Optimality and the Linear Substitution Calculus*

Pablo Barenbaum[1] and Eduardo Bonelli[2]

1 **Universidad de Buenos Aires, Buenos Aires, Argentina**
Université Paris 7, Paris, France
2 **Universidad Nacional de Quilmes, Buenos Aires, Argentina**
CONICET, Argentina
Stevens Institute of Technology, Hoboken, NJ, USA

Abstract

We lift the theory of optimal reduction to a decomposition of the lambda calculus known as the *Linear Substitution Calculus* (LSC). LSC decomposes β-reduction into finer steps that manipulate substitutions in two distinctive ways: it uses *context rules* that allow substitutions to act "at a distance" and rewrites modulo a set of *equations* that allow substitutions to "float" in a term. We propose a notion of redex family obtained by adapting Lévy labels to support these two distinctive features. This is followed by a proof of the finite family developments theorem (FFD). We then apply FFD to prove an optimal reduction theorem for LSC. We also apply FFD to deduce additional novel properties of LSC, namely an algorithm for standardisation by selection and normalisation of a linear call-by-need reduction strategy. All results are proved in the axiomatic setting of Glauert and Khashidashvili's *Deterministic Residual Structures*.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Rewriting, Lambda Calculus, Explicit Substitutions, Optimal Reduction

Digital Object Identifier 10.4230/LIPIcs.FSCD.2017.9

1 Introduction

The λ-calculus distills the essence of functional programming languages. Programs are represented as syntactic terms, and execution corresponds to repeated simplification of these terms using a reduction rule called *β-reduction*. The study of the λ-calculus has produced a vast body of work, by no means limited to functional programming. It has also played a key role in laying the foundations of modern *rewriting theory*. Rewriting is an abstract model of computation in which rather than syntactic terms and their step-by-step reduction, one considers sets of *arrows* over arbitrary *objects*. The λ-calculus is an example of a rewriting system, but there are many other ones, such as graph rewriting systems or first-order term rewriting systems. The impact of the λ-calculus in rewriting is that its study has suggested generalizations of numerous properties to abstract rewriting frameworks.

There are many variants of the λ-calculus. In its simplest presentation, it consists of a unique *reduction rule β* that models the application of a function to an argument. Despite the conciseness of its definition, the study of the λ-calculus unveils surprisingly rich mathematical structures. One example is its *denotational semantics*, which attempts to provide *models* for the λ-calculus, and motivates the theory of *domains*. Another example arises from attempting to compare *derivations*. Given that computation is modeled by reduction and that there are

* Work partially supported by PICT-2012-2747 (Ministerio de Ciencia, Tecnología e Innovación Productiva, Argentina) and LIA INFINIS.

2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017).
Editor: DaleMiller; Article No. 9; pp. 9:1–9:41
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

multiple ways to reduce a term, how do these choices compare? This requires analyzing the *derivation space* of a term. The derivation space of a term is the set of all derivations, *i.e.* sequence of (composable) β-reduction steps, starting from that term. Establishing whether a particular choice produces derivations that are "better" than others in any reasonable sense involves *comparing* the resulting derivations. This, in turn, involves tracking steps around in order to relate the steps of one derivation to those of another one, hence determining that they *correspond* to each other. *Theories of residuals* attempt to provide a framework for analyzing the derivation space.

An example in the λ-calculus follows in order to provide a better intuition on what is meant by a theory of residuals. Consider the term $(I\,\Delta)\,(I\,x)$, where Δ stands for $\lambda x.x\,x$ and I for $\lambda x.x$. Below we depict the derivation space of this term. As mentioned, a study of the *structure* of this space involves understanding how derivations are related and, since derivations are built from β-steps, how β-steps from one derivation are related to those of another.

$$\begin{tikzcd}
 & (I\,\Delta)\,x \arrow[rr, "\mathsf{R}_2"] & & \Delta\,x \arrow[ddr, "\mathsf{R}_3"] & \\
(I\,\Delta)\,(I\,x) \arrow[ur, "\mathsf{R}_1"] \arrow[dr, "\mathsf{R}_4"] & & & x\,(I\,x) \arrow[dr, "\mathsf{R}_8"] & \\
 & \Delta\,(I\,x) \arrow[uurr, "\mathsf{R}_5"] \arrow[r, "\mathsf{R}_6"] & (I\,x)\,(I\,x) \arrow[ur, "\mathsf{R}_7"] \arrow[dr, "\mathsf{R}_9"] & & x\,x \\
 & & & (I\,x)\,x \arrow[ur, "\mathsf{R}_{10}"] &
\end{tikzcd}$$

An example of a derivation is $\mathsf{R}_4;\mathsf{R}_6;\mathsf{R}_7;\mathsf{R}_8$. It consists of four β-steps denoted R_4, R_6, R_7, and R_8. Notice that the derivation $\mathsf{R}_4;\mathsf{R}_6;\mathsf{R}_7;\mathsf{R}_8$ essentially performs the same steps as the derivation $\mathsf{R}_4;\mathsf{R}_6;\mathsf{R}_9;\mathsf{R}_{10}$ since the derivations $\mathsf{R}_7;\mathsf{R}_8$ and $\mathsf{R}_9;\mathsf{R}_{10}$ do the same computational work, namely they reduce the two copies of $(I\,x)$ in $(I\,x)\,(I\,x)$, only in a different order (reducible subterms such as $(I\,x)$ are called *redexes*). This suggests *algebraic principles* over derivations, such as $\mathsf{R}_7;\mathsf{R}_8 \simeq \mathsf{R}_9;\mathsf{R}_{10}$. Not all steps can be commuted. For example, R_4 cannot be commuted with R_6 because the former *creates* the latter. Note also that, we write $\mathsf{R}_7;\mathsf{R}_8 \simeq \mathsf{R}_9;\mathsf{R}_{10}$ and not $\mathsf{R}_7;\mathsf{R}_8 \simeq \mathsf{R}_8;\mathsf{R}_7$ because R_9 is the form that R_8 adopts when it is fired from $(I\,x)\,(I\,x)$ rather than from $x\,(I\,x)$; we say that R_8 is a *residual* or what is left of R_9 after R_7. As may be gleamed from this preliminary discussion, it soon becomes clear that any prospective algebraic principles must arise from identifying β-steps and tracking them along derivations. Such *theories of residuals* mark and track β-steps. However, this is just the starting point of an analysis of the structure of the derivation space since, when one attempts to prove properties of derivations, one realizes that more general principles are required. The principles include the following, presented in increasing level of complexity:

