A Nonstandard Standardization Theorem

Beniamino Accattoli
Carnegie Mellon University
United States

Eduardo Bonelli
Univ. Nac. de Quilmes and CONICET
Argentina

Delia Kesner
Univ. Paris-Diderot, SPC, PPS, CNRS
France

Carlos Lombardi
Univ. Nac. de Quilmes and Univ. de Buenos Aires
Univ. Paris-Diderot, SPC, PPS, CNRS
Argentina and France

Abstract

Standardization is a fundamental notion for connecting programming languages and rewriting calculi. Since both programming languages and calculi rely on substitution for defining their dynamics, *explicit substitutions* (ES) help further close the gap between theory and practice.

This paper focuses on standardization for the *linear substitution calculus*, a calculus with ES capable of mimicking reduction in λ-calculus and linear logic *proof-nets*. For the latter, proof-nets can be formalized by means of a simple *equational theory* over the linear substitution calculus.

Contrary to other extant calculi with ES, our system can be equipped with a *residual theory* in the sense of Lévy, which is used to prove a left-to-right standardization theorem for the calculus with ES but without the equational theory. Such a theorem, however, does not lift from the calculus with ES to proof-nets, because the notion of left-to-right derivation is not preserved by the equational theory. We then relax the notion of left-to-right standard derivation, based on a *total order* on redexes, to a more liberal notion of standard derivation based on *partial orders*.

Our proofs rely on Gonthier, Lévy, and Melliès' axiomatic theory for standardization. However, we go beyond merely applying their framework, revisiting some of its key concepts: we obtain uniqueness (modulo) of standard derivations in an abstract way and we provide a coinductive characterization of their key abstract notion of *external redex*. This last point is then used to give a simple proof that *linear head reduction*—a *nondeterministic* strategy having a central role in the theory of linear logic—is standard.

Categories and Subject Descriptors F.3.2 [*Logics and Meanings of Programs*]: Semantics of Programming Languages - Operational semantics; F.4.1 [*Mathematical Logic and Formal Languages*]: Mathematical Logic - Lambda calculus and related systems; I.1.3 [*Symbolic and Algebraic Manipulation*]: Languages and Systems - evaluation strategies

General Terms Theory, Languages.

Keywords Lambda-calculus, Standardization, Residuals, Explicit Substitutions, Linear Logic, Proof-Nets.

1. Introduction

In his seminal paper [35] Plotkin introduced the idea of relating a calculus (given by means of an equational theory induced by a rewriting relation) and a programming language (specified as an abstract machine) via a *standardization* theorem: the programming language implements the standard strategy associated to the calculus. Let us recall what standardization is about. The idea is to identify a class of derivations, called standard, that is *complete*: whenever $t \twoheadrightarrow u$ then there is a standard derivation from t to u. Often, it is possible to describe a reduction strategy that produces standard derivations only (the leftmost-outermost strategy in the λ-calculus), and for which the normalization property holds as a corollary of the standardization theorem: if t has a normal form then the standard strategy will reach this normal form. Completeness and normalization are the justifications for Plotkin's approach.

Another approach to close the gap between calculi and programming languages is to resort to *explicit substitutions* (ES), that are refinements of λ-calculus where evaluation is decomposed into small steps, and that can be thought as a framework in which to specify abstract machines as strategies.

It would be natural to expect calculi with ES to enjoy standardization theorems whose induced standard strategies justify abstract machines. Surprisingly, the literature does not present any such relationship. This is due to some inherent difficulties. Calculi with ES are complex and fragile rewriting systems for which already studying normalization is technically demanding. Standardization—that mixes confluence and termination arguments—generally is simply out of scope. In particular, ES calculi present *critical pairs* and thus lack *orthogonality*, a basic syntactic property enjoyed by the λ-calculus and on which most standardization techniques rely on.

This paper makes significant progress on standardization for ES. The starting point is a recent advance in the field of ES. A new generation of calculi with ES *at a distance*, having very simple meta-theories and conceived as *behaviorally equivalent copies* of graphical languages such as linear logic proof-nets, have recently been introduced by Accattoli and Kesner in [6] (see also [2, 3, 5]).

POPL '14, January 22–24, 2014, San Diego, CA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2544-8/14/01…$15.00.
http://dx.doi.org/10.1145/2535838.2535886

Traditional calculi with ES implement the ordinary substitution by percolating ES through the term structure until they reach variable occurrences, on which they finally substitute or get garbage collected. The key feature of distance calculi is that the *percolating process* is factored out from the computational process. Indeed, ES at a distance do not move: their dynamics is defined using contexts (*i.e.* terms with holes) that allows them to act directly on single variable occurrences (via a context isolating the occurrence), without any need to commute with the other constructors in between.

The Framework. In this paper we focus on a specific calculus at a distance, *the linear substitution calculus* $\lambda_{\mathtt{lsub}}$, that is both a slight generalization of a calculus by Robin Milner [33] (related to bigraphs), from which it inherits the *substitution rules* at a distance, and a slight modification of the structural λ-calculus by Accattoli and Kesner [6] (related to proof nets). Such a calculus has some relevant features:

1. *Linear logic and concurrency*: it is tightly connected with the translations of λ-calculus into linear logic [4] and the π-calculus [3]. In particular, reduction rules for ES act as exponential cut-elimination rules in linear logic (resp. replication in the π-calculus).
2. *Linear head reduction (LHR)*: given the deep connection between $\lambda_{\mathtt{lsub}}$ and proof-nets [4], our calculus naturally expresses *linear head reduction*, a notion of evaluation for proof nets that is connected with other fundamental models [3, 5, 14, 15, 17, 29]. Notably, LHR is a strategy for $\lambda_{\mathtt{lsub}}$, while it *cannot* be expressed as a strategy in λ-calculus.
3. *Simplicity* and *expressiveness*: the calculus is simple (only 3 rewriting rules) and it enjoys numerous desirable properties including: a tight correspondence with the lambda calculus (simulating β-reduction), confluence on terms and metaterms, preservation of strong β-normalisation and—as we will show—a theory of residuals.

The linear substitution calculus—as do all other extant calculi with ES—presents critical pairs, and so standardization is non-trivial. However, $\lambda_{\mathtt{lsub}}$ enjoys a unique property, that we call *semantical orthogonality*.

Orthogonality. Orthogonal (first and higher-order) term rewriting systems are defined as left-linear systems without critical pairs. They are the most studied class of rewriting systems, including in particular the λ-calculus. This notion of *syntactic orthogonality* is handy and simple, but it has the drawback of being too restrictive. Fortunately, orthogonality can be defined in a more abstract way, as the fact that residuals—a standard concept in rewriting theory (see Section 3)—are sufficient to close local confluence diagrams. This form of *semantical orthogonality* is a particular property of syntactic orthogonality, but it is of a behavioral nature and more general. The crucial point is that $\lambda_{\mathtt{lsub}}$ is not syntactically orthogonal but—in contrast to any other calculus with ES—is semantically orthogonal, as we prove in this paper. In this sense $\lambda_{\mathtt{lsub}}$ is the first calculus with ES that conservatively refines λ-calculus, preserving its orthogonality. Unfortunately, most techniques for standardization rely on syntactical orthogonality. An exception is the abstract theory of standardization developed by Gonthier, Lévy, and Melliès (GLM) [19], and later refined by Melliès [30, 32]. Such a theory can cope even with non semantically orthogonal calculi as the $\lambda\sigma$-calculus, but it takes a much simpler form in presence of semantical orthogonality.

Left-to-Right Standardization and Beyond. Two first contributions of this paper are 1) a theory of residuals for $\lambda_{\mathtt{lsub}}$ and 2) a proof of the axioms of GLM's theory. These technical results allow us to obtain a *left-to-right*[1] standardization theorem generalizing the well-known standardization theorem for λ-calculus, and capturing the *deterministic* weak (*i.e.* not reducing under abstractions) variant of LHR—that is the variant that actually matches evaluation in the Krivine Abstract Machine and the π-calculus—as a standard strategy. It is pleasing that the properties of left-to-right derivations extend naturally to $\lambda_{\mathtt{lsub}}$. However, such a standardization theorem is not fully satisfying.

A first reason is that LHR (without the weak restriction) is a nondeterministic strategy whose sequences are not necessarily left-to-right standard. However, the nondeterminism of LHR does not affect the final result nor the length of evaluation sequences (technically, LHR enjoys the diamond property). It is then disappointing that LHR is not left-to-right standard, because LHR is a crucial notion in the theory of linear logic.

A second reason for not being satisfied with the left-to-right standardization theorem concerns the close relationship between the linear substitution calculus and linear logic proof-nets, the model behind its inception. Terms and proof-nets are behaviorally equivalent in a strong sense: every term t maps to a proof-net P_t, and every evaluation *step* on t or P_t maps to an evaluation *step* on the other. Additionally, the redexes are in bijection and so concepts such as residuals transfer from terms to proof-nets and viceversa. The interest of proof-nets is that they provide a quotient of terms; remarkably, this quotient can be explicitly characterized by a simple *equational theory* $\sim$ (in the style of [6, 23], and here generated by 3 equations only) turning the behavioral equivalence with linear logic proof-nets into a true isomorphism: for every proof-net P there is a $\sim$-class containing all and only the terms behaviorally equivalent to P. Consequently, any notion or result that is stable by equivalence $\sim$ immediately lifts to proof-nets. Unfortunately, left-to-right derivations in the linear substitution calculus are not stable by $\sim$, because $\sim$ swaps some constructors, inverting the relative position of some redexes. Thus, our left-to-right standardization theorem does not hold for proof-nets. This is quite disappointing because proof nets provide an *operational model*, and one would like to have a notion of standard derivation that can be freely transported from the language to the model and viceversa. We then refine our left-to-right standardization theorem.

Partial Standardization Orders. Standard derivations are defined as those reductions that respect a certain order on redexes, for instance the left-to-right order. In GLM's theory such a total order is replaced by a *partial* order, providing a more general but also subtler setting. To achieve a notion of standard derivation that is stable by $\sim$ we are forced to relax the total left-to-right order into a partial order. In the setting of total orders, given a derivation $\rho : t \twoheadrightarrow u$ there is a *unique* standard derivation ρ' to which ρ standardizes, and moreover a deterministic and normalizing standard strategy can be easily obtained by selecting the *minimum* redex according to the order. When the order is partial, instead, there can be many standard derivations to which a derivation standardizes, *i.e.* uniqueness is lost: a standard derivation is an ordered derivation but only up to swaps of $\prec$-disjoint steps. Moreover, although GLM's theory does support partial orders, it requires a number of axioms to hold, some of which are not enjoyed by $\lambda_{\mathtt{lsub}}$ modulo $\sim$. So we shall switch to the axiomatics of Melliès' PhD thesis [32], which is more general and allows us to prove existence of standard derivations for a suitable partial order. However, alas, $\lambda_{\mathtt{lsub}}$ modulo $\sim$ does not even satisfy the axioms of [32] required to obtain uniqueness of stan-

[1] We distinguish between the *leftmost strategy* (that repeatedly reduces the leftmost redex) from *left-to-right* standardization that re-arranges redexes (not necessarily involving the leftmost redex) from left to right. We also use *leftmost* (considering terms as strings of symbols) for what is sometimes called *leftmost-outermost* (that is relative to terms seen as trees).

dard derivations. Nor does it supply us with the guarantee that reducing *external* redexes—the generalization to partial orders of the notion of minimum redexes in the setting of total orders—leads to normal forms. We circumvent these inadequacies of the axiomatic framework by exploiting the embedding of a partial order into a total order, thus proving uniqueness (modulo swappings) of standard derivations and the normalization of a (specific) external strategy.

Main Contributions. A summary of our results follows:

1. **The Standardization Theorem**. We identify a partial order $\prec_{\mathtt{B}}$ that is related to the concept of exponential box in linear logic proof-nets, and that is stable by $\sim$. We show that Melliès' axioms for existence of a standard derivation hold for $\prec_{\mathtt{B}}$, obtaining a standardization theorem for $\lambda_{\mathtt{lsub}}$ modulo $\sim$.
2. **Uniqueness Modulo**. For partial orders, GLM's theory has some additional axioms that ensures uniqueness modulo independent swaps (see Section 8) but one of these axioms *does not hold* for $\lambda_{\mathtt{lsub}}$. Nonetheless, we prove uniqueness in an abstract way, relying on the embedding of the partial box order into the total left-to-right order.
3. **Coinductive External Redexes**. In GLM's theory external redexes are defined somewhat indirectly, via an extraction process [32]. We clarify this concept by providing a direct and simple *coinductive* characterization: an external redex is a minimal redex that is *persistently minimal*, *i.e.* whose (unique) residual after any other redex is still external.
4. **Applications**. We use the coinductive characterization of external redexes to obtain a *normalization theorem* for the leftmost strategy and to give a concise proof by coinduction of the fact that LHR is a standard strategy.

