The Bang Calculus: an untyped lambda-calculus generalizing Call-By-Name and Call-By-Value

Thomas Ehrhard
CNRS, IRIF, UMR 8243, Univ Paris Diderot, Sorbonne Paris Cité, F-75205 Paris, France
thomas.ehrhard@pps.univ-paris-diderot.fr

Giulio Guerrieri
Aix Marseille Université, CNRS, Centrale Marseille, I2M UMR 7373, F-13453 Marseille, France
giulio.guerrieri@univ-amu.fr

Abstract

We introduce and study the Bang Calculus, an untyped functional calculus in which the promotion operation of Linear Logic is made explicit and where application is a bilinear operation. This calculus, which can be understood as an untyped version of Call-By-Push-Value, subsumes both Call-By-Name and Call-By-Value lambda-calculi, factorizing the Girard's translations of these calculi in Linear Logic. We build a denotational model of the Bang Calculus based on the relational interpretation of Linear Logic and prove an adequacy theorem by means of a resource Bang Calculus whose design is based on Differential Linear Logic.

Keywords lambda-calculus, call-by-name, call-by-value, call-by-push-value, Taylor expansion, relational model, clash, linear logic, box, sigma-reduction, denotational semantics, Girard's translations.

1. Introduction

The λ-calculus is a simple syntactic 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. 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 [36, 37]) is necessary. While CBN λ-calculus [7] 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 [40], 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!).

Linear Logic (LL, [23]) proof-nets provide a setting where this discrepancy could be solved since it is well-know that both CBN and CBV λ-calculi can be faithfully translated in LL proof nets (see below). Proof-nets syntax is extremely expressive and powerful, but it is too general for the computational purpose of representing purely functional programs.

The goal of this paper is to provide an intermediate formalism enjoying at the same time the conceptual simplicity of λ-calculus and the operational expressiveness of LL proof-nets: the *bang calculus*. It is a variant of λ-calculus which is "linear" in the sense that the exponential rules of LL are part of the syntax. The explicit availability of exponential rules in this new syntax allows to express both Girard's encodings of CBN and CBV λ-calculi into LL proof-nets as encodings into the bang calculus (which itself can be encoded into LL proof-nets as we will show in a longer version of this paper). It turns out that this bang calculus was already known in the literature: it is an untyped version of Paul Levy's Call-By-Push-Value calculus (CBPV, [30, 31]), but our motivations for introducing it are quite different from his.

In our opinion, the bang calculus is interesting also for another reason. The aforementioned theoretical notions and results well studied for the CBN λ-calculus (separability, solvability, Böhm trees, classification of λ-theories, full-abstraction, *etc.*), might be adapted and studied in the more general setting of the bang calculus, giving a compelling point of view to analyze the corresponding notions for the CBV λ-calculus. In particular we believe that the bang calculus (which enjoys a nice approximation theory thanks to the notion of Taylor expansion) might be an useful tool to find a satisfactory definition of Böhm tree for CBV λ-calculus, as well as to study the equational theory induced by the relational model. More generally, the bang calculus subsumes in the *same* rewriting system both CBN and CBV λ-calculi, thus it may be a general setting to compare CBN and CBV.

Linear Logic and Girard's translations Girard's Linear Logic (LL, [23]) can be understood as a refinement of intuitionistic logic and λ-calculus in which resource management is made explicit thanks to the introduction of a new pair of dual connectives: 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 enter other exponential boxes or can be opened (by dereliction rules).

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. The first concrete example is the category of coherence spaces and linear maps, and many other examples have been found since Girard's initial discovery, for instance the category **Rel** of sets and relations, which gives rise to the simplest and most canonical denotational

[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.

model of LL: the relational semantics. Recent results [15, 16, 26] show that the relational model faithfully represents normal proof nets.

The presence of exponentials in LL makes it possible to express at a syntactical level notions considered as semantical in the λ-calculus. This is the case in particular for the distinction between the Call-By-Name (CBN) and Call-By-Value (CBV) evaluation mechanism [40]. 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, in CBV a β-redex can be fired only when the argument is a value, *i.e.* a variable or an abstraction. CBN and CBV evaluations have a denotational counterpart called *adequacy* in two different kinds of denotational models. We say that a denotational model is adequate for a calculus if the set of normalizable terms can be characterized by means of their denotations. Adequacy has an important consequence: all observably distinct terms have distinct denotations.

In his seminal article [23], Girard proposes a standard translation of (intuitionistic logic and hence) λ-calculus in multiplicative-exponential LL proof-nets whose semantical counterpart is well known: the Kleisli category of the exponential comonad "!" is cartesian closed thanks to the mentioned strong monoidal structure of "!". This translation $(\cdot)^{\mathsf{CBN}}$ maps the intuitionistic implication $A \Rightarrow B$ to the LL formula $!A^{\mathsf{CBN}} \multimap B^{\mathsf{CBN}}$, and is related to CBN for the following reason: a λ-term reaches a head-normal form by CBN reduction iff its translation in LL proof-nets reaches a normal form for the depth-0 cut-elimination (that is, without reducing within boxes).

A similar result holds for CBV if we use another translation $(\cdot)^{\mathsf{CBV}}$ called "boring" by Girard in [23]: it maps the the intuitionistic implication $A \Rightarrow B$ to the LL formula $!(A^{\mathsf{CBV}} \multimap B^{\mathsf{CBV}})$ (or equivalently $!A^{\mathsf{CBV}} \multimap !B^{\mathsf{CBV}}$). Thus, as deeply studied in [34], the two Girard's logical translations can explain the two different calling mechanisms, bringing them into the scope of the Curry-Howard isomorphism. At the λ-term level, these two translations differ only by the way they use logical exponential (*i.e.* box and dereliction) rules, whereas both of them use multiplicative and structural (*i.e.* contraction and weakening) ingredients in the same way: since in CBN λ-calculus there is no restriction on firing a β-redex (its argument can be freely copied or erased), the translation $(\cdot)^{\mathsf{CBN}}$ puts the argument of every application into a box (see [12, 28, 41]); on the other hand, the translation $(\cdot)^{\mathsf{CBV}}$ puts only values into boxes (see [2])[2] since in CBV λ-calculus values are the only duplicable and discardable λ-terms.

The bang calculus LL proof-nets however are a complicated (and perhaps too expressive for a purely computational purpose) formalism, but from the analysis of Girard's translations it seems worthwhile to enrich the λ-calculus by allowing an explicit use of *boxes* in order to mark 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, see for instance [1, 8, 32, 34]. Because they allow for a quite liberal use of multiplicative rules, 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 calculus and is actually useless as far as we are interested in translating the λ-calculus.

The *bang calculus* we propose here is a linear λ-calculus where all variables are non-linear in the sense that there are no restrictions on the use of variables. In this syntax there is an explicit constructor not only for box but also for its dual operation, *dereliction*. As the exponential rules of LL are part of the syntax, the basic application operation of the bang calculus is a (bi-)linear application[3]. A redex can be fired only when its argument is a box or a variable, boxes and variables being the only discardable and duplicable terms. We decide to focus on an *untyped* calculus in order to capture both untyped CBN and CBV λ-calculi, but in a simply typed version we would use the following LL type constructions: if A is a type then $!A$ is a type, and if A and B are types then $!A \multimap B$ is a type. In this simply typed version, when a term T is typed, all its variables have a type of shape $!A$: if T is an abstraction (resp. a box) then it has a type of shape $!B \multimap C$ (resp. $!B$). From the point of view of structural rules, the modality "!" is used as a polarization operator along the lines of polarized LL [28, 29], even if the fragment of LL corresponding to the simply typed version of the bang calculus is not polarized (but it seems to share some features with polarized LL): *e.g.* the identity term $\lambda x\, x$ can be typed by the unpolarized formula $!X \multimap !X$.

As already mentioned, our (untyped) bang calculus is just the untyped version of a purely functional implicative fragment of Paul Levy's Call-By-Push-Value calculus (CBPV, [30, 31]) as presented in [19]. CBPV is a common framework for CBN and CBV, taking place in a line of research mainly inspired by Moggi's computational λ-calculus [36, 37]. Interestingly, thunk and force operators in CBPV correspond to our box and dereliction, respectively: we propose to view the (untyped) CBPV calculus as a nice factorization of Girard's CBN and CBV translations of (untyped) λ-calculus into LL.

To get a denotational model of the bang calculus, it suffices to find, in a model of LL, an object $\mathcal{U}$ together with two morphisms $\phi\colon !\mathcal{U} \,\&\, (!\mathcal{U} \multimap \mathcal{U}) \to \mathcal{U}$ and $\psi\colon \mathcal{U} \to !\mathcal{U} \,\&\, (!\mathcal{U} \multimap \mathcal{U})$ such that $\psi\,\phi = \mathsf{id}$; we consider here only the case where these morphisms are isos, this can be obtained under standard assumptions about the model of LL. The choice of restricting our attention to models of LL is explained by our motivation above: we see the bang calculus as a decomposition of Girard's CBN and CBV translations of λ-calculus into LL. We consider more specifically the above construction in the case of the relational model $\mathbf{Rel}$ of LL, where the above isos are identities. We present the interpretation of terms in the model $\mathcal{U}$ of $\mathbf{Rel}$ as a typing system in the spirit of non-idempotent intersection types [14, 38].

Clashes, Taylor expansion and σ-reduction Unlike usual pure (CBN and CBV) λ-calculi, reductions in our calculus can lead to *clashes*, that is, terms which make no sense. A typical example of such clashes is a box applied to another term, since "a box is not a function". A clash can nevertheless be interpreted in our model $\mathcal{U}$ and it is easy to see that its interpretation is the trivial one: for instance, in $\mathbf{Rel}$ it is the empty set. So, every term with non-empty interpretation in $\mathbf{Rel}$ (*i.e.* typable in our non-idempotent intersection typing system) has no clashes and cannot reduce to a term with clashes. We prove that the reduction of such terms terminates: it gives the first half-part of an adequacy result for $\mathbf{Rel}$ with respect to the bang calculus.

[2] Actually, variables are translated by exponential axioms, which can be seen as boxes up to η-expansion.

[3] The application of the bang calculus is linear in both the function and the argument: in an algebraic extension of the bang calculus obtained by endowing the set of terms with a structure of vector space, given two scalars α and β and three terms T, S and R, one has $\langle \alpha S + \beta R\rangle T = \alpha\langle S\rangle T + \beta\langle R\rangle T$ and $\langle T\rangle(\alpha S + \beta R) = \alpha\langle T\rangle S + \beta\langle T\rangle R$.

One could prove this termination result by reducibility, but we prefer to use a combinatory approach made possible by the non-idempotent character of our typing system related to **Rel**. We introduce an auxiliary *resource calculus* which "reifies" as resource terms the typing derivations of terms of the bang calculus. The syntax of this resource calculus is derived from differential LL [18, 21] (for which **Rel** is a denotational model) and its reduction rules are non-deterministic: a resource term reduces generally to a finite set of resource terms. In this calculus, each variable occurrence is linear and terms are never duplicated. For that reason, reduction is strongly normalizing. This resource calculus can be interpreted in the model $\mathcal{U}$ in **Rel** and this interpretation is invariant by reduction (sets of resource terms have to be interpreted as the union of the interpretation of their elements). Terms of the bang calculus can then be translated into (generally infinite) sets of resource terms thanks to an operation called *Taylor expansion*. This translation preserves denotations and *simulates* the bang calculus reduction into the resource calculus reduction: these properties are the key observations to prove the first half-part of our adequacy result.

What about the second half-part of the adequacy? Is it true that if a term is normalizable to a normal form without clashes then its interpretation in the model $\mathcal{U}$ in **Rel** is non-empty? We give a positive answer, provided that we extend the notion of reduction in the bang calculus. Indeed, to get a full adequacy we need to add reduction rules in order to reduce terms which are locked or clash-free for bad reasons in the sense that:

- either they contain configurations that "should be redexes" — and give rise to redexes in the proof-net obtained by translating the term— but are not redexes;
- or they contain configurations that "should be clashes" —and give rise to clashes in the proof-net obtained by translating the term— but are not clashes.

These new rules, called *σ-reduction* rules, allow to rearrange terms in such a way that these "hidden" redexes or clashes become explicit. Our σ-reduction is essentially obtained by putting together the σ-reduction rules already studied for CBN λ-calculus [28, 41, 42] and CBV λ-calculus [10, 24, 25]. All denotational models of the bang calculus built from a model of LL are invariant under σ-reduction.

Finally, we point out that our adequacy result holds not only for closed terms but also for the open ones and for a notion of reduction that can reduce under λ.

Related work We have already mentioned other linear λ-calculi [1, 8, 32, 34] subsuming both CBN and CBV and inspired by LL. The bang calculus simplifies their syntax, for instance it does not use explicit substitutions (yielding a more compact syntactic characterization of normal forms), and does not require a clear distinction between non-linear and linear variables. Following a different approach based on the analysis of the duality of the sequent calculus, the $\bar{\lambda}\mu\tilde{\mu}$-calculus [11] provides another setting subsuming both CBN and CBV λ-calculi: it is related to the polarized extension of LL [28, 29], rather than to a decomposition of the two Girard's translations.

Outline In §2 we introduce the syntax of the bang calculus, its reduction rules and the notion of clash, we show also how the bang calculus can simulate both CBN and CBV λ-calculi. §3 describes how to build a denotational model of the bang calculus starting from a model of LL, then a concrete semantics of the bang calculus is introduced: the relational model. In §4 we introduce the resource-sensitive version of the bang calculus and the notion of Taylor expansion, which are used in §5 to prove the adequacy result for the relational semantics of the bang calculus (non-empty semantics is equivalent to termination without clashes).

2. Syntax of the bang calculus

We present here the *bang calculus*. Given a countable set $\mathcal{V}ar$ of *variables* (denoted by $x, y, z, \dots$), the sets $!\Lambda$ of *terms* and $!\Lambda_{\mathsf{v}}$ of *values* are defined by mutual induction:

$$\begin{array}{lrll} (!\Lambda_{\mathsf{v}}) & V, W & ::= & x \mid T^{!} \qquad \textit{values} \\ (!\Lambda) & T, S, R & ::= & V \mid \lambda x\, T \mid \langle T\rangle S \mid \mathrm{der}\, T \qquad \textit{terms} \end{array}$$

Clearly, $!\Lambda_{\mathsf{v}} \subsetneq\ !\Lambda$. Terms of the form $T^{!}$ (resp. $\mathrm{der}\, T$; $\langle T\rangle S$) are called *bangs* or *boxes* (resp. *derelictions*; *linear applications*). 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 $V_1, \dots, V_n \in !\Lambda_{\mathsf{v}}$ and pairwise distinct variables $x_1, \dots, x_n$, $T\{V_1/x_1, \dots, V_n/x_n\}$ denotes the term obtained by the *capture-avoiding simultaneous substitution* of V_i for each free occurrence of x_i in the term T (for all $1 \le i \le n$): so, $(\mathrm{der}\, T)\{V/x\} = \mathrm{der}(T\{V/x\})$ and $T^{!}\{V/x\} = T\{V/x\}^{!}$. Note that values are closed under substitution: $V\{V_1/x_1, \dots, V_n/x_n\} \in !\Lambda_{\mathsf{v}}$ for any $V, V_1, \dots, V_n \in !\Lambda_{\mathsf{v}}$.

Contexts $\mathtt{C}$ and *weak contexts* $\mathtt{W}$ (both with exactly one hole $(\!|\cdot|\!)$) are defined inductively via the following grammar:

$$\begin{array}{l} \mathtt{C} ::= (\!|\cdot|\!) \mid \lambda x\, \mathtt{C} \mid \mathtt{C}M \mid M\mathtt{C} \mid \mathrm{der}\, \mathtt{C} \mid \mathtt{C}^{!} \\ \mathtt{W} ::= (\!|\cdot|\!) \mid \lambda x\, \mathtt{W} \mid \mathtt{W}M \mid M\mathtt{W} \mid \mathrm{der}\, \mathtt{W}\,. \end{array}$$

Clearly, every weak context is a context but the converse does not hold. We use $\mathtt{C}(\!|T|\!)$ for the term obtained by the capture-allowing substitution of the term T for the hole $(\!|\cdot|\!)$ in the context $\mathtt{C}$.

Consider the binary relations (*root-steps*) on $!\Lambda$ defined by (for any $T \in !\Lambda$ and $V \in !\Lambda_{\mathsf{v}}$):

$$\langle \lambda x\, T\rangle V \mapsto_{\mathsf{v}} T\{V/x\} \quad \mathrm{der}(T^{!}) \mapsto_{!} T \quad \mapsto_{\mathsf{b}} := \mapsto_{\mathsf{v}} \cup \mapsto_{!}\ .$$

A term of the form $\langle \lambda x\, T\rangle S$ (resp. $\lambda x\, T$) is called *β-like redex* (resp. *abstraction*). The root-step $\mapsto_{\mathsf{v}}$ says that a β-like redex can be fired only when the argument is a value: variables and boxes are the only erasable or duplicable terms. The root-step $\mapsto_{!}$ opens a box, *i.e.* accesses to its content, destroying its status of availability at will (but its content, in turn, might be a box).

Let $\mathsf{r} \in \{\mathsf{v}, !, \mathsf{b}\}$. The *$\mathsf{r}$-reduction* $\to_{\mathsf{r}}$ (resp. *weak r-reduction* or *r_{w}-reduction* $\to_{\mathsf{r}_{\mathsf{w}}}$) is the closure under contexts (resp. weak contexts) of the binary relation $\mapsto_{\mathsf{r}}$ on $!\Lambda$, *i.e.*

$$\begin{array}{l} T \to_{\mathsf{r}} S \Leftrightarrow \exists\, \mathtt{C}, \exists\, R, Q \in !\Lambda : T = \mathtt{C}(\!|R|\!), S = \mathtt{C}(\!|Q|\!), R \mapsto_{\mathsf{r}} Q \\ T \to_{\mathsf{r}_{\mathsf{w}}} S \Leftrightarrow \exists\, \mathtt{W}, \exists\, R, Q \in !\Lambda : T = \mathtt{W}(\!|R|\!), S = \mathtt{W}(\!|Q|\!), R \mapsto_{\mathsf{r}} Q \end{array} \quad (1)$$

We set $\to_{\mathsf{r}_{\neg\mathsf{w}}} := \to_{\mathsf{r}} \setminus \to_{\mathsf{r}_{\mathsf{w}}}$. The *$\mathsf{r}$-equivalence* $\simeq_{\mathsf{r}}$ is the reflexive-transitive and symmetric closure of the r-reduction $\to_{\mathsf{r}}$.

Clearly, $\to_{\mathsf{r}_{\mathsf{w}}} \subsetneq \to_{\mathsf{r}}$: the only difference between $\to_{\mathsf{r}}$ and $\to_{\mathsf{r}_{\mathsf{w}}}$ is that the latter does not reduce under ! (but both reduce under λ). Values and abstractions, respectively, are closed under b-reduction: $V \to_{\mathsf{b}} T$ implies $T \in !\Lambda_{\mathsf{v}}$, for every $V \in !\Lambda_{\mathsf{v}}$; $\lambda x\, S \to_{\mathsf{b}} T$ implies $T = \lambda x\, R$ with $S \to_{\mathsf{b}} R$.

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

Clearly, $\to_{\mathsf{b}}$ and $\to_{\mathsf{b}_{\mathsf{w}}}$ are not deterministic reductions. We aim to prove strong confluence of $\to_{\mathsf{b}_{\mathsf{w}}}$ and confluence of $\to_{\mathsf{b}}$.

Lemma 1 (Basic properties of reductions). Proof p. 13

1. *$\to_{\mathsf{v}_{\mathsf{w}}}$ is strongly confluent:* ${}_{\mathsf{v}_{\mathsf{w}}}\!\leftarrow \cdot \to_{\mathsf{v}_{\mathsf{w}}} \subseteq (\to_{\mathsf{v}_{\mathsf{w}}} \cdot\, {}_{\mathsf{v}_{\mathsf{w}}}\!\leftarrow) \cup =$.
2. *$\to_{!_{\mathsf{w}}}$ and $\to_{!}$ are strongly normalizing and strongly confluent (separately).*
3. *$\to_{!_{\mathsf{w}}}$ and $\to_{\mathsf{v}_{\mathsf{w}}}$ strongly commute:* ${}_{!_{\mathsf{w}}}\!\leftarrow \cdot \to_{\mathsf{v}_{\mathsf{w}}} \subseteq \to_{\mathsf{v}_{\mathsf{w}}} \cdot\, {}_{!_{\mathsf{w}}}\!\leftarrow$.
   $\to_{!}$ and $\to_{\mathsf{v}}$ commute: ${}_{!}^{*}\!\leftarrow \cdot \to_{\mathsf{v}}^{*} \subseteq \to_{\mathsf{v}}^{*} \cdot\, {}_{!}^{*}\!\leftarrow$.
4. *$\to_{\mathsf{v}}$ is confluent.*

The proof of Lemma 1.4 is a standard adaptation of Tait–Martin-Löf technique using parallel reductions (see [44]).

Proposition 1 (Confluence). *$\to_{\mathsf{b}_\mathsf{w}}$ is strongly confluent and $\to_\mathsf{b}$ is confluent.*

Proof. Since $\to_{!_\mathsf{w}}$ and $\to_{\mathsf{v}_\mathsf{w}}$ are strongly confluent (Lemmas 1.1-2) and strongly commute (Lemma 1.3), then it follows immediately that $\to_{!_\mathsf{w}} \cup \to_{\mathsf{v}_\mathsf{w}}$ is strongly confluent, where by definition $\to_{\mathsf{b}_\mathsf{w}} = \to_{!_\mathsf{w}} \cup \to_{\mathsf{v}_\mathsf{w}}$.