- *Finite Developments*: marking and tracking *sets* of β-steps in a term and showing that their reduction terminates;
- *Finite Family Developments*: marking and tracking *sets* of β-step that may have been created along the way in a derivation and also showing their termination properties;
- *Redex Families*: identifying created β-steps that are related in the sense that they could be shared;
- *Optimal Reduction*: the apex of residual theory.

Optimal reduction characterizes derivations in the derivation space that are shortest in a precise sense and has close ties with Geometry of Interaction [17]. It arose with a clear motivation in the implementation of the λ-calculus since it addresses the concern of avoiding unnecessary β-steps. This same motivation, bridging the gap between programming languages and their implementation, is shared by *Calculi with Explicit Substitutions*.

Calculi with Explicit Substitutions (ES). Substitution in the λ-calculus is a non-trivial metalanguage operation that simultaneously replaces every occurrence of a variable by a given term. In contrast, in actual implementations of functional programming languages it usually takes various steps to perform a substitution. For example, variables might be bound to values in an environment, and looked up in the environment whenever needed. Calculi with ES were introduced to bridge the gap between the λ-calculus and its implementations. They are characterized by the presence of an explicit operator in the object language for modeling substitution. A paradigmatic example calculus with ES is $\lambda\sigma$ [1] which includes, among others, rules $\mathtt{beta}$[1] $(\lambda x.s)\,t \mapsto s[x/t]$, where $s[x/t]$ denotes an ES, and $\mathtt{app}$ $(s\,t)[x/u] \mapsto s[x/u]t[x/u]$, for propagating substitutions over applications. Unfortunately, these rules produce a critical pair rendering $\lambda\sigma$ a syntactically *non*-orthogonal system, a situation common to most known calculi with ES, as depicted below where $\to_{\mathtt{beta}}$ means application of the $\mathtt{beta}$-rule in an arbitrary context:

$$s[x/t][y/u] \;{}_{\mathtt{beta}}\!\leftarrow ((\lambda x.s)\,t)[y/u] \to_{\mathtt{app}} (\lambda x.s)[y/u]t[y/u]$$

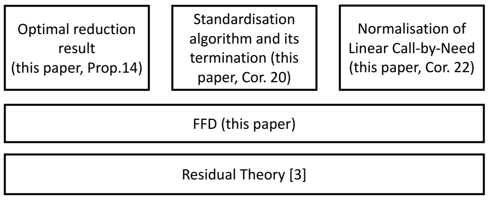
The $\mathtt{beta}$-step in the middle term has been “erased” in the right term because $\mathtt{beta}$ and $\mathtt{app}$ overlap. It is unclear how to devise a reasonable residual theory in such a situation[2]. The λ-calculus is thus set apart from traditional calculi with ES since in the latter the lack of orthogonality makes it impossible to address a proper theory of residuals, let alone optimality.

The Linear Substitution Calculus (LSC). The LSC is a calculus with ES introduced rather recently [6]. It is based on a *contextual* approach: rewriting rules are expressed using contexts, which allows for non-local interactions between subterms, and obviates the need to propagate explicit substitutions. It is also equipped with a relation of *structural equivalence* between terms, which reflects the exact correspondence between terms and their encoding as proof nets (which are graphs), linear logic being the domain in which the LSC was originally conceived.

The fact that the LSC encodes a graph-rewriting system based on proof nets, rather than *ad hoc* syntactic machinery for implementing explicit substitutions, is one of the reasons for it being relatively well-behaved. In particular, the LSC does not suffer from the above mentioned problems of other calculi with ES. Recent work has shown that, even though the LSC is not syntactically orthogonal, it enjoys *semantical orthogonality*, which means that it can be given a sensible *theory of residuals* [4]. On the other hand, not all expected properties of residuals that hold for the λ-calculus turn out to hold for LSC (*e.g. enclave* and *stability* fail [4]). Besides, the very same fact that the LSC encodes a graph-rewriting system is the source of some technical challenges, especially because the encoding is based on two distinctive features: the use of context rules and a notion of structural equivalence. One complication is that the usual tree-like representation of terms and nesting of redexes that guide our intuition in the λ-calculus and first-order term rewriting no longer applies. *E.g.*, in the term $(x\,x)[x/y]$ either of the two occurrences of x might be replaced by y, so there are two redexes $(x\,x)[x/y] \to (y\,x)[x/y]$ and $(x\,x)[x/y] \to (x\,y)[x/y]$. These redexes overlap in the standard tree reading of terms, yet they should by all means be considered “independent” redexes. Another complication is that redex creation may take place at a distance, such as in the step $(x\,y)[x/I] \to (I\,y)[x/I]$, in which the substitution of x by the identity creates a beta-redex. Anyhow, enough properties are satisfied by LSC’s residual theory for it to be a reasonable starting point for following the path set by the λ-calculus: finite developments, finite family developments, redex families and optimal reduction.