The *nonstandard* character of our standardization result comes at least from the following facts: highlight of the first semantical orthogonal calculus with ES being isomorphic to linear logic proof-nets and enjoying standardization; use of a partial (vs total) order on redexes being stable by equivalence classes; proof of uniqueness (modulo) of standard derivations by means of simple arguments and an original application of GLM's theory to a non-syntactical orthogonal case.

Abstract Machines. As mentioned before, we will show that weak LHR is a standard strategy. In parallel, Accattoli, Barenbaum and Mazza [7] studied the relationship between abstract machines and calculi at a distance obtaining that weak LHR is the strategy implemented by the Krivine Abstract Machine, up to a certain notion of structural equivalence. This result refines a similar one by Danos and Regnier [14], and completes the lifting of Plotkins's approach (relating the SECD machine with a standard strategy in the call-by-value λ-calculus) to the granularity of explicit substitutions. Moreover, [7] contains similar correspondences between machines for call-by-value (the CEK and Leroy's ZINC) and call-by-need (Sestoft's machine) and some variations over weak LHR. We are confident that the new understanding of standardization for ES provided by our work will lead to analogous standardization theorems for call-by-value and call-by-need variant of the linear substitution calculus, completing the picture.

Related Work. A brief history of standardization may be found in [30]. Standardization was introduced in 1958 by Curry and Feys [13] for the lambda calculus. Lévy [27] strengthens this result by defining the notion of permutation equivalence and showing that every derivation has a unique standard derivation in each equivalence class. Huet and Lévy [21] extend these ideas to first-order term rewriting without critical pairs and also show that one must abandon the left-to-right order for more general orders in these systems. Klop gives the first abstract proof of standardization and applies these ideas to higher-order rewriting [26]. Boudol [10] extends Huet and Lévy's work to term rewriting systems with critical pairs. Clark and Kennaway [12], Khasidashvili and Glauert [25], Paolini and Ronchi Della Rocca [34], and Xi [41] provide further generalizations along various axes. Gonthier, Lévy and Melliès [19] introduce an axiomatic approach to standardization, later refined by Melliès [30, 32], that has turned out to be convenient to deal with our case, due to its abstract nature. Indeed, simple techniques such as Plotkin's inductive reasoning in [35] or the *proof terms* technique by van Oostrom and De Vrijer [40]—extended to the higher-order case by Bruggink [11]—are not easily adaptable to our equational setting. Takahashi's technique [37] does not apply either. There is no relationship between our results and standardization by levels for linear logic proof-nets[2]: both approaches use partial orders, but such orders are incomparable. This is a further reason to consider our result as nonstandard.

Plan of the Paper. The next section introduces the linear substitution calculus. Section 3 defines residuals and Section 4 shows that residuals are compatible with the equivalence $\sim$ in a very strong sense. Section 5 explains the ideas behind the abstract approach to standardization and Section 6 proves the left-to-right standardization theorem. Section 7 explains how the left-to-right order can be relaxed to cope with the equational theory and Section 8 proves our equational standardization theorem. Section 9 provides uniqueness of standard derivations. Section 10 presents a coinductive characterization of external redexes and Section 11 proves the normalization theorem and the fact that (weak) linear head reduction is standard.

2. The Linear Substitution Calculus

The set of terms of the *linear substitution calculus*, denoted by $\mathcal{T}$, is generated by the following grammar:

$$t \ ::= \ x \ \mid \ tt \ \mid \ \lambda x.t \ \mid \ t[x/t]$$

A term x is called a **variable**, tu an **application**, $\lambda x.t$ an **abstraction** and $t[x/u]$ an **explicit substitution**. The notions of free and bound variables are defined as usual plus $\mathtt{fv}(t[x/u]) := \mathtt{fv}(t) \setminus \{x\} \cup \mathtt{fv}(u)$. We work with the standard notion of α-conversion (*i.e.* renaming of bound variables for abstractions and substitutions). We use C to denote a **context** w.r.t. the previous grammar (*i.e.* a term with a unique occurrence of a designated symbol $\Box$ called the *hole*). We write $C[t]$ for the term obtained by replacing the hole of C by the term t. We write $C[\![u]\!]$ when the free variables of u are not captured by the context C, *i.e.* there are no abstractions or explicit substitutions in C that bind the variables of $\mathtt{fv}(u)$.

The $\lambda_{\mathtt{lsub}}$**-calculus** is given by the set of terms $\mathcal{T}$ and by the **reduction relation** $\rightarrow_{\lambda_{\mathtt{lsub}}}$ defined as the union of $\rightarrow_{\mathtt{dB}}$, $\rightarrow_{\mathtt{ls}}$, and $\rightarrow_{\mathtt{gc}}$, which are the closure by contexts C of the following rewriting rules, where $\mathtt{L}$ denotes a (possibly empty) list of substitutions $[x_1/t_1]\ldots[x_k/t_k]$:

$$\begin{array}{llll} (\lambda x.t)\mathtt{L}u & \mapsto_{\mathtt{db}} & t[x/u]\mathtt{L} & \\ C[\![x]\!][x/u] & \mapsto_{\mathtt{ls}} & C[\![u]\!][x/u] & \\ t[x/u] & \mapsto_{\mathtt{gc}} & t & \text{if } x \notin \mathtt{fv}(t) \end{array}$$

The names $\mathtt{db}$, $\mathtt{ls}$, and $\mathtt{gc}$ stand for *distant beta*, *linear substitution*, and *garbage collection*, respectively. Rule $\mapsto_{\mathtt{db}}$ (resp. $\mapsto_{\mathtt{ls}}$) comes from the structural λ-calculus [6] (resp. Milner's calculus [33]), while $\mapsto_{\mathtt{gc}}$ belongs to both calculi. In $\mathtt{db}$ we may assume w.l.o.g. that $\bigcup_{i=1}^{k}\{x_i\} \cap \mathtt{fv}(u) = \emptyset$ and $x \notin \mathtt{fv}(u) \cup \bigcup_{i=1}^{k} \mathtt{fv}(t_i)$.

Note that the meta-notation $\mathtt{L} = [x_1/t_1]\ldots[x_k/t_k]$ can also be seen as a context $\Box[x_1/t_1]\ldots[x_k/t_k]$. This fact, together with

[2] Such a standardization theorem is folklore in the linear logic community. See [2, 16] for details.

the use of a context C in the second rule, and the global side condition in the third rule, justify the idea of rewriting rules *at a distance*. The substitution context in rule $\mathtt{db}$ is motivated by its encoding in proof-nets, where explicit substitutions are partially free to float (*i.e.* to traverse some term constructors). Such *freedom* is formalized by the forthcoming graphical equivalence and by the study of its properties in Section 4.

The linear substitution calculus enjoys all the properties required of calculi with ES (including simulation of β-reduction, preservation of strong normalisation, confluence on terms and metaterms and full composition), whose proofs are simple and omitted, as they are minor variations over those for Milner's calculus [24], or those of the structural λ-calculus [6].

In order to study residuals we need to fix a precise terminology about redexes. A **redex occurrence** in a term t is either a tuple $\langle D, r\rangle$ where $t = D[r]$ and $r = (\lambda x.s)\mathtt{L}u$ (a $\mathtt{dB}$**-redex**) or a tuple $\langle D, r, C\rangle$ where $t = D[r]$ and $r = C[\![x]\!][x/u]$ (a $\mathtt{ls}$**-redex**), or a tuple $\langle D, r\rangle$ where $t = D[r]$, $r = s[x/u]$ and $x \notin \mathtt{fv}(s)$ (a $\mathtt{gc}$**-redex**). For example, the term $t = (xx)[x/y]$ has two different redex occurrences, namely $\langle D_1, r_1, C_1\rangle = \langle\Box, (xx)[x/y], \Box x\rangle$ and $\langle D_2, r_2, C_2\rangle = \langle\Box, (xx)[x/y], x\Box\rangle$. The **pattern** of a redex is r, the second component of the tuple; the **box**[3] of the redex is the subterm of the pattern noted u, and the **context** of the redex is D, the first component of the tuple. We use $A, B, \ldots$ for redexes occurrences and $\mathtt{Red}(t)$ for the set of all redex occurrences of t. Given a redex occurrence $A \in \mathtt{Red}(s)$ we write $s \xrightarrow{A} t$ for the **reduction step** obtained by contracting A in s.

The **graphical equivalence** $\sim$ is given by the contextual, transitive, symmetric and reflexive closure of α-conversion (*i.e.* renaming of bound variables) and the following axioms:

$$\begin{array}{llll} t[x/u][y/v] & \approx_{\mathtt{CS}} & t[y/v][x/u] & x \notin \mathtt{fv}(v) \;\&\; y \notin \mathtt{fv}(u) \\ (\lambda y.t)[x/u] & \approx_{\sigma_1} & \lambda y.t[x/u] & y \notin \mathtt{fv}(u) \\ (tv)[x/u] & \approx_{\sigma_2} & t[x/u]v & x \notin \mathtt{fv}(v) \end{array}$$

This equivalence characterizes exactly the representation of terms as proof-nets, in the sense that $t \sim u$ iff t and u map to the same proof-net [4].

The $\lambda_{\widetilde{\mathtt{lsub}}}$**-calculus** is given by the set of terms $\mathcal{T}$ and by the **reduction relation** $\rightarrow_{\lambda_{\widetilde{\mathtt{lsub}}}}$, defined by the rewriting rules $\{\mathtt{db}, \mathtt{gc}, \mathtt{ls}\}$ modulo the equations $\{\mathtt{CS}, \sigma_1, \sigma_2\}$, *i.e.* , $t \rightarrow_{\lambda_{\widetilde{\mathtt{lsub}}}} u$ iff $t \sim t' \rightarrow_{\lambda_{\mathtt{lsub}}} u' \sim u$. Thus in particular $t \rightarrow_{\lambda_{\mathtt{lsub}}} t'$ implies $t \rightarrow_{\lambda_{\widetilde{\mathtt{lsub}}}} t'$.

Given any reduction relation $\mathcal{R}$, an $\mathcal{R}$-**derivation** is a (possibly empty) sequence of reduction steps having the form $s = t_0 \rightarrow_{\mathcal{R}} t_1 \rightarrow_{\mathcal{R}} \ldots$. An $\mathcal{R}$-derivation is **finite** iff the associated sequence is finite. We write $d, e, \ldots$ for derivations and $d : s \twoheadrightarrow_{\mathcal{R}} t$ when we stress the **source** s and **target** t of the (finite) derivation. We write ϵ for the empty derivation and $A; d : s \twoheadrightarrow t$ for the derivation composing $s \xrightarrow{A} u$ to the left of $d : u \twoheadrightarrow t$. A reduction relation $\mathcal{R}$ is said to be **terminating** iff every $\mathcal{R}$-derivation is finite. An element t is said to be in $\mathcal{R}$**-normal form** iff there is no u such that $t \rightarrow_{\mathcal{R}} u$.

3. Residuals

The residuals of a redex A after a reduction step $t \xrightarrow{B} u$ are the traces of the redex A that appear in the target term u. The crucial property that residuals need to satisfy is that any local confluence diagram $u_1 \xleftarrow{A_1} t \xrightarrow{A_2} u_2$ can be closed by reducing in u_1 (resp. u_2) all and only the residuals of the redex A_2 after A_1 (resp. of A_1 after A_2). When this happens for every possible local confluence diagram we say that the system is **semantically orthogonal**. The subtlety in the study of residuals is that redexes may be duplicated or erased along the way and duplication may even nest two residuals of the same redex. For further references the reader may consult [20, 27, 31, 38].

Another simpler notion of orthogonality is *syntactical orthogonality* [21], and happens when the system is left-linear and has no critical pairs. Syntactical orthogonality implies semantic orthogonality but the converse does not hold. In particular, the $\lambda_{\mathtt{lsub}}$-calculus is not syntactically orthogonal but, as we will show in Section 6, it turns out to be semantically orthogonal, a property that no other calculus with ES enjoys. In order to give an intuition on such a phenomenon let us consider a calculus with ES such as $\lambda\sigma$ [1] or $\lambda\mathtt{x}$ [9] containing at least the following reduction rules:

$$\begin{array}{lll} (\lambda x.t)u & \rightarrow_{\mathtt{b}} & t[x/u] \\ (tu)[x/v] & \rightarrow_{@} & t[x/v]u[x/v] \end{array}$$

The following critical pair arises:

$$(\lambda x.t)[y/v]u[y/v] \;\; {}_{@}\!\leftarrow \;\; ((\lambda x.t)u)[y/v] \;\; \rightarrow_{\mathtt{b}} \;\; t[x/u][y/v]$$

However, the @-step has no residual after the $\mathtt{b}$-step. The $\mathtt{b}$-step also has no residual (note how it would have a residual if the rule were *at a distance*). The diagram can be closed, but only reducing *created* redexes, and so the calculus is not semantically orthogonal.

