The Bang Calculus and the Two Girard's Translations

Giulio Guerrieri
Dipartimento di Informatica – Scienza e Ingegneria (DISI), Università di Bologna
Bologna, Italy
giulio.guerrieri@unibo.it

Giulio Manzonetto
LIPN, UMR 7030, Université Paris 13, Sorbonne Paris Cité, F-93430, Villetaneuse, France
giulio.manzonetto@lipn.univ-paris13.fr

We study the two Girard's translations of intuitionistic implication into linear logic by exploiting the bang calculus, a paradigmatic functional language with an explicit box-operator that allows both the call-by-name and call-by value λ-calculi to be encoded in. We investigate how the bang calculus subsumes both call-by-name and call-by-value λ-calculi from a syntactic and a semantic viewpoint.

1 Introduction

The λ-calculus is a simple framework formalizing many features of functional programming languages. For instance, the λ-calculus can be endowed with two different evaluation mechanisms, call-by-name (CbN) and call-by-value (CbV), which have quite different properties. A CbN discipline re-evaluates an argument each time it is used. On the contrary, a CbV discipline evaluates an argument just once and recalls its value each time it is used. CbN and CbV λ-calculi are usually defined by means of operational rules giving rise to two different rewriting systems on the same set of λ-terms: in CbN there is no restriction on firing a β-redex, whereas in CbV a β-redex can be fired only when the argument is a value, *i.e.* a variable or an abstraction. The standard categorical setting for describing denotational models of the λ-calculus, cartesian closed categories, provides models which are adequate for CbN, but typically not for CbV. For CbV, the introduction of an additional computational monad (in the sense of Moggi [18, 19]) is necessary. While CbN λ-calculus [3] has a rich and refined semantic and syntactic theory featuring advanced concepts such as separability, solvability, Böhm trees, classification of λ-theories, full-abstraction, *etc.*, this is not the case for CbV λ-calculus [21], in the sense that concerning the CbV counterpart of these theoretical notions there are only partial and not satisfactory results (or they do not exist at all!).

Quoting from [13], "the existence of two separate paradigms is troubling" for at least two reasons:

- it makes each language appear arbitrary (whereas a unified language might be more canonical);
- each time we create a new style of semantics, *e.g.* Scott semantics, operational semantics, game semantics, continuation semantics, *etc.*, we always need to do it twice — once for each paradigm.

Girard's Linear Logic (LL, [11]) provides a unifying setting where this discrepancy could be solved since both CbN and CbV λ-calculi can be faithfully translated, via two different translations, into LL proof-nets. Following [13], we can claim that, via these translations, LL proof-nets "subsume" the CbN and CbV paradigms, in the sense that both operational and denotational semantics for those paradigms can be seen as arising, via these translations, from similar semantics for LL.

Indeed, LL can be understood as a refinement of intuitionistic logic (and hence λ-calculus) in which resource management is made explicit thanks to the introduction of a new pair of dual connectives:

Submitted to:
TLLA/Linearity 2018

the exponentials "!" and "?". In proof-nets, the standard syntax for LL proofs, *boxes* (introducing the modality "!") mark the sub-proofs available at will: during cut-elimination, such boxes can be erased (by weakening rules), can be duplicated (by contraction rules), can be opened (by dereliction rules) or can enter other boxes. The categorical counterpart of this refinement is well known: it is the notion of a cartesian $*$-autonomous[1] category, equipped with a comonad endowed with a strong monoidal structure. Every instance of such a kind of structure yields a denotational model of LL.

In his seminal article [11], Girard proposes a standard translation of intuitionistic logic (and hence simply typed λ-calculus) in multiplicative-exponential LL proof-nets whose semantic counterpart is well known: the Kleisli category of the exponential comonad "!" is cartesian closed thanks to the strong monoidal structure of "!". This translation $(\cdot)^{\mathsf{N}}$ maps the intuitionistic implication $A \Rightarrow B$ to the LL formula $!A^{\mathsf{N}} \multimap B^{\mathsf{N}}$. In [11] Girard proposes also another translation $(\cdot)^{\mathsf{V}}$ that he calls "boring": it maps the the intuitionistic implication $A \Rightarrow B$ to the LL formula $!(A^{\mathsf{V}} \multimap B^{\mathsf{V}})$ (or equivalently $!A^{\mathsf{V}} \multimap !B^{\mathsf{V}}$). Since the untyped λ-calculus can be seen as simply typed with only one ground type o satisfying the recursive identity $o = o \Rightarrow o$, the two Girard's translations $(\cdot)^{\mathsf{N}}$ and $(\cdot)^{\mathsf{V}}$ decompose this identity into $o = !o \multimap o$ and $o = !(o \multimap o)$ (or equivalently, $o = !o \multimap !o$), respectively. At the λ-term level, these two translations differ only by the way they use logical exponential rules (*i.e.* box and dereliction), whereas they use multiplicative and structural (*i.e.* contraction and weakening) ingredients in the same way. Because of this difference, the translation $(\cdot)^{\mathsf{N}}$ encodes the CbN λ-calculus into LL proof-nets (in the sense that CbN evaluation $\to_\beta$ is simulated by cut-elimination via $(\cdot)^{\mathsf{N}}$), while $(\cdot)^{\mathsf{V}}$ encodes the CbV λ-calculus into LL proof-nets (CbV evaluation $\to_{\beta^v}$ is simulated by cut-elimination via $(\cdot)^{\mathsf{V}}$). Indeed, since in CbN λ-calculus there is no restriction on firing a β-redex (its argument can be freely copied or erased), the translation $(\cdot)^{\mathsf{N}}$ puts the argument of every application into a box (see [6, 23, 12]); on the other hand, the translation $(\cdot)^{\mathsf{V}}$ puts only values into boxes (see [2]) since in CbV λ-calculus values are the only duplicable and discardable λ-terms. Thus, as deeply studied in [16], the two Girard's logical translations explain the two different evaluation mechanisms, bringing them into the scope of the Curry-Howard isomorphism.

The syntax of multiplicative-exponential LL proof-nets is extremely expressive and powerful, but it is too general and sophisticated for the computational purpose of representing purely functional programs. For instance, simulation of β-reduction on LL proof-nets passes through intermediate states/proof-nets that cannot be expressed as λ-terms, since LL proof-nets have many spurious cuts with axioms that have no counterpart on λ-terms. More in general, LL proof-nets are manipulated in their graphical form, and while this is a handy formalism for intuitions, it is far from practical for formal reasoning.

From the analysis of Girard's translations it seems worthwhile to extend the syntax of the λ-calculus to internalize the insights coming from LL *in a λ-like syntax*. The idea is to enrich the λ-calculus with explicit *boxes* marking the "values" of the calculus, *i.e.* the terms that can be freely duplicated and discarded: such a *linear* λ-calculus subsumes both CbN and CbV λ-calculi, via suitable translations. This, of course, has been done quite early in the history of LL by defining various linear λ-calculi, such as [15, 1, 4, 24, 16]. All these calculi require a clear distinction between linear and non-linear variables, structural rules being freely (and implicitly) available for the latter and forbidden for the former. This distinction complicates the formalism and is actually useless as far as we are interested in subsuming λ-calculi.

Inspired by Ehrhard [9], in [10] it has been introduced an intermediate formalism enjoying at the same time the conceptual simplicity of λ-calculus (without any distinction between linear and non-linear variables) and the operational expressiveness of LL proof-nets: the *bang calculus*. It is a variant of the λ-calculus which is "linear" in the sense that the exponential rules of LL (box and dereliction) are part

[1] Actually the full symmetry of such a category is not really essential as far as the λ-calculus is concerned, it is however quite natural from the LL viewpoint: LL restores the classical involutivity of negation in a constructive setting.

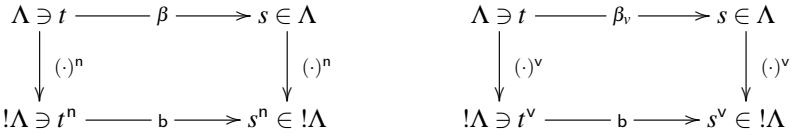
of the syntax, so as to subsume CbN and CbV λ-calculi via two translations $(\cdot)^{\mathsf{n}}$ and $(\cdot)^{\mathsf{v}}$, respectively, from the set Λ of λ-terms to the set $!\Lambda$ of terms of the bang calculus (see §3). These two translations are deeply related to Girard's encodings $(\cdot)^{\mathsf{N}}$ and $(\cdot)^{\mathsf{V}}$ of CbN and CbV λ-calculi into LL proof-nets. Indeed, Girard's translations $(\cdot)^{\mathsf{N}}$ and $(\cdot)^{\mathsf{V}}$ decompose in such a way that the following diagrams commute:

where $(\cdot)^{\circ}$ is a natural translation of the bang calculus into multiplicative-exponential LL proof-nets. Thus, the bang calculus *internalizes* the two Girard's translations in a λ-like calculus instead of LL proof-nets. It subsumes both CbN and CbV λ-calculi in the *same* rewriting system and denotational model, so that it may be a general setting to compare CbN and CbV. The bang calculus can be seen as a metalanguage where the choice of CbN or CbV evaluation depends on the way the term is built up. If we consider the syntax of the λ-calculus as a programming language, issues like CbN versus CbV evaluations affect the way the λ-calculus is translated in this metalanguage, but does not affect the metalanguage itself.