Since $\to_!$ and $\to_\mathsf{v}$ are confluent (Lemmas 1.2 and 1.4) and commute (Lemma 1.3), $\to_! \cup \to_\mathsf{v}$ is confluent according to Hindley-Rosen Lemma [7, Lemma 3.3.5], where by definition $\to_\mathsf{b} = \to_! \cup \to_\mathsf{v}$. □

The *bang* (resp. *weak bang*) *calculus* is the set $!\Lambda$ endowed with the reduction $\to_\mathsf{b}$ (resp. $\to_{\mathsf{b}_\mathsf{w}}$). Prop. 1 ensures that in the bang and weak bang calculi every term has at most one normal form.

2.1 Embedding CBN and CBV λ-calculi in the bang calculus

One of the interests of the bang calculus is that it is a general framework subsuming both call-by-name (CBN *i.e.* ordinary, [7]) and Plotkin's call-by-value (CBV, [40]) λ-calculi. Consider the translations $(\cdot)^{\mathsf{cbn}}$ and $(\cdot)^{\mathsf{cbv}}$ of ordinary λ-terms into $!\Lambda$ defined as follows:

$$\begin{array}{ll} x^{\mathsf{cbn}} := \mathrm{der}\, x & x^{\mathsf{cbv}} := x \\ (\lambda x\, T)^{\mathsf{cbn}} := \lambda x\, T^{\mathsf{cbn}} & (\lambda x\, T)^{\mathsf{cbv}} := (\lambda x\, T^{\mathsf{cbv}})^! \\ (TS)^{\mathsf{cbn}} := \langle T^{\mathsf{cbn}} \rangle S^{\mathsf{cbn}\,!} & (TS)^{\mathsf{cbv}} := \langle \mathrm{der}(T^{\mathsf{cbv}}) \rangle S^{\mathsf{cbv}}\,. \end{array}$$

For instance, setting $\Omega := (\lambda x\, xx)\lambda x\, xx$ (the typical diverging λ-term for CBN and CBV λ-calculi), $\Omega^{\mathsf{cbn}} = \langle \Delta' \rangle \Delta'^!$ and $\Omega^{\mathsf{cbv}} = \langle \mathrm{der}(\Delta^!) \rangle \Delta^!$ (Δ and Δ' are defined in Ex. 1).

Note that the translation $(\cdot)^{\mathsf{cbn}}$ puts the argument of any application into a box: in CBN λ-calculus every λ-term is duplicable or discardable. On the other hand, abstractions are translated by $(\cdot)^{\mathsf{cbv}}$ into abstractions inside boxes, since they are the only λ-terms, besides variables, duplicable or discardable in CBV λ-calculus.

Clearly, $\mathsf{fv}(T^{\mathsf{cbn}}) = \mathsf{fv}(T) = \mathsf{fv}(T^{\mathsf{cbv}})$ for any λ-term T. We denote by $\mathsf{hoc}(T)$ the leftmost variable occurrence (it always exists) in T, where T is a λ-term or a term in $!\Lambda$.

Remark 1. Given a λ-term T and a variable x, there is a natural one-to-one correspondence between the occurrences of x in T and the occurrences of x in T^{cbn} or T^{cbv}. In particular, $\mathsf{hoc}(T)$ corresponds to $\mathsf{hoc}(T^{\mathsf{cbn}})$ and $\mathsf{hoc}(T^{\mathsf{cbv}})$, and conversely. Moreover, the translations $(\cdot)^{\mathsf{cbn}}$ and $(\cdot)^{\mathsf{cbv}}$ preserve the kind of variable occurrence: if a variable occurrence of x is free (resp. bound) in T then the corresponding occurrence in T^{cbn} and T^{cbv} is free (resp. bound), and conversely.

Proof p. 14

Lemma 2 (Substitution). *Let T, S be λ-terms and x be a variable.*

1. *If x has $n \in \mathbb{N}$ free occurrences in T, then $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}\,!}/x\} \to_!^n (T\{S/x\})^{\mathsf{cbn}}$. More precisely: if $\mathsf{hoc}(T)$ is a free occurrence of x in T, then $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}\,!}/x\} \to_{!_\mathsf{w}} \to_{!_{\neg\mathsf{w}}}^{n-1} (T\{S/x\})^{\mathsf{cbn}}$; otherwise, $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}\,!}/x\} \to_{!_{\neg\mathsf{w}}}^n (T\{S/x\})^{\mathsf{cbn}}$.*
2. *If S is a variable or an abstraction, then $T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = (T\{S/x\})^{\mathsf{cbv}}$.*

We denote by $\to_\beta$ and $\to_{\beta^h}$ the ordinary (*i.e.* call-by-name) β-reduction and head β-reduction, respectively. We denote by $\to_{\beta_v}$ and $\to_{\beta_v^h}$ the (call-by-value) β_v-reduction and head β_v-reduction, respectively (informally, the latter reduces the leftmost-outermost β_v-redex not under the scope of a λ, see [40, p. 136]).

Proposition 2 (Simulations). *Let T and S be λ-terms.*

1. *If $T \to_\beta S$ then $T^{\mathsf{cbn}} \to_\mathsf{v} \to_!^* S^{\mathsf{cbn}}$.*
2. *If $T \to_{\beta^h} S$ then $T^{\mathsf{cbn}} \to_{\mathsf{v}_\mathsf{w}} \to_{!_\mathsf{w}}^= \to_{!_{\neg\mathsf{w}}}^* S^{\mathsf{cbn}}$.*
3. *If $T \to_{\beta_v} S$ then $T^{\mathsf{cbv}} \to_! \to_\mathsf{v} S^{\mathsf{cbv}}$.*
4. *If $T \to_{\beta_v^h} S$ then $T^{\mathsf{cbv}} \to_{!_\mathsf{w}} \to_{\mathsf{v}_\mathsf{w}} S^{\mathsf{cbv}}$.*

Proof. Both proof are by induction on the λ-term T.

1. According to the definition of $T \to_\beta S$, there are the following cases:
   - *Root, i.e.* $T := (\lambda x\, R)Q \to_\beta R\{Q/x\} =: S$: then, $T^{\mathsf{cbn}} = \langle \lambda x\, R^{\mathsf{cbn}} \rangle Q^{\mathsf{cbn}\,!} \to_{\mathsf{v}_\mathsf{w}} R^{\mathsf{cbn}}\{Q^{\mathsf{cbn}\,!}/x\} \to_{!_\mathsf{w}}^= \to_{!_{\neg\mathsf{w}}}^* S^{\mathsf{cbn}}$ by Lemma 2.1 (recall that $\to_{\mathsf{v}_\mathsf{w}} \subseteq \to_\mathsf{v}$ and $\to_{!_\mathsf{w}}, \to_{!_{\neg\mathsf{w}}} \subseteq \to_!$).
   - *Abstraction, i.e.* $T := \lambda x\, R \to_\beta \lambda x\, R' =: S$ with $R \to_\beta R'$: by induction hypothesis, $R^{\mathsf{cbn}} \to_\mathsf{v} \to_!^* R'^{\mathsf{cbn}}$, so $T^{\mathsf{cbn}} = \lambda x\, R^{\mathsf{cbn}} \to_\mathsf{v} \to_!^* \lambda x\, R'^{\mathsf{cbn}} = S^{\mathsf{cbn}}$.
   - *Application Left, i.e.* $T := RQ \to_\beta R'Q =: S$ with $R \to_\beta R'$: by induction hypothesis, $R^{\mathsf{cbn}} \to_\mathsf{v} \to_!^* R'^{\mathsf{cbn}}$, so $T^{\mathsf{cbn}} = \langle R^{\mathsf{cbn}} \rangle Q^{\mathsf{cbn}\,!} \to_\mathsf{v} \to_!^* \langle R'^{\mathsf{cbn}} \rangle Q^{\mathsf{cbn}\,!} = S^{\mathsf{cbn}}$.
   - *Application Right, i.e.* $T := QR \to_\beta QR' =: S$ with $R \to_\beta R'$: by induction hypothesis, $R^{\mathsf{cbn}} \to_\mathsf{v} \to_!^* R'^{\mathsf{cbn}}$, so $T^{\mathsf{cbn}} = \langle Q^{\mathsf{cbn}} \rangle R^{\mathsf{cbn}\,!} \to_\mathsf{v} \to_!^* \langle Q^{\mathsf{cbn}} \rangle R'^{\mathsf{cbn}\,!} = S^{\mathsf{cbn}}$.
2. Analogously to the proof of Prop. 2.1, with the difference that the β^h-reduction does not give rise to the case *Application Right*.
3. According to the definition of $T \to_{\beta_v} S$, there are the following cases:
   - *Root, i.e.* $T := (\lambda x\, R)Q \to_{\beta_v} R\{Q/x\} =: S$ where Q is a variable or an abstraction: then $Q^{\mathsf{cbv}} \in !\Lambda_\mathsf{v}$, whence by Lemma 2.2 $T^{\mathsf{cbv}} = \langle \mathrm{der}((\lambda x\, R^{\mathsf{cbv}})^!) \rangle Q^{\mathsf{cbv}} \to_{!_\mathsf{w}} \langle (\lambda x\, R^{\mathsf{cbv}})^! \rangle Q^{\mathsf{cbv}} \to_{\mathsf{v}_\mathsf{w}} R^{\mathsf{cbv}}\{Q^{\mathsf{cbv}}/x\} = S^{\mathsf{cbv}}$ (recall that $\to_{\mathsf{v}_\mathsf{w}} \subseteq \to_\mathsf{v}$ and $\to_{!_\mathsf{w}} \subseteq \to_!$).
   - *Abstraction, i.e.* $T := \lambda x\, R \to_{\beta_v} \lambda x\, R' =: S$ with $R \to_{\beta_v} R'$: by induction hypothesis $R^{\mathsf{cbv}} \to_! \to_\mathsf{v} R'^{\mathsf{cbv}}$, so $T^{\mathsf{cbv}} = (\lambda x\, R^{\mathsf{cbv}})^! \to_! \to_\mathsf{v} (\lambda x\, R'^{\mathsf{cbv}})^! = S^{\mathsf{cbv}}$.
   - *Application Left, i.e.* $T := RQ \to_{\beta_v} R'Q =: S$ with $R \to_{\beta_v} R'$: by induction hypothesis $R^{\mathsf{cbv}} \to_! \to_\mathsf{v} R'^{\mathsf{cbv}}$, therefore one has $T^{\mathsf{cbv}} = \langle \mathrm{der}(R^{\mathsf{cbv}}) \rangle Q^{\mathsf{cbv}} \to_! \to_\mathsf{v} \langle \mathrm{der}(R'^{\mathsf{cbv}}) \rangle Q^{\mathsf{cbv}} = S^{\mathsf{cbv}}$.
   - *Application Right, i.e.* $T := QR \to_{\beta_v} QR' =: S$ with $R \to_{\beta_v} R'$: by induction hypothesis, $R^{\mathsf{cbv}} \to_! \to_\mathsf{v} R'^{\mathsf{cbv}}$, therefore $T^{\mathsf{cbv}} = \langle \mathrm{der}(Q^{\mathsf{cbv}}) \rangle R^{\mathsf{cbv}} \to_! \to_\mathsf{v} \langle \mathrm{der}(Q^{\mathsf{cbv}}) \rangle R'^{\mathsf{cbv}} = S^{\mathsf{cbv}}$.
4. Analogously to the proof of Prop. 2.3, with the difference that the β_v^h-reduction does not give rise to the case *Abstraction*. □

So, not only the bang calculus can simulate β- and β_v-reductions, but the simulation is "modular". Indeed, according to Prop. 2.4, the head β_v-reduction can be simulated in the bang calculus by weak b-steps only. Prop. 2.2 says that to embed a head β-reduction sequence (with several β^h-steps) in the bang calculus, some non-weak !-steps are needed, however it can be shown that such non-weak !-steps can always be postponed in terms b-equivalent to the image via $(\cdot)^{\mathsf{cbn}}$ of some λ-term: the "core" of this simulation is given by weak b-steps.

The syntax of bang calculus can also be reformulated allowing the use of explicit substitutions as in [3–5] and both the CBN and CBV versions of these calculi (and their linear variants) can be simulated in the bang calculus. Also the call-by-need λ-calculus (see [6, 27, 34]) can be simulated in the bang calculus.

2.2 Digression: bang calculus and linear logics proof-nets

Bang calculus is deeply related to linear logic (LL) proof-nets [23]. Here we just sketch this relationship, the precise analysis will be

given in a future work: it is not needed to understand the sequel of this paper, except for the key-notion of clash (Def. 1).

Consider the Girard's translations $(\cdot)^{\mathsf{CBN}}$ of untyped call-by-name [12, 28, 41] and $(\cdot)^{\mathsf{CBV}}$ of untyped call-by-value [2] λ-calculi into LL proof-nets based on, respectively, the recursive types identity $o = !o \multimap o$ and $o = !(o \multimap o)$ (or, equivalently, $o = !o \multimap !o$): they differ essentially in handling boxes and derelictions. Since the bang calculus has explicit constructors for boxes and derelictions, it is natural to define a translation $(\cdot)^\circ$ of untyped bang calculus into untyped LL proof-nets that decomposes $(\cdot)^{\mathsf{CBN}}$ and $(\cdot)^{\mathsf{CBV}}$ in such a way that the following diagrams commute (where Λ is the set of λ-terms, LL is the set of untyped proof-nets, $(\cdot)^{\mathsf{cbn}}$ and $(\cdot)^{\mathsf{cbv}}$ are defined in §2.1):

So, the bang calculus allows to internalize the two Girard's translations in a λ-like calculus instead of LL proof-nets.

Moreover, b-reduction steps correspond via $(\cdot)^\circ$ to cut-elimination steps, and weak b-reduction steps to cut-elimination steps at depth 0; !-reduction steps correspond via $(\cdot)^\circ$ to dereliction/box cut-elimination steps, and v-reduction steps to multiplicative and structural cut-elimination steps.

Actually, proof-nets that are targets of $(\cdot)^{\mathsf{CBN}}$ and $(\cdot)^{\mathsf{CBV}}$ can be typed using the recursive types identity (where & has to be intended as a sort of cartesian product, from a categorical viewpoint):

$$o = (!o \multimap o) \,\&\, !o\,. \tag{2}$$

It is interesting to consider terms of the bang calculus whose translation $(\cdot)^\circ$ into LL proof-nets is typable according to Eq. (2): this leads naturally to the notion of clash.

Definition 1 (Clash)**.** *A* clash *is a term of* $!\Lambda$ *having one of the following forms:*

$$\langle V\rangle T \qquad \mathrm{der}(\lambda x\, T) \qquad \langle T\rangle \lambda x\, S\,.$$

Let $T \in !\Lambda$*:* T *is* clash-free *iff it contains no clash;* T *is* clash-free at depth 0 *iff each clash occurring in* T *is under the scope of a* !*.*

E.g., $\langle \lambda z\, x\rangle(\langle x\rangle y)^!$ is clash-free at depth 0 but not clash-free.

It can be shown that $T \in !\Lambda$ is clash-free (resp. clash-free at depth 0) when T° is typable (resp. typable at depth 0) according to Eq. (2); in particular, *all terms that are the* $(\cdot)^{\mathsf{cbn}}$ *or* $(\cdot)^{\mathsf{cbv}}$ *translation of a (untyped)* λ*-term are* typable according to Eq. (2) and hence *clash-free*. A semantic counterpart of this fact will be detailed in §5, in particular we will show that all terms b-reducing to terms that are not clash-free at depth 0 have an empty semantics in the relational model: in a way, b-normal forms containing clashes at depth 0 can be seen as "meaningless" terms.

Observe that b- and $\mathsf{b_w}$-reductions might create new clashes, even at depth 0: the term $T := \langle \lambda x\, \langle \mathrm{der}\, x\rangle y\rangle z^!$ (with $x \neq y$) is clash-free but $T \to_{\mathsf{b_w}} \langle \mathrm{der}(z^!)\rangle y \to_{\mathsf{b_w}} \langle z\rangle y$ where $\langle z\rangle y$ is not clash-free (at depth 0).

Clashes are defined in a purely syntactic way so that this notion makes sense also if one ignores the translation $(\cdot)^\circ$ of bang calculus into LL proof-nets. Intuitively, the bang calculus is based on a clear distinction between values (the only terms that can be duplicated, discarded and accessed through dereliction) and abstractions (the only terms representing functions): these two sets of terms are disjoint and closed under b-reduction. In a way, clashes "violate" this distinction: they apply a value (which is not a function) to something else, or try to access to an abstraction (which is not a value) through a dereliction, or apply a function to an abstraction (which is not a value). It is a natural (and still open) question to find a type system characterizing all and only the terms that do not reduce to a term containing a clash or a clash at depth 0. This amounts to better understanding the logical meaning of the "&" in Eq. (2).

2.3 Extending the bang calculus: σ-reduction

The reduction $\to_{\mathsf{v}}$ fires a β-like redex $\langle \lambda x\, T\rangle S$ only when the argument S is a value, *i.e.* a term of a special form. Such a kind of reduction gives rise to the problem of "premature" normal forms, an issue impacting on termination and hence the notion of observational equivalence. Consider the following terms (Δ is defined in Ex. 1, and $A_x := \langle \mathrm{der}\, x\rangle x$):

$$\begin{aligned} &T := \langle\langle \lambda x \Delta\rangle A_x\rangle \Delta^! && S := \langle \Delta\rangle\big(\langle \lambda x \Delta^!\rangle A_x\big) \\ &P := \mathrm{der}\,\big(\langle \lambda x\, (\langle \Delta\rangle \Delta^!)^!\rangle A_x\big) \end{aligned} \tag{3}$$

They are clash-free and either b-normal (T and S) or $\mathsf{b_w}$-normal (P), but one would expect that they diverge for the $\mathsf{b_w}$-reduction. Indeed, from a semantic point of view, the interpretations of T, S and P are empty in the relational model, as well as the interpretation of the $\mathsf{b_w}$-diverging term $\langle \Delta\rangle \Delta^!$, see §3. From a syntactic point of view, in T, S and P the argument of λx is A_x which is not (and cannot reduce to) a value; but in T and S there is an "hidden" v-redex whose function is Δ and argument is $\Delta^!$ (a value); while in P there is an "hidden" !-redex between der and $(\langle \Delta\rangle \Delta^!)^!$. In a way, the hyper-sequential structure of terms may conceal some v- and !-redexes, which become explicit in a more parallel syntax such as LL proof-nets via the translation $(\cdot)^\circ$ sketched in §2.2. A similar phenomenon appears in CBV λ-calculus [5, 10, 25, 39, 43], but not in CBN λ-calculus where there is no restriction on firing β-redexes.

Premature b-/$\mathsf{b_w}$-normal forms can conceal not only b-/$\mathsf{b_w}$-redexes (and hence divergence) but also clashes (Def. 1). Consider the following terms, with x, y, z pairwise distinct variables and $A_y := \langle \mathrm{der}\, y\rangle y$:

$$R := \mathrm{der}(\langle \lambda y\, \lambda x\, z\rangle A_y) \qquad Q := \langle A_x\rangle(\langle \lambda y\, \lambda x\, z\rangle A_y) \tag{4}$$

R and Q are b-normal and clash-free, but actually they contain an "hidden" clash at depth 0 of the form $\mathrm{der}(\lambda x\, T)$ and $\langle T\rangle \lambda x\, S$ respectively, since it is reasonable to consider their b-normal subterm $\langle \lambda y\, \lambda x\, z\rangle A_y$ as a sort of abstraction: $\lambda x\, z$ is concealed by the dummy application of λy to A_y. Indeed, from a semantic viewpoint, the interpretations of R and Q are empty in the relational model as well as the semantics of $\mathrm{der}(\lambda x\, z)$ and $\langle A_x\rangle \lambda x\, z$ (see §3.3).

Premature $\mathsf{b_w}$-normal forms as T, S, R, Q, P in Eq. (3)-(4) stand in the way of adequacy for any semantics of the bang calculus. To avoid this problem, σ*-reduction* is introduced: it shuffles constructors so as to unveil hidden v- and !-redexes and hidden clashes.