The formal development of residuals is based on a notion of (well-)labeled term to be introduced next. Note that since $\lambda_{\mathtt{lsub}}$ is not syntactically orthogonal the technique of underlining the redex pattern [26] cannot be applied, *e.g.* in $\underline{(xx)[x/s]}$ the underlining does not distinguish between the two occurrences of the $\mathtt{ls}$ redexes.

Labels. In order to follow redexes along a derivation we mark them with special symbols called **labels**, denoted $\alpha, \beta, \gamma, \ldots$. The obtained set of labeled terms, denoted by $\mathcal{T}_{\mathcal{L}}$, is generated by the following grammar.

$$t ::= x \mid x^{\alpha} \mid tt \mid \lambda x.t \mid \lambda x^{\alpha}.t \mid t[x/t] \mid t[x^{\alpha}/t]$$

The notations $x^{(\alpha)}$, $\lambda x^{(\alpha)}.t$ and $t[x^{(\alpha)}/t]$ mean that x may or may not be labeled. We write $\mathtt{Lab}(t)$ to denote the **set of all the labels** of t and t° to denote the term obtained from t by **removing all its labels**. Thus for example $((x^{\alpha}y^{\beta})[y/\lambda z^{\gamma}.z])^{\circ} = (xy)[y/\lambda z.z]$.

We extend the meta-notation $\mathtt{L}$ to lists of possibly labeled substitutions, and C to possibly labeled contexts. Similarly, the notions of free and bound variables are extended to labeled terms as expected together with their corresponding notion of α-conversion. We use $\mathtt{flv}(t)$ to denote the subset of $\mathtt{fv}(t)$ having at least one labeled occurrence, *e.g.* $\mathtt{flv}(x^{\alpha}y^{\beta}[y/z]) = \{x\}$.

Labeled reduction $\xrightarrow{\alpha}$ on labeled terms is defined as the contextual closure of the following rewriting rules:

$$\begin{array}{llll} (\lambda x^{\alpha}.t)\mathtt{L}u & \overset{\alpha}{\mapsto}_{\mathtt{dB}} & t[x/u]\mathtt{L} & \\ C[\![x^{\alpha}]\!][x/u] & \overset{\alpha}{\mapsto}_{\mathtt{ls}} & C[\![u]\!][x/u] & \\ t[x^{\alpha}/u] & \overset{\alpha}{\mapsto}_{\mathtt{gc}} & t & x \notin \mathtt{fv}(t) \end{array}$$

Definition of redexes for terms naturally extends to labeled ones. A **labeled redex** A is a redex having a pattern of the form $(\lambda x^{\alpha}.t)\mathtt{L}u$, $C[\![x^{\alpha}]\!][x/u]$ or $t[x^{\alpha}/u]$, and α is called the **label of the redex** A. The **anchor** of a redex (labeled or not) is the variable possibly carrying its label. We will usually associate the labels α, β and γ to the redex names A, B and C respectively; occasionally we will write A_{α} to emphasize that α is the label of the redex A. We write $\mathtt{Red}_{\alpha}(t)$ for all the redexes of t labeled with α.

In order to show some key properties required by the axiomatic approach (*cfr.* Section 6) we will work with a subset of labeled terms, written $\mathcal{T}_{\mathcal{WL}}$, called **well-labeled terms**, and defined by:

- $x \in \mathcal{T}_{\mathcal{WL}}$ and $x^{\alpha} \in \mathcal{T}_{\mathcal{WL}}$

[3] When terms are represented as linear logic proof-nets, what we call *box* corresponds exactly to the exponential box.

- If $t \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{flv}(t)$ then $\lambda x.t \in \mathcal{T}_{\mathcal{WL}}$
- If $t, u \in \mathcal{T}_{\mathcal{WL}}$, then $tu \in \mathcal{T}_{\mathcal{WL}}$
- If $(\lambda x.t)\mathtt{L}, u \in \mathcal{T}_{\mathcal{WL}}$, then $(\lambda x^{\alpha}.t)\mathtt{L}u \in \mathcal{T}_{\mathcal{WL}}$
- If $t, u \in \mathcal{T}_{\mathcal{WL}}$, then $t[x/u] \in \mathcal{T}_{\mathcal{WL}}$
- If $t, u \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{fv}(t)$, then $t[x^{\alpha}/u] \in \mathcal{T}_{\mathcal{WL}}$.

Note that $\lambda x^{\alpha}.x$, $\lambda x.x^{\alpha}$ and $x[x^{\alpha}/u]$ are not in $\mathcal{T}_{\mathcal{WL}}$. Note also that subterms of well-labeled terms are not necessarily well-labeled (*e.g.* the abstraction of a labeled db-redex). Well-labeled terms are stable by reduction and graphical equivalence:

LEMMA 1. *Let $t \in \mathcal{T}_{\mathcal{WL}}$. If $t \xrightarrow{\alpha} u$ or $t \sim u$, then $u \in \mathcal{T}_{\mathcal{WL}}$.*

Residuals. Here we define residuals for the reduction relation $\lambda_{\mathtt{lsub}}$. In Section 4 we will extend them to the $\lambda^{\sim}_{\mathtt{lsub}}$.

A term t can be labeled in different ways, leading to different variants of t. More precisely, we say that t is a **variant** of u iff $t^{\circ} = u^{\circ}$. Thus in particular, t is a variant of itself. If t is a variant of u, then we will consider the *obvious bijection* between the sets of redexes of t and u; sometimes, we will even identify some particular redexes of them which are related by this bijection.

To define the residuals of a redex $A \in \mathtt{Red}(t)$ after a step $t \xrightarrow{B} u$, we consider the **α-lift of** t ***w.r.t.*** A **by** α, written $\mathtt{lift}(t, A, \alpha)$, which is a particular variant of t obtained as follows. Let α be a fresh symbol, *i.e.* $\alpha \notin \mathtt{Lab}(t)$. If the redex A is already labeled in t, then change the label of A to α, otherwise, if A is not already labeled, assign α to A. Let us write $\mathtt{lift}(t, A, \alpha) \xrightarrow{B} u$ for the induced associated reduction step. Then, the **set of residuals of** A **after** B is given by $A\langle B\rangle := \{\mathtt{Red}_{\alpha}(u) \mid \mathtt{lift}(t, A, \alpha) \xrightarrow{B} u\}$.

It is clear that this definition is completely *independent* from the variant used to lift the term t. We write $A\langle B\rangle A'$ for $A' \in A\langle B\rangle$, and $\mathcal{A}\langle B\rangle$ for $\bigcup_{A \in \mathcal{A}} A\langle B\rangle$. For example, taking $v = (x^{\beta}x^{\beta}x^{\beta})[x/y]$, $B = \langle \Box, v, x^{\beta}\Box x^{\beta}\rangle$ (so that $v \xrightarrow{B} (x^{\beta}yx^{\beta})[x/y] = v'$) and $\mathcal{A} = \mathtt{Red}_{\beta}(v) \setminus B$, we have $\mathcal{A}\langle B\rangle = \{B', B''\}$, where $B' = \langle \Box, v', \Box y x^{\beta}\rangle$ and $B'' = \langle \Box, v', x^{\beta}y\Box\rangle$.

Creation. Given a reduction step $t \rightarrow_{\lambda_{\mathtt{lsub}}} u$, the set of redexes in u that are not residuals of redexes in t are said to be **created**. While there are only 3 possible ways to create redexes in λ-calculus [27], in the $\lambda_{\mathtt{lsub}}$-calculus redexes may be created in one of 6 possible ways, we just show below one example for each case, where the reduced redex has label α and the *created* redex is underlined:

$$\begin{array}{llll}
\mathtt{db}\text{ creates }\mathtt{db}: & ((\lambda x^{\alpha}.(\lambda y.s))u)\,v & \rightarrow_{\mathtt{dB}} & \underline{(\lambda y.s)[x/u]v} \\
\mathtt{db}\text{ creates }\mathtt{ls}: & (\lambda x^{\alpha}.x)u & \rightarrow_{\mathtt{dB}} & \underline{x[x/u]} \\
\mathtt{db}\text{ creates }\mathtt{gc}: & (\lambda x^{\alpha}.y)u & \rightarrow_{\mathtt{dB}} & \underline{y[x/u]} \\
\mathtt{ls}\text{ creates }\mathtt{db}: & x^{\alpha}[x/\lambda y.s]u & \rightarrow_{\mathtt{ls}} & \underline{(\lambda y.s)[x/\lambda y.s]u} \\
\mathtt{ls}\text{ creates }\mathtt{gc}: & x^{\alpha}[x/y] & \rightarrow_{\mathtt{ls}} & \underline{y[x/y]} \\
\mathtt{gc}\text{ creates }\mathtt{gc}: & z[y^{\alpha}/x][x/u] & \rightarrow_{\mathtt{gc}} & \underline{z[x/u]}
\end{array}$$

Note that dB-steps can create every kind of redex, ls-steps can create $\{\mathtt{dB}, \mathtt{gc}\}$-redexes, and gc-redexes can only create other gc-redexes. Note also that whenever a labeled redex creates a redex A, then A is not labeled.

Developments. Labels are particularly useful to study *developments*, *i.e.* derivations in which only residuals of an initial set of redexes are contracted. For that, we first extend the concept of residual to finite derivations as follows: $\mathcal{A}\langle\epsilon\rangle := \mathcal{A}$, $\mathcal{A}\langle B; d\rangle := (\mathcal{A}\langle B\rangle)\langle d\rangle$.

Given $\mathcal{A} \subseteq \mathtt{Red}(t)$, a (possibly infinite) derivation $t = t_0 \xrightarrow{A_1} t_1 \xrightarrow{A_2} t_2 \ldots$ is a **development** of $\mathcal{A}$ iff $A_i \in \mathcal{A}\langle A_1; \ldots; A_{i-1}\rangle$ for all i. For example, in the $\lambda_{\mathtt{lsub}}$-calculus, $t_0 = (xx)[x/I] \rightarrow (xI)[x/I] \rightarrow (II)[x/I]$ is a development of the set containing the two ls-redexes of t_0. In the sequel, $\mathcal{A}_1; ...; \mathcal{A}_n$ denotes any derivation $d_1; ...; d_n$ such that d_i is a development of the set of redexes $\mathcal{A}_i$ for all i. The *finite developments (FD)* property states that given a term t and a set of redexes $\mathcal{A} \subseteq \mathtt{Red}(t)$, any development of $\mathcal{A}$ starting at t terminates. This property (proved in Section 6) is one of the key requirements for the axiomatic theory of standardization that we are going to apply. The residual of a derivation over a set of redexes $\mathcal{A}$ is defined as follows: $\epsilon\langle\mathcal{A}\rangle = \epsilon$ and $(B; d)\langle\mathcal{A}\rangle = B\langle e\rangle; d\langle\mathcal{A}\langle B\rangle\rangle$, where e is any development of $\mathcal{A}$.

4. Working with Equivalence Classes

In this section we first define the notion of residual of a redex along an equation, so that we will be able to trace redexes along $\lambda^{\sim}_{\mathtt{lsub}}$-derivations. We then show that the notion of residual defined in Section 3 is well-defined w.r.t equivalence classes, *i.e.* 1) the residual relation yields a unique bijection between two redexes in the same $\sim$-equivalence class and 2) residuals of rewriting steps lift to $\sim$-equivalence classes.

The **labeled graphical equivalence**[4] $\sim$ on *labeled terms* is given by the contextual, transitive, symmetric and reflexive closure of α-conversion and the following axioms:

$$\begin{array}{llll}
t[x^{(\alpha)}/u][y^{(\beta)}/v] & \approx_{\mathtt{CS}} & t[y^{(\beta)}/v][x^{(\alpha)}/u] & x \notin \mathtt{fv}(v)\ \&\ y \notin \mathtt{fv}(u) \\
(\lambda y^{(\beta)}.t)[x^{(\alpha)}/u] & \approx_{\sigma_1} & \lambda y^{(\beta)}.t[x^{(\alpha)}/u] & y \notin \mathtt{fv}(u) \\
(tv)[x^{(\alpha)}/u] & \approx_{\sigma_2} & t[x^{(\alpha)}/u]v & x \notin \mathtt{fv}(v)
\end{array}$$

The axioms are to be understood in such a way that each label occurs either in both sides of the axiom or in none of them. An equational proof of $t \sim u$ is a **derivation tree** ending in $t \sim u$ which uses α-conversion, the axioms $\approx_{\mathtt{CS}}, \approx_{\sigma_1}$ and $\approx_{\sigma_2}$ and the standard inference schemes expressing compatibility of $\sim$ w.r.t. all term constructors and transitivity, symmetry and reflexivity of $\sim$.