It turns out that this bang calculus was already known in the literature: it is an untyped version of the implicative fragment of Paul Levy's Call-By-Push-Value calculus [13, 14]. Interestingly, his work was not motivated by an investigation of the two Girard's translations. This link is not casual, since it holds even when the bang calculus is extended to a PCF-like system, as shown by Ehrhard [9].

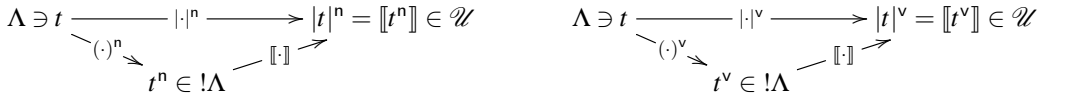
The aim of our paper is to further investigate the way the bang calculus subsumes CbN and CbV λ-calculi, refining and extending some results already obtained in [10].

1. From a syntactic viewpoint, we show in §3 that the bang calculus subsumes in the same rewriting system both CbN and CbV λ-calculi, in the sense that the translations $(\cdot)^{\mathsf{n}}$ and $(\cdot)^{\mathsf{v}}$ from the λ-calculus to the bang calculus are sound and complete with respect to β-reduction and β_v-reduction, respectively (in [10] only soundness was proven, and in a less elegant way). In other words, the diagrams



   commute in the two ways: starting from the β-reduction step $\to_{\beta}$ for the CbN λ-calculus (on the left) or the β_v-reduction step $\to_{\beta^v}$ for the CbV λ-calculus (on the right), and starting from the b-reduction step $\to_{\mathsf{b}}$ of the bang calculus.[2]

2. From a semantic viewpoint, we show in §4 that the *every* LL-based model $\mathscr{U}$ of the bang calculus (as categorically defined in [10]) provides a model for both CbN and CbV λ-calculi (in [10] this was done only for the special case of relational semantics). Moreover, given a λ-term t, we investigate the relation between its interpretations $|t|^{\mathsf{n}}$ in CbN (resp. $|t|^{\mathsf{v}}$ in CbV) and the interpretation $[\![\cdot]\!]$ of its translation t^{n} (resp. t^{v}) into the bang calculus. We prove that the diagram below on the left (for CbN) commutes, whereas we give a counterexample (in the relational semantics) to the commutation of the diagram below on the right (for CbV). We conjecture that there still exists a relationship in CbV between $|t|^{\mathsf{v}}$ and $[\![t^{\mathsf{v}}]\!]$, but it should be more sophisticated than in CbN.



[2] Actually, for the CbV λ-calculus the diagram is slightly more complex, as we will see in §3, but the essence does not change.

$$
\begin{array}{rll}
\textit{Terms}: & T,S,R ::= x \mid \lambda x T \mid \langle T\rangle S \mid \mathrm{der}\, T \mid T^! & (\text{set: } !\Lambda)\\
\textit{Contexts}: & \mathtt{C} ::= (\!|\cdot|\!) \mid \lambda x \mathtt{C} \mid \langle \mathtt{C}\rangle T \mid \langle T\rangle \mathtt{C} \mid \mathrm{der}\, \mathtt{C} \mid \mathtt{C}^! & (\text{set: } !\Lambda_{\mathtt{C}})\\
\textit{Ground contexts}: & \mathtt{G} ::= (\!|\cdot|\!) \mid \lambda x \mathtt{G} \mid \langle \mathtt{G}\rangle T \mid \langle T\rangle \mathtt{G} \mid \mathrm{der}\, \mathtt{G} & (\text{set: } !\Lambda_{\mathtt{G}})\\
\textit{Root-steps}: & \langle \lambda x T\rangle S^! \mapsto_\ell T\{S/x\} \qquad \mathrm{der}(T^!) \mapsto_{\mathtt{d}} T \qquad \mapsto_{\mathtt{b}} := \mapsto_\ell \cup \mapsto_{\mathtt{d}} &\\
\mathtt{r}\textit{-reduction}: & T \to_{\mathtt{r}} S \Leftrightarrow \exists \mathtt{C} \in !\Lambda_{\mathtt{C}}, \exists T',S' \in !\Lambda : T = \mathtt{C}(\!|T'|\!), S = \mathtt{C}(\!|S'|\!), T' \mapsto_{\mathtt{r}} S' &\\
\mathtt{r}_{\mathtt{g}}\textit{-reduction}: & T \to_{\mathtt{r}_{\mathtt{g}}} S \Leftrightarrow \exists \mathtt{G} \in !\Lambda_{\mathtt{G}}, \exists T',S' \in !\Lambda : T = \mathtt{G}(\!|T'|\!), S = \mathtt{G}(\!|S'|\!), T' \mapsto_{\mathtt{r}} S' &
\end{array}
$$

Figure 1: The bang calculus: its syntax and its reduction rules, where $\mathtt{r} \in \{\ell, \mathtt{d}, \mathtt{b}\}$.

In order to achieve these results in a clearer and simpler way, we have slightly modified (see §2) the syntax and operational semantics of the bang calculus with respect to its original formulation in [10].

Preliminaries and notations. Let $\to_{\mathtt{r}}$ and $\to_{\mathtt{r}'}$ be binary relations on a set X.

- The composition of $\to_{\mathtt{r}}$ and $\to_{\mathtt{r}'}$ is denoted by $\to_{\mathtt{r}}\to_{\mathtt{r}'}$ or $\to_{\mathtt{r}} \cdot \to_{\mathtt{r}'}$. The transpose of $\to_{\mathtt{r}}$ is denoted by ${}_{\mathtt{r}}\!\leftarrow$. The reflexive-transitive (resp. reflexive) closure of $\to_{\mathtt{r}}$ is denoted by $\to_{\mathtt{r}}^*$ (resp. $\to_{\mathtt{r}}^=$). The $\mathtt{r}$-*equivalence* $\simeq_{\mathtt{r}}$ is the reflexive-transitive and symmetric closure of $\to_{\mathtt{r}}$.
- Let $t \in X$: t is $\mathtt{r}$-*normal* if there is no term s such that $t \to_{\mathtt{r}} s$; t is $\mathtt{r}$-*normalizable* if there is a $\mathtt{r}$-normal term s such that $t \to_{\mathtt{r}}^* s$, and we then say that s is a $\mathtt{r}$-*normal form of* t.
- $\to_{\mathtt{r}}$ is *confluent* if ${}_{\mathtt{r}}^*\!\leftarrow \cdot \to_{\mathtt{r}}^* \subseteq \to_{\mathtt{r}}^* \cdot {}_{\mathtt{r}}^*\!\leftarrow$. From confluence it follows that: $t \simeq_{\mathtt{r}} s$ iff $t \to_{\mathtt{r}}^* r\ {}_{\mathtt{r}}^*\!\leftarrow s$ for some term r; and any $\mathtt{r}$-normalizable term has a *unique* $\mathtt{r}$-normal form.

2 Syntax and reduction rules of the bang calculus

The syntax and operational semantics of the *bang calculus* are defined in Fig. 1.

Terms are built up from a countably infinite set $\mathscr{V}ar$ of *variables* (denoted by $x,y,z,\ldots$). Terms of the form $T^!$ (resp. $\lambda x T$; $\langle T\rangle S$; $\mathrm{der}\, T$) are called or *boxes* (resp. *abstractions*; *(linear) applications*; *derelictions*). The set of boxes is denoted by $!\Lambda_!$. The set of free variables of a term T, denoted by $\mathsf{fv}(T)$, is defined as expected, λ being the only binding constructor. All terms are considered up to α-conversion. Given $T,S \in !\Lambda$ and a variable x, $T\{S/x\}$ denotes the term obtained by the *capture-avoiding substitution* of S (and not $S^!$) for each free occurrence of x in T: so, $T^!\{S/x\} = (T\{S/x\})^! \in !\Lambda_!$.

Contexts $\mathtt{C}$ and *ground contexts* $\mathtt{G}$ (both with exactly one hole $(\!|\cdot|\!)$) are defined in Fig. 1. All ground contexts are contexts but the converse fails: $(\!|\cdot|\!)^!$ is a non-ground context. We write $\mathtt{C}(\!|T|\!)$ for the term obtained by the capture-allowing substitution of the term T for the hole $(\!|\cdot|\!)$ in the context $\mathtt{C}$.