[1] $\lambda\sigma$ is actually based on de Bruijn indices, we use variable names for expository purposes.
[2] There are some attempts at addressing residual theories for syntactically non-orthogonal systems [14, 24].

Goal and value of the paper. This paper attempts to reclaim, for the LSC, the status of the λ-calculus by providing a theory of optimality for it. The key technical result on which it builds is a *Finite Family Developments (FFD)* theorem (Thm. 4 on page 9). FFD is used as a tool to develop various novel results, including optimal reduction itself, termination of standardisation procedures, and normalisation strategies. All results in this paper are proved in an axiomatic setting, namely *Deterministic Residual Structures* [16], whose axioms LSC is shown to comply with.



The reader will no doubt realize the technical nature of this paper. Standardisation, normalisation and, most notably, optimal reduction are known to be technical in themselves. The LSC is not much of an aid in this sense, its use of context rules and rewriting modulo a set of equations only seem to make matters yet more technical. We are well aware of this fact and have strived to present the material in such a way that the reader is able to see through the technicalities and perceive the value of this paper, namely how it manages to lift the theory of optimal reduction to *refinements* of the λ-calculus.

Structure of this paper. Sec. 2 defines LSC. We also review the definition of residuals for LSC and present *Deterministic Residual Structures* [16]. The Lévy labeled LSC is presented in Sec. 3. The *Finite Family Developments Theorem* is addressed in Sec. 4, its proof broken down into three principles. Sec. 5 addresses optimal reduction: we recall the notion of *Deterministic Residual Structure* from [16] and then prove that our labeled LSC is an instance of such structure. Sec. 6 introduces standardisation by selection (of multi-redexes) and proves termination. Sec. 7 studies a linear call-by-need strategy and proves that it normalizes. We conclude in Sec. 8. Proofs of all results are included in the extended version.

Related Work. The literature on FD is quite extensive; the reader is invited to consult [26, Ch. 4]. Some abstract notions of rewriting establish FD as an axiom [25, 22, 16]. For classical references to FFD there is [19, 12]. FFD generalizes Hyland-Wadsworth labels which records the depth of the labels [26, 8.4.4]. Also, it is referred to as *Generalized Finite Developments* in [20]. FFD was extended to higher-order rewriting [27, 15]. LSC was introduced by Milner [**?**] and then adopted by Accattoli and Kesner [6, 4] although similar ideas were also developed by de Bruijn and Nederpelt (see [8] for additional references). LSC has somewhat revived the explicit substitutions community given its success in explaining results in the classical λ-calculus (*e.g.* cost models, call-by-value solvability, call-by-value on open terms, linear head reduction and abstract machines, etc.) [9, 3, 7, 5, 8]. Regarding standardisation for LSC, [4] proves the existence and uniqueness of standard derivations. However, standardisation algorithms are not studied. Residuals for calculi with ES have also been studied by Melliès [22, 23] where he developed a general theory of rewriting and applied it, among others, to $\lambda\sigma$ [1]. Regarding labels, ES and sharing there is some work [21, 13], however it all addresses weak reduction. We should also mention [28] which uses a calculus of ES and suggests an optimal reduction result for it. However, no proofs are supplied.

2 The Linear Substitution Calculus

Given *variables* $x, y, z, \ldots$, the set $\mathcal{T}$ of **terms** is defined by the grammar:

$$t, s \quad ::= \quad x \mid t\,s \mid \lambda x.t \mid t[x/s]$$

A term of the form $t[x/s]$ is called a **substitution**. The notion of free and bound variables is defined as usual, in particular $\lambda x.t$ and $t[x/s]$ bind all free occurrences of x in t. We write

$\mathsf{fv}(s)$ (resp. $\mathsf{bv}(s)$) for the set of free (resp. bound) variables of s. A **context** is a term with a unique occurrence of a singled-out variable $\Box$ called a hole. If $\mathtt{C}$ is a context, then $\mathtt{C}\langle t\rangle$ is the term resulting from replacing the hole in $\mathtt{C}$ with t (possibly resulting in the capture of free variables of t in $\mathtt{C}$). We write $\mathtt{C}\langle\!\langle t\rangle\!\rangle$ when the free variables of t are not captured by $\mathtt{C}$.

Terms are considered up to a set of **structural equations** that allow commuting some substitutions around, in order to quotient out the order imposed by the fact that terms are trees rather than graphs, and to reflect more closely their correspondence with proof-nets. **Structural equivalence**, written $t \sim s$, is the reflexive, symmetric, transitive, and contextual closure of the following axioms:

$$\begin{array}{rcll} (\lambda x.t)[y/s] & \sim_\lambda & \lambda x.t[y/s] & \text{if } x \neq y \text{ and } x \notin \mathsf{fv}(s) \\ (t\,s)[x/u] & \sim_@ & t[x/u]\,s & \text{if } x \notin \mathsf{fv}(s) \\ t[x/s][y/u] & \sim_{com} & t[y/u][x/s] & \text{if } x \neq y,\ x \notin \mathsf{fv}(u), \text{ and } y \notin \mathsf{fv}(s) \end{array}$$