Residuals Along Equations. We now propose a notion of trace of a redex along an equational proof which is similar to that used for derivations. Although the intuitive idea is quite natural, one needs to guarantee that this notion is independent from the particular syntactic proof used to show that a term t is equivalent to another term u, a property that we call *well-definedness*.

Given $t \sim u$, $A \in \mathtt{Red}(t)$ and $\alpha \notin \mathtt{Lab}(t)$, we consider the labeled equation $\mathtt{lift}(t, A, \alpha) \sim u$. The **set of residuals of** A **after** $t \sim u$, is given by $A\langle t \sim u\rangle := \{\mathtt{Red}_{\alpha}(u) \mid \mathtt{lift}(t, A, \alpha) \sim u\}$. Again, this definition is *independent* from the variant used to lift the term t. We write $A\langle t \sim u\rangle A'$ iff $A' \in A\langle t \sim u\rangle$ and we extend this notion to sets of redexes as expected, in which case we write $\mathcal{A}\langle t \sim u\rangle\mathcal{A}'$, where $\mathcal{A} \subseteq \mathtt{Red}(t)$ and $\mathcal{A}' \subseteq \mathtt{Red}(u)$. For example, given $v = (x^{\alpha}x^{\alpha})[x/y][x'/y'] \sim (z^{\alpha}[x'/y']z^{\alpha})[z/y] = v'$ and $B = \langle \Box[x'/y'], (x^{\alpha}x^{\alpha})[x/y], \Box x^{\alpha}\rangle$, we have $B\langle v \sim v'\rangle = \{\langle \Box, v', \Box[x'/y']z^{\alpha}\rangle\}$.

Well-Definedness. Despite the simplicity of the equivalence relation $\sim$, the result of well-definedness is not as straightforward as it may seem. In principle, it could be shown by introducing proof-nets and showing that the proof nets corresponding to two $\sim$-equivalent terms s and t are identical: thus there is a bijection between redexes in s and redexes in t, since there is a bijection between redexes in s (resp t) and redexes in their proof-net representation. We prefer, however, to avoid introducing proof-nets here: on one hand because they are only apparently simpler than terms, on the other hand, to resort to a unique formalism, namely terms, to develop our ideas.

Note that our equations do not duplicate/erase/rename labels, so that any redex has a unique residual along the equivalence. Then well-definedness can be stated as the fact that every equational proof of $t \sim u$ induces the same bijection of redexes. The proof

[4] By abuse of notation we use the same symbol both for the equivalence relation on labeled and unlabeled terms.

of this property is based on the identification of three structural invariants of labeled redexes with respect to the equivalence $\sim$, it is omitted for lack of space.

LEMMA 2 (**Well-Definedness**). *Let $t, u \in \mathcal{T}_{\mathcal{WL}}$ s.t. $t \sim u$. Then there is unique bijection $\phi : \texttt{Red}(t) \mapsto \texttt{Red}(u)$ s.t. $\phi(A) = A'$ iff $A\langle t \sim u\rangle A'$.*

Note that uniqueness guarantees that if $t \overset{d_1}{\sim} u$ and $t \overset{d_2}{\sim} u$ are two derivation trees of $t \sim u$, then $A\langle t \overset{d_1}{\sim} u\rangle A'$ iff $A\langle t \overset{d_2}{\sim} u\rangle A'$.

The last result concerns preservation of residuals by means of $\sim$-equivalence classes, where below we write $t \sim_\phi u$ to emphasize that ϕ is the (unique) bijection given by Lemma 2.

LEMMA 3. *Let $t \sim_\phi u$. Consider $A, B \in \texttt{Red}(t)$. If $t \xrightarrow{A} t'$, then:*

1. **Simulation***: $\exists u'$ s.t. $u \xrightarrow{\phi(A)} u'$ and*
2. **Same equivalence target***: $t' \sim u'$,* i.e. *$\exists \xi$ s.t. $t' \sim_\xi u'$, and*
3. **Preservation of residuals***: if $B\langle A\rangle B'$, then $\phi(B)\langle\phi(A)\rangle\xi(B')$.*

5. Abstract Standardization

A *standard* derivation is a *canonical* element of a class of derivations, where canonicity is expressed by means of *completeness*: whenever $t \twoheadrightarrow u$ then there is a standard derivation from t to u. In λ-calculus, a derivation is standard if redexes are selected from *left-to-right*. It is easy to see that the *right-to-left* order, instead, does not provide completeness. Consider the derivation $t = (\lambda x.xx)(II) \rightarrow_\beta (II)(II) \rightarrow_\beta I(II)$: it reduces both redexes in t, but it can only be obtained by reducing these redexes according to a left-to-right order, as the right-to-left order would instead give $(\lambda x.xx)(II) \rightarrow_\beta (\lambda x.xx)I \rightarrow_\beta II$. The left-to-right orientation is specific to the λ-calculus and may not be appropriate for other term rewriting systems, as for example the first-order TRS containing the unique rewriting rule $f(x, a) \rightarrow x$. The abstract theory of standardization then replaces the left-to-right order with an abstract order $\prec$, capturing what we call *the action principle*.

The Action Principle. Let us introduce two crucial concepts. Firstly, reduction of a redex A may **directly act** on another redex B (*i.e.* duplicate/erase it). For example in λ-calculus, if $t = (\lambda x.xx)(II) \rightarrow_\beta (II)(II)$, then reduction of the leftmost redex A in t duplicates the redex $B = II$. Secondly, reduction of a redex A may **indirectly act** on a redex B by 1) *creating* a new redex C that may directly (or indirectly act) on (a residual of) B, or 2) changing the set of redexes that can act on B or on which B can act. An example of the first case: if $t = (\lambda x.x(II))M \rightarrow_\beta M(II)$, where $M = \lambda x.xx$, then reduction of the leftmost redex A in t indirectly acts on the redex $B = II$ by creating the redex $M(II)$ which can duplicate (the residual of) B. An example of the second case will be given at the end of Section 8.

Standard derivations are defined abstractly according to the **action principle**: for every redex A, reduction of A forbids future reductions of redexes which *may have acted (directly or indirectly) on* A before. The nesting order $\prec$ essentially has to induce this principle by capturing as much action as possible (we will see that $\prec$ has to capture direct action, while in general it might capture only some cases of indirect action). The abstract theory of standardization re-organizes derivations according to the action principle, relying on some axioms about the interaction between the nesting order and residuals. Then, whenever the axioms are satisfied, the standardization theorem follows: an abstract reasoning proves that $\prec$-ordered derivations are complete [19, 30, 32].

Total and Partial Orders. The left-to-right order has a fundamental property, it is a *total* order, *i.e.* given two coinitial different redexes A and B, then either $A \prec B$ or $B \prec A$. In such a case the abstract theory is quite simple.

Total orders have a limited descriptive power, though. Consider a head-normal form $t = xuv$. The left-to-right order forces the redexes in u to be reduced before those in v. However, it is clear that u and v cannot inter*act* in any way, and consequently they could be reduced in parallel even if this parallelism is not captured by the left-to-right order. In other words, the left-to-right order captures more than the action principle. Standardization can then be refined by switching to a *partial* order. For instance, the partial *tree order* $\prec_{tree}$, where $A \prec_{tree} B$ if B is a sub-term of A (so that A and B are unrelated if B is to the right of A), captures the parallelism of redexes in t and admits a standardization theorem [30, 32]. The abstract theory of standardization for partial orders is more sophisticated, so for the moment we stay with total orders and shall return to partial orders in Section 7.

In the rest of this section we introduce the necessary notions to deal with the abstract standardization theorem for total orders. In Section 6 we will adapt the left-to-right order to the $\lambda_{\texttt{lsub}}$-calculus and obtain a left-to-right standardization theorem. From Section 7 to the end of the paper, we will motivate and study a relaxed variant of this order, introducing the framework for partial orders and prove a more general standardization theorem for the $\lambda^{\sim}_{\texttt{lsub}}$-calculus.

The Abstract Theory. In this paper we resort to the abstract framework developed by Gonthier, Lévy, and Melliès in [19], but we adopt the more general formulation given by Melliès in his PhD thesis [32], as the axioms in [19] will not work when we shall later relax the left-to-right order. The framework is based on *Abstract Rewrite Systems* (ARS), a general formalism encompassing first and higher-order rewriting, plus a *residual relation* $\_\langle\_\rangle\_$ and a *partial order* $\prec$ on redexes, verifying some axioms. The axioms are divided into two groups: The *basic axioms*, which guarantee a well-behaved theory of residuals, and the *standardization axioms*, which concern the interaction of the order with the residual relation.

To formally define the axioms we introduce the following notion. An **Abstract Rewrite System (ARS)** is a tuple of the form $(\mathcal{O}, \mathcal{R}, \mathit{Src}(\_), \mathit{Tgt}(\_), \_\langle\_\rangle\_, \_ \prec \_)$ where $\mathcal{O}$ is called the set of **objects**, $\mathcal{R}$ is called the set of **redexes** , $\mathit{Src}(\_)$ and $\mathit{Tgt}(\_)$ are functions from redexes to objects that we call **source** and **target** functions (resp.), $\_\langle\_\rangle\_$ is a family of binary relations indexed by the set of redexes that we call the **residual relation** and $\prec$ is a partial order on redexes. We say that two redexes A and B are **coinitial** if $\mathit{Src}(A) = t = \mathit{Src}(B)$ for some term t. Note that the $\lambda_{\texttt{lsub}}$-calculus may be seen as an ARS where $\mathcal{O}$ is the set of terms $\mathcal{T}$ defined in Section 2, whereas the $\lambda^{\sim}_{\texttt{lsub}}$-calculus is an ARS where $\mathcal{O}$ is the set of $\sim$-equivalence classes generated by $\mathcal{T}$ modulo the graphical equivalence $\sim$. The basic axioms are[5]:

- **Autoerasure (AE)**. For any redex A, $A\langle A\rangle = \emptyset$.
- **Finite residuals (FR)**. Let A, B be coinitial redexes. Then the set $\{C \mid A\langle B\rangle C\}$ is finite.
- **Uniqueness of ancestors (UA)**. Let A, B, C be coinitial redexes. Then, $B\langle A\rangle B'$ and $C\langle A\rangle C'$ and $B' = C'$ imply $B = C$.
- **Finite Developments (FD)**. Let $\mathcal{A}$ be a set of coinitial redexes. Then any development of $\mathcal{A}$ terminates.
- **Semantic Orthogonality (SO)**. Let A, B be coinitial redexes. Then $\mathit{Tgt}(B\langle A\rangle) = \mathit{Tgt}(A\langle B\rangle)$ and the relations $\langle A; B\langle A\rangle\rangle$ and $\langle B; A\langle B\rangle\rangle$ are exactly the same, where $A; B\langle A\rangle$

[5] In [32] **Autoerasure** is called **A**, **Finite residuals** is called **B**, **Uniqueness of ancestors** is called **D**, and **Semantic Orthogonality** is called **PERM**.

is the derivation that contracts A and then develops $B\langle A\rangle$; likewise for $B; A\langle B\rangle$ (cf. end of Sec. 3).

We consider two standardization axioms:

- **Linearity**. Let A and B be coinitial redexes s.t. $A \not\prec B$. Then $\exists! B'$ s.t. $B\langle A\rangle B'$.
- **Context-freeness**. Let A, B, C be coinitial redexes s.t. $B\langle A\rangle B'$ and $C\langle A\rangle C'$. If $A \not\prec C$ then $(B \prec C \Leftrightarrow B' \prec C')$.

The linearity axiom captures *direct action*: if A does not nest B then it cannot directly act (*i.e.* duplicate/erase) on B, and so B has exactly one residual after A. The context-freeness axiom forbids a form of *indirect action* of the second kind: whenever A cannot act on C (*i.e.* $A \not\prec C$) then it cannot grant or remove the power to act on C to any other coinitial redex.

