini and the third author proved that $\mathcal{M}^c \models \mathtt{I} = \mathtt{ZB}$, where $\mathtt{B} \equiv \lambda\mathtt{xyz.x(yz)}$ is the composition operator. Using the approximation theorem, we have that $\mathcal{A}(\mathtt{I}) = \{\lambda\mathtt{x}.\Omega, \lambda\mathtt{x.x}\}$, while it is easy to check that $\mathcal{A}(\mathtt{ZB}) = \{\lambda\mathtt{x}_1 \ldots \mathtt{x}_n.\Omega \mid n \geq 1\}$. So, for example, we have $\vdash_{\mathbb{C},\simeq} \mathtt{I} : [[\,] \to [\,]] \to [\,] \to [\,]$, while this type is not derivable for $\mathtt{ZB}$.

As a corollary, we have the following incompleteness result.

Theorem 5.14. No model $\mathcal{M}^{\mathbb{C},\simeq}$ is complete with respect to Plotkin's operational semantics.

Proof:
In [28], it is proved that $\mathcal{M}^c$ is adequate with respect to Plotkin's operational semantics and that $\mathcal{M}^c \models \mathtt{I} = \mathtt{ZB}$ holds. So the incompleteness follows. □

Conclusions and further works

We built a new class of models of the call-by-value λ-calculus, based on parametric intersection types system. Through a careful study of such a system we extended the reduction rule of the calculus, in such a way that the new obtained calculus is conservative with respect to the original Plotkin's calculus. A key property of all the models in this class is that they satisfy an approximation theorem: the interpretation of a term is the collection of the interpretations of its approximants, which are normal forms in an extended language. This theorem supplies a powerful tool to study the theory induced by a model. Unfortunately, the problem of building a satisfactory denotational model of λ_v-calculus is not completely solved yet, since we prove that all models in our class are adequate, but not complete for Plotkin operational semantics (and therefore not fully abstract). Hence, the problem of finding the paradigmatic call-by-value calculus remains open. In the present paper we study the properties that are common to all the models in our class: we would like to continue in exploring different call-by-value theories, by playing with the equivalence relation on types, and using the approximation theorem. A further problem is to characterize our class of model from the mathematical point of view. The type system we define is clearly inspired by the relational semantics, which is based on the category of sets and relations, as the model in [11], but the two constructions are clearly different, and we need to support an alternative description of our models in terms on domain equation, in order to be able to compare the two approaches.

References

[1] Church A. The Calculi of Lambda Conversion, volume 6 of *Annals of Mathematical Studies*. Princeton University Press, Princeton, 1941. Reprinted by University Microfilms Inc., Ann Arbor, MI in 1963 and by Klaus Reprint Corp., New York in 1965.

[2] Plotkin GD. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1975. **1**(2):125–159. URL `https://doi.org/10.1016/0304-3975(75)90017-1`.

[3] Carraro A, Guerrieri G. A Semantical and Operational Account of Call-by-Value Solvability. In: Muscholl A (ed.), Foundations of Software Science and Computation Structures, volume 8412 of *Lecture Notes in Computer Science*. Springer-Verlag, 2014 pp. 103–118. ISBN:9783642548291.

[4] Guerrieri G, Paolini L, Ronchi Della Rocca S. Standardization and Conservativity of a Refined Call-by-Value lambda-Calculus. *Logical Methods in Computer Science*, 2017. **13**(4):1–27. `www.lmcs-online.org`

[5] Wadsworth CP. The Relation Between Computational and Denotational Properties for Scott's $\mathcal{D}_\infty$-Models of the Lambda-Calculus. *SIAM Journal of Computing*, 1976. **5**(3):488–521. doi:10.1137/0205036.

[6] Hyland JME. A Syntactic Characterization of the Equality in Some Models of the Lambda Calculus. *Journal of the London Mathematical Society*, 1976. **2-12**(3):361–370. doi:10.1112/jlms/s2-12.3.361.

[7] Egidi L, Honsell F, Ronchi Della Rocca S. Operational, Denotational and Logical Descriptions: a case study. *Fundamenta Informaticae*, 1992. **16**(2):149–170. URL `http://www.iospress.nl,Fax:01131206203419`.

[8] Paolini L, Ronchi Della Rocca S. The Parametric Parameter Passing λ-calculus. *Information and Computation*, 2004. **189**(1):87–106. doi:10.1016/j.ic.2003.08.003.

[9] Paolini L, Piccolo M, Ronchi Della Rocca S. Logical Semantics for Stability. *Electr. Notes Theor. Comput. Sci.*, 2009. **249**:429–449. URL `https://doi.org/10.1016/j.entcs.2009.07.101`.

[10] Ehrhard T. Collapsing non-idempotent intersection types. In: CSL'12, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*. 2012 pp. 259–273. doi:10.4230/LIPIcs.CSL.2012.259.

[11] Ehrhard T, Guerrieri G. The Bang Calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, Edinburgh, United Kingdom, September 5-7, 2016 pp. 174–187. doi:10.1145/2967973.2968608.