Reductions in the bang calculus are defined in Fig. 1 as follows: given a *root-step* rule $\mapsto_{\mathtt{r}} \subseteq !\Lambda \times !\Lambda$, we define the $\mathtt{r}$-*reduction* $\to_{\mathtt{r}}$ (resp. $\mathtt{r}_{\mathtt{g}}$-*reduction* $\to_{\mathtt{r}_{\mathtt{g}}}$) as the closure of $\mapsto_{\mathtt{r}}$ under contexts (resp. ground contexts). Note that $\to_{\mathtt{r}_{\mathtt{g}}} \subsetneq \to_{\mathtt{r}}$ as $!\Lambda_{\mathtt{G}} \subsetneq !\Lambda_{\mathtt{C}}$: the only difference between $\to_{\mathtt{r}}$ and $\to_{\mathtt{r}_{\mathtt{g}}}$ is that the latter does not reduce under $!$ (but both reduce under λ). The root-steps used in the bang calculus are $\mapsto_\ell$ and $\mapsto_{\mathtt{d}}$ and $\mapsto_{\mathtt{b}} := \mapsto_\ell \cup \mapsto_{\mathtt{d}}$. From the definitions in Fig. 1 it follows that $\to_{\mathtt{b}} = \to_\ell \cup \to_{\mathtt{d}}$ and $\to_{\mathtt{b}_{\mathtt{g}}} = \to_{\ell_{\mathtt{g}}} \cup \to_{\mathtt{d}_{\mathtt{g}}}$.

Intuitively, the basic idea behind the root-steps $\mapsto_\ell$ and $\mapsto_{\mathtt{d}}$ is that the box-constructor $!$ marks the only terms that can be erased and duplicated. When the argument of a constructor der is a box $T^!$, the root-step $\mapsto_{\mathtt{d}}$ opens the box, *i.e.* accesses its content T, destroying its status of availability at will (but T,

$$\frac{}{x \Rightarrow_\ell x}\ \mathsf{var} \qquad \frac{T \Rightarrow_\ell S}{\lambda x T \Rightarrow_\ell \lambda x S}\ \lambda \qquad \frac{T \Rightarrow_\ell S}{T^! \Rightarrow_\ell S^!}\ ! \qquad \frac{T \Rightarrow_\ell S}{\mathrm{der}\, T \Rightarrow_\ell \mathrm{der}\, S}\ \mathrm{der} \qquad \frac{T \Rightarrow_\ell S \quad R \Rightarrow_\ell Q}{\langle T\rangle R \Rightarrow_\ell \langle S\rangle Q}\ @ \qquad \frac{T \Rightarrow_\ell S \quad R \Rightarrow_\ell Q}{\langle \lambda x T\rangle R^! \Rightarrow_\ell S\{Q/x\}}\ \ell$$

Figure 2: Parallel ℓ-reduction.

in turn, might be a box). The root-step $\mapsto_\ell$ says that a β-like redex $\langle \lambda x T\rangle S$ can be fired only when its argument is a box, *i.e.* $S = R^!$: if it so, the content R of the box S replaces any free occurrence of x in T.

Example 1. Let $\Delta := \lambda x\langle x\rangle x^!$ and $\Delta' := \lambda x\langle \mathrm{der}(x^!)\rangle x^!$. Then, $\Delta' \to_{\mathsf{d_g}} \Delta$ and $\langle\Delta\rangle\Delta^! \to_{\ell_\mathsf{g}} \langle\Delta\rangle\Delta^! \to_{\ell_\mathsf{g}} \dots$ and $\langle \mathrm{der}(\Delta'^!)\rangle\Delta'^! \to_{\mathsf{d_g}} \langle\Delta'\rangle\Delta'^! \to_{\ell_\mathsf{g}} \langle \mathrm{der}(\Delta'^!)\rangle\Delta'^! \to_{\mathsf{d_g}} \dots$ Note that $(\langle\Delta\rangle\Delta^!)^!$ is $\mathsf{b_g}$-normal but not b-normalizable.

The *bang* (resp. *ground bang*) *calculus* is the set $!\Lambda$ endowed with the reduction $\to_\mathsf{b}$ (resp. $\to_{\mathsf{b_g}}$).

Quasi-strong confluence of $\mathsf{b_g}$-reduction and confluence of b-reduction. To prove the confluence of $\to_\mathsf{b}$ (Prop. 4.2), first we show that $\to_\ell$ is confluent (Lemma 3.4). The latter is proved by a standard adaptation of Tait–Martin-Löf technique — as improved by Takahashi [25] — based on parallel reduction. For this purpose, we introduce *parallel ℓ-reduction*, denoted by $\Rightarrow_\ell$, a binary relation on $!\Lambda$ defined by the rules in Fig. 2. Intuitively, $\Rightarrow_\ell$ reduces simultaneously a number of ℓ-redexes existing in a term. It is immediate to check that $\Rightarrow_\ell$ is reflexive and $\to_\ell \subseteq \Rightarrow_\ell \subseteq \to_\ell^*$, whence $\Rightarrow_\ell^* = \to_\ell^*$.

For any term T, we denote by T^* the term obtained by reducing *all* ℓ-redexes in T simultaneously. Formally, T^* is defined by induction on $T \in !\Lambda$ as follows:

$$x^* := x \qquad (\lambda x T)^* := \lambda x T^* \qquad (T^!)^* := (T^*)^! \qquad (\mathrm{der}\, T)^* := \mathrm{der}(T^*)$$
$$(\langle T\rangle S)^* := \langle T^*\rangle S^* \text{ if } T \neq \lambda x R \text{ or } S \notin !\Lambda_! \qquad (\langle \lambda x T\rangle S^!)^* := T^*\{S^*/x\}$$

Lemma 2 (Development)**.** *Let $T, S \in !\Lambda$. If $T \Rightarrow_\ell S$, then $S \Rightarrow_\ell T^*$.*

Lemma 2 is the key ingredient to prove the confluence of $\to_\ell$ (Lemma 3.4), indeed it entails that $\Rightarrow_\ell$, and hence $\to_\ell^*$, are strongly confluent.

The next lemma lists a series of good rewriting properties of ℓ-, ℓ_g-, d- and $\mathsf{d_g}$-reductions that will be used to prove quasi-strong confluence of $\to_{\mathsf{b_g}}$ and confluence of $\to_\mathsf{b}$ (Prop. 4).

Lemma 3 (Basic properties of reductions)**.**

1. *$\to_{\ell_\mathsf{g}}$ is quasi-strongly confluent: ${}_{\ell_\mathsf{g}}\!\leftarrow \cdot \to_{\ell_\mathsf{g}} \subseteq (\to_{\ell_\mathsf{g}} \cdot {}_{\ell_\mathsf{g}}\!\leftarrow) \cup =$.*
2. *$\to_{\mathsf{d_g}}$ and $\to_\mathsf{d}$ are quasi-strongly confluent (separately).*
3. *$\to_{\mathsf{d_g}}$ and $\to_{\ell_\mathsf{g}}$ strongly commute: ${}_{\mathsf{d_g}}\!\leftarrow \cdot \to_{\ell_\mathsf{g}} \subseteq \to_{\ell_\mathsf{g}} \cdot {}_{\mathsf{d_g}}\!\leftarrow$.*
   $\to_\mathsf{d}$ quasi-strongly commutes over $\to_\ell$: ${}_{\mathsf{d}}\!\leftarrow \cdot \to_\ell \subseteq \to_\ell \cdot {}^*_{\mathsf{d}}\!\leftarrow$.
   $\to_\mathsf{d}$ and $\to_\ell$ commute: ${}^*_{\mathsf{d}}\!\leftarrow \cdot \to_\ell^* \subseteq \to_\ell^* \cdot {}^*_{\mathsf{d}}\!\leftarrow$.
4. *$\to_\ell$ is confluent: ${}^*_\ell\!\leftarrow \cdot \to_\ell^* \subseteq \to_\ell^* \cdot {}^*_\ell\!\leftarrow$.*

Proposition 4 (Quasi-strong confluence of $\to_{\mathsf{b_g}}$ and confluence of $\to_\mathsf{b}$)**.**