A derivation $d : t \twoheadrightarrow u$ is said to be obtained from a derivation $e : t \twoheadrightarrow u$ by a **standardizing permutation**, written $e \rightsquigarrow d$ (noted $d \angle e$ in [32]), if d is obtained from e by swapping two consecutive redexes which form an inversion w.r.t. $\prec$ in d; more precisely, $e \rightsquigarrow d$ if $e = f; B; A'; g$ and $d = f; A; h; g$, and $A \prec B$, where $A\langle B\rangle A'$ and h develops $B\langle A\rangle$. We also say that d is **more standard** than e. The classic notion of Lévy's permutation equivalence on derivations, adapted to total orders, can be seen as the equivalence generated by standardization, indeed $d : t \twoheadrightarrow u$ is **permutation equivalent** to $e : t \twoheadrightarrow u$, written $d \equiv e$, iff $d(\rightsquigarrow \cup \rightsquigarrow^{-1})^* e$. In the sequel we write $\equiv_{\prec}$ when we wish to emphasize the underlying order $\prec$.

A derivation d is **standard** if it is a $\rightsquigarrow$-normal form. The standardization theorem then follows by specializing Thm. 4.7 in [32] to total orders:

THEOREM 1 (**Abstract Total Standardization).** *Consider any ARS equipped with a total order $\prec$ and satisfying the basic and the standardization axioms. Then for any derivation $d : t \twoheadrightarrow u$ there exists a standard derivation $e : t \twoheadrightarrow u$ such that $d \rightsquigarrow^* e$.*

6. The Left-to-Right Standardization Theorem

In this section we prove a standardization theorem for the $\lambda_{\mathtt{lsub}}$-calculus relative to a total order. This result is an extension of the left-to-right standardization theorem for λ-calculus.

The Left-to-Right Order. Given two redexes A and B in t we say that A **is left-to-right nesting** or **on the left of** B, written $A \prec_{\mathtt{L}} B$, if the anchor of A is to the left of the anchor of B (looking at t as a string of symbols). Clearly, $\prec_{\mathtt{L}}$ is a total order so that $A \not\prec_{\mathtt{L}} B$ and $A \neq B$ imply $B \prec_{\mathtt{L}} A$.

In order to prove standardization for the $\lambda_{\mathtt{lsub}}$-calculus by means of Theorem 1 we need to verify the basic and the standardization axioms.

Basic Axioms. The first three basic axioms are trivially true. By using (well-)labeled terms to trace residuals we can prove the finite development axiom.

PROPOSITION 1 **(FD).** *Let $t \in \mathcal{T}$ and let $\mathcal{A} \subseteq \mathtt{Red}(t)$. Then any $\lambda_{\mathtt{lsub}}$-development of $\mathcal{A}$ terminates.*

Proof. Consider any $t \in \mathcal{T}$ and *lift* every redex in $\mathcal{A} \subseteq \mathtt{Red}(t)$ with a different label belonging to some arbitrary set $\mathbb{L}$. It is clear that the resulting term is a *well-labeled* term. Now, let us consider the labeled reduction relation $\rightarrow_{\mathbb{L}} := \bigcup_{\alpha \in \mathbb{L}} \xrightarrow{\alpha}$. To prove the **(FD)** property it is sufficient to show that the reduction relation $\rightarrow_{\mathbb{L}}$ terminates on well-labeled terms. Termination is proved by means of a measure that strictly decreases with every reduction step (similar to that one used in the proof of Lemma 2 in [6]).

The next proposition proves the semantical orthogonality property for $\lambda_{\mathtt{lsub}}$, that is expressed as a form of local confluence for the labeled system. Again we resort to well-labeled terms.

PROPOSITION 2 **(SO).** *The $\lambda_{\mathtt{lsub}}$-calculus endowed with the left-to-right order satisfies* **Semantic Orthogonality**.

Proof. The proof uses well-labeled terms to trace residuals in such a way that axiom (SO) can be reformulated as follows:

The reduction relations $\xrightarrow{\alpha}$ and $\xrightarrow{\gamma}$ locally commute, i.e. *if $t, u_1, u_2 \in \mathcal{T}_{\mathcal{WL}}$, $t \xrightarrow{\alpha} u_1$ and $t \xrightarrow{\gamma} u_2$ then there exists v s.t. $u_1 \overset{\gamma}{\twoheadrightarrow} v$ and $u_2 \overset{\alpha}{\twoheadrightarrow} v$.*

This alternative statement can be proved by induction on the relations $\xrightarrow{\alpha}$ and $\xrightarrow{\gamma}$.

Standardization Axioms. Totality of the order $\prec_{\mathtt{L}}$ provides very simple proofs of these axioms:

PROPOSITION 3. *The $\lambda_{\mathtt{lsub}}$-calculus endowed with the left-to-right order satisfies* **Linearity** *and* **Context-freeness**.

Proof. **Linearity**. By totality of $\prec_{\mathtt{L}}$, we have to show that if $B \prec_{\mathtt{L}} A$ in t then $\exists! B'$ s.t. $B\langle A\rangle B'$. Now, if A is a $\mathtt{db}$-redex this is obvious, as no redex is duplicated/erased by a $\mathtt{db}$-step. If A is a $\{\mathtt{gc}, \mathtt{ls}\}$-redex then it can only act on redexes whose anchor is in its box, *i.e.* on redexes on its right, and thus not on B.
Context-freeness. If $A \not\prec_{\mathtt{L}} C$ then $C \prec_{\mathtt{L}} A$. Assume $B \prec_{\mathtt{L}} C$ (and so $B \prec_{\mathtt{L}} C \prec_{\mathtt{L}} A$). Then, A is on the right of both B and C. It is easily seen that A can only change the order between redexes on its right; consequently $B' \prec_{\mathtt{L}} C'$. The other direction is by contraposition. Assume $B \not\prec_{\mathtt{L}} C$, that is $C \prec_{\mathtt{L}} B$. We have to prove that $B' \not\prec_{\mathtt{L}} C'$, *i.e.* $C' \prec_{\mathtt{L}} B'$. There are two cases. If $C \prec_{\mathtt{L}} B \prec_{\mathtt{L}} A$ then we reason as in the previous direction, getting $C' \prec_{\mathtt{L}} B'$. Otherwise, we have $C \prec_{\mathtt{L}} A \prec_{\mathtt{L}} B$. Now, the only case that is not immediate is when A is a $\mathtt{ls}$-step. It is enough to observe that a $\mathtt{ls}$-step can only move the redexes in its box at most where the step itself was; hence, B' can at most be where A was (while the position of C is left unchanged), and so $C' \prec_{\mathtt{L}} B'$.

We can then conclude with our first standardization theorem:

COROLLARY 1 **(Left-to-right Standardization for $\lambda_{\mathtt{lsub}}$).** *If $t \twoheadrightarrow_{\lambda_{\mathtt{lsub}}} u$ then there is a $\prec_{\mathtt{L}}$-standard $\lambda_{\mathtt{lsub}}$-derivation from t to u.*

Proof. It follows from Theorem 1, whose hypothesis are given by Propositions 2, 1, and 3 (the first three basic axioms are immediately seen to hold).

7. Towards Equational Standardization

Proof-Nets. The linear substitution calculus has been designed to mimic the representation of λ-calculus in linear logic proof-nets [18], where β-reduction is decomposed into small steps. The relationship between the two formalisms occurs at the static and the dynamic levels: every term can be mapped to a proof-net, and every proof-net can be mapped to an $\sim$-equivalence class of terms, as defined in Section 2. Moreover, there is a bijection ϕ between the redexes of a term t and the redexes of its corresponding proof-net PN_t which induces a *strong bisimulation* between terms and proof-nets: if $t \rightarrow_{\lambda_{\mathtt{lsub}}} u$ by reducing a redex A, then $PN_t \rightarrow_{PN} PN_u$ by reducing $\phi(A)$, and if $PN_t \rightarrow_{PN} R$ then there exists a term u s.t. $t \rightarrow u$ and $R = PN_u$.

The graphical equivalence $\sim$ introduced in Section 2 then inherits a remarkably strong property: it is a *strong bisimulation* between $\lambda_{\mathtt{lsub}}$ and itself, *i.e.* $t \sim u \rightarrow v$ implies that $\exists\, r$ s.t. $t \rightarrow r \sim v$ (the proof is similar to that of the structural λ-calculus [6]). It, moreover, induces a bijection of redexes, and so it is possible to mimic derivations via $\sim$ as follows: given a derivation $d : t \to^n u$ and a term $t' \sim t$ we can unambiguously refer to the projection $d' : t' \to^n u'$ (with $u' \sim u$) of d on t'. Since d and d' essentially reduce the same redexes at each step, one expects any reasonable notion of standardization to apply without distinction to both derivations in the sense that either both are standard or none of them is. This would imply, in particular, that our standardization theorem for $\lambda_{\mathtt{lsub}}$ (Corollary 1) also applies to proof-nets. Unfortunately, the left-to-right order does not meet this requirement. Indeed, a left-to-right derivation for $t[x/u][y/v]$ (where $y \notin \mathtt{fv}(u)$) does not project to a left-to-right derivation for $t[y/v][x/u]$, obtained by applying the equation $t[x/u][y/v] \sim_{\mathtt{CS}} t[x/u][y/v]$.

Relaxing the Order. We are therefore going to relax the total left-to-right order $\prec_{\mathtt{L}}$ to a partial order $\prec_{\mathtt{B}}$ that will be stable by $\sim$. The definition of $\prec_{\mathtt{B}}$ shall be guided by the action principle. Because of the linearity axiom, it is mandatory that $\prec_{\mathtt{B}}$ captures direct action between redexes. For that, a simple diagrammatic intuition, due to Klop [26] and then explored by Melliès [30], turns out to be extremely helpful. The idea relies on the analysis of local confluence diagrams. Whenever a redex is duplicated (resp. erased), then the standard derivation should be the longest (resp. shortest) side of the diagram. Consider for example the diagram in Fig.1-a. The *standard* way to get from the source to the sink of the diagram is the longest (*i.e.* down-below) side, because the $\mathtt{ls}$-redex *acts on* (*i.e. nests*) the redexes in s [6]. The standard derivation associated to the diagram in Fig.1-b —which is an erasing case— is instead the shortest side, because the erasing redex *acts on* the redexes in s.

The previous two cases are just reformulations of similar cases in λ-calculus, and amount to what can be described as *nesting as subterms*. The novelty of $\lambda_{\mathtt{lsub}}$ is that *direct action* (and then nesting of redexes) can also happen *at a distance*. Consider now the example of the diagram in Fig.1-c. The duplicated $\mathtt{ls}$-redex on y is not syntactically contained in the acting $\mathtt{ls}$-redex on x. Worse, the same diagram applies to terms like $(x[x/y]yz)[y/z]$, where $[x/y]$ and $[y/z]$ are no longer next to each other. According to the action principle, our order has to impose that the $\mathtt{ls}$-redex on x nests the $\mathtt{ls}$-redex on y so that the standard side of the diagram is the longest one. Diagram in Fig.1-d is the version at a distance of the erasing diagram, requiring the same notion of nesting at a distance. All these intuitions lead to the partial order of the next section, that will be used in Section 8 to show standardization of the $\lambda_{\mathtt{lsub}}^{\sim}$-calculus.

8. The Box Standardization Theorem

We now generalize the notions introduced in Section 5 to the partial **box order**, and then prove the standardization theorem for the $\lambda_{\mathtt{lsub}}^{\sim}$-calculus with respect to this order.

The Box Order. Let $A, B \in \mathtt{Red}(t)$. Then,

- A **immediately boxes** B, noted $A \prec^1_{\mathtt{B}} B$ if the anchor of B (*i.e.* the variable possibly carrying a label) is in the box of A, *i.e.* if the pattern of A is any of $(\lambda x.t)\mathtt{L}u$, $C[\![x]\!][x/u]$ or $t[x/u]$, then the anchor of B appears in u.
- A **boxes** B, noted $A \prec_{\mathtt{B}} B$ if $A(\prec^1_{\mathtt{B}})^+ B$ (we use $A \preceq_{\mathtt{B}} B$ for $A(\prec^1_{\mathtt{B}})^* B$);
- A **and** B **are disjoint**, noted $A \parallel B$, if $A \not\preceq_{\mathtt{B}} B$ and $B \not\preceq_{\mathtt{B}} A$.

Note that $A \prec_{\mathtt{B}} B$ implies $A \prec_{\mathtt{L}} B$. Additionally, note the transitive closure in the definition of $\prec_{\mathtt{B}}$—unnecessary for $\prec_{\mathtt{L}}$—which impacts on the proofs of the standardization axioms for $\prec_{\mathtt{B}}$.