▶ **Definition 1.** The LSC is the pair $\langle \mathcal{T}, \rightarrow\rangle$ where $\rightarrow$ is defined by the rules $\{\mathsf{db}, \mathsf{ls}\}$ modulo the equations $\{\sim_\lambda, \sim_@, \sim_{com}\}$, *i.e.* $t \rightarrow u$ if and only if $t \sim t'(\rightarrow_{\mathsf{db}} \cup \rightarrow_{\mathsf{ls}})u' \sim u$. Here $\rightarrow_{\mathsf{db}}$ is $\mathtt{C}\langle\mapsto_{\mathsf{db}}\rangle$ (*i.e.* the contextual closure of $\mapsto_{\mathsf{db}}$) and $\rightarrow_{\mathsf{ls}}$ is $\mathtt{C}\langle\mapsto_{\mathsf{ls}}\rangle$, $\mapsto_{\mathsf{db}}$ and $\mapsto_{\mathsf{ls}}$ being[3]:

$$(\lambda x.t)\mathtt{L}\ s \mapsto_{\mathsf{db}} t[x/s]\mathtt{L} \qquad\qquad \mathtt{C}\langle\!\langle x\rangle\!\rangle[x/t] \mapsto_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x/t]$$

▶ Remark. Originally LSC includes $\rightarrow_{\mathsf{gc}}$, defined as the contextual closure of: $t[x/s] \mapsto_{\mathsf{gc}} t$, if $x \notin \mathsf{fv}(t)$. However, in the literature it is often ignored: dropping it simplifies the metatheory (*e.g.* LSC with $\rightarrow_{\mathsf{gc}}$ does not enjoy *stability* [4]; *cf.* Rem. 4) at no loss of generality since $\rightarrow_{\mathsf{gc}}$ can be postponed past $\rightarrow_{\mathsf{db}}$ and $\rightarrow_{\mathsf{ls}}$.

A **LSC-step** $(\mathsf{R}, \mathsf{S}, \ldots)$ is either a pair of the form $\langle \mathtt{C}, (\lambda x.t)\mathtt{L}\ s\rangle$ (a db-step) or a triple of the form $\langle \mathtt{D}, \mathtt{C}\langle\!\langle x\rangle\!\rangle[x/t], \mathtt{C}\rangle$ (an ls-step). Steps, as defined here, are often also called *redexes*. We write $\mathsf{src}(\mathsf{R})$ and $\mathsf{tgt}(\mathsf{R})$ for the source and target of R, respectively. Two redexes are said to be **coinitial** (resp. **cofinal**) if their sources (resp. targets) coincide. A **derivation** $(\rho, \sigma, \ldots)$ is a sequence of steps $\mathsf{R}_1 \ldots \mathsf{R}_n$ s.t. $\mathsf{src}(\mathsf{R}_i) = \mathsf{tgt}(\mathsf{R}_{i-1})$ for $i \in 2..n$. We write ϵ for the empty derivation and $t \twoheadrightarrow s$ if there is a derivation from t to s and say that t is its source and s its target (empty derivations are assumed to be indexed by terms). *E.g.*:

$$\begin{array}{cccccc} & (\lambda x.\lambda y.xyx)II & \rightarrow_{\mathsf{db}} & (\lambda y.xyx)[x/I]I & \rightarrow_{\mathsf{ls}} & (\lambda y.Iyx)[x/I]I \\ \rightarrow_{\mathsf{db}} & (\lambda y.z[z/y]x)[x/I]I & \sim & ((\lambda y.z[z/y]x)I)[x/I] & \rightarrow_{\mathsf{db}} & (z[z/y]x)[y/I][x/I] \end{array}$$

Residuals for LSC [4]. Given markers $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$, **marked terms**[4] are defined as follows, where α ranges over markers: $t, s ::= x \mid x^\alpha \mid t\,s \mid \lambda x.t \mid \lambda x^\alpha.t \mid t[x/s]$. Since markers are intended to mark redexes, we consider only *well-marked terms*: terms where marks are only placed on redexes (*cf.* [4]). For example, $\lambda x^{\mathbf{a}}.x$ and $\lambda x.x^{\mathbf{a}}$ are not well-marked. **Marked reduction** $\xrightarrow{\alpha}$ on well-marked terms is defined as the contextual closure of the following rewriting rules, where the contexts below are also well-marked:

$$(\lambda x^{\mathbf{a}}.t)\mathtt{L}s \overset{\mathbf{a}}{\mapsto}_{\mathsf{dB}} t[x/s]\mathtt{L} \qquad\qquad \mathtt{C}\langle\!\langle x^{\mathbf{a}}\rangle\!\rangle[x/t] \overset{\mathbf{a}}{\mapsto}_{\mathsf{ls}} \mathtt{C}\langle\!\langle t\rangle\!\rangle[x/t]$$

We write $\mathtt{Red}(s)$ (resp. $\mathtt{Red}_{\mathbf{a}}(s)$) for the set of redexes (resp. marked $\mathbf{a}$) in s. The **set of residuals of** R **after** S is given by $\mathsf{R}/\mathsf{S} := \{\mathtt{Red}_{\mathbf{a}}(u') \mid \mathtt{mark}(t, \mathsf{R}, \mathbf{a}) \xrightarrow{\mathsf{S}} u'\}$, where

[3] For the $\mapsto_{\mathsf{db}}$ rule we have opted to use the more familiar $t\mathtt{L}$ rather than $\mathtt{L}\langle t\rangle$.

[4] [4] speaks of *labeled terms*, we use "marked" to stress that they are not to be confused with Lévy labels introduced in Sec. 3.