1. *The reduction $\to_{\mathsf{b_g}}$ is quasi-strongly confluent, i.e. ${}_{\mathsf{b_g}}\!\leftarrow \cdot \to_{\mathsf{b_g}} \subseteq (\to_{\mathsf{b_g}} \cdot {}_{\mathsf{b_g}}\!\leftarrow) \cup =$.*
2. *The reduction $\to_\mathsf{b}$ is confluent: ${}^*_\mathsf{b}\!\leftarrow \cdot \to_\mathsf{b}^* \subseteq \to_\mathsf{b}^* \cdot {}^*_\mathsf{b}\!\leftarrow$.*

$$
\begin{array}{rcl}
\lambda\textit{-terms}: & t,s,r ::= v \mid ts & (\text{set: } \Lambda)\\
\lambda\textit{-values}: & v ::= x \mid \lambda x\, t & (\text{set: } \Lambda_v)\\
\lambda\textit{-contexts}: & \mathtt{C} ::= (\!|\cdot|\!) \mid \lambda x \mathtt{C} \mid \mathtt{C}t \mid t\mathtt{C} & (\text{set: } \Lambda_{\mathtt{C}})\\
\textit{CbN ground } \lambda\textit{-contexts}: & \mathtt{N} ::= (\!|\cdot|\!) \mid \lambda x \mathtt{N} \mid \mathtt{N}t & (\text{set: } \Lambda_{\mathtt{N}})\\
\textit{CbV ground } \lambda\textit{-contexts}: & \mathtt{V} ::= (\!|\cdot|\!) \mid \mathtt{V}t \mid t\mathtt{V} & (\text{set: } \Lambda_{\mathtt{V}})
\end{array}
$$

$$
\textit{Root-steps}: \quad (\lambda x t)S \mapsto_\beta t\{S/x\}\ (\text{CbN}) \qquad (\lambda x t)v \mapsto_{\beta^v} t\{v/x\}\ (\text{CbV})
$$

$$
\begin{array}{rc}
\mathsf{r}\textit{-reduction}: & t \to_{\mathsf{r}} s \Leftrightarrow \exists \mathtt{C} \in \Lambda_{\mathtt{C}}, \exists t', s' \in \Lambda : t = \mathtt{C}(\!|t'|\!), s = \mathtt{C}(\!|s'|\!), t' \mapsto_{\mathsf{r}} s'\\
\beta_{\mathsf{g}}\textit{-reduction}: & t \to_{\beta_{\mathsf{g}}} s \Leftrightarrow \exists \mathtt{N} \in \Lambda_{\mathtt{N}}, \exists t', s' \in \Lambda : t = \mathtt{N}(\!|t'|\!), s = \mathtt{N}(\!|s'|\!), t' \mapsto_{\beta} s'\\
\beta^v_{\mathsf{g}}\textit{-reduction}: & t \to_{\beta^v_{\mathsf{g}}} s \Leftrightarrow \exists \mathtt{V} \in \Lambda_{\mathtt{V}}, \exists t', s' \in \Lambda : t = \mathtt{V}(\!|t'|\!), s = \mathtt{V}(\!|s'|\!), t' \mapsto_{\beta^v} s'
\end{array}
$$

Figure 3: The CbN and CbV λ-calculi: their syntax and reduction rules, where $\mathsf{r} \in \{\beta, \beta^v\}$.

3 The bang calculus with respect to CbN and CbV λ-calculi, syntactically

One of the interests of the bang calculus is that it is a general framework where both *call-by-name* (CbN *i.e.* ordinary, [3]) and Plotkin's *call-by-value* (CbV, [21]) λ-calculi can be embedded.[3] Syntax and reduction rules of CbN and CbV λ-calculi are in Fig. 3: β reduction $\to_\beta$ (resp. β^v-reduction $\to_{\beta^v}$) is the reduction for the CbN (resp. CbV) λ-calculus. CbN and CbV λ-calculi share the same term syntax (the set Λ of λ-terms of CbV and CbN λ-calculi can be seen as a subset of $!\Lambda$), whereas $\to_{\beta^v}$ is just the restriction of $\to_\beta$ allowing to fire a β-redex $(\lambda x T)S$ only when S is a λ-value, *i.e.* variable or abstraction. Ground β-(resp. β^v-)reduction $\to_{\beta_{\mathsf{g}}}$ (resp. $\to_{\beta^v_{\mathsf{g}}}$) is an interesting restriction of β-(resp. β^v-)reduction:

- $\to_{\beta_{\mathsf{g}}}$ contains head β-reduction and weak head β-reduction, two well-known evaluation strategies for CbN λ-calculus (both reduce the β-redex in head position, the latter does not reduce under λ's);
- $\to_{\beta^v_{\mathsf{g}}}$ contains (weak) head β^v-reduction (aka left reduction in [21, p. 136]), the well-known evaluation strategy for CbV λ-calculus firing the leftmost-outermost β^v-redex (if any) not under λ's.

CbN and *CbV translations* are two functions $(\cdot)^{\mathsf{n}}\colon \Lambda \to !\Lambda$ and $(\cdot)^{\mathsf{v}}\colon \Lambda \to !\Lambda$, respectively, translating λ-terms into terms of the bang calculus as follows:

$$
\begin{array}{lll}
x^{\mathsf{n}} := x & (\lambda x\, t)^{\mathsf{n}} := \lambda x\, t^{\mathsf{n}} & (ts)^{\mathsf{n}} := \langle t^{\mathsf{n}} \rangle s^{\mathsf{n}!}\\
x^{\mathsf{v}} := x^{!} & (\lambda x\, t)^{\mathsf{v}} := (\lambda x\, t^{\mathsf{v}})^{!} & (ts)^{\mathsf{v}} := \langle \mathsf{der}\, t^{\mathsf{v}} \rangle s^{\mathsf{v}}\,.
\end{array}
$$

Example 5. Let $\Omega := (\lambda xxx)\lambda xxx$, the typical diverging λ-term for CbN and CbV λ-calculi: one has $\Omega^{\mathsf{n}} = \langle \Delta \rangle \Delta^{!}$ and $\Omega^{\mathsf{v}} = \langle \mathsf{der}(\Delta'^{!}) \rangle \Delta'^{!}$, which are not b_{g}- nor b-normalizable (Δ and Δ' are defined in Ex. 1).

For any λ-term t, t^{n} and t^{v} are just different decorations of t by means of the monadic operators ! and der (the latter does not occur in t^{n}). Note that the translation $(\cdot)^{\mathsf{n}}$ puts the argument of any application into a box: in CbN λ-calculus any λ-term is duplicable or discardable. On the other hand, only λ-values (*i.e.* abstractions and variables) are translated by $(\cdot)^{\mathsf{v}}$ into boxes, as they are the only λ-terms duplicable or discardable in CbV λ-calculus. Moreover, for any λ-term t, t^{v} is ℓ-normal, so if $t^{\mathsf{v}} \to_{\mathsf{d}} S_0 \to_\ell S$ then the only ℓ-redex in S_0 has been created by the step $t^{\mathsf{v}} \to_{\mathsf{d}} S_0$ and is absent in t^{v}.

Lemma 6 (Substitution)**.** *Let t, s be λ-terms and x be a variable.*

[3] Here, with CbN or CbV λ-calculus we refer to the whole calculus and its general reduction rules, not only to CbN or CbV (deterministic) evaluation strategy in the λ-calculus.

$$\begin{array}{lll} \textit{target of CbN translation into } !\Lambda: & T,S ::= x \mid \langle T\rangle S^{!} \mid \lambda x T & (\text{set: } !\Lambda^{\mathsf{n}}) \\ \textit{target of CbV translation into } !\Lambda: & M,N ::= U^{!} \mid \langle \mathrm{der} M\rangle N \mid \langle U\rangle M \quad (\text{set: } !\Lambda^{\mathsf{v}}) & U ::= x \mid \lambda x M \quad (\text{set: } !\Lambda_{v}^{\mathsf{v}}). \end{array}$$

Figure 4: Targets of CbN and CbV translations into the bang calculus.

1. *One has that* $t^{\mathsf{n}}\{s^{\mathsf{n}}/x\} = (t\{s/x\})^{\mathsf{n}}$.
2. *If* s *is such that* $s^{\mathsf{v}} = R^{!}$ *for some* $R \in !\Lambda$*, then* $t^{\mathsf{v}}\{R/x\} = (t\{s/x\})^{\mathsf{v}}$.

Note that the hypothesis about s in Lemma 6.2 is fulfilled if and only if s is a λ-value.

We can now show that the CbN (resp. CbV) translation from the CbN (resp. CbV) λ-calculus into the bang calculus is *sound* and *complete*: said differently, the target of the CbN (resp. CbV) translation into the bang calculus is a *conservative* extension of the CbN (resp. CbV) λ-calculus.