The key property of the box order is that it is stable by the equivalence $\sim$. For example, for $t[x/u][y/v]$ with $y \notin \mathtt{fv}(u)$ the redexes in u and the redexes in v are not related by $\prec_{\mathtt{B}}$, so that $\prec_{\mathtt{B}}$ is stable by the permuting axiom $t[y/v][x/u] \sim_{\mathtt{CS}} t[x/u][y/v]$ (where $y \notin \mathtt{fv}(u)$ & $x \notin \mathtt{fv}(v)$). More precisely, given $s \sim t$, the bijection between $\mathtt{Red}(s)$ and $\mathtt{Red}(t)$ defined in Section 4 is order-preserving. To show this property it is sufficient to remark that symbols inside boxes never go in/out these boxes by means of the equivalence relation. Thus, we get:

LEMMA 4 (**Preservation of the Box Order by Equivalence**). *Let $t, u \in \mathcal{T}$ s.t. $t \sim_\phi u$, where ϕ is the bijection specified in Lemma 2. Then, ϕ commutes with $\prec_{\mathtt{B}}$,* i.e. *$A \prec_{\mathtt{B}} B$ iff $\phi(A) \prec_{\mathtt{B}} \phi(B)$.*

Several remarks on $\prec_{\mathtt{B}}$ are in order:

1. *Disjoint redexes may superpose*: $A \parallel B$ does not necessarily imply that A and B are syntactically disjoint. Examples: the redexes A_α and B_β are disjoint but 1) syntactically superposed in $(x^\alpha x^\beta)[x/y]$, and 2) syntactically nested in $(\lambda z^\alpha.x^\beta[x/z])y$. However, disjoint redexes always strongly locally commute in the following sense: if $t_0 \xrightarrow{\alpha} t_1$ and $t_0 \xrightarrow{\gamma} t_2$ then there exists t_3 s.t. $t_1 \xrightarrow{\gamma} t_3$ and $t_2 \xrightarrow{\alpha} t_3$. Note that this is just a particular case of axiom **(SO)** where the diagram can be closed by using just one reduction step from t_i to t_3.
2. *Indirect action*: the box order fails to capture indirect action, in the sense that $A \parallel B$ implies that A cannot directly act on B but it can still indirectly act on it (and viceversa). For example, in the following derivation $A_\alpha \parallel B_\beta$ and A_α creates a redex that can act on B_β:
$$x^\alpha[x/\lambda y.y]((\lambda z^\beta.z)u) \xrightarrow{\alpha}_{\mathtt{ls}} (\lambda y.y)[x/\lambda y.y]((\lambda z^\beta.z)u)$$
This apparently odd fact—that is not specific to the box order—will not forbid the standardization theorem. Its consequences, and an easy way to deal with them, will be discussed in Section 10. Note that by definition the box order is the *minimum standardization order*, in the sense that it is the transitive closure of the relation capturing direct action only.
3. *The order on λ-terms*: in [30] (pp. 74-75) Melliès considers three orders on λ-terms, the *left-to-right* order plus two refinements called the *tree* and the *argument* order. When restricted to λ-terms, our order coincides with the argument order.
4. *Linear logic interpretation*: we explained that there are two kinds of nesting in $\lambda_{\mathtt{lsub}}$, namely, *subterm nesting* and *nesting at a distance*. In the definition of the box order they are both captured by the notion of *immediately boxes*. In terms of proof-nets, however, they have different interpretations, despite they are both related to the concept of exponential box[7].

Standardization Up to Square Equivalence. A consequence of switching to partial orders is that the notion of standard derivation has to be refined. In fact, partial orders force to work *modulo* the exchanges of disjoint redexes, even to simply define what is an ordered (*i.e.* standard) sequence; this is why abstract standardization is a form of rewriting modulo.

Standardizing permutations are defined as in Section 5. However, to deal with disjoint redexes and adequately represent their

[6] Remark that otherwise one would lose completeness, as there would not be any standard derivation from $C[\![x]\!][x/s]$ to the term $C[\![s']\!][x/s]$, which is the intermediate term in the sequence $C[\![x]\!][x/s] \rightarrow\!\!\rightarrow C[\![s']\!][x/s']$

[7] A subterm nests B when B is contained in the !-box of A, while A nests B at a distance when the cut corresponding to B crosses an auxiliary conclusion of the box of A.

a)
$$\begin{tikzcd}
C\llbracket x\rrbracket[x/s] \arrow[rr] \arrow[d] & & C\llbracket x\rrbracket[x/s'] \arrow[d] \\
C\llbracket s\rrbracket[x/s] \arrow[r] & {} \arrow[r] & C\llbracket s'\rrbracket[x/s']
\end{tikzcd}$$

b)
$$\begin{tikzcd}
t[x/s] \arrow[r] \arrow[d] & t[x/s'] \arrow[dl] \\
t &
\end{tikzcd}$$

c)
$$\begin{tikzcd}
x[x/y][y/z] \arrow[rr] \arrow[d] & & x[x/z][y/z] \arrow[d] \\
y[x/y][y/z] \arrow[r] & {} \arrow[r] & z[x/z][y/z]
\end{tikzcd}$$

d)
$$\begin{tikzcd}
x[x'/y][y/z] \arrow[r] \arrow[d] & x[x'/z][y/z] \arrow[dl] \\
x[y/z] &
\end{tikzcd}$$

Figure 1. Some standardization diagrams.

parallelism induced by the partial order, a new concept is necessary. A derivation d is said to be obtained from a derivation e by **square permutation**, written $d\Diamond^1 e$, if $d = f; A; B'; g$ and $e = f; B; A'; g$ and $A \parallel B$, where $A\langle B\rangle A'$ and $B\langle A\rangle B'$. We write $\Diamond$ for the equivalence relation generated by $\Diamond^1$, and call it **square equivalence**. We use $\rightsquigarrow_\Diamond$ for the relation $\rightsquigarrow$ modulo $\Diamond$ and we generalize the permutation equivalence relation $\equiv_\prec$ as $(\rightsquigarrow_\Diamond \cup \rightsquigarrow_\Diamond^{-1})^*$.

A derivation d is **standard** if it is a $\rightsquigarrow$-normal form modulo $\Diamond$. The standardization theorem (Thm. 4.7 in [32]) then follows:

THEOREM 2 (**Abstract Partial Standardization).** *Consider any ARS equipped with a partial order that enjoys the basic and the standardization axioms. Then for every derivation d there exists a standard derivation e s.t. d* $\rightsquigarrow_\Diamond^*$ *e.*

Results. Our goal is to prove a standardization theorem for $\lambda^{\sim}_{\mathtt{lsub}}$ relative to $\prec_{\mathtt{B}}$. The proof consists in first obtaining the theorem for $\lambda_{\mathtt{lsub}}$ and then lifting it to $\lambda^{\sim}_{\mathtt{lsub}}$.

The basic axioms do not mention the order, and thus they still hold for $\lambda_{\mathtt{lsub}}$. The standardization axioms are proved by the following proposition:

PROPOSITION 4. *The $\lambda_{\mathtt{lsub}}$-calculus endowed with the box order $\prec_{\mathtt{B}}$ satisfies* **Linearity** *and* **Context-freeness***.*

Proof. **Linearity** is easy, by just remarking that a redex B can be duplicated or erased by A only if its anchor is in the box of A, case in which $A \prec_{\mathtt{B}} B$. **Context-freeness** requires a detailed study of the interaction between the box order $\prec_{\mathtt{B}}$ and the reduction relation. In particular the proof is involved due to the transitive clause defining the box order, and it is omitted for lack of space.

Then, the previous proposition and Theorem 2 give:

COROLLARY 2 (**Box Standardization for $\lambda_{\mathtt{lsub}}$).** *If $t \twoheadrightarrow_{\lambda_{\mathtt{lsub}}} u$ then there exists a $\prec_{\mathtt{B}}$-standard $\lambda_{\mathtt{lsub}}$-derivation from t to u.*

In order to lift the previous result to $\lambda^{\sim}_{\mathtt{lsub}}$ we use the results on equivalence classes developed in Section 4.

PROPOSITION 5. *The $\lambda^{\sim}_{\mathtt{lsub}}$-calculus endowed with the box order $\prec_{\mathtt{B}}$ satisfies the basic and the standardization axioms.*

Proof. By Lemma 3 and Lemma 4 the notion of residual and the box order lift to $\sim$-equivalence classes preserving their properties. Moreover, the **FD** axiom can be shown exactly as done for $\lambda_{\mathtt{lsub}}$ in Proposition 1, by means of a measure that strictly decreases with every reduction step and remains equal for every pair of equivalent pairs. Therefore, all axioms hold for $\lambda^{\sim}_{\mathtt{lsub}}$ and we can conclude.

Proposition 5 allows to apply Theorem 2 again and finally obtain our *nonstandard*, equational standardization theorem.

COROLLARY 3 (**Box Standardization for $\lambda^{\sim}_{\mathtt{lsub}}$).** *If $t \twoheadrightarrow_{\lambda^{\sim}_{\mathtt{lsub}}} u$ then there is a $\prec_{\mathtt{B}}$-standard $\lambda^{\sim}_{\mathtt{lsub}}$-derivation from t to u.*

The next section will strengthen the result by showing uniqueness (modulo $\Diamond$) of the obtained standard derivation. We conclude this section with a remark.

	Redex Order	Existence	Uniqueness
$\lambda_{\mathtt{lsub}}$	Total, left-to-right (Sec. 6)	Thm. 1	Thm. 3
$\lambda_{\mathtt{lsub}}$	Partial, box (Sec. 8)	Cor. 2	Thm. 4
$\lambda^{\sim}_{\mathtt{lsub}}$	Partial, box (Sec. 8)	Cor. 3	Thm. 5

Figure 2. Summary of results on standardization

Variation on the Context-Freeness Axiom. We use in this paper **Context-freeness** as it appears in Melliès's thesis [32]. In GLM's theory [19], however, the axiom appears in a stronger, symmetric form which does not hold for $\lambda_{\mathtt{lsub}}$. Indeed,

- **Strong context-freeness**. Let A, B, C be coinitial redexes s.t. $B\langle A\rangle B'$ and $C\langle A\rangle C'$. If $A \not\prec C$ then $((B \prec C \Leftrightarrow B' \prec C')$ and $(C \prec B \Leftrightarrow C' \prec B'))$.

Consider $t = (\lambda x^\alpha . y^\gamma[y/x])(z^\beta[z/u])$, where $A_\alpha \parallel C_\gamma$, $C_\gamma \parallel B_\beta$, and $A_\alpha \prec_{\mathtt{B}} B_\beta$. The reduction step of the redex A_α yields the term $y^\gamma[y/x][x/(z^\beta[z/u]]$, where $C'_\gamma \prec_{\mathtt{B}} B'_\beta$, contradicting the strong form of the axiom (but not the weaker one we use) and providing an example of the second kind of indirect action, as A_α changes who can act on (the residual of) B_β.

9. Uniqueness

The standardization results obtained up to now concern the *existence* of a standard derivation. In this section the theorems are strengthened by proving *uniqueness* of such derivations. We first discuss the simple case of the left-to-right order, and then provide an original proof of uniqueness (modulo) for the box order.

The axiomatic theory in Mellies' thesis provides uniqueness of standard derivations when two further axioms, **Enclave** and **Stability**, hold (Theorem 4.5 in [32]).

Axiom **Enclave** has two parts. Let A, B, B' be coinitial redexes s.t. $B\langle A\rangle B'$.

1. **Creation**: If $B \prec A$ and A creates a redex C, then $B' \prec C$.
2. **Nesting**: Let C, C' be redexes. If $B \prec A \prec C$ and $C\langle A\rangle C'$, then $B' \prec C'$.

In terms of the action principle **Enclave** forbids yet another form of indirect action: if B can directly act on A then A cannot indirectly act on B. Consider the *creation* part: if $C \prec B'$ or $C \parallel B'$ then C may act on B', and so A would indirectly act on B ; similarly for the *nesting* part.

PROPOSITION 6. *The $\lambda_{\mathtt{lsub}}$-calculus endowed with the left-to-right order $\prec_{\mathtt{L}}$ satisfies* **Enclave***.*

Proof. Creation: a simple inspection of the cases of creation (Section 3) shows that a redex A can create a redex C only on its right or at most where it was, so that C cannot nest any redex that was on the left of A. *Nesting*: similarly, it is easily seen that A can only move redexes on its right and at most where it was.

In the case of a total standardization order the axiom **Stability** disappears, because its statement (see [32]) assumes two $\prec$-disjoint redexes. Thus, by applying Melliès' theorem we obtain:

THEOREM 3 (**Uniqueness for $\lambda_{\texttt{lsub}}$).** *If $t \twoheadrightarrow_{\lambda_{\texttt{lsub}}} u$ then there exists a unique $\prec_{\texttt{L}}$-standard $\lambda_{\texttt{lsub}}$-derivation from t to u.*

Switching to partial orders, uniqueness is necessarily relaxed to uniqueness modulo $\Diamond$ (as standard derivations are defined up to $\Diamond$).