Consider the binary relations (*root-steps*) on $!\Lambda$ defined by (for any $T, S, R \in !\Lambda$):

$$\begin{aligned} \langle\langle \lambda x\, T\rangle R\rangle S &\mapsto_{\sigma_1} \langle \lambda x \langle T\rangle S\rangle R && x \notin \mathsf{fv}(S) \\ \langle \lambda y\, \lambda x\, T\rangle S &\mapsto_{\sigma_2} \lambda x \langle \lambda y\, T\rangle S && x \notin \mathsf{fv}(S) \cup \{y\} \\ \langle T\rangle(\langle \lambda x\, S\rangle R) &\mapsto_{\sigma_3} \langle \lambda x \langle T\rangle S\rangle R && x \notin \mathsf{fv}(T) \\ \mathrm{der}(\langle \lambda x\, T\rangle S) &\mapsto_{\sigma_4} \langle \lambda x\, \mathrm{der}\, T\rangle S \\ \mapsto_\sigma &:= \mapsto_{\sigma_1} \cup \mapsto_{\sigma_2} \cup \mapsto_{\sigma_3} \cup \mapsto_{\sigma_4} && \mapsto_{\mathsf{b}\sigma} := \mapsto_{\mathsf{b}} \cup \mapsto_\sigma \end{aligned}$$

The above conditions about free variables can always be fulfilled by α-conversion. For any $\mathsf{r} \in \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma, \mathsf{b}\sigma\}$, the r-*reduction* $\to_{\mathsf{r}}$ (resp. *weak* r-*reduction* or $\mathsf{r_w}$-*reduction* $\to_{\mathsf{r_w}}$) is the closure of $\mapsto_{\mathsf{r}}$ under contexts (resp. weak contexts), see Eq. (1); the r-*equivalence* $\simeq_{\mathsf{r}}$ is the reflexive-transitive and symmetric closure of the r-reduction $\to_{\mathsf{r}}$. Observe that values and abstractions are closed under $\mathsf{r_w}$- and r-reductions.

The *extended* (resp. *extended weak*) *bang calculus* is the set $!\Lambda$ endowed with the reduction $\to_{\mathsf{b}\sigma}$ (resp. $\to_{\mathsf{b}\sigma_{\mathsf{w}}}$).

Example 2. Let T, S, R, Q, P be as in Eq. (3)-(4).

$$T \to_{\sigma_{1_\mathsf{w}}} B \;{}_{\sigma_{3_\mathsf{w}}}\!\leftarrow S \qquad P \to_{\sigma_{4_\mathsf{w}}} \langle \lambda x\, \mathrm{der}((\langle\Delta\rangle\Delta^!)^!)\rangle A_x \to_! B$$

where $B \coloneqq \langle \lambda x\, \langle\Delta\rangle\Delta^!\rangle A_x$ is neither b-/b_w- nor $\mathsf{b}\sigma$-/$\mathsf{b}\sigma_\mathsf{w}$-normalizable. Besides, $R \to_{\sigma_{2_\mathsf{w}}} \mathrm{der}(\lambda x\langle \lambda y\, z\rangle A_y)$, which is a ($\mathsf{b}\sigma$-normal) clash; analogously for $Q \to_{\sigma_{2_\mathsf{w}}} \langle A_x\rangle \lambda x\langle \lambda y\, z\rangle A_y$.

Ex. 2 shows that the σ_w-reduction can "unveil" hidden b_w-redexes and hidden clashes at depth 0, transforming clash-free at depth 0 b_w-normal forms into terms which are not b_w-normalizable or not clash-free at depth 0.

Note that $\to_{\sigma_1} \cup \to_{\sigma_2}$ is known as Regnier's σ-reduction for the CBN λ-calculus [28, 41, 42], while $\to_{\sigma_1} \cup \to_{\sigma_3}$ is already used in CBV λ-calculus [10, 24, 25]. This means in particular that not only Plotkin's CBV λ-calculus [40] but also its extension with σ-rules [10, 24, 25] can be simulated in the extended bang calculus. Interestingly, in the λ-calculus, $\to_{\sigma_1} \cup \to_{\sigma_2}$ is contained in the (CBN) β-equivalence, while $\to_{\sigma_1} \cup \to_{\sigma_3}$ are not contained in the (CBV) β_v-equivalence.

Proof p. 15 **Proposition 3.** *$\to_\sigma$ and $\to_{\sigma_\mathsf{w}}$ are strongly normalizing.*

Proof. For any term T, we define two sizes $\mathsf{s}(T)$ and $\#(T)$ by induction on $T \in !\Lambda$:

$$\begin{aligned}
\mathsf{s}(x) &\coloneqq 1 \\
\mathsf{s}(\lambda x\, T) &\coloneqq \mathsf{s}(T) + 1 \\
\mathsf{s}(\langle T\rangle S) &\coloneqq \mathsf{s}(T) + \mathsf{s}(S) \\
\mathsf{s}(\mathrm{der}\, T) &\coloneqq \mathsf{s}(T) \\
\mathsf{s}(T^!) &\coloneqq \mathsf{s}(T)
\end{aligned}
\qquad
\begin{aligned}
\#(x) &\coloneqq 1 \\
\#(\lambda x\, T) &\coloneqq \#(T) + \mathsf{s}(T) \\
\#(\langle T\rangle S) &\coloneqq \#(T) + \#(S) + 2\,\mathsf{s}(T)\,\mathsf{s}(S) - 1 \\
\#(\mathrm{der}\, T) &\coloneqq \#(T) + \mathsf{s}(T) \\
\#(T^!) &\coloneqq \#(T)
\end{aligned}$$

Note that $\mathsf{s}(T) > 0$ and $\#(T) > 0$ for any $T \in !\Lambda$. It is easy to check that if $T \to_\sigma S$ (in particular, if $T \to_{\sigma_\mathsf{w}} S$) then $\mathsf{s}(T) = \mathsf{s}(S)$ and $\#(T) > \#(S)$. □

Let $\mathsf{r} \in \{\sigma, \sigma_\mathsf{w}, \mathsf{b}\sigma, \mathsf{b}\sigma_\mathsf{w}\}$. The reduction $\to_\mathsf{r}$ is not confluent: for $x \neq y$, the only r-reduction sequences starting from $T \coloneqq \langle\langle \lambda x\, \lambda y\, z\rangle(\langle \mathrm{der}\, x\rangle x)\rangle(\langle \mathrm{der}\, y\rangle y)$ are (recall that $\to_{\sigma_\mathsf{w}} \subseteq \to_\mathsf{r}$)

$$T \to_{\sigma_{1_\mathsf{w}}} \langle \lambda x\, \langle \lambda y\, z\rangle(\langle \mathrm{der}\, y\rangle y)\rangle(\langle \mathrm{der}\, x\rangle x) =: S$$
$$T \to_{\sigma_{2_\mathsf{w}}} \langle \lambda y\, \langle \lambda x\, z\rangle(\langle \mathrm{der}\, x\rangle x)\rangle(\langle \mathrm{der}\, y\rangle y) =: R$$

with S and R r-normal but $S \neq R$. We conjecture that $\to_\mathsf{r}$ is confluent modulo $\sim_\mathsf{sh}$ (*i.e.* ${}^*_\mathsf{r}\!\leftarrow \cdot \sim_\mathsf{sh} \cdot \to^*_\mathsf{r} \subseteq \to^*_\mathsf{r} \cdot \sim_\mathsf{sh} \cdot {}^*_\mathsf{r}\!\leftarrow$), where $\sim_\mathsf{sh}$ is the least equivalence relation on $!\Lambda$ closed under contexts such that $\langle \lambda y\langle \lambda x\, T\rangle R\rangle S \sim_\mathsf{sh} \langle \lambda x\langle \lambda y\, T\rangle S\rangle R$ for any $T, S, R \in !\Lambda$ with $x \notin \mathsf{fv}(S)$, $y \notin \mathsf{fv}(R)$ and $x \neq y$. Observe that $\sim_\mathsf{sh}$ is contained in the σ-equivalence $\simeq_\sigma$ (the reflexive-transitive and symmetric closure of $\to_\sigma$), since

$$\langle \lambda y\langle \lambda x\, T\rangle R\rangle S \;{}_{\sigma_1}\!\leftarrow \langle\langle \lambda y \lambda x\, T\rangle S\rangle R \to_{\sigma_2} \langle \lambda x\langle \lambda y\, T\rangle S\rangle R.$$

It is worth noting that, for any $T, S \in !\Lambda$, if $T \to_\sigma S$ or $T \sim_\mathsf{sh} S$, then the translations T° and S° into LL proof-nets (see §2.2) are equal up to multiplicative cut-elimination steps.

Remark 2. It is easy to check that clashes *at depth 0* are preserved by $\mathsf{b}\sigma_\mathsf{w}$-reduction: if T contains a clash at depth 0 and $T \to_{\mathsf{b}\sigma_\mathsf{w}} S$ then S contains a clash at depth 0. Terms R and Q in Eq. (4) show that clashes at depth 0 are not preserved by $\mathsf{b}\sigma_\mathsf{w}$-expansion (it is one of the interests of σ_w-reduction to unveil "hidden" clashes at depth 0): the fact that S contains a clash at depth 0 and $T \to_{\mathsf{b}\sigma_\mathsf{w}} S$ does not imply that T contains a clash at depth 0, see Ex. 2.

Clashes (not at depth 0) are not preserved by $\mathsf{b}\sigma_\mathsf{w}$-reduction (or $\mathsf{b}\sigma_\mathsf{w}$-expansion): *e.g.* $\langle \lambda x\, y\rangle(\langle z\rangle z)^!$ contains a clash (not at depth 0) but $\langle \lambda x\, y\rangle(\langle z\rangle z)^! \to_{\mathsf{b}_\mathsf{w}} y$ where y is clash-free.

3. Linear-logic based denotational semantics of bang calculus

The denotational models we are interested in this paper are those induced by a model of LL, as explained in [19]. We recall the basic definitions and notations, see [19, 35] for more details. A model of LL consists of the following data:

A symmetric monoidal closed category, SMCC for short, $(\mathcal{L}, \otimes, 1, \lambda, \rho, \alpha, \sigma)$. We use $X \multimap Y$ for the object of linear morphisms from X to Y, $\mathsf{ev} \in \mathcal{L}((X \multimap Y) \otimes X, Y)$ for the evaluation morphism and cur for the linear curryfication map $\mathcal{L}(Z \otimes X, Y) \to \mathcal{L}(Z, X \multimap Y)$. For convenience, and because it is the case in the concrete models we consider, we assume this SMCC to be a $*$-autonomous category with dualizing object $\bot$. We use $X^\bot$ for the object $X \multimap \bot$ of $\mathcal{L}$ (the *dual*, or *linear negation*, of X). This operation $(\_)^\bot$ is a functor $\mathcal{L}^{\mathsf{op}} \to \mathcal{L}$.

The category $\mathcal{L}$ is cartesian with terminal object $\top$, product $\&$, projections pr_i. By $*$-autonomy $\mathcal{L}$ is cocartesian with initial object 0, coproduct $\oplus$ and injections in_i.

We are given a comonad $!\_ \colon \mathcal{L} \to \mathcal{L}$ with counit $\mathsf{der}_X \in \mathcal{L}(!X, X)$ (*dereliction*) and comultiplication $\mathsf{dig}_X \in \mathcal{L}(!X, !!X)$ (*digging*) together with a strong symmetric monoidal structure (Seely isos m^0 and m^2) for the functor $!\_$, from the symmetric monoidal category $(\mathcal{L}, \&)$ to the symmetric monoidal category $(\mathcal{L}, \otimes)$ satisfying an additional coherence condition wrt. dig.

We use $?\_$ for the "De Morgan dual" of $!\_$, *i.e.* $?X = (!(X^\bot))^\bot$ and similarly for morphisms. It is a monad on $\mathcal{L}$.

3.1 Specific assumptions

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

We need first to assume that the unique morphism in $\mathcal{L}(0, \top)$ is an iso. To simplify, let us simply assume that $0 = \top$. It follows that for any two objects X and Y we have a morphism $\mathsf{t}\,\mathsf{i} \in \mathcal{L}(X, Y)$ 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 morphism, to be understood as the undefined.

In order to define the object where our terms will be interpreted, we assume that the category $\mathcal{L}$ is equipped with a notion of "embedding-retraction pairs", following a standard approach. We use $\mathcal{L}_\subseteq$ for the corresponding category. It is equipped with a functor $\mathsf{F} : \mathcal{L}_\subseteq \to \mathcal{L}^{\mathsf{op}} \times \mathcal{L}$ such that $\mathsf{F}(X) = (X, X)$ and for which we use the notation $(\phi^-, \phi^+) = \mathsf{F}(\phi)$ and assume that $\phi^-\,\phi^+ = \mathsf{id}_X$. We assume furthermore that 0 is the initial object of $\mathcal{L}_\subseteq$ and that $\mathcal{L}_\subseteq$ has all countable directed colimits and that the functor $\mathsf{E} = \mathsf{pr}_2\,\mathsf{F} : \mathcal{L}_\subseteq \to \mathcal{L}$ is continuous. We also assume that all the basic operations on objects ($\otimes$, $\oplus$, $(\_)^\bot$ and $!\_$) are continuous functors[4] from $\mathcal{L}_\subseteq$ to itself and that the following equations hold: if $\phi_i \in \mathcal{L}_\subseteq(X_i, Y_i)$ for $i = 1, 2$ then $(\phi_1 \otimes \phi_2)^- = \phi_1^- \otimes \phi_2^-$, $(\phi_1 \oplus \phi_2)^- = \phi_1^- \oplus \phi_2^-$, $(\phi_1 \otimes \phi_2)^+ = \phi_1^+ \otimes \phi_2^+$, $(\phi_1 \oplus \phi_2)^+ = \phi_1^+ \oplus \phi_2^+$, and if $\phi \in \mathcal{L}_\subseteq(X, Y)$ we assume that $(\phi^\bot)^- = (\phi^+)^\bot$, $(\phi^\bot)^+ = (\phi^-)^\bot$ and $(!\phi)^- = !(\phi^-)$, $(!\phi)^+ = !(\phi^+)$.

Remark 3. Many models of LL satisfy these additional assumptions (most of them do actually): relational model, Scott model, (hyper-

[4] Notice that $(\_)^\bot$ is a *covariant* endofunctor on $\mathcal{L}_\subseteq$: if $\phi \in \mathcal{L}_\subseteq(X, Y)$ then $\phi^\bot \in \mathcal{L}_\subseteq(X^\bot, Y^\bot)$.

)coherence space model, all models based on Indexed LL [9], probabilistic coherence space model [13].

3.2 Denotational semantics of bang calculus

In particular, we can consider the operation $\Phi : \mathcal{L}_\subseteq \to \mathcal{L}_\subseteq$ defined on objects by $\Phi(X) = !X \;\&\; (!X \multimap X)$ and similarly on morphisms. As a composition of continuous functors, Φ is a continuous functor and therefore the family $(\Phi^n(0))_{n\in\mathbb{N}}$ is directed in $\mathcal{L}_\subseteq$ and its colimit $\mathcal{U}$ satisfies $\mathcal{U} \simeq \Phi(\mathcal{U})$; we assume that this isomorphism is an equality of objects (this will be the case in the relational model), therefore $\mathcal{U} = !\mathcal{U} \;\&\; (!\mathcal{U} \multimap \mathcal{U})$.

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

If T is a variable then $T = x_i$ for some $1 \le i \le n$, assume that $i = n$ to simplify notations. Let $\mathsf{w} \in \mathcal{L}((!\mathcal{U})^{\otimes(n-1)}, 1)$ be the weakening morphism (indeed, $(!\mathcal{U})^{\otimes n}$ is a !-coalgebra, see [19]), then we have $\mathsf{w} \otimes \mathsf{id}_{!\mathcal{U}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U})$ (keeping implicit the monoidality isos $1 \otimes !\mathcal{U} \simeq !\mathcal{U}$) and $0_{(!\mathcal{U})^{\otimes n}, !\mathcal{U}\multimap\mathcal{U}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \multimap \mathcal{U})$. So, we set

$$[\![x_n]\!]_{\vec{x}} = \langle \mathsf{w} \otimes \mathsf{id}_{!\mathcal{U}}, 0_{(!\mathcal{U})^{\otimes n}, !\mathcal{U}\multimap\mathcal{U}} \rangle \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \;\&\; (!\mathcal{U} \multimap \mathcal{U})) = \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})\,.$$

If $T = \lambda y\, S$ (we can assume $y \neq x_i$ for all $1 \le i \le n$) then by inductive hypothesis we have $[\![S]\!]_{\vec{x},y} \in \mathcal{L}((!\mathcal{U})^{\otimes n} \otimes !\mathcal{U}, \mathcal{U})$ whence $\mathsf{cur}\,([\![S]\!]_{\vec{x},y}) \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \multimap \mathcal{U})$, so we set

$$[\![\lambda y\, S]\!]_{\vec{x}} = \langle 0_{(!\mathcal{U})^{\otimes n}, !\mathcal{U}}, \mathsf{cur}\,([\![S]\!]_{\vec{x},y}) \rangle \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \;\&\; (!\mathcal{U} \multimap \mathcal{U})) = \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})\,.$$

If $T = \langle S \rangle R$, by inductive hypothesis we have $[\![S]\!]_{\vec{x}}, [\![R]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})$ and, since $\mathcal{U} = !\mathcal{U} \;\&\; (!\mathcal{U} \multimap \mathcal{U})$, $\mathsf{pr}_1 [\![R]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U})$ and $\mathsf{pr}_2\, [\![S]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \multimap \mathcal{U})$. Let $\mathsf{c} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, (!\mathcal{U})^{\otimes n} \otimes (!\mathcal{U})^{\otimes n})$ be the contraction morphism, we set $[\![\langle S \rangle R]\!]_{\vec{x}} = \mathsf{ev}\ (\mathsf{pr}_2\, [\![S]\!]_{\vec{x}} \otimes \mathsf{pr}_1\, [\![R]\!]_{\vec{x}})\ \mathsf{c} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})$.

If $T = S^!$, by inductive hypothesis $[\![S]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})$, so $([\![S]\!]_{\vec{x}})^! = !([\![S]\!]_{\vec{x}})\ \mathsf{h} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U})$ where $\mathsf{h} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !((!\mathcal{U})^{\otimes n}))$ is the coalgebra structure map of $(!\mathcal{U})^{\otimes n}$. Then we have $0_{(!\mathcal{U})^{\otimes n}, !\mathcal{U}\multimap\mathcal{U}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \multimap \mathcal{U})$ and we set

$$[\![S^!]\!]_{\vec{x}} = \langle ([\![S]\!]_{\vec{x}})^!, 0_{(!\mathcal{U})^{\otimes n}, !\mathcal{U}\multimap\mathcal{U}} \rangle \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U} \;\&\; (!\mathcal{U} \multimap \mathcal{U})) = \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})\,.$$

Last if $T = \mathrm{der}\, S$, by inductive hypothesis we have $[\![S]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})$ hence $\mathsf{pr}_1\, [\![S]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U})$ and we set $[\![\mathrm{der}\, S]\!]_{\vec{x}} = \mathsf{der}\ \mathsf{pr}_1\, [\![S]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})$.

Theorem 1 (Invariance)**.** *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 $T \to_{\mathsf{b}\sigma} S$ then $[\![T]\!]_{\vec{x}} = [\![S]\!]_{\vec{x}}$.*

Proof (Sketch). The proof that $T \to_\sigma S$ implies $[\![T]\!]_{\vec{x}} = [\![S]\!]_{\vec{x}}$ relies on the commutative diagrams holding since the category $\mathcal{L}$ is a model of LL. The proof that $T \to_{\mathsf{b}} S$ implies $[\![T]\!]_{\vec{x}} = [\![S]\!]_{\vec{x}}$ is easy and based on the following substitivity property. Let T be a term, V be a value, $\vec{x}$ be a repetition-free list of variables containing all the free variables of V, and x be a variable which does not occur in $\vec{x}$ and such that $\vec{x}, x$ contains all the free variables of T. Then we know that $[\![T]\!]_{\vec{x},x} \in \mathcal{L}((!\mathcal{U})^{\otimes n} \otimes !\mathcal{U}, \mathcal{U})$ (where $n \in \mathbb{N}$ is the length of $\vec{x}$), $[\![V]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, !\mathcal{U})$ and that $[\![T\{V/x\}]\!]_{\vec{x}} \in \mathcal{L}((!\mathcal{U})^{\otimes n}, \mathcal{U})$. A simple induction on T shows that $[\![T\{V/x\}]\!]_{\vec{x}} = [\![T]\!]_{\vec{x},x}\ \left((!\mathcal{U})^{\otimes n} \otimes [\![V]\!]_{\vec{x}}\right)\ \mathsf{c}$ where c is the contraction morphism mentioned above. The proof uses crucially the fact that $[\![V]\!]_{\vec{x}}$ is a coalgebra morphism, see [19]. □

The general notion of denotational model for the bang calculus presented here and obtained from any denotational model $\mathcal{L}$ of LL (with the assumptions detailed in §3.1) is a particular case of Moggi's semantics of computations based on monads [36, 37], if one keeps in mind that the functor "!" defines a strong monad on the Kleisli category $\mathcal{L}_!$ of $\mathcal{L}$.

3.3 Relational semantics: the model of sets and relations

We focus from now on a particular instance of the categorical structure presented above. We take for $\mathcal{L}$ the category $\mathbf{Rel}$ of sets and relations: the objects of $\mathbf{Rel}$ are sets, and the set of morphisms from the set X to the set Y is $\mathbf{Rel}(X,Y) = \mathcal{P}(X \times Y)$ (composition is the usual composition of relations). This category is a well known model of LL, setting $1 = \{*\}$ (a singleton set), $X \otimes Y = X \times Y$ (with obvious monoidal isomorphisms), $X \multimap Y = X \times Y$, $\bot = 1$ so that $X^\bot = X$ (up to obvious iso). As to the additive structure, we take $\top = 0 = \emptyset$ and $X \;\&\; Y = X \oplus Y = (\{1\} \times X) \cup (\{2\} \times Y)$; in the special case where $X \cap Y = \emptyset$, we set $X \;\&\; Y = X \oplus Y = X \cup Y$ (no ambiguity arises with the previous definition, up to obvious iso). Then one checks easily that $0_{X,Y} = \emptyset$.