$\mathtt{mark}(t, \mathsf{R}, \mathbf{a})$ denotes the result of marking redex R in t with $\mathbf{a}$. Given steps S and T such that $\mathsf{tgt}(\mathsf{S}) = \mathsf{src}(\mathsf{T})$, we say that S **creates** T if there is no R such that $\mathsf{R}/\mathsf{S} = \mathsf{T}$. A **multistep** is a non-empty finite set $\mathcal{M}$ of coinitial steps. The residual relation may be extended to multisteps as expected: $\mathcal{M}/\mathsf{S} \stackrel{\text{def}}{=} \bigcup_{\mathsf{R}\in\mathcal{M}} \mathsf{R}/\mathsf{S}$. Also, we may define the residual of a set of steps after a derivation: $\mathcal{M}/\epsilon \stackrel{\text{def}}{=} \mathcal{M}$, and $\mathcal{M}/\mathsf{S}\sigma \stackrel{\text{def}}{=} (\mathcal{M}/\mathsf{S})/\sigma$. Examples of the residual relation follow [4]: let $v = (x^{\mathbf{b}}x^{\mathbf{b}}x^{\mathbf{c}}y^{\mathbf{c}})[x/y][y/w]$, $\mathsf{S} = \langle\Box[y/w], (x^{\mathbf{b}}x^{\mathbf{b}}x^{\mathbf{c}}y^{\mathbf{c}})[x/y], x^{\mathbf{b}}\Box\, x^{\mathbf{c}}y^{\mathbf{c}}\rangle$ (so that $v \xrightarrow{\mathsf{S}} (x^{\mathbf{b}}y\,x^{\mathbf{c}}y^{\mathbf{c}})[x/y][y/w]$), and $\mathsf{R} = \langle\Box[y/w], (x^{\mathbf{b}}x^{\mathbf{b}}x^{\mathbf{c}}y^{\mathbf{c}})[x/y], \Box\, x^{\mathbf{b}}x^{\mathbf{c}}y^{\mathbf{c}}\rangle$. Observe that $\mathtt{mark}(v, \mathsf{R}, \mathbf{a}) = (x^{\mathbf{a}}x^{\mathbf{b}}x^{\mathbf{c}}y^{\mathbf{c}})[x/y][y/w] \xrightarrow{\mathsf{S}} (x^{\mathbf{a}}yx^{\mathbf{c}}x^{\mathbf{c}})[x/y][y/w]$. Therefore, if $\mathcal{M} = \{\mathsf{R}\}$, then $\mathcal{M}/\mathsf{S} = \{\mathsf{R}'\}$ where $\mathsf{R}' = \langle\Box\,[y/w], (x^{\mathbf{b}}y\,x^{\mathbf{c}}y^{\mathbf{c}})[x/y], \Box\, y\, x^{\mathbf{c}}y^{\mathbf{c}}\rangle$. Suppose now that $\mathcal{M} = \mathtt{Red}_{\mathbf{c}}(v)$. Then a similar analysis for each $\mathsf{R} \in \mathcal{M}$ yields $\mathcal{M}/\mathsf{S} = \{\mathsf{R}_1, \mathsf{R}_2\}$ where $\mathsf{R}_1 = \langle\Box\,[y/w], (x^{\mathbf{b}}y\,x^{\mathbf{c}}y^{\mathbf{c}})[x/y], x^{\mathbf{b}}y\,\Box\, y^{\mathbf{c}}\rangle$ and $\mathsf{R}_2 = \langle\Box\,, v, (x^{\mathbf{b}}y\,x^{\mathbf{c}}\Box)[x/y]\rangle$.

▶ Remark. Structural equivalence $\sim$ can be lifted to well-marked terms and the residual relation on steps shown to pass the equations in the sense that they induce a bijection between the steps they relate. Moreover, $\sim$ is a strong bisimulation with respect to $\to$ [4].

Marks are useful to study *developments*. For any $\mathcal{M} \subseteq \mathtt{Red}(t)$, a (possibly infinite) derivation from t, $\rho = \mathsf{R}_1\mathsf{R}_2\ldots$, is a **development** of $\mathcal{M}$ iff $\mathsf{R}_\mathsf{i} \in \mathcal{M}/\mathsf{R}_1\ldots\mathsf{R}_{\mathsf{i}-1}$ for all i. A development ρ of $\mathcal{M}$ is said to be **complete** if it is *maximal*, *i.e.* if there is no non-empty derivation σ s.t. $\rho\sigma$ is also a development of $\mathcal{M}$. Note that if ρ is finite, then $\mathcal{M}/\rho = \emptyset$. *E.g.* $t_0 = (xx)[x/t] \to (xt)[x/t] \to (tt)[x/t]$ is a complete development of the set containing the two ls-steps of t_0.

An Abstract Framework: Deterministic Residual Structures

Abstract rewriting frameworks, such as *Orthogonal Axiomatic Rewrite Systems* [22] and *Deterministic Residual Structures (DRS)* [16], single out properties that well-behaved residuals should enjoy. LSC with the above defined notion of residual satisfies the properties of both[5] of these frameworks. Here we briefly describe DRS since they shall be used when we address the applications of FFD to LSC.

An **Abstract Rewrite System** (ARS) is a tuple $\langle \mathit{Obj}, \mathit{Stp}, \mathsf{src}, \mathsf{tgt}\rangle$ of *objects*, *steps* and functions src and tgt that return the source and target, resp., of a step. Moreover, we assume that ARSs are finitely branching, *i.e.* that there is only a finite number of steps having the same object as source.