Interestingly, for the partial box order both points of **Enclave** fail for $\lambda_{\texttt{lsub}}$. For *creation* consider the step:

$$s = ((\lambda x^{\beta}.x)(y[z^{\alpha}/z']))[z'/u] \xrightarrow{\alpha} ((\lambda x^{\beta}.x)y)[z'/u] = t$$

Let us call A_α and B_β the two labeled redexes in s, and B'_β (resp. C) the residual of B_β (resp. the created gc-redex on z') in t. We have $B_\beta \prec_{\texttt{B}} A_\alpha$ but $B'_\beta \not\prec_{\texttt{B}} C$. For *nesting* it is enough to consider a redex C inside u in the counter-example for *creation*.

So the box order does not capture the indirect action expressed by **Enclave**, and Mellies' axiomatics cannot provide uniqueness modulo $\Diamond$ for $\lambda_{\texttt{lsub}}^{\sim}$. However, we can obtain it via another argument. In particular, our alternative proof shows that **Enclave** (together with **Stability**) is a sufficient but by no means necessary condition for uniqueness. We first show uniqueness modulo $\Diamond$ for $\lambda_{\texttt{lsub}}$, and then lift this result to $\lambda_{\texttt{lsub}}^{\sim}$.

THEOREM 4 (**Uniqueness Modulo for $\lambda_{\texttt{lsub}}$).** *If $t \twoheadrightarrow_{\lambda_{\texttt{lsub}}} u$ then there exists a $\prec_{\texttt{B}}$-standard $\lambda_{\texttt{lsub}}$-derivation from t to u that is unique modulo $\Diamond$.*

Proof. (Sketch) Let e be the $\prec_{\texttt{B}}$-standard $\lambda_{\texttt{lsub}}$-derivation given by Corollary 2 and f be the unique $\prec_{\texttt{L}}$-standard derivation given by Corollary 1. We prove that $e \Diamond f$, from which the statement follows. Consider the leftmost contracted redex C in $e = B_1; \ldots ; B_n$ (*i.e.* the leftmost redex in $\{A \in \texttt{Red}(t) \mid A\langle B_1; \ldots ; B_i\rangle B_{i+1}, 0 \leq i < n\}$) and let e' be $C; e_1; e_2$ where $e_1 = (B_1; \ldots ; B_i)\langle C\rangle$ and $C\langle B_1; \ldots ; B_i\rangle B_{i+1}$ and $e_2 = B_{i+2}; \ldots ; B_n$. It is not difficult to prove, given that e is $\prec_{\texttt{B}}$-standard, that $e' \Diamond e$, $|e_1| = i$ and so $|e_1; e_2| = |e| - 1$, where $|e|$ denotes the number of steps in e. By induction hypothesis we obtain a $\prec_{\texttt{L}}$-standard derivation f' s.t. $f' \Diamond (e_1; e_2)$. Therefore $C; f'$ is also $\prec_{\texttt{L}}$-standard and moreover $(C; f') \Diamond (C; e_1; e_2) \Diamond e$. Thm. 3 implies $C; f' = f$.

Note that the proof does not depend concretely on $\lambda_{\texttt{lsub}}^{\sim}$, but only on the fact that $\prec_{\texttt{B}}$ can be embedded in a total order admitting a standardization theorem, which is a fully *abstract* argument.

By exploiting again the results on equivalence classes, we obtain the strongest result of the paper:

THEOREM 5 (**Uniqueness Modulo for $\lambda_{\texttt{lsub}}^{\sim}$).** *If $t \twoheadrightarrow_{\lambda_{\texttt{lsub}}^{\sim}} u$ then there exists a $\prec_{\texttt{B}}$-standard $\lambda_{\texttt{lsub}}^{\sim}$-derivation from t to u that is unique modulo $\Diamond$.*

Proof. By Lemma 3 and Theorem 4 using the stability of the $\prec_{\texttt{B}}$-order by the equivalence $\sim$ given by Lemma 4.

10. External Redexes, Coinductively

When the nesting order $\prec$ is total, standardization is relatively easy, because a derivation $d : t \twoheadrightarrow u$ may be standardized by simply selecting the minimum redex A among the redexes in t that are reduced in d and then repeating the process for the residual of this derivation after A. However, when the order is partial—as our box order, and more generally in Gonthier-Levy-Melliès' approach—there may be many $\prec$-minimal redexes among the redexes in t that are reduced in d; and randomly selecting one of them does not necessarily produce a standard derivation. This is due to the fact that the partial box order—as it is usually the case with partial orders—fails to properly capture indirect action. Indeed, even if two minimal redexes A and B are necessarily disjoint, A may *create* a redex which will nest (a residual of) B.

We illustrate this situation in the simpler setting of the λ-calculus using the box order (called argument order by Melliès in [30]). Let I^α stand for $\lambda x^\alpha.x$ and A_α for $I^\alpha\, I$ (and similarly for B_β). Consider the derivation:

$$(I^\alpha\, I)\,(I^\beta\, I) \quad \xrightarrow{\beta}_\beta \quad (I^\alpha\, I)\, I \quad \xrightarrow{\alpha}_\beta \quad I\, I \quad \rightarrow_\beta \quad I$$

Note that both A_α and B_β are minimal in the initial term, and the derivation picks at every step a minimal redex. However, the derivation d is not standard. This can be seen by permuting A_α and B_β, which yields the following $\Diamond$-equivalent derivation, where the two last steps form an inversion for the box order:

$$(I^\alpha\, I)\,(I^\beta\, I) \quad \xrightarrow{\alpha}_\beta I\,(I^\beta\, I) \quad \xrightarrow{\beta}_\beta I\, I \quad \rightarrow_\beta \quad I$$

A standardizing permutation swapping them produces the following standard derivation:

$$(I^\alpha\, I)\,(I^\beta\, I) \quad \xrightarrow{\alpha}_\beta I\,(I^\beta\, I) \quad \rightarrow_\beta (I^\beta\, I) \quad \xrightarrow{\beta}_\beta \quad I$$

This example shows that selecting mimimal redexes does not necessarily give standard derivations when the order is partial. A similar example in $\lambda_{\texttt{lsub}}$ is obtained by considering the derivations of $x^\alpha[x/\lambda y.y]((\lambda z^\beta.z)u)$.

The solution to this problem is to select an *external* redex [8, 10, 19, 21, 28, 32, 38, 39], *i.e.* a minimal one on which no other redex can indirectly act. The definition of *external redex for a derivation* in GLM's theory is given via an extraction process. In particular, the definition mentions the nesting order only indirectly. Alternatively, external redexes can be defined coinductively as *persistently* minimal redexes, as hinted in [38], *i.e.* the minimal redexes whose (unique) residual after any other redex is still persistently minimal.

We first recall the definition of external redex from [32], that requires two preliminary definitions. A redex A **traverses** a coinitial derivation d becoming B, written $A \xmapsto{d} B$, iff:

- For $d = \epsilon$, $A \xmapsto{\epsilon} B$ iff $B = A$.
- For $d = C; e$, $A \xmapsto{C;e} B$ iff $C \not\preceq A$ and $A' \xmapsto{e} B$, where $A\langle C\rangle A'$.

A redex A in t can be **extracted** from a derivation $d : t \twoheadrightarrow u$, written $A \curvearrowleft d$, if there exist d_1, d_2 s.t. $d = d_1; A'; d_2$ and $A \xmapsto{d_1} A'$. A redex A in t is **external for** $d : t \twoheadrightarrow u$ if $A \in \mathit{Ext}_\prec(d) := \{B \mid \forall e \equiv_\prec d \,.\, B \curvearrowleft e\}$.

The following result corresponds to Lemma 4.36 in [32].

LEMMA 5 (**External Gives Standard).** *Consider any ARS equipped with a partial order $\prec$ that enjoys the basic and the standardization axioms. If $d = A_1; \ldots ; A_n$ and $A_i \in \mathit{Ext}_\prec(A_i; \ldots ; A_n)$ $(1 \leq i \leq n)$, then d is $\prec$-standard.*

External redexes generalize leftmost redexes in λ-calculus, whose key properties are 1) no other redex can act on it, and 2) its unique residual after any other redex is still the leftmost redex. This suggests to define external redexes coinductively as follows.

Let $d : t \twoheadrightarrow u$ be a derivation. The **set of starting redexes** of d is $d^1 := \{B \mid \exists e \text{ s.t. } d \equiv_\prec B; e\}$. A redex A in t is **$\prec$-external for** d if d is not empty and:

1. *Minimality*: A is minimal in d^1, and
2. *Persistency*: whenever $d \equiv_\prec B; e$ and $A\langle B\rangle A'$ (A' is unique by minimality), then A' is $\prec$-external in e.

The next two technical lemmas are used to relate the two notions of externality. Their proofs are easy:

LEMMA 6. *Let A be $\prec$-external for d. Then:*

1. *$A \curvearrowleft d$, and*
2. *If $e \equiv_\prec d$ then $e^1 = d^1$ and A is $\prec$-external for e.*

LEMMA 7. *Let $A \curvearrowleft d$,* i.e. *$d = e; A'; f$ and $A \xmapsto{e} A'$. Then:*

1. *There exists g s.t. $d \equiv_{\prec} A; g$.*
2. *If $d = B; d'$ (with $B \neq A$) then $B \not\preceq A$ and if $A\langle B\rangle A''$ then $A'' \curvearrowleft d'$.*

We conclude the section with the equivalence of the two notions of external redex.

PROPOSITION 7. *A is $\prec$-external for d iff $A \in Ext_{\prec}(d)$.*

Proof. $\Rightarrow$) Let $e \equiv_{\prec} d$. Lemma 6.2 implies that A is $\prec$-external for e, and Lemma 6.1 gives $A \curvearrowleft e$, *i.e.* $A \in Ext_{\prec}(d)$. $\Leftarrow$) By coinduction on the definition of $\prec$-external. $A \in Ext_{\prec}(d)$ implies $A \curvearrowleft d$ so that Lemma 7.1 gives $A \in d^1$. Let $B \in d^1$ so that $d \equiv_{\prec} B; e$ and $A \curvearrowleft B; e$. If $B \neq A$ then Lemma 7.2 gives $B \not\preceq A$, that implies A minimal in d^1, and $A'' \curvearrowleft e$ if $A\langle B\rangle A''$. Thus, $A'' \in Ext_{\prec}(e)$. By the coinductive hypothesis A'' is $\prec$-external for e, and thus A is $\prec$-external for d.

11. Applications

In this section we apply the new characterization of external redexes to obtain a normalization theorem, and to prove that (weak) linear head reduction is standard.

A Normalizing Strategy for $\lambda_{\mathtt{lsub}}$ and $\lambda^{\sim}_{\mathtt{lsub}}$. Using the coinductive reformulation of external redexes, we can now provide an easy proof that the leftmost strategy, which always reduces the leftmost redex, is normalizing for $\lambda_{\mathtt{lsub}}$ and $\lambda^{\sim}_{\mathtt{lsub}}$, *i.e.* it reaches a normal form, if any. Let us say that a redex A is **universally $\prec$-external**, *i.e.* external with respect to any reduction step (and thus wrt any derivation), when the following holds:

1. *Universal minimality*: A is $\prec$-minimal in $\mathtt{Red}(t)$.
2. *Universal persistency*: for any $B \in \mathtt{Red}(t)$, if $A\langle B\rangle A'$ then A' is universally $\prec$-external.

THEOREM 6 (**Normalization**).

1. *The leftmost redex is universally $\prec_{\mathtt{L}}$-external $\lambda_{\mathtt{lsub}}$.*
2. *The leftmost strategy is normalizing for $\lambda_{\mathtt{lsub}}$ and $\lambda^{\sim}_{\mathtt{lsub}}$.*

Proof. 1) By coinduction. *Left-to-right order*: the leftmost redex is $\prec_{\mathtt{L}}$-minimum and by the **Enclave** axiom for $\prec_{\mathtt{L}}$ (Proposition 6) its residual after any other redex is still minimum and thus leftmost. We conclude using the coinductive hypothesis. 2) For $\lambda_{\mathtt{lsub}}$ it follows by Theorem 5.2 in Mellies' thesis [32] (p. 137, formulating abstractly an argument by Huet and Lévy [22]), stating that any strategy reducing $\prec_{\mathtt{L}}$-external redexes is normalizing. For $\lambda^{\sim}_{\mathtt{lsub}}$, let t be a term having an $\lambda^{\sim}_{\mathtt{lsub}}$-normal form. Since $\sim$ can be postponed (*cfr.* Lemma 3) and it is the identity on normal forms, then t has also a normal form in $\lambda_{\mathtt{lsub}}$. The leftmost strategy $\lambda_{\mathtt{lsub}}$-normalizes t. Since $\lambda_{\mathtt{lsub}} \subseteq \lambda^{\sim}_{\mathtt{lsub}}$, then the leftmost strategy also $\lambda^{\sim}_{\mathtt{lsub}}$-normalizes t.