[12] Curien PL, Herbelin H. The duality of computation. In: Odersky M, Wadler P (eds.), Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00). ACM. 2000 pp. 233–243. ISBN:1-58113-202-6. doi:10.1145/357766.351262.

[13] Dyckhoff R, Lengrand S. Call-by-Value lambda-calculus and LJQ. *Journal of Logic and Computation*, 2007. **17**(6):1109–1134. URL `https://hal.archives-ouvertes.fr/hal-00150284`.

[14] Herbelin H, Zimmermann S. An Operational Account of Call-by-Value Minimal and Classical lambda-Calculus in "Natural Deduction" Form. In: Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, volume 5608 of *Lecture Notes in Computer Science*. Springer-Verlag, 2009 pp. 142–156. doi:10.1007/978-3-642-02273-9_12.

[15] Accattoli B, Paolini L. Call-by-Value Solvability, Revisited. In: Functional and Logic Programming, volume 7294 of *Lecture Notes in Computer Science*. Springer-Verlag 2012 pp. 4–16. ISBN:978-3-642-29821-9.

[16] Accattoli B, Sacerdoti Coen C. On the Relative Usefulness of Fireballs. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015. IEEE Computer Society 2015 pp. 141–155. ISBN:978-1-4799-8875-4.

[17] Moggi E. Computational lambda-calculus and monads. Technical report, Edinburgh University, 1988. Tech. Report ECS-LFCS-88-66.

[18] Moggi E. Computational Lambda-Calculus and Monads. In: Proceedings of the 4th Symposium on Logic in Computer Science (LICS'89). IEEE Computer Society. 1989 pp. 14–23. ISBN:0-8186-1954-6.

[19] Sabry A, Felleisen M. Reasoning About Programs in Continuation-passing Style. *SIGPLAN Lisp Pointers*, 1992. **V**(1):288–298. doi:10.1145/141478.141563.

[20] Sabry A, Felleisen M. Reasoning about programs in continuation-passing style. *Lisp and Symbolic Computation*, 1993. **6**(3-4):289–360. doi:10.1007/BF01019462.

[21] Maraist J, Odersky M, Turner DN, Wadler P. Call-by-name, call-by-value, call-by-need and the linear lambda calculus. *Electronic Notes in Theoretical Computer Science*, 1995. **1**:370–392. URL `https://doi.org/10.1016/S1571-0661(04)00022-2`.

[22] Sabry A, Wadler P. A Reflection on Call-by-Value. *ACM Transactions on Programming Languages and Systems*, 1997. **19**(6):916–941.

[23] Maraist J. Call-by-name, call-by-value, call-by-need and the linear lambda calculus. *Theoretical Computer Science*, 1999. **228**(1–2):175–210. doi:10.1016/S0304-3975(98)00358-2.

[24] Guerrieri G. Head reduction and normalization in a call-by-value lambda-calculus. In: 2nd International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE 2015), volume 46 of *OpenAccess Series in Informatics (OASIcs)* 2015 pp. 3–17. ISBN:978-3-939897-94-1.

[25] Accattoli B, Guerrieri G. Open Call-by-Value. In: Programming Languages and Systems – 14th Asian Symposium (APLAS 2016), volume 10017 of *Lecture Notes in Computer Science*. Springer-Verlag, 2016 pp. 206–226.

[26] Ehrhard T, Guerrieri G. The Bang Calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming (PPDP 2016). ACM. ISBN 978-1-4503-4148-6, 2016 pp. 174–187. URL `https://arxiv.org/abs/1701.08186`.

[27] Barendregt H. The Lambda Calculus: Its Syntax and Semantics, volume 103 of *Studies in logic and the foundation of mathematics*. North-Holland, revised edition, 1984. ISBN:0444867481, 9780444867483.

[28] Ronchi Della Rocca S, Paolini L. The Parametric λ-Calculus: a Metamodel for Computation. Texts in Theoretical Computer Science: An EATCS Series. Springer, 2004. doi:10.1007/978-3-662-10394-4.

[29] Hindley JR, Longo G. Lambda calculus models and extensionality. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 1980. **26**:289–310. URL `https://doi.org/10.1002/malq.19800261902`.

[30] Regnier L. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 1994. **126**(2):281–292. URL `https://doi.org/10.1016/0304-3975(94)90012-4`.

[31] de Carvalho D. Execution time of λ-terms via denotational semantics and intersection types. *Mathematical Structures in Computer Science*, 2018. **28**(7):1169–1203. URL `https://doi.org/10.1017/S0960129516000396`.

[32] Paolini L, Piccolo M, Ronchi Della Rocca S. Essential and relational models. *Mathematical Structures in Computer Science*, 2017. **27**(5):626–650. URL `https://doi.org/10.1017/S0960129515000316`.

[33] Breuvart F, Manzonetto G, Ruoppolo D. Relational Graph Models at Work. *Logical Methods in Computer Science*, 2018. **14**(3):1–43 doi:10.23638/LMCS-14(3:2)2018.

[34] Guerrieri G. Personal communication, 2017.

[35] Kerinec E, Manzonetto G, Pagani M. Revisiting Call-by-value Böhm trees in light of their Taylor expansion. *CoRR*, 2018. **abs/1809.02659**.