Notation. We use $[a_1, \dots, a_n]$ for the finite multiset made of $a_1, \dots, a_n$ (with $n \in \mathbb{N}$) taking multiplicities into account; so, $[\,]$ stands for the empty multiset. Given a finite family $(a_i)_{i\in I}$ and a predicate P on I, $[a_i \mid P(i)]$ denotes the finite multiset made of the a_i's such that $P(i)$ holds, taking multiplicities into account. We use $\uplus$ for multiset sum. The set of finite multisets over a set A is denoted by $\mathcal{M}_\mathrm{f}(A)$.

The exponential is defined by $!X = \mathcal{M}_\mathrm{f}(X)$ and, given $f \in \mathbf{Rel}(X,Y)$, we set $!f = \{([a_1, \dots, a_n], [b_1, \dots, b_n]) \mid \forall i\ (a_i, b_i) \in f\}$ and this operation is easily seen to be functorial. Its comonad structure is given by the natural transformations $\mathsf{der}_X = \{([a], a) \mid a \in X\} \in \mathbf{Rel}(!X, X)$ and $\mathsf{dig}_X = \{(m_1 \uplus \dots \uplus m_k, [m_1, \dots, m_k]) \mid m_1, \dots, m_k \in !X\} \in \mathbf{Rel}(!X, !!X)$. The Seely isomorphisms have obvious definitions.

The category $\mathbf{Rel}_\subseteq$ is such that $\mathbf{Rel}_\subseteq(X,Y)$ is a singleton $\{\eta_{X,Y}\}$ if $X \subseteq Y$ and is empty otherwise, so $\mathbf{Rel}_\subseteq$ is a partially ordered class with least element $\emptyset$. Clearly, each connective of LL defines a continuous functional on this partially ordered class, so the operation $\Phi(X) = !X \;\&\; (!X \multimap X)$ has a least fixpoint $\mathcal{U}$ (see §3.2). Concretely, $\mathcal{U}$ is defined as $\mathcal{U} := \bigcup_{n\in\mathbb{N}} \mathcal{U}_n$, where

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

Clearly, $\mathcal{U} = \mathcal{M}_\mathrm{f}(\mathcal{U}) \cup (\mathcal{M}_\mathrm{f}(\mathcal{U}) \times \mathcal{U}) = !\mathcal{U} \;\&\; (!\mathcal{U} \multimap \mathcal{U})$ with $\mathcal{M}_\mathrm{f}(\mathcal{U}) \cap (\mathcal{M}_\mathrm{f}(\mathcal{U}) \times \mathcal{U}) = \emptyset$. So, given a term T and a repetition-free list $\vec{x}$ of variables containing all the free variables of T, one has $[\![T]\!]_{\vec{x}} \subseteq (\mathcal{M}_\mathrm{f}(\mathcal{U}))^n \times \mathcal{U}$, where n is the length of $\vec{x}$: via Thm. 1, it is immediate to check if T is $\mathsf{b}\sigma$-equivalent to a value (resp. an abstraction), then $[\![T]\!]_{\vec{x}} \subseteq (\mathcal{M}_\mathrm{f}(\mathcal{U}))^n \times \mathcal{M}_\mathrm{f}(\mathcal{U})$ (resp. $[\![T]\!]_{\vec{x}} \subseteq (\mathcal{M}_\mathrm{f}(\mathcal{U}))^n \times (\mathcal{M}_\mathrm{f}(\mathcal{U}) \times \mathcal{U})$). Intuitively, the interpretation of values (resp. abstractions) is included $\mathcal{M}_\mathrm{f}(\mathcal{U})$ (resp. $\mathcal{M}_\mathrm{f}(\mathcal{U}) \times \mathcal{U}$).

The semantics of terms in this relational model can be presented as a non-idempotent type system (Prop. 4): the relational semantics of a term T turns out to be just the set of conclusions of the type derivations of T. In this typing system, judgments have the shape $\Gamma \vdash T : a$ where T is a term, $a \in \mathcal{U}$ and Γ is a *context* (that is, a partial function from $\mathcal{V}ar$ to $\mathcal{M}_\mathrm{f}(\mathcal{U})$ whose domain is finite and includes $\mathsf{fv}(T)$); if moreover T is a value, then $a \in \mathcal{M}_\mathrm{f}(\mathcal{U})$. The sum of contexts $\Gamma \uplus \Delta$ is defined pointwise using the multiset sum, when Γ and Δ have the same domain: if $x \in \mathsf{dom}(\Gamma) = \mathsf{dom}(\Delta)$, then $(\Gamma \uplus \Delta)(x) = \Gamma(x) \uplus \Delta(x)$. A context Γ such that $\mathsf{dom}(\Gamma) = \{x_1, \dots, x_n\}$ with $x_i \neq x_j$ and $\Gamma(x_i) = m_i$ for any $1 \le i \neq j \le n$ is often written as $\Gamma = x_1 : m_1, \dots, x_n : m_n$. The typing rules of this system are in Fig. 1.

$$\frac{}{\Gamma_0, x:m \vdash x:m}\,\mathsf{ax} \qquad \frac{\Gamma, x:m \vdash T:a}{\Gamma \vdash \lambda x\, T:(m,a)}\,\lambda \qquad \frac{\Gamma \vdash T:[a]}{\Gamma \vdash \mathrm{der}\, T:a}\,\mathrm{der}$$

$$\frac{\Gamma_1 \vdash T:a_1 \quad \overset{n\in\mathbb{N}}{\cdots} \quad \Gamma_n \vdash T:a_n}{\Gamma_0 \uplus \cdots \uplus \Gamma_n \vdash T^!:[a_1,\ldots,a_n]}\,! \qquad \frac{\Gamma \vdash T:(m,a) \quad \Gamma' \vdash S:m}{\Gamma \uplus \Gamma' \vdash \langle T\rangle S:a}\,@$$

Figure 1. Non-idempotent intersection type system for the bang calculus, where $m \in \mathcal{M}_{\mathrm{f}}(\mathcal{U})$ and $a \in \mathcal{U}$. In the rules ax and !, $\Gamma_0 = x_1:[\,],\ldots,x_n:[\,]$ and moreover, for the rule !, $\mathsf{fv}(T) \subseteq \mathsf{dom}(\Gamma_0)$.

The type system in Fig. 1 can be seen as a non-idempotent intersection type system. A finite multiset $[a_1,\ldots,a_n]$ can naturally be seen as a type $a_1 \wedge \cdots \wedge a_n$, where the connective $\wedge$ is associative and commutative but not idempotent ($a \wedge a \neq a$), and has $[\,]$ as its neutral element. Intuitively, a typed term $T:[a,a,b]$ means that T can be used exactly twice as data of type a and exactly once as data of type b (see the rule ! in Fig. 1, introducing the only non-linear construct of the bang calculus). A pair (m,a) can be seen as an arrow type $m \to a$ or more precisely a linear implication $m \multimap a$. The rule @ in Fig. 1 introduces the linear application. For further investigations about the relationship between relational model and non-idempotent intersection types, see [38].

Proposition 4 (Relational semantics via type derivations)**.** *Let $T \in !\Lambda$ and $\vec{x} = (x_1,\ldots,x_n)$ be a repetition-free list of variables which contains all the free variables of T, with $n \in \mathbb{N}$. Let $\vec{m} = (m_1,\ldots,m_n) \in (\mathcal{M}_{\mathrm{f}}(\mathcal{U}))^n$ and $a \in \mathcal{U}$. Then, $(\vec{m},a) \in [\![T]\!]_{\vec{x}}$ iff the type judgment $x_1:m_1,\ldots,x_n:m_n \vdash T:a$ is derivable.*

Proof. By straightforward induction on $T \in !\Lambda$. □

Using Prop. 4, one checks easily that $[\![\langle\Delta\rangle\Delta']\!] = \emptyset = [\![\langle\Delta'\rangle\Delta'^!]\!]$ (see Ex. 1) and $[\![T]\!]_x = [\![S]\!]_x = [\![R]\!]_{y,z} = [\![Q]\!]_{x,y,z} = [\![P]\!]_x = \emptyset$, where T, S, R, Q, P are defined as in Eq. (3)-(4).

It is worth noting that the type system in Fig. 1 is a generalization of System R [14] for CBN λ-calculus and the type system used in [17] for CBV λ-calculus. As a consequence, using Prop. 4, it can be shown that the relational semantics of the bang calculus contains the relational semantics of both CBN and CBV λ-calculi. More precisely, for any λ-term T and any repetition-free list of variables $\vec{x} = (x_1,\ldots,x_n)$ containing all the free variables of T, one has

$$[\![T^{\mathsf{cbn}}]\!]_{\vec{x}} = [\![T]\!]^{\mathsf{cbn}}_{\vec{x}} \qquad [\![T^{\mathsf{cbv}}]\!]_{\vec{x}} = [\![T]\!]^{\mathsf{cbv}}_{\vec{x}}$$

where $[\![T]\!]^{\mathsf{cbn}}_{\vec{x}}$ (resp. $[\![T]\!]^{\mathsf{cbv}}_{\vec{x}}$) is the interpretation of the λ-term T in the relational model for CBN (resp. CBV) λ-calculus defined in [14] (resp. [10, 17]) using $\{[\,]\}$ as set of atoms.

4. Resource bang calculus and Taylor expansion

We introduce the *resource bang calculus*, a resource-sensitive version of the bang calculus whose terms can be used to denote typing derivations in the typing system described above. The set $!\Lambda^{\mathsf{res}}$ of *resource terms* is defined by induction as follows (note the absence of boxes, replaced by bags):

$$(!\Lambda^{\mathsf{res}}) \qquad t,s,r ::= x \mid [t_1,\ldots,t_n] \mid \lambda x\, t \mid \langle t\rangle s \mid \mathrm{der}\, t$$

where $[t_1,\ldots,t_n]$ is a multiset of resource terms (with $n \in \mathbb{N}$), called *bag*. The set of free variables of a resource 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 the bags $b_1,\ldots,b_n$ and the pairwise distinct variables $x_1,\ldots,x_n$, $t\{b_1/x_1,\ldots,b_n/x_n\}$ denotes the term obtained by the *capture-avoiding simultaneous substitution* of b_i for each free occurrence of x_i in the term t (for all $1 \leq i \leq n$). For any resource term t and variable x, the number of free occurrences of x in t is denoted by $|t|_x$.

Notation. Let $n, m \in \mathbb{N}$: we set $\overline{n} := \{1,\ldots,n\}$ (in particular, $\overline{0} = \emptyset$), and $\overline{n}^{\,\overline{m}}$ is the (finite) set of functions from $\overline{m}$ to $\overline{n}$.

The set of finite subsets of $!\Lambda^{\mathsf{res}}$ is denoted by $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$ and $t, s, r, \ldots$ range over it. As $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$ is isomorphic to the free module generated by $!\Lambda^{\mathsf{res}}$ over the boolean semiring $\{0,1\}$ with $1+1=1$, we use algebraic notations for operations on $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$, *e.g.* $+$ stands for set union, 0 stands for $\emptyset$, a singleton set is confused with its element. To avoid ambiguity, $*$ denotes the multiset sum of bags: *e.g.*, given $t,s,r \in !\Lambda^{\mathsf{res}}$, $[t,s]+[r] \neq [t,s]*[r] = [t,s,r]$.

The above syntactic constructs of the resource bang calculus are extended to $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$ by (multi-)linearity: *e.g.*, given $t := \sum_{i\in I} t_i$ and $s := \sum_{j\in J} s_j$ (the summands being resource terms), we set $[t,s] := \sum_{i\in I, j\in J}[t_i,s_j]$, and $\lambda x\, t := \sum_{i\in I} \lambda x\, t_i$. Note that $*$ distributes over $+$: the multiset sum $*$ of bags behaves like a product.

Resource contexts c (with exactly one hole $(\!|\cdot|\!)$) are defined inductively via the following grammar:

$$\mathsf{c} ::= (\!|\cdot|\!) \mid \lambda x\, \mathsf{c} \mid \mathsf{c}M \mid M\mathsf{c} \mid \mathrm{der}\, \mathsf{c} \mid [\mathsf{c}, t_1, \ldots, t_n] \quad (n \in \mathbb{N})$$

We use $\mathsf{c}(\!|t|\!)$ for the resource term obtained by the capture-allowing substitution of the resource term t for the hole $(\!|\cdot|\!)$ in the resource context c. We set $\mathsf{c}(\!|t+s|\!) := \mathsf{c}(\!|t|\!) + \mathsf{c}(\!|s|\!)$.

Consider the following binary relations (*root-steps*) from $!\Lambda^{\mathsf{res}}$ to $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$ (for any $t,s,r,s_1,\ldots,s_n \in !\Lambda^{\mathsf{res}}$, $y \in \mathcal{V}ar$ and $n \in \mathbb{N}$):

$$\langle\lambda x\, t\rangle[s_1,\ldots,s_n] \mapsto_{\mathsf{v}} \sum_{\mathsf{f}\in\overline{k}^{\,\overline{n}}} t'\{m_1/x_1,\ldots,m_k/x_k\}$$

$$\langle\lambda x\, t\rangle y \mapsto_{\mathsf{ax}} t\{y/x\} \qquad \mathrm{der}\,[t_1,\ldots,t_n] \mapsto_{[\cdot]} \begin{cases} t_1 & \text{if } n=1 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} \langle\langle\lambda x\, t\rangle r\rangle s &\mapsto_{\sigma_1} \langle\lambda x\langle t\rangle s\rangle r && x \notin \mathsf{fv}(s) \\ \langle\lambda y\, \lambda x\, t\rangle s &\mapsto_{\sigma_2} \lambda x\langle\lambda y\, t\rangle s && x \notin \mathsf{fv}(s)\cup\{y\} \\ \langle t\rangle(\langle\lambda x\, s\rangle r) &\mapsto_{\sigma_3} \langle\lambda x\langle t\rangle s\rangle r && x \notin \mathsf{fv}(t) \\ \mathrm{der}(\langle\lambda x\, t\rangle s) &\mapsto_{\sigma_4} \langle\lambda x\, \mathrm{der}\, t\rangle s \end{aligned}$$

$$\mapsto_{\mathsf{b}} := \mapsto_{\mathsf{v}} \cup \mapsto_{\mathsf{ax}} \cup \mapsto_{[\cdot]}$$

$$\mapsto_{\sigma} := \mapsto_{\sigma_1} \cup \mapsto_{\sigma_2} \cup \mapsto_{\sigma_3} \cup \mapsto_{\sigma_4} \qquad \mapsto_{\mathsf{b}\sigma} := \mapsto_{\mathsf{b}} \cup \mapsto_{\sigma}$$

where, in the definition of $\mapsto_{\mathsf{v}}$, t' is the *x-linearization* of t, *i.e.* the resource term obtained from t by replacing the $k = |t|_x$ free occurrences of x in t by the fresh pairwise distinct variables $x_1,\ldots,x_k$, and $m_j := [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = j]$ for any $1 \leq j \leq k$. This means that the step $\mapsto_{\mathsf{v}}$ from $!\Lambda^{\mathsf{res}}$ to $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$ cannot duplicate nor erase any s_i in each summand on the right-hand-side of $\mapsto_{\mathsf{v}}$.

For any $\mathsf{r} \in \{\mathsf{v}, \mathsf{ax}, [\cdot], \mathsf{b}, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma, \mathsf{b}\sigma\}$, the *r-reduction* $\to_{\mathsf{r}}$ is the closure under contexts of the binary relation $\mapsto_{\mathsf{r}}$ from $!\Lambda^{\mathsf{res}}$ to $\overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$, *i.e.*

$$t \to_{\mathsf{r}} s \Leftrightarrow \exists\, \mathsf{c}, \exists\, r,q \in !\Lambda : t = \mathsf{c}(\!|r|\!),\ s = \mathsf{c}(\!|q|\!),\ r \mapsto_{\mathsf{r}} q \qquad (5)$$

Example 3. $\langle\lambda x\, \mathrm{der}\, x\rangle[y,z] \to_{\mathsf{v}} \mathrm{der}\,[y,z] \to_{[\cdot]} 0$. Moreover, $\langle\lambda x\langle\mathrm{der}\, x\rangle x\rangle[y,z] \to_{[\cdot]} \langle\mathrm{der}\,[y,z]\rangle[\,]+\langle\mathrm{der}\,[\,]\rangle[y,z]+\langle\mathrm{der}\,[z]\rangle[y] + \langle\mathrm{der}\,[y]\rangle[z]$ where $\langle\mathrm{der}\,[y,z]\rangle[\,] \to_{[\cdot]} 0 \,_{[\cdot]}\!\leftarrow \langle\mathrm{der}\,[\,]\rangle[y,z]$ and $\langle\mathrm{der}\,[z]\rangle[y] \to_{[\cdot]} \langle z\rangle[y]$ and $\langle\mathrm{der}\,[y]\rangle[z] \to_{[\cdot]} \langle y\rangle[z]$.

Note the different behavior of $\langle\lambda x\,\langle x\rangle x\rangle y \to_{\mathsf{ax}} \langle y\rangle y$ and $\langle\lambda x\,\langle x\rangle x\rangle[y] \to_{\mathsf{v}} \langle[y]\rangle[\,] + \langle[\,]\rangle[y]$.

Just as the bang calculus is related to LL, the resource bang calculus is inspired by differential LL without boxes [21]. It can be shown that both call-by-name [21, 22] and call-by-value [10, 17] resource calculi can be embedded in the resource bang calculus.

Lemma 3 (Strong normalization)**.** *There is no infinite sequence $(t_i)_{i\in\mathbb{N}}$ of resource terms such that: for any $i \in \mathbb{N}$ there is $s \in \overline{2}\langle!\Lambda^{\mathsf{res}}\rangle$ with $t_{i+1} \in s$ and $t_i \to_{\mathsf{b}\sigma} s$.* Proof p. 15

Proof. For any $t \in !\Lambda^{\mathsf{res}}$, let $\|t\| := (|t|_{\mathrm{der},\lambda}, \mathsf{s}(t), \#(t))$, where $|t|_{\mathrm{der},\lambda}$ is the number of occurrences of λ and der in t, and $\mathsf{s}(t)$ and

$$\frac{}{\Gamma_0, x:m \vdash x:m}\,\mathsf{ax} \qquad \frac{\Gamma, x:m \vdash t:a}{\Gamma \vdash \lambda x\, t:(m,a)}\,\lambda \qquad \frac{\Gamma \vdash t:[a]}{\Gamma \vdash \mathrm{der}\, t:a}\,\mathrm{der}$$

$$\frac{\Gamma_1 \vdash t:a_1 \quad \overset{n\in\mathbb{N}}{\cdots} \quad \Gamma_n \vdash t:a_n}{\Gamma_0 \uplus \ldots \uplus \Gamma_n \vdash [t_1,\ldots,t_n]:[a_1,\ldots,a_n]}\,[\cdot] \qquad \frac{\Gamma \vdash t:(m,a) \quad \Gamma' \vdash s:m}{\Gamma \uplus \Gamma' \vdash \langle t\rangle s:a}\,@$$

Figure 2. Non-idempotent intersection type system for the resource bang calculus, where $m \in \mathcal{M}_{\mathrm{f}}(\mathcal{U})$ and $a \in \mathcal{U}$. In the rules ax and $[\cdot]$, $\Gamma_0 = x_1:[],\ldots,x_n:[]$; in the rule $[\cdot]$, $\mathsf{fv}(t) \subseteq \mathsf{dom}(\Gamma_0)$.

$\#(t)$ are defined as follows:

$$\begin{aligned}
\mathsf{s}(x) &:= 1\\
\mathsf{s}(\lambda x\, t) &:= \mathsf{s}(t) + 1\\
\mathsf{s}(\langle t\rangle s) &:= \mathsf{s}(t) + \mathsf{s}(s)\\
\mathsf{s}(\mathrm{der}\, t) &:= \mathsf{s}(t)\\
\mathsf{s}([t_1,\ldots,t_n]) &:= \mathsf{s}(t_1) + \cdots + \mathsf{s}(t_n)
\end{aligned}$$

$$\begin{aligned}
\#(x) &:= 1\\
\#(\lambda x\, t) &:= \#(t) + \mathsf{s}(t)\\
\#(\langle t\rangle s) &:= \#(t) + \#(s) + 2\,\mathsf{s}(t)\,\mathsf{s}(s) - 1\\
\#(\mathrm{der}\, t) &:= \#(t) + \mathsf{s}(t)\\
\#([t_1,\ldots,t_n]) &:= \#(t_1) + \cdots + \#(t_n)
\end{aligned}$$

It is easy to prove that: if $t \to_{\mathsf{b}} s$ and $t' \in s$, then $|t|_{\mathrm{der},\lambda} > |t'|_{\mathrm{der},\lambda}$; if $t \to_{\sigma} s$ and $t' \in s$, then $|t|_{\mathrm{der},\lambda} = |t'|_{\mathrm{der},\lambda}$, $\mathsf{s}(t) = \mathsf{s}(t')$ and $\#(t) > \#(t')$. Therefore, if $t \to_{\mathsf{b}\sigma} s$ and $t' \in s$, then $\|t\| > \|t'\|$ according to the lexicographical order on $\mathbb{N}^3$. □

The non-idempotent type system in Fig. 1 for the bang calculus has been adapted in Fig. 2 to the resource bang calculus, providing a denotational model to it. Along the lines of Prop. 4, given a resource term t and a repetition-free list $\vec{x} = (x_1,\ldots,x_n)$ of variables containing all the free variables of t, the *relational semantics* of t is

$$[\![t]\!]_{\vec{x}} := \big\{((m_1,\ldots,m_n),a) \in (\mathcal{M}_{\mathrm{f}}(\mathcal{U}))^n \times \mathcal{U} \mid x_1:m_1,\ldots,x_n:m_n \vdash t:a \text{ is derivable}\big\}.$$

Given a repetition-free list $\vec{x}$ of variables containing all free variables of the resource terms t and s, we set $[\![t+s]\!]_{\vec{x}} := [\![t]\!]_{\vec{x}} \cup [\![s]\!]_{\vec{x}}$. So, sets of resource terms are interpreted as the union of the interpretations of their elements.