Weak Linear Head Reduction is Standard. The leftmost strategy captures a relevant strategy in the theory of linear logic, the *weak* (*i.e.* not under abstractions) variant of *linear head reduction*.

To introduce such a notion let us first define weak head contexts by the grammar $W ::= \Box \mid Wt \mid W[x/t]$. **Weak linear head reduction** (WLHR) $\rightarrow_W$ is defined as the closure under *weak head contexts* of the rewriting rules $\{\mapsto_{\mathtt{db}}, \mapsto_{\mathtt{Wls}}\}$, where $\mapsto_{\mathtt{Wls}}$ is the following variation on $\mapsto_{\mathtt{ls}}$:

$$W[\![x]\!][x/u] \;\mapsto_{\mathtt{Wls}}\; W[\![u]\!][x/u]$$

Alternatively, $\rightarrow_W$ can be defined as the rewriting relation reducing only the $\{\mathtt{dB}, \mathtt{ls}\}$-redexes whose anchor is in a weak head context.

This notion of reduction is deterministic and tightly related to the π-calculus [3] and Krivine Abstract Machine [7, 14].

PROPOSITION 8 (**Weak Linear Head Reduction is Standard).**

1. *Every term $t \in \mathcal{T}$ has at most one $\rightarrow_W$ redex, and if it does then it is the leftmost redex of t.*
2. *$\rightarrow_W$-derivations are $\prec_{\mathtt{L}}$-standard (and thus also $\prec_{\mathtt{B}}$-standard).*

Proof. 1) By induction on t. 2) By Lemma 5 and Theorem 6.

Linear Head Reduction is Standard. The head strategy for $\lambda_{\mathtt{lsub}}$ historically arose as a special proof-nets strategy[8] [29]. Given the behavioral equivalence between $\lambda_{\mathtt{lsub}}$ and proof-nets, it can be easily expressed in our setting. For that, let us introduce the set of **head contexts**, that are generated by the grammar $H ::= \Box \mid \lambda x.H \mid H\,t \mid H[x/t]$. **Linear head reduction** (LHR) $\rightarrow_H$ is defined as the closure under *head contexts* of the rewriting rules $\{\mapsto_{\mathtt{db}}, \mapsto_{\mathtt{Hls}}\}$, where $\mapsto_{\mathtt{Hls}}$ is the following variation of the rewriting rule $\mapsto_{\mathtt{ls}}$:

$$H[\![x]\!][x/u] \;\mapsto_{\mathtt{Hls}}\; H[\![u]\!][x/u]$$

Alternatively, $\rightarrow_H$ can be defined as the rewriting relation reducing only the $\{\mathtt{dB}, \mathtt{ls}\}$-redexes whose anchor is inside a head context.

Historically, linear head reduction has been defined on ordinary λ-terms (*i.e.* without explicit substitutions) [14, 29] by using a notion of *virtual redex* (equivalent to a head-redex up to σ-equivalence [36]). The alternative approach used here, already in use in [2, 3, 5], improves the original presentation in several aspects: there is no need to work with virtual redexes, there is a one-to-one correspondence between LHR in proof-nets and LHR in $\lambda_{\mathtt{lsub}}$, and LHR can be seen as a strategy of $\lambda_{\mathtt{lsub}}$ although it is not a strategy of β-reduction. Moreover, this presentation of LHR admits a factorization theorem, and can be easily generalized to take place inside arguments of terms [2].

Linear head reduction is another motivation for considering the partial box order on redexes, as we now explain. A crucial point of our definition is that that $\rightarrow_H$—in contrast to $\rightarrow_W$—is nondeterministic, for instance we have both $(\lambda x.y[y/w])z \rightarrow_H (\lambda x.w[y/w])z$ and $(\lambda x.y[y/w])z \rightarrow_H y[y/w][x/z]$. This fact is not a drawback, rather a plus, as it faithfully mimicks the parallelism of cut-elimination in proof-nets. A simple case analysis shows that LHR enjoys the diamond property, *i.e.* any two steps commute and no duplication/erasure is involved. Therefore, the nondeterminism is harmless. In particular, all maximal LHR derivations (if any) have the same length. Concerning standardization, however, not all LHR derivations are standard with respect to the left-to-right order $\prec_{\mathtt{L}}$, consider:

$$(\lambda x.y[y/w])z \rightarrow_H (\lambda x.w[y/w])z \rightarrow_H w[y/w][x/z]$$

This fact is disappointing because the nondeterminism of LHR is only apparent, and so one would like to consider LHR sequences as standard, without having to re-organize them. The box order turns LHR into a standard strategy, giving to the presentation of LHR with explicit substitutions a solid status.

THEOREM 7 (**Linear Head Reduction is Standard).**

1. *Linear head redexes are universally $\prec_{\mathtt{B}}$-external.*
2. *Linear head reduction is $\prec_{\mathtt{B}}$-standard.*

Proof. 1) By coinduction on the definition of universally $\prec$-external. *Universal minimality*: the anchor of $\rightarrow_H$-redexes is out of all boxes, so they can never be nested by another redex. *Universal persistency*: it is easily seen that no redex can move the anchor of a $\rightarrow_H$-redex inside a box. So the residual of a $\rightarrow_H$-redex is a $\rightarrow_H$-redex and we conclude by the coinductive hypothesis. 2) A

[8] The one reducing only cuts out of all !-boxes that do not involve the auxiliary conclusion of any box.

universally external redex is easily seen to be external for every coinitial derivation. Then we conclude applying Lemma 5.

12. Conclusions

We study standardization for the linear substitution calculus, a calculus with explicit substitutions, that is not syntactically orthogonal and that it is equipped with an equational theory that makes it isomorphic to linear logic proof-nets. Our main result is a standardization theorem, *nonstandard* because it is based on a partial rather than a total order, it lifts to equivalence classes (*i.e.* proof-nets), and it provides a notion of standard derivation which departs from the one *by levels* in the linear logic literature.

Along the way, we provided other results: 1) a theory of residuals lifting to equivalence classes; 2) a simple left-to-right standardization theorem; 3) a coinductive characterization of the notion of external redex; 4) a normalization theorem; 5) a simple proof that (weak) linear head reduction is standard.

Last, we believe that a further contribution of this paper is the fine analysis of Gonthier, Lévy, and Melliès' axiomatic framework, which gives new intuitions on their complex notions.

Acknowledgments

We are grateful to Paul-André Melliès for fruitful discussions. This work was partially funded by the French-Argentinian Laboratory in Computer Science **INFINIS** (`www.infinis.org`) and by the Qatar National Research Fund under grant NPRP 09-1107-1-168.

References

[1] M. Abadi, L. Cardelli, P. L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 4(1):375–416, 1991.

[2] B. Accattoli. An abstract factorization theorem for explicit substitutions. In *RTA*, pages 6–21, 2012.

[3] B. Accattoli. Evaluating functions as processes. In *TERMGRAPH*, pages 41–55, 2013.

[4] B. Accattoli. *Jumping around the box: graphical and operational studies on λ-calculus and Linear Logic*. PhD thesis, *La Sapienza* University of Rome, february 2011.

[5] B. Accattoli and U. Dal Lago. On the invariance of the unitary cost model for head reduction. In *RTA*, pages 22–37, 2012.

[6] B. Accattoli and D. Kesner. The structural λ-calculus. In *CSL*, pages 381–395, 2010.

[7] B. Accattoli, P. Barenbaum, and D. Mazza. Distilling abstract machines. Submitted, 2013. `https://sites.google.com/site/beniaminoaccattoli/machines.pdf?attredirects=0`.

[8] H. P. Barendregt, R. Kennaway, J. W. Klop, and M. R. Sleep. Needed reduction and spine strategies for the lambda calculus. *Information and Computation*, 75(3):191–231, 1987.

[9] R. Bloo and K. Rose. Preservation of strong normalization in named lambda calculi with explicit substitution and garbage collection. In *Computing Science in the Netherlands*, pages 62–72. Netherlands Computer Science Research Foundation, 1995.

[10] G. Boudol. Computational semantics of term rewriting systems. In *Algebraic methods in semantics*, pages 169–236. Cambridge University Press, 1986.

[11] H. S. Bruggink. *Equivalence of reductions in higher-order rewriting*. PhD thesis, Utrecht University, 2008.

[12] D. Clark and R. Kennaway. Some properties of non-orthogonal term graph rewriting systems. In *SEGRAGRA*, volume 2 of *ENTCS*, pages 36–45, 1995.

[13] H. Curry and R. Feys. *Combinatory Logic*. Number Vol. 1 in Studies in logic and the foundations of mathematics. North-Holland, 1958.

[14] V. Danos and L. Regnier. Head linear reduction, 2003. `http://iml.univ-mrs.fr/~regnier/articles/pam.ps.gz`.

[15] V. Danos, H. Herbelin, and L. Regnier. Game semantics & abstract machines. In *LICS*, pages 394–405, 1996.

[16] D. de Carvalho, M. Pagani, and L. Tortora de Falco. A semantic measure of the execution time in linear logic. *Theor. Comput. Sci.*, 412(20):1884–1902, 2011.

[17] T. Ehrhard and L. Regnier. Böhm trees, krivine's machine and the taylor expansion of lambda-terms. In *CiE*, pages 186–197, 2006.

[18] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[19] G. Gonthier, J.-J. Lévy, and P.-A. Melliès. An abstract standardisation theorem. In *LICS*, pages 72–81, 1992.

[20] G. Huet. Residual theory in λ-calculus: A formal development. *Journal of Functional Programming*, 4(3):371–394, 1994.

[21] G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems, I and II. In *Computational Logic - Essays in Honor of Alan Robinson*, pages 395–414, 1991.

[22] G. Huet and J.-J. Lévy. Call by Need computations in non-ambiguous linear term rewriting systems. Technical Report 359, INRIA, 1979.

[23] D. Kesner and S. Lengrand. Resource operators for lambda-calculus. *Information and Computation*, 205(4):419–473, 2007.

[24] D. Kesner and S. Ó Conchúir. Milner's lambda calculus with partial substitutions, 2008. `http://www.pps.univ-paris-diderot.fr/~kesner/papers/shortpartial.pdf`.

[25] Z. Khasidashvili and J. R. W. Glauert. Discrete normalization and standardization in deterministic residual structures. In *ALP*, pages 135–149, 1996.

[26] J. W. Klop. *Combinatory Reduction Systems*. Phd thesis, Utrecht University, 1980.

[27] J.-J. Lévy. Réductions correctes et optimales dans le lambda-calcul. Thése d'Etat, Univ. Paris VII, France, 1978.

[28] L. Maranget. Optimal derivations in weak lambda-calculi and in orthogonal terms rewriting systems. In *POPL*, pages 255–269, 1991.

[29] G. Mascari and M. Pedicini. Head linear reduction and pure proof net extraction. *Theoretical Computer Science*, 135(1):111–137, 1994.

[30] P.-A. Melliès. Axiomatic rewriting theory I: A diagrammatic standardization theorem. In *Processes, Terms and Cycles: Steps on the Road to Infinity*, volume 3838 of *LNCS*, pages 554–638. Springer, 2005.

[31] P.-A. Melliès. Axiomatic rewriting theory VI: Residual theory revisited. In *RTA*, pages 24–50, 2002.

[32] P.-A. Melliès. *Description Abstraite de système de réécriture*. PhD thesis, Paris 7 University, 1996.

[33] R. Milner. Local bigraphs and confluence: Two conjectures. *ENTCS*, 175(3):65–73, 2007.

[34] L. Paolini and S. Ronchi Della Rocca. Parametric parameter passing lambda-calculus. *Information and Computation*, 189(1):87–106, 2004.

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

[36] L. Regnier. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 2(126):281–292, 1994.

[37] M. Takahashi. Parallel reductions in λ-calculus. *Inf. Comput.*, 118(1):120–127, 1995.

[38] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[39] V. van Oostrom. Normalisation in weakly orthogonal rewriting. In *RTA*, pages 60–74, 1999.

[40] V. van Oostrom and R. C. de Vrijer. Four equivalent equivalences of reductions. *ENTCS*, 70(6):21–61, 2002.

[41] H. Xi. Upper bounds for standardizations and an application. *Journal of Symbolic Logic*, 64(1):291–303, 1999.