Proposition 7 (Simulation of CbN and CbV λ-calculi)**.** *Let* t *be a* λ*-term.*

1. Conservative extension of CbN λ-calculus:
   Soundness: *If* $t \to_{\beta} t'$ *then* $t^{\mathsf{n}} \to_{\ell} t'^{\mathsf{n}}$ *(and* $t^{\mathsf{n}} \to_{\mathsf{b}} t'^{\mathsf{n}}$*);*
   Completeness: *Conversely, if* $t^{\mathsf{n}} \to_{\mathsf{b}} S$ *then* $t^{\mathsf{n}} \to_{\ell} S = t'^{\mathsf{n}}$ *and* $t \to_{\beta} t'$ *for some* λ*-term* t'.
2. Conservative extension of ground CbN λ-calculus:
   Soundness: *If* $t \to_{\beta_{\mathsf{g}}} t'$ *then* $t^{\mathsf{n}} \to_{\ell_{\mathsf{g}}} t'^{\mathsf{n}}$ *(and* $t^{\mathsf{n}} \to_{\mathsf{b}_{\mathsf{g}}} t'^{\mathsf{n}}$*);*
   Completeness: *Conversely, if* $t^{\mathsf{n}} \to_{\mathsf{b}_{\mathsf{g}}} S$ *then* $t^{\mathsf{n}} \to_{\ell_{\mathsf{g}}} S = t'^{\mathsf{n}}$ *and* $t \to_{\beta_{\mathsf{g}}} t'$ *for some* λ*-term* t'.
3. Conservative extension of CbV λ-calculus:
   Soundness: *If* $t \to_{\beta^{v}} t'$ *then* $t^{\mathsf{v}} \to_{\mathsf{d}} \to_{\ell} t'^{\mathsf{v}}$ *(and hence* $t^{\mathsf{v}} \to_{\mathsf{b}} \to_{\mathsf{b}} t'^{\mathsf{v}}$*);*
   Completeness: *Conversely, if* $t^{\mathsf{v}} \to_{\mathsf{d}} \to_{\ell} S$ *then* $S = t'^{\mathsf{v}}$ *and* $t \to_{\beta^{v}} t'$ *for some* λ*-term* t'.
4. Conservative extension of ground CbV λ-calculus:
   Soundness: *If* $t \to_{\beta_{\mathsf{g}}^{v}} t'$ *then* $t^{\mathsf{v}} \to_{\mathsf{d}_{\mathsf{g}}} \to_{\ell_{\mathsf{g}}} t'^{\mathsf{v}}$ *(and hence* $t^{\mathsf{v}} \to_{\mathsf{b}_{\mathsf{g}}} \to_{\mathsf{b}_{\mathsf{g}}} t'^{\mathsf{v}}$*);*
   Completeness: *Conversely, if* $t^{\mathsf{v}} \to_{\mathsf{d}_{\mathsf{g}}} \to_{\ell_{\mathsf{g}}} S$ *then* $S = t'^{\mathsf{v}}$ *and* $t \to_{\beta_{\mathsf{g}}^{v}} t'$ *for some* λ*-term* t'.

So, the bang calculus can simulate β- and β^{v}-reductions via $(\cdot)^{\mathsf{n}}$ and $(\cdot)^{\mathsf{v}}$ and, conversely, ℓ-reductions in the targets of $(\cdot)^{\mathsf{n}}$ and $(\cdot)^{\mathsf{v}}$ correspond to β- and β^{v}-reductions. Also, these simulations are:

- *modular*, in the sense that *ground* β-reduction (including head β-reduction and weak head β-reduction) is simulated by *ground* ℓ-reduction, and vice-versa (Prop. 7.2); *ground* β^{v}-reduction (including head β^{v}-reduction) is simulated by *ground* d- and ℓ-reductions, and vice-versa (Prop. 7.4);
- *quantitative sensitive*, meaning that *one step* of (ground) β-reduction corresponds exactly, via $(\cdot)^{\mathsf{n}}$, to *one step* of (ground) ℓ-reduction, and vice-versa; *one step* of (ground) β^{v}-reduction corresponds exactly, via $(\cdot)^{\mathsf{v}}$, to *one step* of (ground) ℓ-reduction, and vice-versa.

According to the definition of CbN translation $(\cdot)^{\mathsf{n}}$, the target of $(\cdot)^{\mathsf{n}}$ into the bang calculus can be characterized syntactically: it is the subset $!\Lambda^{\mathsf{n}}$ of $!\Lambda$ defined in Fig. 4. This means that $t^{\mathsf{n}} \in !\Lambda^{\mathsf{n}}$ for any $t \in \Lambda$, and conversely, for any $T \in !\Lambda^{\mathsf{n}}$, $T^{\mathsf{n}} = t$ for some $t \in \Lambda$. Note that in $!\Lambda^{\mathsf{n}}$ the constructor der does not occur and hence reductions $\to_{\ell}$ and $\to_{\ell_{\mathsf{g}}}$ coincide with $\to_{\mathsf{b}}$ and $\to_{\mathsf{b}_{\mathsf{g}}}$, respectively, in $!\Lambda^{\mathsf{n}}$.

So, Prop. 7.1-2 says that $!\Lambda^{\mathsf{n}}$ endowed with the reduction $\to_{\ell}$ (resp. $\to_{\ell_{\mathsf{g}}}$) — which coincides with $\to_{\mathsf{b}}$ (resp. $\to_{\mathsf{b}_{\mathsf{g}}}$) in $!\Lambda^{\mathsf{n}}$ — is *isomorphic* to CbN (resp. ground CbN) λ-calculus. In particular:

Corollary 8 (Preservations with respect to CbN λ-calculus)**.** *Let* $t, s \in \Lambda$.

1. CbN equational theory: $t \simeq_{\beta} s$ *iff* $t^{\mathsf{n}} \simeq_{\ell} s^{\mathsf{n}}$ *iff* $t^{\mathsf{n}} \simeq_{\mathsf{b}} s^{\mathsf{n}}$.

2. CbN normal forms: *t is (ground) β-normal iff t^{n} is (ground) ℓ-normal iff t^{n} is (ground)* b*-normal.*

The correspondence between CbV λ-calculus and bang calculus is slightly more delicate: CbV translation $(\cdot)^{\mathsf{v}}$ gives a sound and complete embedding of $\to_{\beta^v}$ into $\to_{\mathsf{d}}\to_{\ell}$ (and similarly for their ground variants), but it is not complete with respect to generic $\to_{\mathsf{b}}$. Indeed, Ex. 1 and Ex. 5 have shown that $(\lambda xxx)^{\mathsf{v}} = \Delta^! \to_{\mathsf{d}} \Delta^!$, where $\Delta^!$ is b-normal and there is no λ-term t such that $t^{\mathsf{v}} = \Delta^!$. Note that λxxx is β^v-normal but $(\lambda xxx)^{\mathsf{v}} = \Delta^!$ is not b-normal: the analogous of Cor. 8.2 does not hold for $(\cdot)^{\mathsf{v}}$.

Actually, an analogous of Cor. 8.1 for CbV holds: CbV translation preserves β^v-equivalence in a sound and complete way with respect to b-equivalence (see Cor. 10 below). The proof requires a fine analysis of CbV translation $(\cdot)^{\mathsf{v}}$. First, in Fig. 4 we define two subsets $!\Lambda^{\mathsf{v}}$ and $!\Lambda^{\mathsf{v}}_v$ of $!\Lambda$, which include the target of CbV translation $(\cdot)^{\mathsf{v}}$: for any $t \in \Lambda$, $t^{\mathsf{v}} \in !\Lambda^{\mathsf{v}}$; in particular, for any $v \in \Lambda_v$, $v^{\mathsf{v}} = U^!$ for some $U \in !\Lambda^{\mathsf{v}}_v$. We have just shown that $(\cdot)^{\mathsf{v}}$ is not surjective in $!\Lambda^{\mathsf{v}}$. Anyway, it can be shown (Lemma 9.3) that $!\Lambda^{\mathsf{v}}$ is the set of terms in $!\Lambda$ reachable by b-reduction from CbV translations of λ-terms (*i.e.* for any $t \in \Lambda$, if $t^{\mathsf{v}} \to_{\mathsf{b}}^* S$ then $S \in !\Lambda^{\mathsf{v}}$) and b-equivalence in $!\Lambda^{\mathsf{v}}$ preserves β_v-equivalence (Cor. 10). To prove that, we define a forgetful translation $(\cdot)^\dagger \colon !\Lambda^{\mathsf{v}} \cup !\Lambda^{\mathsf{v}}_v \to \Lambda$ transforming terms $M \in !\Lambda^{\mathsf{v}}$ and $U \in !\Lambda^{\mathsf{v}}_v$ into λ-terms:

$$(U^!)^\dagger := U^\dagger \qquad (\langle \mathrm{der} M\rangle N)^\dagger := M^\dagger N^\dagger \qquad (\langle U\rangle M)^\dagger := U^\dagger M^\dagger$$
$$x^\dagger := x \qquad (\lambda x M)^\dagger := \lambda x M^\dagger.$$

Lemma 9 (Properties of the forgetful translation $(\cdot)^\dagger$)**.**

1. $(\cdot)^\dagger$ is a right-inverse of $(\cdot)^{\mathsf{v}}$*: For every $t \in \Lambda$, one has $t^{\mathsf{v}\dagger} = t$.*
2. $(\cdot)^\dagger$ preserves substitution*: $M\{U/x\} \in !\Lambda^{\mathsf{v}}$ with $(M\{U/x\})^\dagger = M^\dagger\{U^\dagger/x\}$, for any $M \in !\Lambda^{\mathsf{v}}$ and $U \in !\Lambda^{\mathsf{v}}_v$.*
3. $(\cdot)^\dagger$ maps b-reduction into β^v-reduction*: For any $M \in !\Lambda^{\mathsf{v}}$ and $T \in !\Lambda$, if $M \to_{\mathsf{b}} T$ then $T \in !\Lambda^{\mathsf{v}}$ and $M^\dagger \to_{\beta^v}^{=} T^\dagger$.*

Through this attentive analysis, we can conclude:

Corollary 10 (Preservation of CbV equational theory)**.** *Let $t, s \in \Lambda$. One has $t \simeq_{\beta^v} s$ iff $t^{\mathsf{v}} \simeq_{\mathsf{b}} s^{\mathsf{v}}$.*

So, Cor. 10 says that CbV translation $(\cdot)^{\mathsf{v}}$ — even if it is a sound but not complete embedding of β^v-reduction into b-reduction — is a sound and complete embedding of β^v-equivalence into b-equivalence.