Proof p. 15 **Lemma 4** (Substitution). *Let $n \in \mathbb{N}$, $t, s_1, \ldots, s_n \in !\Lambda^{\mathsf{res}}$ and $a, b_1, \ldots, b_n \in \mathcal{U}$. Let $x \in \mathcal{V}ar$ with $|t|_x = k$ and $|s_i|_x = 0$ for all $1 \le i \le n$. If $\Gamma_0, x:[b_1,\ldots,b_n] \vdash t:a$ is derivable and, for all $1 \le i \le n$, $\Gamma_i \vdash s_i:b_i$ is derivable with $\mathsf{dom}(\Gamma_0) = \cdots = \mathsf{dom}(\Gamma_n)$, then for any x-linearization t' of t there is $\mathsf{f} \in \overline{k}^{\overline{n}}$ such that*

$$\Gamma_0 \uplus \cdots \uplus \Gamma_n \vdash t'\{m_1/x_1,\ldots,m_k/x_k\}:a$$

is derivable, with $m_j := [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = j]$ for any $1 \le j \le k$.

Lemma 5 (Subject reduction). *Let $t, s \in !\Lambda^{\mathsf{res}}$ and $\vec{x}$ be a repetition-free list of variables which contains all free variables of t and s. If $t \to_{\mathsf{b}\sigma} s$ then $[\![t]\!]_{\vec{x}} \subseteq [\![s]\!]_{\vec{x}}$.*

Proof (Sketch). First, observe that $\to_{\mathsf{b}\sigma} = \to_{\mathsf{b}} \cup \to_{\sigma}$.

The proof that $t \to_{\mathsf{b}} s$ implies $[\![t]\!]_{\vec{x}} \subseteq [\![s]\!]_{\vec{x}}$ (by induction on the definition of $t \to_{\mathsf{b}} s$) follows easily from Lemma 4 (for the root-step $\mapsto_{\mathsf{v}}$) and from the induction hypothesis (for the other cases); it is trivial for the root-step of $\mapsto_{\mathsf{ax}}$.

The proof that $t \to_{\sigma} s$ implies $[\![t]\!]_{\vec{x}} \subseteq [\![s]\!]_{\vec{x}}$ is left to the reader: for any $((m_1,\ldots,m_n),a) \in [\![t]\!]_{\vec{x}}$, it consists only of a local rearrangement of some pieces of the derivation of $x_1:m_1,\ldots,x_n:m_n \vdash t:a$. □

Actually, it can be shown (but this out of our scope) that if $t \to_{\mathsf{b}\sigma} s$ then $[\![t]\!]_{\vec{x}} = [\![s]\!]_{\vec{x}}$, and this is true not only for the relational semantics but also for any model of the bang calculus coming from a model of differential LL [18].

Taylor expansion Taylor expansion is a notion relating bang calculus and resource bang calculus both semantically (via relational model) and operationally (via reduction). Formally, the *Taylor expansion $\mathcal{T}(S)$ of a term S* is defined by induction on $S \in !\Lambda$ as follows:

$$\begin{aligned}
\mathcal{T}(x) &:= \{x\}\\
\mathcal{T}(\mathrm{der}\, S) &:= \{\mathrm{der}\, s \mid s \in \mathcal{T}(S)\}\\
\mathcal{T}(\lambda x\, S) &:= \{\lambda x\, s \mid s \in \mathcal{T}(S)\}\\
\mathcal{T}(\langle S\rangle R) &:= \{\langle s\rangle r \mid s \in \mathcal{T}(S),\ r \in \mathcal{T}(R)\}\\
\mathcal{T}(S^!) &:= \{[s_1,\ldots,s_n] \mid n \in \mathbb{N},\ s_1,\ldots,s_n \in \mathcal{T}(S)\}
\end{aligned}$$

Our definition of Taylor expansion is a generalization of analogous deeply studied notions for CBN [20, 22, 33] and CBV [10, 17] λ-calculi. In particular, if for any λ-term S we denote by $\mathcal{T}^{\mathsf{cbn}}(S)$ and $\mathcal{T}^{\mathsf{cbv}}(S)$ the CBN and CBV Taylor expansions of S, respectively, then it is easy to show by straightforward induction on S that $\mathcal{T}^{\mathsf{cbn}}(S) = \mathcal{T}(S^{\mathsf{cbn}})$ and $\mathcal{T}^{\mathsf{cbv}}(S) = \mathcal{T}(S^{\mathsf{cbv}})$.

Observe that, for any $S \in !\Lambda$, $\mathcal{T}(S)$ is infinite iff S has a subterm of shape $T^!$. Besides, if $s \in \mathcal{T}(S)$ then $\mathsf{fv}(s) \subseteq \mathsf{fv}(S)$: thus, a repetition-free list of variables containing all the free variables of S contains also all the free variables of s.

Lemma 6 (Simulating reduction via Taylor expansion). *For all $T, S \in !\Lambda$ and $t \in !\Lambda^{\mathsf{res}}$, if $T \to_{\mathsf{b}\sigma_{\mathsf{w}}} S$ and $t \in \mathcal{T}(T)$, then $t \to_{\mathsf{b}\sigma} s \subseteq \mathcal{T}(S)$ for some $s \in \overline{2}\langle !\Lambda^{\mathsf{res}}\rangle$.* Proof p. 16

One has to be careful when using Lemma 6 because, using its notations, nothing prevents s (which is a sum, *i.e.* a set, of resource terms) from being 0 *i.e.* the empty set.

Lemma 6 is false if we replace "$T \to_{\mathsf{b}\sigma_{\mathsf{w}}} S$" with "$T \to_{\mathsf{b}\sigma} S$" in the hypothesis, *e.g.* take $T := (\langle \lambda y\, y\rangle z)^! \to_{\mathsf{b}\sigma} z^! =: S$ and $t := [\,] \in \mathcal{T}(T)$, then t is $\mathsf{b}\sigma$-normal.

Lemma 7 (Relational semantics via Taylor expansion). *Let $S \in !\Lambda$ and $\vec{x} = (x_1,\ldots,x_n)$ be a repetition-free list of variables (with $n \in \mathbb{N}$) containing all free variables of S. Then, $[\![S]\!]_{\vec{x}} = \bigcup_{s\in\mathcal{T}(S)} [\![s]\!]_{\vec{x}}$.* Proof p. 16

Actually, it can be shown that Lemma 7 holds not only for the relational semantics but also for any model of the bang calculus coming from a model of differential LL which satisfies a version of the Taylor formula [17, 18].

5. Adequacy of Rel for the bang calculus

In this section we prove the computational adequacy of the relational semantics for the extended bang calculus: the interpretation of a term is non empty in the relational model if and only if all the $\mathsf{b}\sigma_{\mathsf{w}}$-reductions starting from this term terminates without clashes at depth 0 (Thm. 2). We will show also some consequences of this theorem, about the relationship between normalization and strong normalization for the $\mathsf{b}\sigma_{\mathsf{w}}$-reduction (Cor. 1), the adequacy of the relational semantics with respect to the notion of observational equivalence for the extended bang calculus (Cor. 3), and the soundness of the $\mathsf{b}\sigma$-equivalence with respect this notion of observational equivalence (Cor. 4).

Thanks to the notion of Taylor expansion (in particular, Lemmas 6-7), we are able to prove the left-to-right part of the computational adequacy of the relational semantics for the (extended) bang

calculus (see Thm. 2.1 below) in a purely combinatorial way (along the lines of [10, 17]), without using a reducibility argument.

Actually, with a little more effort we can strengthen the computational adequacy result by proving also Thm. 2.2, a sort of converse of Thm. 2.1. First, we characterize syntactically the $\mathsf{b}\sigma_\mathsf{w}$-normal forms that are clash-free at depth 0. We define inductively the subsets $!\Lambda_\ell$, $!\Lambda_\mathsf{n}$ and $!\Lambda_\mathsf{d}$ (whose elements are denoted by L, N and D, respectively) of $!\Lambda$ as follows (for any $x \in \mathcal{V}ar$ and $V \in !\Lambda_\mathsf{v}$):

$$(!\Lambda_\mathsf{d}) \qquad D ::= \mathrm{der}\, x \mid \mathrm{der}\, D \mid \langle D\rangle V \mid \langle D\rangle D'$$
$$(!\Lambda_\mathsf{n}) \qquad N ::= V \mid D \mid \langle \lambda x\, N\rangle D$$
$$(!\Lambda_\ell) \qquad L ::= N \mid \lambda x\, L$$

Note that all terms in $!\Lambda_\mathsf{d}$ are not closed (they have a free "head derelicted variable"), moreover they have at least one occurrence of der and they are neither a value nor a β-like redex nor an abstraction. Clearly, $!\Lambda_\mathsf{d} \subsetneq !\Lambda_\mathsf{n}$ and $!\Lambda_\mathsf{v} \subsetneq !\Lambda_\mathsf{n} \subsetneq !\Lambda_\ell$, with $!\Lambda_\mathsf{d} \cap !\Lambda_\mathsf{v} = \emptyset$.

Lemma 8 (Syntactic characterization of clash-free at depth 0 $\mathsf{b}\sigma_\mathsf{w}$-normal forms)**.** *Let T be a term.*

1. *T is $\mathsf{b}\sigma_\mathsf{w}$-normal, clash-free at depth 0 and is neither a value nor a β-like redex nor an abstraction iff $T \in !\Lambda_\mathsf{d}$.*
2. *T is $\mathsf{b}\sigma_\mathsf{w}$-normal, clash-free at depth 0 and is not an abstraction iff $T \in !\Lambda_\mathsf{n}$.*
3. *T is $\mathsf{b}\sigma_\mathsf{w}$-normal and clash-free at depth 0 iff $T \in !\Lambda_\ell$.*

Proof. Statement (3) is an immediate consequence of statements (1)-(2). Let us prove statements (1)-(2).

1. (⇐): By straightforward induction on $T \in !\Lambda_\mathsf{d}$, recalling that values are $\mathsf{b}\sigma_\mathsf{w}$-normal since $\to_{\mathsf{b}\sigma_\mathsf{w}}$ does not reduce under !.
   (⇒): By induction on $T \in !\Lambda$. The cases where T is a variable, an abstraction or a box are impossible by hypothesis, hence T is a dereliction or an application.
   Suppose $T := \mathrm{der}\, S$: by hypothesis, S is $\mathsf{b}\sigma_\mathsf{w}$-normal and clash-free at depth 0; moreover, S is neither a β-like redex (otherwise T would not be σ_w-normal) nor an abstraction (otherwise T would not be clash-free at depth 0) nor a box (otherwise T would not be b_w-normal). If S is a variable, then $T \in !\Lambda_\mathsf{d}$ by definition of $!\Lambda_\mathsf{d}$. Otherwise S is not a value and then, by induction hypothesis, $S \in !\Lambda_\mathsf{d}$; thus, $T \in !\Lambda_\mathsf{d}$ by definition of $!\Lambda_\mathsf{d}$.
   Suppose $T := \langle S\rangle R$: by hypothesis, S and R are $\mathsf{b}\sigma_\mathsf{w}$-normal and clash-free at depth 0; moreover, S is neither a value (otherwise T would not be clash-free at depth 0) nor a β-like redex (otherwise T would not be σ_w-normal) nor an abstraction (otherwise T would be a β-like redex). Also, R is neither an abstraction (otherwise T would not be clash-free at depth 0) nor a β-like redex (otherwise T would not be σ_w-normal). By induction hypothesis, $S \in !\Lambda_\mathsf{d}$. If R is a value, then $T \in !\Lambda_\mathsf{d}$ by definition of $!\Lambda_\mathsf{d}$. Otherwise R is not a value and then, by induction hypothesis, $R \in !\Lambda_\mathsf{d}$; therefore, $T \in !\Lambda_\mathsf{d}$ by definition of $!\Lambda_\mathsf{d}$.
2. (⇐): By straightforward induction on $T \in !\Lambda_\mathsf{n}$, recalling that values are $\mathsf{b}\sigma_\mathsf{w}$-normal since $\to_{\mathsf{b}\sigma_\mathsf{w}}$ does not reduce under !, and using Lemma 8.1.
   (⇒): By induction on $T \in !\Lambda$.
   The case where T is an abstraction is impossible by hypothesis. If T is a value (i.e. a box or a variable), then $T \in !\Lambda_\mathsf{n}$ by definition of $!\Lambda_\mathsf{n}$.
   Suppose T is a β-like redex, *i.e.* $T := \langle \lambda x\, S\rangle R$: by hypothesis, S and R are $\mathsf{b}\sigma_\mathsf{w}$-normal and clash-free at depth 0; moreover, R is neither a value (otherwise T would not be b_w-normal) nor an abstraction (otherwise T would not be clash-free at depth 0) nor a β-like redex (otherwise T would not be σ_w-normal). Also, S is not an abstraction (otherwise T would not be σ_w-normal). According to Lemma 8.1, $R \in !\Lambda_\mathsf{d}$. By induction hypothesis, $S \in !\Lambda_\mathsf{n}$. Therefore, $T \in !\Lambda_\mathsf{n}$ by definition of $!\Lambda_\mathsf{n}$.
   Finally, if T is neither a value nor an abstraction nor a β-like redex, then $T \in !\Lambda_\mathsf{d}$ according to Lemma 8.1, hence $T \in !\Lambda_\mathsf{n}$ since $!\Lambda_\mathsf{d} \subseteq !\Lambda_\mathsf{n}$. □

Terms in $!\Lambda_\mathsf{v}$, $!\Lambda_\mathsf{d}$, $!\Lambda_\mathsf{n}$ and $!\Lambda_\ell$ have a peculiar semantic behavior, summarized in the following remark and lemma.

Remark 4. Using Prop. 4, in particular the rules ax and (the 0-ary version of) ! in Fig. 1, it is immediate to check that, given a value V and a repetition-free list $\vec{x} = (x_1, \dots, x_n)$ of variables containing all the free variables of V (with $n \in \mathbb{N}$), one has $((\overset{n}{[\,]}, [\,]), [\,]) \in [\![V]\!]_{\vec{x}}$.

For certain values, $((\overset{n}{[\,]}, [\,]), [\,])$ is the only element in their semantics, *e.g.* $[\![(\langle\Delta\rangle\Delta^!)^!]\!] = \{((\,), [\,])\} = [\![(\langle\Delta'\rangle\Delta'^!)^!]\!]$ (Δ and Δ' are defined in Ex. 1).

Lemma 9 (Semantics of clash-free at depth 0 $\mathsf{b}\sigma_\mathsf{w}$-normal forms)**.** *Let T be a term and $\vec{x} = (x_1, \dots, x_n)$ be a repetition-free list of variables containing all the free variables of T (with $n \in \mathbb{N}$).* Proof p. 16

1. *If $T \in !\Lambda_\mathsf{d}$, then for any $a \in \mathcal{U}$ there exist $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that $((m_1, \dots, m_n), a) \in [\![T]\!]_{\vec{x}}$.*
2. *If $T \in !\Lambda_\ell$ (in particular, if $T \in !\Lambda_\mathsf{n}$), then there are $a \in \mathcal{U}$ and $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that $((m_1, \dots, m_n), a) \in [\![T]\!]_{\vec{x}}$.*

In a way, Lemma 9.1 means that every $T \in !\Lambda_\mathsf{d}$ can be seen as both a value (since $((m_1, \dots, m_n), m) \in [\![T]\!]_{\vec{x}}$ for any $m \in \mathcal{M}_\mathrm{f}(\mathcal{U})$) and a function (since $((m_1, \dots, m_n), (m, a)) \in [\![T]\!]_{\vec{x}}$ for any $(m, a) \in \mathcal{M}_\mathrm{f}(\mathcal{U}) \times \mathcal{U}$).

In §2.3 we have shown that the reduction $\to_{\mathsf{b}\sigma_\mathsf{w}}$ (unlike $\to_{\mathsf{b}_\mathsf{w}}$) is not confluent and does not have the property of the unique normal form: this is the reason why, in the sequel, we talk about the $\mathsf{b}\sigma_\mathsf{w}$-normal form*s* of a given $\mathsf{b}\sigma_\mathsf{w}$-normalizable term. As already mentioned in §2.3, we conjecture that $\to_{\mathsf{b}\sigma_\mathsf{w}}$ is confluent modulo the equivalence relation $\sim_{\mathsf{sh}}$ on $!\Lambda$.

Theorem 2 (Computational adequacy)**.** *Let $S \in !\Lambda$ and let $\vec{x}$ be a repetition-free list of variables containing all free variables of S.*

1. *If $[\![S]\!]_{\vec{x}} \neq \emptyset$, then S is strongly $\mathsf{b}\sigma_\mathsf{w}$-normalizable (in particular, S is strongly b_w-normalizable) and every term T such that $T \simeq_{\mathsf{b}\sigma} S$ (in particular, S and its b_w- and $\mathsf{b}\sigma_\mathsf{w}$-normal forms) is clash-free at depth 0.*
2. *If S is $\mathsf{b}\sigma_\mathsf{w}$-normalizable and one of its $\mathsf{b}\sigma_\mathsf{w}$-normal forms is clash-free at depth 0, then $[\![S]\!]_{\vec{x}} \neq \emptyset$.*

Proof. 1. First, we prove that S is strongly $\mathsf{b}\sigma_\mathsf{w}$-normalizable. By hypothesis, there is $(\vec{m}, a) \in [\![S]\!]_{\vec{x}}$. According to Lemma 7, there exists $s \in \mathcal{T}(S)$ such that $(\vec{m}, a) \in [\![s]\!]_{\vec{x}}$. By Lemma 6, if $S \to_{\mathsf{b}\sigma_\mathsf{w}} S'$ then $s \to_{\mathsf{b}\sigma_\mathsf{w}} s'$ for some $s' \subseteq \mathcal{T}(S)$. By subject reduction (Lemma 5) $(\vec{m}, a) \in [\![s']\!]_{\vec{x}}$, hence $s' \neq 0$ and there exists $s_1 \in s'$ such that $(\vec{m}, a) \in [\![s_1]\!]_{\vec{x}}$. Therefore, if there was an infinite $\mathsf{b}\sigma_\mathsf{w}$-reduction sequence $S \to_{\mathsf{b}\sigma_\mathsf{w}} S_1 \to_{\mathsf{b}\sigma_\mathsf{w}} S_2 \to_{\mathsf{b}\sigma_\mathsf{w}} \dots$ then there would be an infinite sequence $(s_i)_{i\in\mathbb{N}}$ of resource terms such that: for any $i \in \mathbb{N}$ there is $s' \in \overline{2}\langle !\Lambda^{\mathsf{res}}\rangle$ such that $s_i \to_{\mathsf{b}\sigma} s'$ and $s_{i+1} \in s'$; but this is impossible since the resource bang calculus is strongly normalizing (Lemma 3). So, S is strongly $\mathsf{b}\sigma_\mathsf{w}$-normalizable.
   Now we prove that every term T such that $T \simeq_{\mathsf{b}\sigma} S$ is clash-free at depth 0. According to the invariance of the relational semantics (Thm. 1), it is enough to prove that S is clash-free at depth 0. It is easy to check that clashes are not typable by means of the rules in Fig. 1, thus all terms typable in this typing system are clash-free at depth 0. We are done, since S is typable in this typing system, by Prop. 4.

2. Let T be a $\mathsf{b}\sigma_{\mathsf{w}}$-normal form of S that is clash-free at depth 0 (it exists by hypothesis). Then, $[\![T]\!]_{\vec{x}} \neq \emptyset$ according to the syntactic and semantic characterization of clash-free at depth 0 $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms (Lemmas 8.3 and 9.2). By invariance of the relational semantics (Thm. 1), $[\![S]\!]_{\vec{x}} = [\![T]\!]_{\vec{x}} \neq \emptyset$. □

Thm. 2.1 does not hold if we replace "clash-free at depth 0" with "clash-free": indeed, $(\langle x\rangle y)^!$ is clash-free at depth 0 but is not clash-free, and $((),()),()) \in [\![(\langle x\rangle y)^!]\!]_{x,y}$ (use Prop. 4 and the rule ! in Fig. 1). Also, Thm. 2 does not hold if we replace "strongly $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable" with "strongly $\mathsf{b}\sigma$-normalizable": *e.g.* $(\langle\Delta\rangle\Delta^!)^!$ is $\mathsf{b}\sigma_{\mathsf{w}}$-normal but it is not $\mathsf{b}\sigma$-normalizable (see Ex. 1), and $((),()) \in [\![(\langle\Delta\rangle\Delta^!)^!]\!]$ (again, use Prop. 4 and the rule ! in Fig. 1).

Thm. 2 has some notable consequences, the first one is that normalization and strong normalization coincide for the $\mathsf{b}\sigma_{\mathsf{w}}$-reduction, in the set of terms that are not $\mathsf{b}\sigma$-equivalent to any term containing a clash at depth 0.