▶ **Definition 2** (Deterministic Residual Structure)**.** A DRS is an ARS endowed with a residual relation _/_ satisfying the following axioms:

1. Unique ancestor. If $\mathsf{R} \in \mathsf{R}_1/\mathsf{S}$ and $\mathsf{R} \in \mathsf{R}_2/\mathsf{S}$ then $\mathsf{R}_1 = \mathsf{R}_2$.
2. Acyclicity. If $\mathsf{R} \neq \mathsf{S}$ and $\mathsf{R}/\mathsf{S} = \emptyset$ then $\mathsf{S}/\mathsf{R} \neq \emptyset$.
3. Finite Developments (FD). Let ρ, σ be derivations and $\mathcal{M}$ be a set of coinitial steps.
   a. Finite. If ρ a *development* of $\mathcal{M}$, then ρ is finite.
   b. Cofinal. If ρ, σ are *complete developments* of $\mathcal{M}$, then ρ and σ end in the same term.
   c. Equivalent. If ρ, σ are complete developments of $\mathcal{M}$ then they induce the same residual relation, *i.e.* $\mathsf{R}/\rho = \mathsf{R}/\sigma$ for every step R coinitial with ρ.

In the case of LSC: Acyclicity is immediate; Unique ancestor and Finite Developments are proved in [4]. We next introduce some definitions proper to any DRS.

[5] Although, see comment on enclave and stability in the introduction.

Each multistep determines a "super-step" by taking (any) complete development of that set. Its target is well-defined by axiom COFINAL. A **multistep reduction** D is a sequence of multisteps $\mathcal{M}_1 \ldots \mathcal{M}_n$ s.t. $\mathsf{src}(\mathcal{M}_i) = \mathsf{tgt}(\mathcal{M}_{i-1})$ for $i \in 2..n$.

Define $\tau_1 \,\mathsf{R}\, \sigma \, \tau_2 \equiv^1 \tau_1 \,\mathsf{S}\, \rho \, \tau_2$, where σ is a complete development of S/R and ρ is a complete development of R/S. We define **permutation equivalence**, $\equiv$, as the reflexive and transitive closure of $\equiv^1$. Note that $\rho \equiv \sigma$ implies $/\rho = /\sigma$ (*i.e.* they induce the same residual relation).

Let $\mathcal{X}$ be a set of objects in a DRS. An object s is **$\mathcal{X}$-normalizing** if there is a derivation from s to an object in $\mathcal{X}$. We call a step $\mathsf{R} \in \mathtt{Red}(s)$ **$\mathcal{X}$-needed** if at least one residual of it is contracted in any reduction from s to an object in $\mathcal{X}$. *E.g.* for $\mathcal{X}$ the set of normal forms, the underlined step is needed in $\lambda x.\underline{II}$, but not if $\mathcal{X}$ is the set of abstractions.

A **redex with history** in a DRS is a non-empty derivation, usually written $\rho\mathsf{R}$ to single out the last step. We write $\mathsf{Hist}(t) \overset{\text{def}}{=} \{\rho\mathsf{R} \mid \mathsf{src}(\rho) = t\}$ for the set of redexes with history whose source is the object t. The **copy relation** between coinitial redexes with history, written $\rho\mathsf{R} \leq \sigma\mathsf{S}$ is defined to hold if and only if there is a derivation τ such that $\rho\tau \equiv \sigma$ and $\mathsf{S} \in \mathsf{R}/\tau$. The reflexive, symmetric and transitive closure of $\leq$, written $\leftrightsquigarrow$, is called the **family relation**. Its equivalence classes are called **redex families**. *E.g.*. if $\rho : \Delta(\underline{II}I) \to \Delta(II)$ and $\sigma : \underline{\Delta(III)} \to (\underline{II}I)(III) \to (II)(III)$, then $\sigma(\overline{II})(III)$ is in the family of $\rho\Delta(\overline{II})$ since $\sigma(\overline{II})(III) \leftrightsquigarrow \rho\Delta(\overline{II})$.

Let $\mathcal{F}$ be a set of redex families $\{\mathsf{Fam}_{\leftrightsquigarrow}(\rho_i)\}_{i \in I}$ such that $\rho_i \in \mathsf{Hist}(t)$, for some fixed t. A **family development** of $\mathcal{F}$ is a pair $_\tau|\rho$ where the first component is a "history" $\tau : t \twoheadrightarrow t'$ and the second component is a (possibly infinite) derivation $\rho = \mathsf{R}_1\mathsf{R}_2 \ldots$ from t' such that for every index $i \geq 1$, we have $\mathsf{Fam}_{\leftrightsquigarrow}(\tau\mathsf{R}_1 \ldots \mathsf{R}_i) \in \mathcal{F}$. Usually the history τ is empty, ρ starts from t, and we identify $_\epsilon|\rho$ with ρ. A family development $_\tau|\rho$ of $\mathcal{M}$ is said to be **complete** if it is *maximal*, *i.e.* if there is no non-empty derivation σ s.t. $_\tau|\rho\sigma$ is also a family development of $\mathcal{F}$.

3 The Labeled LSC

Given *initial labels* $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$ including a distinguished one "$\bullet$", we define **labels** $\mathcal{L}$ as:

$$\alpha, \beta ::= \mathbf{a} \mid \lceil\alpha\rceil \mid \lfloor\alpha\rfloor \mid \mathtt{db}(\alpha) \mid \alpha\,\beta$$

We assume juxtaposition $\alpha\,\beta$ to be associative. Labels that are *not* of the form $\alpha\,\beta$ are called **atomic labels**. Labels of the form $\mathtt{db}(\alpha)$ will be used to leave a trace indicating that a db-step was contracted (*cf.* Rem. 3). Similarly, "$\bullet$" will be used to leave a trace indicating the place in which an ls-step was contracted. The remaining labels play a similar rôle to that of Lévy labels for λ-calculus.