4 The bang calculus with respect to CbN and CbV λ-calculi, semantically

The denotational models we are interested in this paper are those induced by a model of LL. We recall the basic definitions and notations, see [17, 9, 10] for more details.

Linear logic based denotational semantics of bang calculus. We consider a $*$-autonomous category, namely an SMCC $(\mathscr{L}, \otimes, 1, \lambda, \rho, \alpha, \sigma)$ with a dualizing object $\bot$. We use $X \multimap Y$ for the linear exponential object, $\mathsf{ev} \in \mathscr{L}((X \multimap Y) \otimes X, Y)$ for the evaluation morphism and cur for the linear curryfication map $\mathscr{L}(Z \otimes X, Y) \to \mathscr{L}(Z, X \multimap Y)$. We use $X^\bot$ for the object $X \multimap \bot$ of $\mathscr{L}$ (the *linear negation* of X). This operation $(\_)^\bot$ is a functor $\mathscr{L}^{\mathsf{op}} \to \mathscr{L}$. The category $\mathscr{L}$ is cartesian with terminal object $\top$, product &, projections pr_i. By $*$-autonomy $\mathscr{L}$ is cocartesian with initial object 0, coproduct $\oplus$ and injections in_i.

We are given a comonad $!\_ \colon \mathscr{L} \to \mathscr{L}$ with counit $\mathsf{der}_X \in \mathscr{L}(!X, X)$ (*dereliction*) and comultiplication $\mathsf{dig}_X \in \mathscr{L}(!X, !!X)$ (*digging*), and a strong symmetric monoidal structure (Seely isos $\mathsf{m}^0 \in \mathscr{L}(1, !\top)$ and

$\mathsf{m}^2_{X,Y} \in \mathscr{L}(!X \otimes !Y, !(X \,\&\, Y)))$ for the functor $!\_$, from the symmetric monoidal category $(\mathscr{L}, \&)$ to the symmetric monoidal category $(\mathscr{L}, \otimes)$ satisfying an additional coherence condition with respect to dig.

We present now the additional structures and assumptions needed to interpret the bang calculus.

First, we assume that the unique morphism in $\mathscr{L}(0, \top)$ is an iso (to simplify, assume just $0 = \top$). It follows that for any two objects X and Y we have a morphism $\mathsf{it} \in \mathscr{L}(X,Y)$ where t is the unique morphism $X \to \top$ and i is the unique morphism $0 \to Y$. We use $0_{X,Y}$ for this specific zero morphism, satisfying for all $f \in \mathscr{L}(Y,Z)$ and $g \in \mathscr{L}(X,Y)$ the equation $f\, 0_{X,Y} = 0_{X,Z} = 0_{Y,Z} g$.

These assumptions are satisfied by many models, like the relational model, Scott model, (hyper-) coherence space model, all models based on Indexed LL [5], probabilistic coherence space model [7].

A model of the bang calculus is an object $\mathscr{U}$ of $\mathscr{L}$ satisfying the equation $\mathscr{U} \cong\, !\mathscr{U} \,\&\, (!\mathscr{U} \multimap \mathscr{U})$ (we assume this iso to be an equality). Note that this entails both $!\mathscr{U} \lhd \mathscr{U}$ and $!\mathscr{U} \multimap \mathscr{U} \lhd \mathscr{U}$.

Given a term T and a repetition-free list of variables $\vec{x} = (x_1, \ldots, x_n)$ which contains all the free variables of T, we can define a morphism $[\![T]\!]_{\vec{x}} \in \mathscr{L}((!\mathscr{U})^{\otimes n}, \mathscr{U})$, the denotational semantics of T. The definition is by induction on $T \in !\Lambda$, with $\vec{x} = (x_1, \ldots, x_n)$ such that $\mathsf{fv}(T) \subseteq \{x_1, \ldots, x_n\}$:

- $[\![x_i]\!]_{\vec{x}} = \mathsf{w}_{\mathscr{U}}^{\otimes i-1} \otimes\, \mathsf{der}_{\mathscr{U}} \otimes \mathsf{w}_{\mathscr{U}}^{\otimes n-i}$ where $\mathsf{w}_{\mathscr{U}} \in \mathscr{L}(!\mathscr{U}, 1)$ is the weakening and we keep implicit the monoidality isos $1 \otimes \mathscr{U} \simeq \mathscr{U}$,
- $[\![\lambda y\, S]\!]_{\vec{x}} = \langle 0_{(!\mathscr{U})^{\otimes n}, !\mathscr{U}}, \mathsf{cur}\,([\![S]\!]_{\vec{x},y})\rangle$, where we assume wlog $y \notin \vec{x}$,
- $[\![\langle S \rangle R]\!]_{\vec{x}} = \mathsf{ev}\,(\mathsf{pr}_2\,[\![S]\!]_{\vec{x}} \otimes \mathsf{pr}_1\,[\![R]\!]_{\vec{x}})\,\mathsf{c}$, where $\mathsf{c} \in \mathscr{L}((!\mathscr{U})^{\otimes n}, (!\mathscr{U})^{\otimes n} \otimes (!\mathscr{U})^{\otimes n})$ is the contraction,
- $[\![S^!]\!]_{\vec{x}} = \langle ([\![S]\!]_{\vec{x}})^!, 0_{(!\mathscr{U})^{\otimes n}, !\mathscr{U} \multimap \mathscr{U}}\rangle$, for $([\![S]\!]_{\vec{x}})^! = !([\![S]\!]_{\vec{x}})\,\mathsf{h}$ where $\mathsf{h} \in \mathscr{L}((!\mathscr{U})^{\otimes n}, !((!\mathscr{U})^{\otimes n}))$ is the coalgebra structure map of $(!\mathscr{U})^{\otimes n}$,
- $[\![\mathrm{der}\, S]\!]_{\vec{x}} = \mathsf{der}_{\mathscr{U}}\ \mathsf{pr}_1\,[\![S]\!]_{\vec{x}}$.

Theorem 11 (Invariance, [10])**.** *Let $T, S \in !\Lambda$ and $\vec{x}$ be a repetition-free list of variables which contains all free variables of T and S. If $T \simeq_{\mathsf{b}} S$ then $[\![T]\!]_{\vec{x}} = [\![S]\!]_{\vec{x}}$.*

The general notion of denotational model for the bang calculus presented here and obtained from any denotational model $\mathscr{L}$ of LL (with the assumptions detailed above) is a particular case of Moggi's semantics of computations based on monads [18, 19], if one keeps in mind that the functor "!" defines a strong monad on the Kleisli category $\mathscr{L}_!$ of $\mathscr{L}$.

Call-by-name. A model of the CbN λ-calculus is a reflexive object in a cartesian closed category. The category $\mathscr{L}$ being $*$-autonomous, its Kleisli $\mathscr{L}_!$ over the comonad $(!, \mathsf{dig}, \mathsf{der})$ is cartesian closed. The category $\mathscr{L}_!$ (whose objects are the same as $\mathscr{L}$ and morphisms are given by $\mathscr{L}_!(A,B) = \mathscr{L}(!A, B)$) has composition $f \circ g$ defined as $f\, !g\, \mathsf{dig}$ and identities A given by der_A. In $\mathscr{L}_!$, products $A \,\&\, B$ are preserved, with projections $\pi_i := \mathsf{pr}_i\, \mathsf{der}_{!(A\&B)}$ ($i \in \{1,2\}$); the exponential object $A \Rightarrow B$ is $!A \multimap B$ (this is the semantic counterpart of Girard's CbN translation) and has an evaluation morphism $\mathsf{Ev} = \mathsf{ev}(\mathsf{der}_{!A \multimap B} \otimes \mathsf{id}_{!A})(\mathsf{m}^2)^{-1} \in \mathscr{L}(!((!A \multimap B) \,\&\, !A), B)$. This defines an exponentiation since for all $f \in \mathscr{L}(!(C \,\&\, A), B)$ there is a unique morphism $\Lambda(f) = \mathsf{cur}(f\, \mathsf{m}^2) \in \mathscr{L}(!C, !A \multimap B)$ satisfying $\mathsf{Ev} \circ \langle \Lambda(f), A\rangle = f$.