Corollary 1 (Normalizability for the $\mathsf{b}\sigma_{\mathsf{w}}$-reduction)**.** *Let S be a term such that every term $T \simeq_{\mathsf{b}\sigma} S$ is clash-free at depth 0: S is $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable iff S is strongly $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable.*

Proof. **⇐:** Trivial.

⇒: Let $\vec{x}$ be a repetition-free list of variables containing all the free variables of S. If S is not strongly $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable then $[\![S]\!]_{\vec{x}} = \emptyset$ according to Thm. 2.1, and hence, by Thm. 2.2, either S is not $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable, or S is $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable but one of its $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms is not clash-free at depth 0. The latter case is impossible by hypothesis. □

A first clue that the several $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms (if any) of a term are not so different is given by the following corollary of Thm. 2:

Corollary 2. *If $S \in !\Lambda$ has a $\mathsf{b}\sigma_{\mathsf{w}}$-normal form that is clash-free at depth 0, then all $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms of S are clash-free at depth 0.*

Proof. Let $\vec{x}$ be a repetition-free list of variables containing all the free variables of S. According to Thm. 2.2, $[\![S]\!]_{\vec{x}} \neq \emptyset$ and hence, by Thm. 2.1, all the $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms are clash-free at depth 0. □

According to Cor. 2, given a $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable term, for its $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms there are only two cases: either all of them are clash-free at depth 0 (with non-empty semantics), or all of them contain a clash at depth 0 (*i.e.* are "meaningless", with empty semantics).

Another remarkable consequence of Thm. 2 is the adequacy of the relational semantics with respect to a suitable notion of observational equivalence in the (extended) bang calculus: if two terms have the same interpretation in the relational model, then it is impossible to observationally distinguish between them (Cor. 3).

Definition 2 (Halting, observational preorder and equivalence)**.** *Let T be a term. We say that T* halts *if T is $\mathsf{b}\sigma_{\mathsf{w}}$-normalizable and its $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms are clash-free at depth 0.*

The observational preorder *(resp.* observational equivalence*) is a preorder $\lesssim$ (resp. equivalence $\cong$) on $!\Lambda$ defined by: $T \lesssim S$ (resp. $T \cong S$) if, for any context $\mathtt{C}$, $\mathtt{C}(\!|T|\!)$ halts implies that $\mathtt{C}(\!|S|\!)$ halts (resp. $\mathtt{C}(\!|T|\!)$ halts iff $\mathtt{C}(\!|S|\!)$ halts).*

Morally, two terms are observationally equivalent if one cannot "observe" any difference between their behaviors.

Corollary 3 (Observational adequacy)**.** *Let $T, S \in !\Lambda$ and $\vec{x}$ be a repetition-free list of variables containing all the free variables of T and S.*

1. *If $[\![T]\!]_{\vec{x}} \subseteq [\![S]\!]_{\vec{x}}$ then $T \lesssim S$.*
2. *If $[\![T]\!]_{\vec{x}} = [\![S]\!]_{\vec{x}}$ then $T \cong S$.*

Proof. 1. Let $\mathtt{C}$ be a context. Let $\vec{y}$ be a repetition-free list of variables containing all the free variable of $\mathtt{C}(\!|T|\!)$ and $\mathtt{C}(\!|S|\!)$. If $\mathtt{C}(\!|T|\!)$ halts, then $[\![\mathtt{C}(\!|T|\!)]\!]_{\vec{y}} \neq \emptyset$ by Thm. 2.2. Using Prop. 4, it is easy to check that $[\![T]\!]_{\vec{x}} \subseteq [\![S]\!]_{\vec{x}}$ implies $[\![\mathtt{C}(\!|T|\!)]\!]_{\vec{y}} \subseteq [\![\mathtt{C}(\!|S|\!)]\!]_{\vec{y}}$, hence $[\![\mathtt{C}(\!|S|\!)]\!]_{\vec{y}} \neq \emptyset$. By Thm. 2.1, $\mathtt{C}(\!|S|\!)$ halts. Therefore, $T \lesssim S$.

2. Immediate consequence of Cor. 3.1. □

It is natural to wonder whether the relational model is fully abstract for the (extended) bang calculus, *i.e.* whether the converse of Cor. 3 holds. The issue will be studied in a future work.

A weaker syntactic reformulation of Cor. 3 is the soundness of the $\mathsf{b}\sigma$-equivalence with respect to the observational equivalence: any two $\mathsf{b}\sigma$-equivalent terms are observationally indistinguishable.

Corollary 4 (Soundness of $\mathsf{b}\sigma$-equivalence wrt observational equivalence)**.** *Let $T, S \in !\Lambda$. If $T \simeq_{\mathsf{b}\sigma} S$ then $T \cong S$. In particular, for any two $\mathsf{b}\sigma_{\mathsf{w}}$-normal forms R and Q of T (if any), one has $R \cong T \cong Q$.*

Proof. According to the invariance of the relational semantics (Thm. 1), from $T \simeq_{\mathsf{b}\sigma} S$ it follows that $[\![T]\!]_{\vec{x}} = [\![S]\!]_{\vec{x}}$ (where $\vec{x}$ is a repetition-free list of variables containing all the free variables of T and S). Hence, $T \cong S$ by Cor. 3.2. □

The converse of Cor. 4 does not hold. Indeed, consider the terms $\langle\Delta\rangle\Delta^!$ (see Ex. 1) and $\langle\Delta_3\rangle\Delta_3^!$, where $\Delta_3 := \lambda x\,\langle\langle \mathrm{der}\, x\rangle x\rangle x$: $\langle\Delta\rangle\Delta^! \cong \langle\Delta_3\rangle\Delta_3^!$ (by Cor. 3.2, as $[\![\langle\Delta\rangle\Delta^!]\!] = \emptyset = [\![\langle\Delta_3\rangle\Delta_3^!]\!]$) but it can be shown that $\langle\Delta\rangle\Delta^! \not\simeq_{\mathsf{b}\sigma} \langle\Delta_3\rangle\Delta_3^!$.

Acknowledgement This work has been partially supported by the French-Chinese project ANR-11-IS02-0002 and NSFC 61161130530 *Locali*, and by the A*MIDEX project ANR-11-IDEX-0001-02 funded by the "Investissements d'Avenir" French Government program, managed by the French National Research Agency (ANR).

References

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

[2] B. Accattoli. Proof nets and the call-by-value λ-calculus. *Theoretical Computer Science*, 606:2–24, 2015.

[3] B. Accattoli, P. Barenbaum, and D. Mazza. Distilling abstract machines. In J. Jeuring and M. M. T. Chakravarty, editors, *Proceedings of the 19th ACM SIGPLAN international conference on Functional programming*, pages 363–376. ACM, 2014.

[4] B. Accattoli and D. Kesner. Preservation of Strong Normalisation modulo permutations for the structural lambda-calculus. *Logical Methods in Computer Science*, 8(1), 2012.

[5] B. Accattoli and L. Paolini. Call-by-Value Solvability, Revisited. In *FLOPS*, pages 4–16, 2012.

[6] Z. M. Ariola, J. Maraist, M. Odersky, M. Felleisen, and P. Wadler. A Call-by-need Lambda Calculus. In *Proceedings of the 22Nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '95, pages 233–246, New York, NY, USA, 1995. ACM.

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

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

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

[10] A. Carraro and G. Guerrieri. A semantical and operational account of call-by-value solvability. In A. Muscholl, editor, *Foundations of Software Science and Computation Structures - FOSSACS 2014*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 2014.

[11] P.-L. Curien and H. Herbelin. The duality of computation. In M. Odersky and P. Wadler, editors, *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00)*, pages 233–243. ACM, 2000.

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

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

[14] D. de Carvalho. Execution Time of lambda-Terms via Denotational Semantics and Intersection Types. *Mathematical Structures in Computer Science*, 2009. To appear.

[15] D. de Carvalho. The relational model is injective for Multiplicative Exponential Linear Logic. *CoRR*, abs/1502.02404, 2015.

[16] D. de Carvalho and L. Tortora de Falco. The relational model is injective for multiplicative exponential linear logic (without weakenings). *Ann. Pure Appl. Logic*, 163(9):1210–1236, 2012.

[17] T. Ehrhard. Collapsing non-idempotent intersection types. In *CSL*, pages 259–273, 2012.

[18] T. Ehrhard. An introduction to Differential Linear Logic: proof-nets, models and antiderivatives. *Mathematical Structures in Computer Science*, 2015. To appear.

[19] T. Ehrhard. Call-By-Push-Value from a Linear Logic point of view. In *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Proceedings*, volume 9632 of *Lecture Notes in Computer Science*, pages 202–228. Springer, 2016.

[20] T. Ehrhard and L. Regnier. Böhm Trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms. In *CiE*, volume 3988 of *LNCS*, pages 186–197. Springer, 2006.

[21] T. Ehrhard and L. Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2):166–195, 2006.

[22] T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theoretical Computer Science*, 403(2-3):347–372, 2008.

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

[24] G. Guerrieri. Head reduction and normalization in a call-by-value lambda-calculus. In N. Nishida et al., editors, *Workshop on Rewriting Techniques for Program Transformations and Evaluation, WPTE 2015*, volume 46 of *OASICS*, pages 3–17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[25] G. Guerrieri, L. Paolini, and S. Ronchi Della Rocca. Standardization of a Call-By-Value Lambda-Calculus. In T. Altenkirch, editor, *Typed Lambda Calculi and Applications, TLCA 2015*, volume 38 of *LIPIcs*, pages 211–225. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[26] G. Guerrieri, L. Pellissier, and L. Tortora de Falco. Computing Connected Proof(-Structure)s From Their Taylor Expansion. In D. Kesner and B. Pientka, editors, *1st International Conference on Formal Structures for Computation and Deduction, FSCD 2016*, volume 52 of *LIPIcs*, pages 20:1–20:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[27] D. Kesner. Reasoning About Call-by-need by Means of Types. In B. Jacobs and C. Löding, editors, *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Proceedings*, volume 9634 of *Lecture Notes in Computer Science*, pages 424–441. Springer, 2016.

[28] O. Laurent. Polarized proof-nets and λμ-calculus. *Theoretical Computer Science*, 290(1):161–188, 2003.

[29] O. Laurent and L. Regnier. About Translations of Classical Logic into Polarized Linear Logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), Proceedings*, pages 11–20. IEEE Computer Society, 2003.

[30] P. B. Levy. Call-by-Push-Value: A Subsuming Paradigm. In J.-Y. Girard, editor, *Typed Lambda Calculi and Applications, TLCA'99, Proceedings*, volume 1581 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 1999.

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

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

[33] G. Manzonetto and M. Pagani. Böhm's Theorem for Resource Lambda Calculus through Taylor Expansion. In *TLCA 2011*, pages 153–168, 2011.

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

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

[36] E. Moggi. Computational Lambda-Calculus and Monads. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89)*, pages 14–23. IEEE Computer Society, 1989.

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

[38] L. Paolini, M. Piccolo, and S. Ronchi Della Rocca. Essential and relational models. *Mathematical Structures in Computer Science*, FirstView:1–25, 2015.

[39] L. Paolini and S. Ronchi Della Rocca. Call-by-value Solvability. *ITA*, 33(6):507–534, 1999.

[40] G. D. Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

[41] L. Regnier. *Lambda calcul et réseaux*. PhD thesis, Université Paris 7, 1992.

[42] L. Regnier. Une équivalence sur les lambda-termes. *Theor. Comput. Sci.*, 126(2):281–292, 1994.

[43] S. Ronchi della Rocca and L. Paolini. *The Parametric λ-calculus: a Metamodel for Computation*. Texts in Theoretical Computer Science. Springer, Berlin, 2004.

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

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

Figure 3. Parallel v-reduction.

The enumeration of propositions, theorems, lemmas already stated in the body of the article is unchanged.

Omitted proofs and remarks of Section 2

We adapt in a standard way the Tait–Martin-Löf technique, more precisely the variant introduced by Takahashi [44], to prove the confluence of $\to_{\mathsf{v}}$ (Lemma 1.4). For this purpose, we introduce the *parallel* v*-reduction*, denoted by $\Rightarrow_{\mathsf{v}}$, a binary relation on $!\Lambda$ defined by the rules in Fig. 3.

Remark 5. It is immediate to check that $\Rightarrow_{\mathsf{v}}$ is reflexive and $\to_{\mathsf{v}} \subseteq \Rightarrow_{\mathsf{v}} \subseteq \to_{\mathsf{v}}^*$, hence $\Rightarrow_{\mathsf{v}}^* = \to_{\mathsf{v}}^*$.

Lemma 10 (Substitution Lemma for parallel v-reduction)**.** *Let $T, S \in !\Lambda$, $V, W \in !\Lambda_{\mathsf{v}}$ and $x \in \mathcal{V}ar$. If $T \Rightarrow_{\mathsf{v}} S$ and $V \Rightarrow_{\mathsf{v}} W$, then $T\{V/x\} \Rightarrow_{\mathsf{v}} S\{W/x\}$.*

Proof. By induction on the derivation of $T \Rightarrow_{\mathsf{v}} S$. □

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

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

Lemma 11 (Development)**.** *Let $T, S \in !\Lambda$. If $T \Rightarrow_{\mathsf{v}} S$, then $S \Rightarrow_{\mathsf{v}} T^*$.*

Proof. By induction on the derivation of $T \Rightarrow_{\mathsf{v}} S$. Let us consider its last rule r.

If $\mathsf{r} = \mathsf{var}$, then $T = x = S$ and $T^* = x$, therefore $S \Rightarrow_{\mathsf{v}} T^*$.

If $\mathsf{r} = !$, then $T = R^{!}$ and $S = Q^{!}$ with $R \Rightarrow_{\mathsf{v}} Q$. By induction hypothesis, $Q \Rightarrow_{\mathsf{v}} R^*$ and hence $S = Q^{!} \Rightarrow_{\mathsf{v}} (R^*)^{!} = T^*$.

The case where $\mathsf{r} = \lambda$ or $\mathsf{r} = \mathrm{der}$ is analogous to the case where $\mathsf{r} = !$.

If $\mathsf{r} = \mathsf{v}$, then $T = \langle \lambda x\, R\rangle V$ and $S = Q\{W/x\}$ with $R \Rightarrow_{\mathsf{v}} Q$ and $V \Rightarrow_{\mathsf{v}} W$. By induction hypothesis, $Q \Rightarrow_{\mathsf{v}} R^*$ and $W \Rightarrow_{\mathsf{v}} V^*$, hence $S = Q\{W/x\} \Rightarrow_{\mathsf{v}} R^*\{V^*/x\} = T^*$, according to Lemma 10.

Finally, if $\mathsf{r} = @$, then there are two subcases:

- either $T = \langle R\rangle Q$ and $S = \langle R'\rangle Q'$ with $R \Rightarrow_{\mathsf{v}} R'$, $Q \Rightarrow_{\mathsf{v}} Q'$, and $R \neq \lambda x\, P$ or $Q \notin !\Lambda_{\mathsf{v}}$; by induction hypothesis, $R' \Rightarrow_{\mathsf{v}} R^*$ and $Q' \Rightarrow_{\mathsf{v}} Q^*$, therefore $S = \langle R'\rangle Q' \Rightarrow_{\mathsf{v}} \langle R^*\rangle Q^* = T^*$;
- or $T = \langle \lambda x\, R\rangle V$ and $S = \langle \lambda x\, Q\rangle W$ with $R \Rightarrow_{\mathsf{v}} Q$ and $V \Rightarrow_{\mathsf{v}} W$; by induction hypothesis, $Q \Rightarrow_{\mathsf{v}} R^*$ and $W \Rightarrow_{\mathsf{v}} V^*$, hence $S = \langle \lambda x\, Q\rangle W \Rightarrow_{\mathsf{v}} R^*\{V^*/x\} = T^*$. □

Lemma 12 (Substitution Lemma for reductions)**.** *Let $T, S, R \in !\Lambda$, $V \in !\Lambda_{\mathsf{v}}$ and $x \in \mathcal{V}ar$.*

1. *For any $\mathsf{r} \in \{\mathsf{v}, !, \mathsf{b}, \mathsf{v_w}, !_{\mathsf{w}}, \mathsf{b_w}\}$, if $T \to_{\mathsf{r}} S$ then $T\{V/x\} \to_{\mathsf{r}} S\{V/x\}$.*
2. *For any $\mathsf{r} \in \{\mathsf{v}, !, \mathsf{b}\}$, if $S \to_{\mathsf{r}} R$ then $T\{S^{!}/x\} \to_{\mathsf{r}}^* T\{R^{!}/x\}$.*

Proof. 1. By induction on $T \in !\Lambda$.

Let us consider the cases where $\mathsf{r} \in \{\mathsf{v_w}, !_{\mathsf{w}}, \mathsf{b_w}\}$. T cannot be a value because variables and boxes are r-normal. Therefore, there are only the following cases:

- *Root-step for $T \to_{\mathsf{r}} S$.* Since $\mathsf{r} \in \{\mathsf{v}, !, \mathsf{b}\}$ and $\mapsto_{\mathsf{b}} = \mapsto_{!} \cup \mapsto_{\mathsf{v}}$, there are two subcases:
  - $T := \mathrm{der}(S^{!}) \mapsto_{!} S$, so $T\{V/x\} = \mathrm{der}(S\{V/x\}^{!}) \to_{!} S\{V/x\}$.
  - $T := \langle \lambda y\, R\rangle W \mapsto_{\mathsf{v}} R\{W/y\} =: S$ and we can suppose without loss of generality that $y \notin \mathsf{fv}(V) \cup \{x\}$. Notice that $W\{V/x\} \in !\Lambda_{\mathsf{v}}$, thus $T\{V/x\} = \langle \lambda y\, R\{V/x\}\rangle W\{V/x\} \to_{\mathsf{v}} R\{V/x\}\{W\{V/x\}/y\} = R\{W/y\}\{V/x\} = S\{V/x\}$.
- $T := \mathrm{der}\, R \to_{\mathsf{r}} \mathrm{der}\, Q =: S$ with $R \to_{\mathsf{r}} Q$: by induction hypothesis, $R\{V/x\} \to_{\mathsf{r}} Q\{V/x\}$ and hence $T\{V/x\} = \mathrm{der}(R\{V/x\}) \to_{\mathsf{r}} \mathrm{der}(Q\{V/x\}) = S\{V/x\}$.
- $T := \lambda y\, R \to_{\mathsf{r}} \lambda y\, Q =: S$ with $R \to_{\mathsf{r}} Q$: we can suppose without loss of generality that $y \notin \mathsf{fv}(V) \cup \{x\}$. By induction hypothesis, $R\{V/x\} \to_{\mathsf{r}} Q\{V/x\}$ and hence $T\{V/x\} = \lambda y\, R\{V/x\} \to_{\mathsf{r}} \lambda y\, Q\{V/x\} = S\{V/x\}$.
- $T := \langle Q\rangle R \to_{\mathsf{r}} \langle Q\rangle P =: S$ with $R \to_{\mathsf{r}} P$. By induction hypothesis, $R\{V/x\} \to_{\mathsf{r}} P\{V/x\}$ and so $T\{V/x\} = \langle Q\{V/x\}\rangle R\{V/x\} \to_{\mathsf{r}} \langle Q\{V/x\}\rangle P\{V/x\} = S\{V/x\}$.
- $T := \langle Q\rangle R \to_{\mathsf{r}} \langle P\rangle R =: S$ with $Q \to_{\mathsf{r}} P$. By induction hypothesis, $Q\{V/x\} \to_{\mathsf{r}} P\{V/x\}$ and so $T\{V/x\} = \langle Q\{V/x\}\rangle R\{V/x\} \to_{\mathsf{r}} \langle P\{V/x\}\rangle R\{V/x\} = S\{V/x\}$.

Now, let us consider the case where $\mathsf{r} \in \{\mathsf{v}, !, \mathsf{b}\}$. The poof is perfectly analogous to the one above, the only novelty is a new case:

- $T := R^{!} \to_{\mathsf{r}} Q^{!} =: S$ with $R \to_{\mathsf{r}} Q$: by induction hypothesis, $R\{V/x\} \to_{\mathsf{r}} Q\{V/x\}$ and hence $T\{V/x\} = (R\{V/x\})^{!} \to_{\mathsf{r}} (Q\{V/x\})^{!} = S\{V/x\}$.

2. By straightforward induction on $T \in !\Lambda$. □

Lemma 13. *Let $\to_1, \to_2$ be two binary relations on a set X such that ${}_1\!\leftarrow \cdot \to_2 \subseteq \to_2 \cdot {}_1^*\!\leftarrow$. Then they commute, i.e. ${}_1^*\!\leftarrow \cdot \to_2^* \subseteq \to_2^* \cdot {}_1^*\!\leftarrow$.*

Proof. For every $t, u \in X$, $\to \subseteq X^2$ and $n \in \mathbb{N}$, we write $t \to^n u$ if there exist $v_0, \ldots, v_n \in X$ such that $t = v_0$, $u = v_n$ and $v_i \to v_{i+1}$ for all $0 \leq i < n$. We prove the following stronger statement, in order to apply the right induction hypothesis: if $t \to_1^* u_1$ and $t \to_2^m u_2$ then there exists $t' \in X$ such that $u_2 \to_1^* t'$ and $u_1 \to_2^m t'$. Let $t \to_1^n u_1$: the proof is by induction on (m, n) with the lexicographical order on $\mathbb{N}^2$.