The set of **labeled terms** ($\mathcal{T}^\ell$), **labeled contexts** and **labeled substitution contexts** are defined by the following grammar:

$$\begin{array}{rcl}
t, s, u, r, q & ::= & x^\alpha \mid \lambda^\alpha x.t \mid @^\alpha(t, s) \mid t[x/s] \\
\mathtt{C} & ::= & \Box \mid \lambda^\alpha x.\mathtt{C} \mid @^\alpha(\mathtt{C}, t) \mid @^\alpha(t, \mathtt{C}) \mid \mathtt{C}[x/t] \mid t[x/\mathtt{C}] \\
\mathtt{L} & ::= & \Box \mid \mathtt{L}[x/t]
\end{array}$$

Note that substitutions are not labeled. The **external label** of a term t, written $\ell(t)$, is the label decorating the outermost node of t, jumping over substitutions:

$$\begin{array}{rclcrcl}
\ell(x^\alpha) & \overset{\text{def}}{=} & \alpha & \quad & \ell(\lambda^\alpha x.t) & \overset{\text{def}}{=} & \alpha \\
\ell(@^\alpha(t, s)) & \overset{\text{def}}{=} & \alpha & & \ell(t[x/s]) & \overset{\text{def}}{=} & \ell(t)
\end{array}$$

We also define the following operation $\alpha : t$ for adding a label to an (already labeled) term, jumping over substitutions:

$$\begin{array}{rcl@{\qquad}rcl} \alpha : x^{\beta} & \stackrel{\text{def}}{=} & x^{\alpha\beta} & \alpha : (\lambda^{\beta}x.t) & \stackrel{\text{def}}{=} & \lambda^{\alpha\beta}x.t \\ \alpha : @^{\beta}(t,s) & \stackrel{\text{def}}{=} & @^{\alpha\beta}(t,s) & \alpha : (t[x/s]) & \stackrel{\text{def}}{=} & (\alpha : t)[x/s] \end{array}$$

Note that we have $\ell(t\mathtt{L}) = \ell(t)$ and $\alpha : (t\mathtt{L}) = (\alpha : t)\mathtt{L}$.

We shall require one more operation on labels. The **outermost (resp. innermost) atomic label** of a label α is written $\uparrow(\alpha)$ (resp. $\downarrow(\alpha)$) and defined as:

$$\uparrow(\alpha) \stackrel{\text{def}}{=} \begin{cases} \uparrow(\alpha_1) & \text{if } \alpha = \alpha_1\alpha_2 \\ \alpha & \text{otherwise} \end{cases} \qquad \downarrow(\alpha) \stackrel{\text{def}}{=} \begin{cases} \downarrow(\alpha_2) & \text{if } \alpha = \alpha_1\alpha_2 \\ \alpha & \text{otherwise} \end{cases}$$

These functions are well-defined modulo associativity of juxtaposition. We also write $\uparrow(t)$ for $\uparrow(\ell(t))$.

A labeled term is **initially labeled** if all its labels are initial and pairwise distinct. A labeled term $t \in \mathcal{T}^{\ell}$ is a **variant** of an (unlabeled) term $t_0 \in \mathcal{T}$ if erasing all the labels from t yields t_0. We say that two labeled terms $t, s \in \mathcal{T}^{\ell}$ are variants of each other if they are variants of the same unlabeled term. Similarly, we may say that two labeled steps (resp. derivations) are variants of an unlabeled step (resp. derivation), or of each other. Sometimes we write t^{ℓ} to stand for a labeled variant of an unlabeled term t, and similarly for labeled steps and labeled derivations.

Redex names $\mathcal{RN}$ are defined as follows, where α' stands for the sort of atomic labels $\mu, \nu, \xi ::= \mathtt{db}(\alpha) \mid \alpha' \bullet \alpha'$. Note that, although we often identify redex names with the labels that represent them, they should be regarded as being of different sorts.

▶ **Definition 3** (Labeled LSC). LLSC is the pair $\langle \mathcal{T}^{\ell}, \to_{\ell} \rangle$, where $\to_{\ell} \stackrel{\text{def}}{=} \to_{\ell\,\mathsf{db}} \cup \to_{\ell\,\mathsf{ls}}$, and $\to_{\ell\,\mathsf{db}} \stackrel{\text{def}}{=} \mathtt{C}\langle \mapsto_{\mathsf{db}} \rangle$ and $\to_{\ell\,\mathsf{ls}} \stackrel{\text{def}}{=} \mathtt{C}\langle \mapsto_{\mathsf{ls}} \rangle$. Relations $\mapsto_{\mathsf{db}}$ and $\mapsto_{\mathsf{ls}}$ are defined as:

$$\begin{array}{rcl} @^{\alpha}((\lambda^{\beta}x.t)\mathtt{L}, s) & \mapsto_{\mathsf{db}} & \alpha\lceil\mathtt{db}(\beta)\rceil : t[x/\lfloor\mathtt{db}(\beta)\rfloor : s]\mathtt{L} \\ \mathtt{C}\langle\!\langle x^{\alpha} \rangle\!\rangle[x/t] & \mapsto_{\mathsf{ls}} & \mathtt{C}\langle \alpha\bullet : t \rangle[x/t] \end{array}$$