The identity $\mathscr{U} = !\mathscr{U} \,\&\, (!\mathscr{U} \multimap \mathscr{U})$ entails $!\mathscr{U} \multimap \mathscr{U} \lhd \mathscr{U}$ in $\mathscr{L}$ via $\mathsf{lam} := \langle 0_{!\mathscr{U} \multimap \mathscr{U}, !\mathscr{U}}, \mathsf{id}_{!\mathscr{U} \multimap \mathscr{U}}\rangle \in \mathscr{L}(!\mathscr{U} \multimap \mathscr{U}, \mathscr{U})$ and $\mathsf{app} := \mathsf{pr}_2 \in \mathscr{L}(\mathscr{U}, !\mathscr{U} \multimap \mathscr{U})$, since $\mathsf{app}\ \mathsf{lam} = \mathsf{id}_{!\mathscr{U} \multimap \mathscr{U}}$. So, $\mathscr{U}$ is a reflexive object (*i.e.* $!\mathscr{U} \multimap \mathscr{U} \lhd \mathscr{U}$) in $\mathscr{L}_!$ via $\mathsf{app}_{\mathsf{n}} := \mathsf{der}_{!\mathscr{U} \multimap \mathscr{U}}\, !\mathsf{app} \in \mathscr{L}_!(\mathscr{U}, !\mathscr{U} \multimap \mathscr{U})$ and $\mathsf{lam}_{\mathsf{n}} := \mathsf{der}_{\mathscr{U}}\, !\mathsf{lam} \in \mathscr{L}_!(!\mathscr{U} \multimap \mathscr{U}, \mathscr{U})$, since $\mathsf{app}_{\mathsf{n}} \circ \mathsf{lam}_{\mathsf{n}} = !\mathscr{U} \multimap \mathscr{U}$. Therefore, the interpretation of a λ-term t can be defined as usual as a morphism $|t|^{\mathsf{n}}_{\vec{x}} \in \mathscr{L}_!(\mathscr{U}^k, \mathscr{U})$, with $\vec{x} = (x_1, \ldots, x_k)$ such that $\mathsf{fv}(t) \subseteq \{x_1, \ldots, x_k\}$:

$$|x_i|^{\mathsf{n}}_{\vec{x}} = \pi^k_i, \qquad |\lambda y\, t|^{\mathsf{n}}_{\vec{x}} = \mathsf{lam}_{\mathsf{n}} \circ \Lambda(|t|^{\mathsf{n}}_{\vec{x},y}), \qquad |ts|^{\mathsf{n}}_{\vec{x}} = \mathsf{Ev} \circ \langle \mathsf{app}_{\mathsf{n}} \circ |t|^{\mathsf{n}}_{\vec{x}}, |s|^{\mathsf{n}}_{\vec{x}}\rangle.$$

Summing up, the object $\mathscr{U}$ provides both a model of the bang calculus and a model of the CbN λ-calculus. The relation between the two is elegant: the semantics $|t|^{\mathsf{n}}$ in the CbN model of the λ-calculus of a λ-term t decomposes into the semantics $[\![\cdot]\!]$ in the model of the bang calculus of the CbN translation t^{n} of t.

Theorem 12 (Factorization of CbN semantics)**.** *For every λ-term t, $[\![t^{\mathsf{n}}]\!]_{\vec{x}} = |t|^{\mathsf{n}}_{\vec{x}}$ (up to Seely's isos).*

Call-by-value. Models of the CbV λ-calculus can be defined using the so-called "boring" translation by Girard of the linear implication into Linear Logic — just recall that the functor "!" defines a strong monad on the Kleisli category $\mathscr{L}_!$. It is therefore enough to find an object X satisfying $!X \multimap !X \lhd X$ (equivalently, $!(X \multimap X) \lhd X$). This is the case for our object $\mathscr{U}$ since $!\mathscr{U} \multimap \mathscr{U} \lhd \mathscr{U}$ and $!\mathscr{U} \lhd \mathscr{U}$ entail $!\mathscr{U} \multimap !\mathscr{U} \lhd \mathscr{U}$ (by the variance of $!\mathscr{U} \multimap \_$) via the morphisms $\mathsf{lam}_{\mathsf{v}} = \langle 0_{!\mathscr{U}\multimap!\mathscr{U},!\mathscr{U}}, \mathsf{cur}(\langle \mathsf{ev}, 0_{(!\mathscr{U}\multimap!\mathscr{U})\otimes!\mathscr{U},\mathscr{U}}\rangle)\rangle$ and $\mathsf{app}_{\mathsf{v}} = \mathsf{cur}(\mathsf{pr}_1\ \mathsf{ev})\mathsf{pr}_2$. Following e.g. [8], it is possible to define the interpretation of a λ-term t as a morphism $|t|^{\mathsf{v}}_{\vec{x}} \in \mathscr{L}((!\mathscr{U})^{\otimes k}, !\mathscr{U})$, with $\vec{x} = (x_1, \ldots, x_k)$ such that $\mathsf{fv}(t) \subseteq \{x_1, \ldots, x_k\}$:

$$|x_i|^{\mathsf{v}}_{\vec{x}} = \mathsf{w}_{\mathscr{U}}^{\otimes i-1} \otimes \mathsf{id}_{!\mathscr{U}} \otimes \mathsf{w}_{\mathscr{U}}^{\otimes k-i}, \qquad |\lambda y\, t|^{\mathsf{v}}_{\vec{x}} = (\mathsf{lam}_{\mathsf{v}}\ \mathsf{cur}(|t|^{\mathsf{v}}_{\vec{x},y}))^!, \qquad |ts|^{\mathsf{v}}_{\vec{x}} = \mathsf{ev}((\mathsf{app}_{\mathsf{v}}\, |t|^{\mathsf{v}}_{\vec{x}}) \otimes (|s|^{\mathsf{v}}_{\vec{x}}))\,\mathsf{c}\,.$$

We now have two possible ways of interpreting the CbV λ-calculus in our model $\mathscr{U}$: either by translating a λ-term t into $t^{\mathsf{v}} \in !\Lambda$ and then compute $[\![t^{\mathsf{v}}]\!]$, or by computing directly $|t|^{\mathsf{v}}$. It is natural to wonder whether the two interpretations $[\![t^{\mathsf{v}}]\!]$ and $|t|^{\mathsf{v}}$ are related, and in what way. In [10] the authors conjectured that, at least in the case of a particular relational model $\mathscr{U}$ satisfying $\mathscr{U} = !\mathscr{U} \cup (!\mathscr{U} \times \mathscr{U}) = !\mathscr{U}\ \&\ (!\mathscr{U} \multimap \mathscr{U})$, the two interpretations coincide. We show that the situation is actually more complicated than expected.

The *relational model* $\mathscr{U}$ introduced in [10] admits the following concrete description as a type system. The set $\mathscr{U}$ of *types* and the set $!\mathscr{U}$ of *finite multisets* over $\mathscr{U}$ are generated by the following grammar:

$$\text{(set: } \mathscr{U}) \quad \alpha, \beta, \gamma ::= a \mid a \multimap \alpha \qquad\qquad \text{(set: } !\mathscr{U}) \quad a, b, c ::= [\alpha_1, \ldots, \alpha_n] \quad \text{for any } n \geq 0.$$

Environments Γ are functions from variables to $!\mathscr{U}$ whose *support* $\mathsf{supp}(\Gamma) = \{x \mid \Gamma(x) \neq []\}$ is finite. We write $x_1 : a_1, \ldots, x_n : a_n$ for the environment Γ satisfying $\Gamma(x_i) = a_i$ and $\Gamma(y) = []$ for $y \notin \vec{x}$. The multiset union $a + b$ is extended to environments pointwise, namely $(\Gamma + \Delta)(x) = \Gamma(x) + \Delta(x)$.