If $m = 0$ or $n = 0$, we conclude easily.

Let $m, n > 0$: so, there exist $u_1', u_2' \in X$ such that $t \to_1 u_1'$, $t \to_2 u_2'$, $u_1' \to_1^{n1} u_1$ and $u_2' \to_2^{m1} u_2$. By hypothesis, there is $v \in X$ such that $u_1' \to_2 v$ and $u_2' \to_1^* v$. By induction hypothesis applied to u_2', there is $v' \in X$ such that $u_2 \to_1^* v'$ and $v \to_2^{m1} v'$; thus $u_1' \to_2^m v'$, so there exists $s \in X$ such that $v' \to_1^* s$ and $u_1 \to_2^m s$ by applying the induction hypothesis to u_1, therefore $u_2 \to_1^* s$. □

Lemma 1 (Basic properties of reductions)**.** See p. 3

1. *$\to_{\mathsf{v_w}}$ is strongly confluent (i.e. ${}_{\mathsf{v_w}}\!\leftarrow \cdot \to_{\mathsf{v_w}} \subseteq (\to_{\mathsf{v_w}} \cdot {}_{\mathsf{v_w}}\!\leftarrow) \cup =$).*
2. *$\to_{!_{\mathsf{w}}}$ and $\to_{!}$ are strongly normalizing and strongly confluent.*
3. *$\to_{!_{\mathsf{w}}}$ and $\to_{\mathsf{v_w}}$ strongly commute (i.e. ${}_{!_{\mathsf{w}}}\!\leftarrow \cdot \to_{\mathsf{v_w}} \subseteq \to_{\mathsf{v_w}} \cdot {}_{!_{\mathsf{w}}}\!\leftarrow$).*
   $\to_{!}$ and $\to_{\mathsf{v}}$ commute (i.e. ${}_{!}^*\!\leftarrow \cdot \to_{\mathsf{v}}^* \subseteq \to_{\mathsf{v}}^* \cdot {}_{!}^*\!\leftarrow$).
4. *$\to_{\mathsf{v}}$ is confluent.*

Proof. 1. Let $T, T_1, T_2 \in !\Lambda$ be such that $T_1\ {}_{\mathsf{v_w}}\!\leftarrow T \to_{\mathsf{v_w}} T_2$ and $T_1 \neq T_2$: we prove by induction on $T \in !\Lambda$ that there is $T' \in !\Lambda$ such that $T_1 \to_{\mathsf{v_w}} T'\ {}_{\mathsf{v_w}}\!\leftarrow T_2$. Since $\to_{\mathsf{v_w}}$ does not reduce under !, there are only the following cases:

- *Root-step for $T \to_{\mathsf{v_w}} T_1$ and non-Root-step for $T \to_{\mathsf{v_w}} T_2$, i.e. $T := \langle \lambda x\, S\rangle V \mapsto_{\mathsf{v}} S\{V/x\} =: T_1$ and $T \to_{\mathsf{v_w}}$*

$\langle \lambda x\, R\rangle V \;=:\; T_2$ with $S \to_{\mathsf{v}_\mathsf{w}} R$. Notice that V is v_w-normal since $\to_{\mathsf{v}_\mathsf{w}}$ does not reduce under !, hence $T \to_{\mathsf{v}_\mathsf{w}} T_2$ via a non-root-step implies $S \to_{\mathsf{v}_\mathsf{w}} R$ by necessity. By Lemma 12.1, $T_1 \to_{\mathsf{v}_\mathsf{w}} R\{V/x\}\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T_2$.

- der-*step for both* $T \to_{\mathsf{v}_\mathsf{w}} T_1$ *and* $T \to_{\mathsf{v}_\mathsf{w}} T_2$, *i.e.* $T_1 := \operatorname{der} S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T \to_{\mathsf{v}_\mathsf{w}} \operatorname{der} S_2 =: T_2$ with $T := \operatorname{der} S$ and $S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow S \to_{\mathsf{v}_\mathsf{w}} S_2$. By induction hypothesis, there exists $S' \in !\Lambda$ such that $S_1 \to_{\mathsf{v}_\mathsf{w}} S'\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow S_2$, whence $T_1 \to_{\mathsf{v}_\mathsf{w}} \operatorname{der} S'\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T_2$.
- λ-*step for both* $T \to_{\mathsf{v}_\mathsf{w}} T_1$ *and* $T \to_{\mathsf{v}_\mathsf{w}} T_2$, *i.e.* $T_1 := \lambda x\, S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T \to_{\mathsf{v}_\mathsf{w}} \lambda x\, S_2 =: T_2$ with $T := \lambda x\, S$ and $S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow S \to_{\mathsf{v}_\mathsf{w}} S_2$: analogous to the previous case.
- *Application Left for both* $T \to_{\mathsf{v}_\mathsf{w}} T_1$ *and* $T \to_{\mathsf{v}_\mathsf{w}} T_2$, *i.e.* $T_1 := \langle S_1\rangle R\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T \to_{\mathsf{v}_\mathsf{w}} \langle S_2\rangle R =: T_2$ with $T := \langle S\rangle R$ and $S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow S \to_{\mathsf{v}_\mathsf{w}} S_2$. By induction hypothesis, there exists $S' \in !\Lambda$ such that $S_1 \to_{\mathsf{v}_\mathsf{w}} S'\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow S_2$, whence $T_1 \to_{\mathsf{v}_\mathsf{w}} \langle S'\rangle R\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T_2$.
- *Application Right for both* $T \to_{\mathsf{v}_\mathsf{w}} T_1$ *and* $T \to_{\mathsf{v}_\mathsf{w}} T_2$, *i.e.* $T_1 := \langle R\rangle S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T \to_{\mathsf{v}_\mathsf{w}} \langle R\rangle S_2 =: T_2$ with $T := \langle R\rangle S$ and $S_1\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow S \to_{\mathsf{v}_\mathsf{w}} S_2$: analogous to the previous case.
- *Application Left for both* $T \to_{\mathsf{v}_\mathsf{w}} T_1$ and *Application Right* $T \to_{\mathsf{v}_\mathsf{w}} T_2$, *i.e.* $T := \langle S\rangle R$ and $T_1 := \langle S'\rangle R\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T \to_{\mathsf{v}_\mathsf{w}} \langle S\rangle R' =: T_2$ with $S \to_{\mathsf{v}_\mathsf{w}} S'$ and $R \to_{\mathsf{v}_\mathsf{w}} R'$. Then, $T_1 \to_{\mathsf{v}_\mathsf{w}} \langle S'\rangle R'\, {}_{\mathsf{v}_\mathsf{w}}\!\leftarrow T_2$.

2. For any $T \in !\Lambda$, let $|T|_!$ be the number of occurrences of ! in T. It is immediate to check that if $T \to_! S$ (in particular, if $T \to_{!_\mathsf{w}} S$) then $|T|_! > |S|_!$. This proves that $\to_{!_\mathsf{w}}$ and $\to_!$ are strongly normalizing.
   Now we prove that $\to_!$ is strongly confluent; the proof of strong confluence of $\to_{!_\mathsf{w}}$ is analogous. Let $T, T_1, T_2 \in !\Lambda$ be such that $T_1\, {}_!\!\leftarrow T \to_! T_2$ and $T_1 \neq T_2$: we prove by induction on $T \in !\Lambda$ that there exists $T' \in !\Lambda$ such that $T_1 \to_! T'\, {}_!\!\leftarrow T_2$. Cases:
   - *Root-step for* $T \to_! T_1$ and *non-Root-step for* $T \to_! T_2$, *i.e.* $T := \operatorname{der}(T_1^!) \mapsto_! T_1$ and $T \to_! \operatorname{der}(S^!) =: T_2$ with $T_1 \to_! S$. Then, $T_1 \to_! S\, {}_!\!\leftarrow T_2$.
   - der-*step for both* $T \to_! T_1$ *and* $T \to_! T_2$, *i.e.* $T_1 := \operatorname{der} S_1\, {}_!\!\leftarrow T \to_! \operatorname{der} S_2 =: T_2$ with $T := \operatorname{der} S$ and $S_1\, {}_!\!\leftarrow S \to_! S_2$. By induction hypothesis, there exists $S' \in !\Lambda$ such that $S_1 \to_! S'\, {}_!\!\leftarrow S_2$, whence $T_1 \to_! \operatorname{der} S'\, {}_!\!\leftarrow T_2$.
   - λ-*step for both* $T \to_! T_1$ *and* $T \to_! T_2$, *i.e.* $T_1 := \lambda x\, S_1\, {}_!\!\leftarrow T \to_! \lambda x\, S_2 =: T_2$ with $T := \lambda x\, S$ and $S_1\, {}_!\!\leftarrow S \to_! S_2$: analogous to the previous case.
   - !-*step for both* $T \to_! T_1$ *and* $T \to_! T_2$, *i.e.* $T_1 := S_1^!\, {}_!\!\leftarrow T \to_! S_2^! =: T_2$ with $T := S^!$ and $S_1\, {}_!\!\leftarrow S \to_! S_2$: analogous to the previous case.
   - *Application Left for both* $T \to_! T_1$ *and* $T \to_! T_2$, *i.e.* $T_1 := \langle S_1\rangle R\, {}_!\!\leftarrow T \to_! \langle S_2\rangle R =: T_2$ with $T := \langle S\rangle R$ and $S_1\, {}_!\!\leftarrow S \to_! S_2$. By induction hypothesis, there exists $S' \in !\Lambda$ such that $S_1 \to_! S'\, {}_!\!\leftarrow S_2$, whence $T_1 \to_! \langle S'\rangle R\, {}_!\!\leftarrow T_2$.
   - *Application Right for both* $T \to_! T_1$ *and* $T \to_! T_2$, *i.e.* $T_1 := \langle R\rangle S_1\, {}_!\!\leftarrow T \to_! \langle R\rangle S_2 =: T_2$ with $T := \langle R\rangle S$ and $S_1\, {}_!\!\leftarrow S \to_! S_2$: analogous to the previous case.
   - *Application Left for both* $T \to_! T_1$ and *Application Right* $T \to_! T_2$, *i.e.* $T := \langle S\rangle R$ and $T_1 := \langle S'\rangle R\, {}_!\!\leftarrow T \to_! \langle S\rangle R' =: T_2$ with $S \to_! S'$ and $R \to_! R'$. Then, $T_1 \to_! \langle S'\rangle R'\, {}_!\!\leftarrow T_2$.

3. First we show that $\to_{\mathsf{v}_\mathsf{w}}$ and $\to_{!_\mathsf{w}}$ strongly commute. Let $T, T_1, T_2 \in !\Lambda$ be such that $T_1\, {}_{!_\mathsf{w}}\!\leftarrow T \to_{\mathsf{v}_\mathsf{w}} T_2$: we prove by induction on $T \in !\Lambda$ that there exists $T' \in !\Lambda$ such that $T_1 \to_{\mathsf{v}_\mathsf{w}} T'\, {}_{!_\mathsf{w}}\!\leftarrow T_2$. Since $\to_{\mathsf{v}_\mathsf{w}}$ and $\to_{!_\mathsf{w}}$ do not reduce under !, the only interesting case is the following:
   - *Root-step for* $T \to_{\mathsf{v}_\mathsf{w}} T_2$ and *non-Root-step for* $T \to_{!_\mathsf{w}} T_1$, *i.e.* $T := \langle \lambda x\, S\rangle V \to_{!_\mathsf{w}} \langle \lambda x\, R\rangle V =: T_1$ and $T \mapsto_{\mathsf{v}_\mathsf{w}} S\{V/x\} =: T_2$ with $S \to_{!_\mathsf{w}} R$ (recall that V is $!_\mathsf{w}$-normal, so $T \to_{!_\mathsf{w}} T_1$ implies $S \to_{!_\mathsf{w}} R$ by necessity). By Lemma 12.1, $T_1 \to_{\mathsf{v}_\mathsf{w}} R\{V/x\}\, {}_{!_\mathsf{w}}\!\leftarrow T_2$.

   The other cases are analogous to the ones in the proof of Lemma 1.1 not involving root-steps.

   Now we show that $\to_\mathsf{v}$ and $\to_!$ commute. According to Lemma 13, it is sufficient to prove that ${}_!\!\leftarrow \cdot \to_\mathsf{v} \subseteq \to_\mathsf{v} \cdot {}_!\!\leftarrow^*$. Let $T, T_1, T_2 \in !\Lambda$ be such that $T_1\, {}_!\!\leftarrow T \to_\mathsf{v} T_2$: we prove by induction on $T \in !\Lambda$ that there exists $T' \in !\Lambda$ such that $T_1 \to_\mathsf{v} T'\, {}_!\!\leftarrow^* T_2$. The only interesting cases are the following:
   - *Root-step for* $T \to_! T_1$ and *non-Root-step for* $T \to_\mathsf{v} T_2$, *i.e.* $T := \operatorname{der}(T_1^!) \mapsto_! T_1$ and $T \to_\mathsf{v} \operatorname{der}(S^!) =: T_2$ with $T_1 \to_\mathsf{v} S$. Then, $T_1 \to_\mathsf{v} S\, {}_!\!\leftarrow T_2$.
   - *Root-step for* $T \to_\mathsf{v} T_2$ and *non-Root-step for* $T \to_! T_1$. There are two subcases:
     - $T := \langle \lambda x\, S\rangle V \to_! \langle \lambda x\, R\rangle V =: T_1$ and $T \mapsto_\mathsf{v} S\{V/x\} =: T_2$ with $S \to_! R$. Then, $T_1 \to_\mathsf{v} R\{V/x\}\, {}_!\!\leftarrow T_2$.
     - $T := \langle \lambda x\, S\rangle R^! \to_! \langle \lambda x\, S\rangle Q^! =: T_1$ and $T \mapsto_\mathsf{v} S\{R^!/x\} =: T_2$ with $R \to_! Q$. Then, $T_1 \to_\mathsf{v} S\{Q^!/x\}\, {}_!\!\leftarrow^* T_2$ according to Lemma 12.2.

   The other cases are analogous to the ones in the proof of Lemma 1.1 not involving root-steps, paying attention that now the inductive hypothesis is different. The only novelty is a new case:
   - !-*step for both* $T \to_! T_1$ *and* $T \to_\mathsf{v} T_2$, *i.e.* $T := S^!$ and $T_1 := S_1^!\, {}_!\!\leftarrow T \to_\mathsf{v} S_2^! =: T_2$ with $S_1\, {}_!\!\leftarrow S \to_\mathsf{v} S_2$. By induction hypothesis, there is $S' \in !\Lambda$ such that $S_1 \to_\mathsf{v} S'\, {}_!\!\leftarrow^* S_2$, hence $T_1 \to_\mathsf{v} S'^!\, {}_!\!\leftarrow^* T_2$.

4. From Lemma 11 it follows that $\Rightarrow_\mathsf{v}$ is strongly confluent: indeed, if $S\, {}_\mathsf{v}\!\Leftarrow T \Rightarrow_\mathsf{v} R$, then $S \Rightarrow_\mathsf{v} T^*\, {}_\mathsf{v}\!\Leftarrow R$. In particular, this means that $\Rightarrow_\mathsf{v}^*$ is strongly confluent. But $\Rightarrow_\mathsf{v}^* = \to_\mathsf{v}^*$ according to Remark 5. Thus, $\to_\mathsf{v}$ is confluent. □

Embedding CBN and CBV λ-calculi in the bang calculus

Lemma 2 (Substitution). *Let T and S be λ-terms, let x be a variable.* See p. 4

1. *If x has $n \in \mathbb{N}$ free occurrences in T, then $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_!^n (T\{S/x\})^{\mathsf{cbn}}$. More precisely: if $\mathsf{hoc}(T)$ is a free occurrence of x in T, then $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_{!_\mathsf{w}} \to_{!_{\neg\mathsf{w}}}^{n-1} (T\{S/x\})^{\mathsf{cbn}}$; otherwise, $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_{!_{\neg\mathsf{w}}}^n (T\{S/x\})^{\mathsf{cbn}}$.*
2. *If S is a variable or an abstraction, then $T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = (T\{S/x\})^{\mathsf{cbv}}$.*

Proof. The proofs of both points are by induction on the λ-term T. Let us denote by $|R|_x$ the number of free occurrences of the variable x in R, for any λ-term R.

1. We just prove that if x has $n \in \mathbb{N}$ free occurrences in T, then $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_!^n (T\{S/x\})^{\mathsf{cbn}}$. The proof that "if $\mathsf{hoc}(T)$ is a free occurrence of x in T, then $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_{!_\mathsf{w}} \to_{!_{\neg\mathsf{w}}}^{n-1} (T\{S/x\})^{\mathsf{cbn}}$; otherwise, $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_{!_{\neg\mathsf{w}}}^n (T\{S/x\})^{\mathsf{cbn}}$," is similar, recalling that $\mathsf{hoc}(T)$ is the only variable occurrence in T^{cbn} that is not under the scope of any !.
   If $T = x$ then $|T|_x = 1$ and $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} = \operatorname{der}(S^{\mathsf{cbn}!}) \to_! S^{\mathsf{cbn}} = (T\{S/x\})^{\mathsf{cbn}}$.
   If $T = y \neq x$, then $|T|_x = 0$ and $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} = \operatorname{der} y = (T\{S/x\})^{\mathsf{cbn}}$.

If $T = \lambda y\, R$, we can suppose without loss of generality that $y \notin \mathsf{fv}(S) \cup \{x\}$, so $|R|_x = |T|_x = n$ and, by induction hypothesis, $R^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_!^n (R\{S/x\})^{\mathsf{cbn}}$. Thus, $T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} = \lambda y\,(R^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\}) \to_!^n \lambda y\,(R\{S/x\})^{\mathsf{cbn}} = (T\{S/x\})^{\mathsf{cbn}}$.
If $T = RQ$, then $|T|_x = h+k$ where $|R|_x := h$ and $|Q|_x := k$; by induction hypothesis, $R^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_!^h (R\{S/x\})^{\mathsf{cbn}}$ and $Q^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} \to_!^k (Q\{S/x\})^{\mathsf{cbn}}$. Hence,

$$\begin{aligned} T^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\} &= \langle R^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\}\rangle (Q^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\})^! \\ &\to_!^h \langle (R\{S/x\})^{\mathsf{cbn}}\rangle (Q^{\mathsf{cbn}}\{S^{\mathsf{cbn}!}/x\})^! \\ &\to_!^k \langle (R\{S/x\})^{\mathsf{cbn}}\rangle (Q\{S/x\})^{\mathsf{cbn}!} \\ &= (T\{S/x\})^{\mathsf{cbn}}. \end{aligned}$$

2. First, note that, as S is a variable or an abstraction, then $S^{\mathsf{cbv}} \in !\Lambda_{\mathsf{v}}$ and hence $T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\}$ is defined.
If $T = x$ then $T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = S^{\mathsf{cbv}} = (T\{S/x\})^{\mathsf{cbv}}$.
If $T = y \neq x$, then $T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = y = (T\{S/x\})^{\mathsf{cbv}}$.
If $T = \lambda y\, R$, we can suppose without loss of generality that $y \notin \mathsf{fv}(S) \cup \{x\}$. By induction hypothesis, $R^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = (R\{S/x\})^{\mathsf{cbv}}$. Thus, $T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = (\lambda y\,(R^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\}))^! = (\lambda y\,(R\{S/x\})^{\mathsf{cbv}})^! = (T\{S/x\})^{\mathsf{cbv}}$.
If $T = RQ$, then $R^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = (R\{S/x\})^{\mathsf{cbv}}$ and $Q^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} = (Q\{S/x\})^{\mathsf{cbv}}$ by induction hypothesis. So,

$$\begin{aligned} T^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\} &= \langle \mathrm{der}(R^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\})\rangle (Q^{\mathsf{cbv}}\{S^{\mathsf{cbv}}/x\}) \\ &= \langle \mathrm{der}((R\{S/x\})^{\mathsf{cbv}})\rangle (Q\{S/x\})^{\mathsf{cbv}} \\ &= (T\{S/x\})^{\mathsf{cbv}}. \end{aligned}$$

□

Extending the bang calculus: σ-reduction

See p. 6

Proposition 3. *$\to_\sigma$ and $\to_{\sigma_{\mathsf{w}}}$ are strongly normalizing.*

Proof. For any term T, we define two sizes $\mathsf{s}(T)$ and $\#(T)$ by induction on $T \in !\Lambda$:

$$\begin{aligned} \mathsf{s}(x) &:= 1 \\ \mathsf{s}(\lambda x\, T) &:= \mathsf{s}(T) + 1 \\ \mathsf{s}(\langle T\rangle S) &:= \mathsf{s}(T) + \mathsf{s}(S) \\ \mathsf{s}(\mathrm{der}\, T) &:= \mathsf{s}(T) \\ \mathsf{s}(T^!) &:= \mathsf{s}(T) \end{aligned}$$

$$\begin{aligned} \#(x) &:= 1 \\ \#(\lambda x\, T) &:= \#(T) + \mathsf{s}(T) \\ \#(\langle T\rangle S) &:= \#(T) + \#(S) + 2\,\mathsf{s}(T)\,\mathsf{s}(S) - 1 \\ \#(\mathrm{der}\, T) &:= \#(T) + \mathsf{s}(T) \\ \#(T^!) &:= \#(T) \end{aligned}$$