On the one hand, the relational model $\mathscr{U}$ for the CbV λ-calculus inteprets a λ-term t using $|\cdot|^{\mathsf{v}}$, which gives $|t|^{\mathsf{v}}_{\vec{x}} = \{(\Gamma, \beta) \mid \Gamma \vdash_v t : \beta\}$ where $\vdash_v$ is the type system below (note that if $\Gamma \vdash_v t : \beta$ then $\beta \in !\mathscr{U}$):

$$\frac{}{x : a \vdash_v x : a}\ \text{ax} \qquad \frac{\Gamma \vdash_v t : [a \multimap b] \qquad \Delta \vdash_v s : a}{\Gamma + \Delta \vdash_v ts : b}\ \text{app} \qquad \frac{\Gamma_i, y : a_i \vdash_v t : b_i \quad n \geq 0}{\sum_{i=1}^n \Gamma_i \vdash_v \lambda y\, t : [a_1 \multimap b_1, \ldots, a_n \multimap b_n]}\ \text{lam}$$

On the other hand, the relational model $\mathscr{U}$ for the bang calculus interprets a term $t \in !\Lambda$ using $[\![\cdot]\!]$, which gives $[\![T]\!]_{\vec{x}} = \{(\Gamma, \beta) \mid \Gamma \vdash_! T : \beta\}$ where $\vdash_!$ is the type system defined as follows:

$$\frac{}{x : [\alpha] \vdash_! x : \alpha}\ ax_! \qquad \frac{\Gamma \vdash_! T : a \multimap \beta \qquad \Delta \vdash_! S : a}{\Gamma + \Delta \vdash_! \langle T \rangle S : \beta}\ @ \qquad \frac{\Gamma_i \vdash_! T : \beta_i \quad n \geq 0}{\sum_{i=1}^n \Gamma_i \vdash_! T^! : [\beta_1, \ldots, \beta_n]}\ !$$

$$\frac{\Gamma \vdash_! T : [\alpha]}{\Gamma \vdash_! \mathrm{der}\, T : \alpha}\ \text{der} \qquad \frac{\Gamma, x : a \vdash_! T : \beta}{\Gamma \vdash_! \lambda x T : a \multimap \beta}\ \lambda$$

What is the interpretation $[\![t^{\mathsf{v}}]\!]_{\vec{x}}$ of the CbV translations of λ-terms? Easy calculations show that in the type system $\vdash_!$ the rules below — the ones needed to interpret terms of the form t^{v} — can be derived:

$$\frac{}{x : a \vdash_! x^! : a}\ \text{ax} \qquad \frac{\Gamma \vdash_! t^{\mathsf{v}} : [a \multimap \beta] \qquad \Delta \vdash_! s^{\mathsf{v}} : a}{\Gamma + \Delta \vdash_! \langle \mathrm{der} t^{\mathsf{v}} \rangle s^{\mathsf{v}} : \beta}\ \text{app} \qquad \frac{\Gamma_i, y : a_i \vdash_! t^{\mathsf{v}} : \beta_i \quad n \geq 0}{\sum_i \Gamma_i \vdash_! (\lambda y\, t^{\mathsf{v}})^! : [a_1 \multimap \beta_1, \ldots, a_n \multimap \beta_n]}\ \text{lam}$$

Intuitively, the type system $\vdash_v$ is obtained from the restriction of $\vdash_!$ to the image of $(\cdot)^\vee$ by substituting arbitrary types β with multisets b of types. So, given a λ-term t, the two interpretations $|t|^\vee_{\vec{x}}$ and $[\![t^\vee]\!]_{\vec{x}}$ can be different: for $\alpha \in \mathscr{U} \setminus !\mathscr{U}$ (*e.g.* take $\alpha = [\,] \multimap [\,]$), one has $([(a \multimap \alpha] + a) \multimap \alpha] \in [\![(\lambda xxx)^\vee]\!] \setminus |\lambda xxx|^\vee$.

Theorem 13 (Relational semantics for CbV)**.** *In the relational model $\mathscr{U}$, we have $|t|^\vee \subsetneq [\![t^\vee]\!]$.*

The counterexample shows that in general neither $[\![t^\vee]\!] = \langle |t|^\vee, 0\rangle$ nor $\mathsf{pr}_1[\![t^\vee]\!] = |t|^\vee$ hold. We believe that $|t|^\vee$ can be obtained from $[\![t^\vee]\!]$ by iterating the application of pr_1 to $[\![\cdot]\!]$ along the structure of t, but how to express this formally? Usually in these situations one defines a logical relation between the two interpretations, but this is complicated by the fact that we are in the untyped setting so there is no type hierarchy to base our induction. We plan to investigate whether the (syntactic) logical relations introduced by Pitts in [20] can give an inspiration to define semantic logical relations in the untyped setting. Another source of inspiration might be the study of other concrete LL based models of the CbV λ-calculus, such as Scott domains and coherent semantics [22].

References

[1] S. Abramsky (1993): *Computational Interpretations of Linear Logic*. *TCS* 111(1&2), pp. 3–57.

[2] B. Accattoli (2015): *Proof nets and the call-by-value λ-calculus*. *TCS* 606, pp. 2–24.

[3] H. P. Barendregt (1984): *The Lambda Calculus: Its Syntax and Semantics*. *Studies in Logic and the Foundation of Mathematics* 103, North-Holland, Amsterdam.

[4] P. N. Benton & P. Wadler (1996): *Linear Logic, Monads and the Lambda Calculus*. In: *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science*, IEEE Computer Society, pp. 420–431.

[5] A. Bucciarelli & T. Ehrhard (2001): *On phase semantics and denotational semantics: the exponentials*. *Annals of Pure and Applied Logic* 109(3), pp. 205–241.

[6] V. Danos: *La Logique Linéaire appliqué à l'étude de divers processus de normalisation (principalement du λ-calcul)*. Ph.D. thesis, Université Paris 7.

[7] V. Danos & T. Ehrhard (2011): *Probabilistic coherence spaces as a model of higher-order probabilistic computation*. *Information and Computation* 152(1), pp. 111–137.

[8] T. Ehrhard (2012): *Collapsing non-idempotent intersection types*. In: *CSL*, pp. 259–273.

[9] T. Ehrhard (2016): *Call-By-Push-Value from a Linear Logic point of view*. In: *ESOP 2016*, *Lecture Notes in Computer Science* 9632, Springer, pp. 202–228.

[10] T. Ehrhard & G. Guerrieri (2016): *The Bang Calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value*. In J. Cheney & G. Vidal, editors: *PPDP'16*, ACM, pp. 174–187.

[11] J.-Y. Girard (1987): *Linear Logic*. *Theoretical Computer Science* 50(1), pp. 1–102.

[12] O. Laurent (2003): *Polarized proof-nets and $\lambda\mu$-calculus*. *Theoretical Computer Science* 290(1), pp. 161–188.

[13] P. B. Levy (1999): *Call-by-Push-Value: A Subsuming Paradigm*. In J.-Y. Girard, editor: *TLCA'99*, *Lecture Notes in Computer Science* 1581, Springer, pp. 228–242.

[14] P. B. Levy (2006): *Call-by-push-value: Decomposing call-by-value and call-by-name*. *Higher-Order and Symbolic Computation* 19(4), pp. 377–414.

[15] P. Lincoln & J. C. Mitchell (1992): *Operational aspects of linear lambda calculus*. In: *Proceedings of the Seventh Annual Symposium on Logic in Computer Science (LICS '92)*, IEEE Computer Society, pp. 235–246.

[16] J. Maraist, M. Odersky, D. N. Turner & P. Wadler (1999): *Call-by-name, Call-by-value, Call-by-need and the Linear lambda Calculus*. *Theoretical Computer Science* 228(1-2), pp. 175–210.

[17] P.-A. Melliès (2009): *Categorical semantics of linear logic*. In: *Interactive models of computation and program behaviour*, *Panoramas et Synthèses* 27, Société Mathématique de France.

[18] E. Moggi (1989): *Computational Lambda-Calculus and Monads*. In: *Proc. of the Fourth Annual Symposium on Logic in Computer Science (LICS '89)*, pp. 14–23.

[19] E. Moggi (1991): *Notions of Computation and Monads*. *Inf. Comput.* 93(1), pp. 55–92.

[20] A. M. Pitts (1993): *Computational Adequacy via "Mixed" Inductive Definitions*. In S. D. Brookes, M. G. Main, A. Melton, A. W. Mislove & D. A. Schmidt, editors: *MFPS'93*, *LNCS* 802, Springer, pp. 72–82.

[21] G. D. Plotkin (1975): *Call-by-name, call-by-value and the λ-calculus*. *TCS* 1(2), pp. 125–159.

[22] A. Pravato, S. Ronchi Della Rocca & L. Roversi (1999): *The call-by-value λ-calculus: a semantic investigation*. *Mathematical Structures in Computer Science* 9(5), pp. 617–650.

[23] L. Regnier (1992): *Lambda calcul et réseaux*. Ph.D. thesis, Université Paris 7.

[24] S. Ronchi Della Rocca & L. Roversi (1997): *Lambda Calculus and Intuitionistic Linear Logic*. *Studia Logica* 59(3), pp. 417–448.

[25] M. Takahashi (1995): *Parallel Reductions in lambda-Calculus*. *Inf. Comput.* 118(1), pp. 120–127.