Note that $\mathsf{s}(T) > 0$ and $\#(T) > 0$ for any $T \in !\Lambda$. It is easy to check that if $T \to_\sigma S$ (in particular, if $T \to_{\sigma_{\mathsf{w}}} S$) then $\mathsf{s}(T) = \mathsf{s}(S)$ and $\#(T) > \#(S)$. □

Omitted proofs and remarks of Section 4

See p. 8

Lemma 3 (Strong normalization)**.** *There is no infinite sequence $(t_i)_{i\in\mathbb{N}}$ of resource terms such that: for any $i \in \mathbb{N}$ there is $s \in \overline{2}\langle !\Lambda^{\mathsf{res}}\rangle$ with $t_{i+1} \in s$ and $t_i \to_{\mathsf{b}\sigma} s$.*

Proof. For any $t \in !\Lambda^{\mathsf{res}}$, let $\|t\| := (|t|_{\mathrm{der},\lambda}, \mathsf{s}(t), \#(t))$, where $|t|_{\mathrm{der},\lambda}$ is the number of occurrences of λ and der in t, and $\mathsf{s}(t)$ and $\#(t)$ are defined as follows:

$$\begin{aligned} \mathsf{s}(x) &:= 1 \\ \mathsf{s}(\lambda x\, t) &:= \mathsf{s}(t) + 1 \\ \mathsf{s}(\langle t\rangle s) &:= \mathsf{s}(t) + \mathsf{s}(s) \\ \mathsf{s}(\mathrm{der}\, t) &:= \mathsf{s}(t) \\ \mathsf{s}([t_1, \dots, t_n]) &:= \mathsf{s}(t_1) + \cdots + \mathsf{s}(t_n) \end{aligned}$$

$$\begin{aligned} \#(x) &:= 1 \\ \#(\lambda x\, t) &:= \#(t) + \mathsf{s}(t) \\ \#(\langle t\rangle s) &:= \#(t) + \#(s) + 2\,\mathsf{s}(t)\,\mathsf{s}(s) - 1 \\ \#(\mathrm{der}\, t) &:= \#(t) + \mathsf{s}(t) \\ \#([t_1, \dots, t_n]) &:= \#(t_1) + \cdots + \#(t_n) \end{aligned}$$

It is easy to prove that: if $t \to_{\mathsf{b}} s$ and $t' \in s$, then $|t|_{\mathrm{der},\lambda} > |t'|_{\mathrm{der},\lambda}$; if $t \to_\sigma s$ and $t' \in s$, then $|t|_{\mathrm{der},\lambda} = |t'|_{\mathrm{der},\lambda}$, $\mathsf{s}(t) = \mathsf{s}(t')$ and $\#(t) > \#(t')$. Therefore, if $t \to_{\mathsf{b}\sigma} s$ and $t' \in s$, then $\|t\| > \|t'\|$ according to the lexicographical order on $\mathbb{N}^3$. □

Lemma 14. *Let $t \in !\Lambda^{\mathsf{res}}$, $a \in \mathcal{U}$, $x \in \mathcal{V}ar$ and $m \in \mathcal{M}_{\mathrm{f}}(\mathcal{U})$.*

1. *Let $s_1, \dots, s_n \in !\Lambda^{\mathsf{res}}$, with $n \in \mathbb{N}$. If $|t|_x = 1$, and $\Gamma, x : m \vdash t : a$ and $\Delta \vdash [s_1, \dots, s_n] : m$ with $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Delta)$, then $\Gamma \uplus \Delta \vdash t\{[s_1, \dots, s_n]/x\} : a$.*
2. *Let $k := |t|_x$ and let t' be a x-linearizaiton of t. If $\Gamma, x : m \vdash t : a$ is derivable, then there are $m_1, \dots, m_k \in \mathcal{M}_{\mathrm{f}}(\mathcal{U})$ such that $\Gamma, x_1 : m_1, \dots, x_k : m_k \vdash t' : a$ is derivable and $\biguplus_{i=1}^k m_i = m$.*

Proof. Both points are proved by straightforward induction on $t \in !\Lambda^{\mathsf{res}}$. □

Lemma 4 (Substitution)**.** *Let $n \in \mathbb{N}$, $t, s_1, \dots, s_n \in !\Lambda^{\mathsf{res}}$ and $a, b_1, \dots, b_n \in \mathcal{U}$. Let $x \in \mathcal{V}ar$ with $|t|_x = k$ and $|s_i|_x = 0$ for all $1 \le i \le n$. If $\Gamma_0, x : [b_1, \dots, b_n] \vdash t : a$ is derivable and, for all $1 \le i \le n$, $\Gamma_i \vdash s_i : b_i$ is derivable with $\mathsf{dom}(\Gamma_0) = \cdots = \mathsf{dom}(\Gamma_n)$, then for any x-linearization t' of t there is $\mathsf{f} \in \overline{k}^{\overline{n}}$ such that*

$$\Gamma_0 \uplus \cdots \uplus \Gamma_n \vdash t'\{m_1/x_1, \dots, m_k/x_k\} : a$$

is derivable, where $m_j := [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = j]$ for any $1 \le j \le k$.

See p. 9

Proof. Immediate consequence of Lemmas 14.1-2. □

Taylor expansion

Remark 6. Given $S \in !\Lambda$, one has $\mathsf{fv}(s) \subseteq \mathsf{fv}(S)$ for any resource term $s \in \mathcal{T}(S)$.

Lemma 15 (Simulating substitution via Taylor expansion)**.** *Let $T, S \in !\Lambda$, $x \in \mathcal{V}ar$, $t \in \mathcal{T}(T)$ and $s_1, \dots, s_n \in \mathcal{T}(S)$ with $n \in \mathbb{N}$ and $|t|_x = k$ (the number of free occurrences of x in t). For any $\mathsf{f} \in \overline{k}^{\overline{n}}$, one has $t'\{[s_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\} \in \mathcal{T}(T\{S^!/x\})$ where t' is a x-linearization of t.*

Proof. By induction on $T \in !\Lambda$.

If $T = x$, then $T\{S^!/x\} = S^!$, $t = x$ and $k = 1$. The only element of $\overline{1}^{\overline{n}}$ is the constant function f defined by $\mathsf{f}(i) = 1$ for any $1 \le i \le n$ (in particular, if $n = 0$ then f is the empty function since $\overline{0} = \emptyset$). Therefore, $t'\{[s_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\} = [s_1, \dots, s_n] \in \mathcal{T}(S^!) = \mathcal{T}(T\{S^!/x\})$.

If $T = y \neq x$, then $T\{S^!/x\} = y$, $t = y$ and $k = 0$. If $n \neq 0$ then $\overline{0}^{\overline{n}} = \emptyset$ (as $\overline{0} = \emptyset$) and hence the statement is vacuously true. Otherwise, $n = 0$ and the only element of $\overline{0}^{\overline{0}}$ is the empty function;

therefore, $t'\{[s_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\} = y \in \mathcal{T}(y) = \mathcal{T}(T\{S^!/x\})$.

The other cases follow easily from the induction hypothesis. For instance, if $T = \lambda y\, R$ then we can suppose without loss of generality that $y \notin \mathsf{fv}(S) \cup \{x\}$, thus $T\{S^!/x\} = \lambda y\, R\{S^!/x\}$ and $t = \lambda y\, r$ with $r \in \mathcal{T}(R)$, $y \notin \bigcup_{i=1}^{n} \mathsf{fv}(s_i) \cup \{x\}$ (see Remark 6) and $|r|_x = |t|_x = k$; so, any x-linearization of t has the shape $t' = \lambda y\, r'$ where r' is a x-linearization of r. Let $\mathsf{f} \in \overline{k}^{\overline{n}}$. By inductive hypothesis, $r'\{[s_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\} \in \mathcal{T}(R\{S^!/x\})$, hence $t'\{[s_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\} = \lambda y\, r'\{[s_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [s_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\} \in \mathcal{T}(\lambda y\, R\{S^!/x\}) = \mathcal{T}(T\{S^!/x\})$. □

Lemma 6 (Simulating reduction via Taylor expansion). *For all $T, S \in\ !\Lambda$ and $t \in\ !\Lambda^{\mathsf{res}}$, if $T \to_{\mathsf{b}\sigma_\mathsf{w}} S$ and $t \in \mathcal{T}(T)$, then $t \to_{\mathsf{b}\sigma} s \subseteq \mathcal{T}(S)$ for some $s \in \overline{2}\langle !\Lambda^{\mathsf{res}}\rangle$.*

See p. 9

Proof. By induction on the definition of $T \in\ !\Lambda$.

If $T \mapsto_{\mathsf{b}\sigma} S$ (root-step) then there are the following cases:

- v*-root-step with a bang as argument, i.e.* $T := \langle \lambda x\, R\rangle Q^!$ $\mapsto_{\mathsf{v}} R\{Q^!/x\} =: S$; then, $t = \langle \lambda x\, r\rangle [q_1, \dots, q_n]$ where $r \in \mathcal{T}(R)$ and $q_1, \dots, q_n \in \mathcal{T}(Q)$, for some $n \in \mathbb{N}$. Let $k = |r|_x$ (the number of free occurrences of x in r), let r' be a x-linearization of r and $s := \sum_{\mathsf{f} \in \overline{k}^{\overline{n}}} r'\{[q_i \mid i \in \overline{n},\ \mathsf{f}(i) = 1]/x_1, \dots, [q_i \mid i \in \overline{n},\ \mathsf{f}(i) = k]/x_k\}$. So, $t \to_{\mathsf{v}} s$, and $s \subseteq \mathcal{T}(S)$ by Lemma 15.
- v*-root-step with a variable as argument, i.e.* $T := \langle \lambda x\, R\rangle y$ $\mapsto_{\mathsf{v}} R\{y/x\} =: S$; then, $t = \langle \lambda x\, r\rangle y \to_{\mathsf{ax}} r\{y/x\}$ for some $r \in \mathcal{T}(R)$. It is easy to check (by induction on $r \in\ !\Lambda^{\mathsf{res}}$) that $r\{y/x\} \in \mathcal{T}(R\{y/x\})$, hence $r\{y/x\} \subseteq \mathcal{T}(S)$ if we see $r\{y/x\}$ as a singleton set.
- !*-root-step, i.e.* $T := \mathrm{der}(S^!) \mapsto_! S$; then $t = \mathrm{der}\,[s_1, \dots, s_n]$ where $n \in \mathbb{N}$ and $s_1, \dots, s_n \in \mathcal{T}(S)$. If $n \neq 1$ then $t \to_{[\cdot]} 0$ where $0 \subseteq \mathcal{T}(S)$ since $0 = \emptyset$. Otherwise $t = \mathrm{der}\,[s_1] \to_{[\cdot]} s_1$ and $s_1 \subseteq \mathcal{T}(S)$ if we see s_1 as a singleton set.
- σ_1*-root-step, i.e.* $T := \langle\langle \lambda x\, R\rangle Q\rangle P \mapsto_{\sigma_1} \langle \lambda x\, \langle R\rangle P\rangle Q =: S$ with $x \notin \mathsf{fv}(P)$; then, $t = \langle\langle \lambda x\, r\rangle q\rangle p$ for some $r \in \mathcal{T}(R)$, $q \in \mathcal{T}(Q)$ and $p \in \mathcal{T}(P)$. By Remark 6, $x \notin \mathsf{fv}(p)$, thus $t \mapsto_{\sigma_1} \langle \lambda x\, \langle r\rangle p\rangle q =: s$ with $s \subseteq \mathcal{T}(S)$ if we see s as a singleton set.
- σ_2*-root-step, i.e.* $T := \langle \lambda y\, \lambda x\, R\rangle Q \mapsto_{\sigma_2} \lambda x\, \langle \lambda y\, R\rangle Q =: S$ with $x \notin \mathsf{fv}(Q) \cup \{y\}$; then, $t = \langle \lambda y\, \lambda x\, r\rangle q$ with $r \in \mathcal{T}(R)$, $q \in \mathcal{T}(Q)$ and $p \in \mathcal{T}(P)$. By Remark 6, $x \notin \mathsf{fv}(q) \cup \{y\}$, so $t \mapsto_{\sigma_2} \lambda x\, \langle \lambda y\, r\rangle q =: s$ and $s \subseteq \mathcal{T}(S)$ if we see s as a singleton set.
- σ_3*-root-step, i.e.* $T := \langle R\rangle(\langle \lambda x\, P\rangle Q) \mapsto_{\sigma_3} \langle \lambda x\, \langle R\rangle P\rangle Q =: S$ with $x \notin \mathsf{fv}(R)$; then, $t = \langle r\rangle(\langle \lambda x\, p\rangle q)$ for some $r \in \mathcal{T}(R)$, $q \in \mathcal{T}(Q)$ and $p \in \mathcal{T}(P)$. By Remark 6, $x \notin \mathsf{fv}(r)$, thus $t \mapsto_{\sigma_3} \langle \lambda x\, \langle r\rangle p\rangle q =: s$ with $s \subseteq \mathcal{T}(S)$ if we see s as a singleton set.

If $T \to_{\mathsf{b}\sigma_\mathsf{w}} S$ is not a root-step, then the inductive hypothesis applies straightforwardly. Note that T is not a box by hypothesis. □

Lemma 7 (Relational semantics via Taylor expansion). *Let $S \in\ !\Lambda$ and $\vec{x} = (x_1, \dots, x_n)$ be a repetition-free list of variables (with $n \in \mathbb{N}$) containing all free variables of S. Then, $[\![S]\!]_{\vec{x}} = \bigcup_{s \in \mathcal{T}(S)} [\![s]\!]_{\vec{x}}$.*

See p. 9

Proof. By straightforward induction on $S \in\ !\Lambda$, using Prop. 4. □

Omitted proofs and remarks of Section 5

Lemma 16 (Weakening in contexts). *Let $T \in\ !\Lambda$, $x \notin \mathsf{fv}(T)$, and $(x_1, \dots, x_n)$ be a repetition-free list of variables containing all the free variables of T. If there exist $a \in \mathcal{U}$ and $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that the type judgment $x_1 : m_1, \dots, x_n : m_n \vdash T : a$ is derivable, then $x_1 : m_1, \dots, x_n : m_n, x : [\,] \vdash T : a$ is derivable.*

Proof. By straightforward induction on $T \in\ !\Lambda$. □

Lemma 9 (Semantics of clash-free at depth 0 $\mathsf{b}\sigma_\mathsf{w}$-normal forms). *Let T be a term and $\vec{x} = (x_1, \dots, x_n)$ be a repetition-free list of variables containing all the free variables of T (with $n \in \mathbb{N}$).*

See p. 10

1. *If $T \in\ !\Lambda_\mathsf{d}$, then for any $a \in \mathcal{U}$ there exist $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that $((m_1, \dots, m_n), a) \in [\![T]\!]_{\vec{x}}$.*
2. *If $T \in\ !\Lambda_\ell$ (in particular, if $T \in\ !\Lambda_\mathsf{n}$), then there are $a \in \mathcal{U}$ and $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that $((m_1, \dots, m_n), a) \in [\![T]\!]_{\vec{x}}$.*

Proof. By Proposition 4, for both points it is enough to prove that the type judgment $x_1 : m_1, \dots, x_n : m_n \vdash T : a$ is derivable using the rules in Fig. 1.

1. Proof by induction on the definition of $T \in\ !\Lambda_\mathsf{d}$. Let $a \in \mathcal{U}$.
   If $T = \mathrm{der}\, x$ then $x = x_i$ and $T = \mathrm{der}\, x_i$ for some $1 \leq i \leq n$ by hypothesis, so
   $$\dfrac{\dfrac{}{x_1 : [\,], \dots, x_i : [a], \dots, x_n : [\,] \vdash x_i : [a]}\ {\scriptstyle \mathsf{ax}}}{x_1 : [\,], \dots, x_i : [a], \dots, x_n : [\,] \vdash \mathrm{der}\, x_i : a}\ \mathrm{der}\ .$$
   If $T = \mathrm{der}\, D$ with $D \in\ !\Lambda_\mathsf{d}$, then by induction hypothesis there exist $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that the type judgment $x_1 : m_1, \dots, x_n : m_n \vdash D : [a]$ is derivable. Hence,
   $$\dfrac{\begin{array}{c}\vdots\\ x_1 : m_1, \dots, x_n : m_n \vdash D : [a]\end{array}}{x_1 : m_1, \dots, x_n : m_n \vdash \mathrm{der}\, D : a}\ \mathrm{der}\ .$$
   If $T = \langle D\rangle V$ with $D \in\ !\Lambda_\mathsf{d}$ and $V \in\ !\Lambda_\mathsf{v}$, then by induction hypothesis for some $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ the type judgment $x_1 : m_1, \dots, x_n : m_n \vdash D : ([\,], a)$ is derivable. Thus,
   $$\dfrac{\begin{array}{c}\vdots\\ x_1 : m_1, \dots, x_n : m_n \vdash D : ([\,], a)\end{array} \qquad \dfrac{}{x_1 : [\,], \dots, x_n : [\,] \vdash V : [\,]}\ !}{x_1 : m_1, \dots, x_n : m_n \vdash \langle D\rangle V : a}\ @\ .$$
   If $T = \langle D\rangle D'$ with $D, D' \in\ !\Lambda_\mathsf{d}$, then by induction hypothesis for some $m_1, m_1' \dots, m_n, m_n' \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ the type judgments $x_1 : m_1, \dots, x_n : m_n \vdash D : ([\,], a)$ and $x_1 : m_1', \dots, x_n : m_n' \vdash D' : [\,]$ are derivable. Therefore,
   $$\dfrac{\begin{array}{c}\vdots\\ x_1 : m_1, \dots, x_n : m_n \vdash D : ([\,], a)\end{array} \qquad \begin{array}{c}\vdots\\ x_1 : m_1', \dots, x_n : m_n' \vdash D' : [\,]\end{array}}{x_1 : m_1 \uplus m_1', \dots, x_n : m_n \uplus m_n' \vdash \langle D\rangle D' : a}\ @\ .$$
2. First, we prove by induction on $T \in\ !\Lambda_\mathsf{n}$ the following preliminary statement: for any $T \in\ !\Lambda_\mathsf{n}$ and any repetition-free list $(x_1, \dots, x_n)$ containing all the free variables of T, there exist $a \in \mathcal{U}$ and $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that the type judgment $x_1 : m_1, \dots, x_n : m_n \vdash T : a$ is derivable. If $T \in\ !\Lambda_\mathsf{v}$ (resp. $T \in\ !\Lambda_\mathsf{d}$) then the statement holds by Remark 4 (resp. Lemma 9.1) and Proposition 4. Finally, if $T = \langle \lambda x\, N\rangle D$ with $N \in\ !\Lambda_\mathsf{n}$ and $D \in\ !\Lambda_\mathsf{d}$, then we can suppose without loss of generality that $x \notin \mathsf{fv}(D)$; by induction hypothesis there exist $a \in \mathcal{U}$ and $m_1, \dots, m_n \in \mathcal{M}_\mathrm{f}(\mathcal{U})$ such that the

type judgment $x_1 : m_1, \dots, x_n : m_n \vdash N : a$ is derivable, where $(x_1, \dots, x_n)$ is a repetition-free list of variables containing all the free variables of N and D (and hence of T); moreover, by Lemma 16, we can suppose without loss of generality that $x = x_i$ for some $1 \leq i \leq n$; by Lemma 9.1, there exist $m'_1, \dots, m'_{i-1}, m'_{i+1}, \dots, m'_n \in \mathcal{M}_{\mathrm{f}}(\mathcal{U})$ such that the type judgment $\Gamma' \vdash D : m_i$ is derivable where $\Gamma' = x_1 : m'_1, \dots, x_{i-1} : m'_{i-1}, x'_{i+1} : m'_{i+1}, \dots, x_n : m'_n$, hence

$$\dfrac{\dfrac{\vdots}{x_1 : m_1, \dots, x_n : m_n \vdash N : a}\,\lambda}{\Gamma \vdash \lambda x\, N : (m_i, a)} \qquad \dfrac{\vdots}{\Gamma' \vdash D : m_i}\,@$$
$$\Gamma \uplus \Gamma' \vdash \langle \lambda x\, N \rangle D : a$$

with $\Gamma = x_1 : m_1, \dots, x_{i-1} : m_{i-1}, x_{i+1} : m_{i+1}, \dots, x_n : m_n$. Now we can prove Lemma 9.2. If $T \in !\Lambda_{\mathsf{n}}$, then we are done according to the preliminary statement above. Finally, if $T = \lambda x\, L$ with $L \in !\Lambda_\ell$, then by induction hypothesis there exist $a \in \mathcal{U}$ and $m_1, \dots, m_n \in \mathcal{M}_{\mathrm{f}}(\mathcal{U})$ such that the type judgment $x_1 : m_1, \dots, x_n : m_n \vdash L : a$ is derivable, where $(x_1, \dots, x_n)$ is a repetition-free list of variables containing all the free variables of L (and hence of T); moreover, by Lemma 16, we can suppose without loss of generality that $x = x_i$ for some $1 \leq i \leq n$, hence

$$\dfrac{\dfrac{\vdots}{x_1 : m_1, \dots, x_n : m_n \vdash L : a}\,\lambda}{x_1 : m_1, \dots, x_{i-1} : m_{i-1}, x_{i+1} : m_{i+1}, \dots, x_n : m_n \vdash \lambda x\, L : (m_i, a)}\,.$$

□