The **name** of the db-step above is $\mathtt{db}(\beta)$ and that of the ls-step is $\downarrow(\alpha) \bullet \uparrow(t)$. We write $t \xrightarrow{\mu}_{\ell} s$ whenever there is a step $t \to_{\ell} s$ such that name of the contracted step is μ. An example of a reduction in LLSC follows, it shows how a $\mathtt{db}$ redex can create a $\mathtt{db}$-step:

$$\begin{array}{rl} & @^{\mathbf{a}}(@^{\mathbf{b}}(\lambda^{\mathbf{c}}x.\lambda^{\mathbf{d}}y.x^{\mathbf{e}}, z^{\mathbf{f}}), z^{\mathbf{g}}) \\ \xrightarrow{\mathbf{db(c)}}_{\ell} & @^{\mathbf{a}}((\lambda^{\mathbf{b}\lceil \mathbf{db(c)}\rceil \mathbf{d}}y.x^{\mathbf{e}})[x/z^{\lfloor \mathbf{db(c)}\rfloor \mathbf{f}}], z^{\mathbf{g}}) \\ \xrightarrow{\mathbf{db(b\lceil db(c)\rceil d)}}_{\ell} & x^{\mathbf{a}\lceil \mathbf{db(b\lceil db(c)\rceil d)}\rceil}[y/z^{\lfloor \mathbf{db(b\lceil db(c)\rceil d)}\rfloor \mathbf{g}}][x/z^{\lfloor \mathbf{db(c)}\rfloor \mathbf{f}}] \end{array}$$

The other two forms of redex creation in LSC are when a db-step creates an ls-step (*e.g.* $(\lambda x.x)y \to_{\mathsf{db}} x[x/y]$) and when an ls-step creates a db-step (*e.g.* $(xy)[x/I] \to_{\mathsf{ls}} (Iy)[x/I]$).

Structural equivalence (Sec. 2) can be lifted to labeled terms as expected. The resulting **labeled structural equivalence**, also called $\sim$, is a strong bisimulation with respect to labeled reduction. LLSC is thus well-defined over $\sim$-equivalence classes. Furthermore, given two equivalent terms $t_1 \sim t_2$ there is a bijection f between the set of steps of t_1 and the set of steps of t_2 such that $\mathsf{tgt}(\mathsf{R}) \sim \mathsf{tgt}(f(\mathsf{R}))$. The resulting system LLSC$/\sim$ will also enjoy Church-Rosser and Finite Family Developments that LLSC will be shown to enjoy in Sec. 4.

▶ Remark. Labels of the form $\mathtt{db}(\alpha)$ (not present in Lévy labeling for λ-calculus) are included for technical reasons. Consider the following example in which an ls-step creates a db-step:

$$@^{\mathbf{a}}(x^{\mathbf{b}}, t)[x/\lambda^{\mathbf{c}}y.s] \xrightarrow{\mathbf{b}\bullet\mathbf{c}}_{\ell} @^{\mathbf{a}}(\lambda^{\mathbf{b}\bullet\mathbf{c}}y.s, t)[x/\lambda^{\mathbf{c}}y.s]$$

If the name of the db-step at the right hand side was declared to be the label decorating the λ-node, namely $\mathbf{b} \bullet \mathbf{c}$, it would coincide with the name of the ls-step we have just fired.

4 Finite Family Developments

FFD relies on the following properties of LLSC:
Property 1: Labeled reduction ($\rightarrow_\ell$) is weak Church–Rosser.
Property 2: Residuals of a step have the same name: $\mathsf{S}' \in \mathsf{S}/\rho$ implies S and S' have the same name in any labeling of any LLSC derivation ρ.
Property 3: Creation implies name contribution: if R creates S then $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$, where μ denotes the name of R and ν denotes the name of S. The latter relation is called *name contribution* and is defined as the transitive closure of the following rules:

1. $\mathtt{db}(\beta) \overset{\mathsf{Name}}{\hookrightarrow} \mathtt{db}(\alpha \lceil \mathtt{db}(\beta) \rceil \gamma)$
2. $\mathtt{db}(\beta) \overset{\mathsf{Name}}{\hookrightarrow} \alpha \bullet \lfloor \mathtt{db}(\beta) \rfloor$ where α is any atomic label.
3. $\downarrow(\alpha) \bullet \uparrow(\beta) \overset{\mathsf{Name}}{\hookrightarrow} \mathtt{db}(\alpha \bullet \beta)$

We next set out to prove FFD. Its precise statement is:

▶ **Theorem 4** (FFD). *Let $\mathcal{F}$ be a finite set of redex families in* $\mathsf{Hist}(t)$ *for some term t.*
1. *(FINITE) there is no infinite family development of $\mathcal{F}$;*
2. *(COFINAL) the complete family developments of $\mathcal{F}$ all end in the same term; and*
3. *(EQUIVALENT) any two complete family developments ρ and σ of $\mathcal{F}$ satisfy $\rho \equiv \sigma$,* i.e. *they are permutation equivalent.*

(FINITE). Labeled reduction is clearly not SN since it can simulate β-reduction. However, if we restrict redex names to those that verify a *bounded predicate* [19], then we do obtain SN. A predicate on redex names $P : \mathcal{RN} \to \mathsf{Bool}$ is said to be **bounded** if the set $\{h(\mu) \mid P(\mu) \text{ holds}\}$ is bounded, where the *height $h(\mu)$ of a redex name μ* is the height of μ interpreted as a label, and the height of a label is defined as follows[6]:

$$h(\mathbf{a}) \overset{\text{def}}{=} 1 \qquad h(\alpha\beta) \overset{\text{def}}{=} \max\{h(\alpha), h(\beta)\} \qquad h(\mathbf{f}(\alpha)) \overset{\text{def}}{=} 1 + h(\alpha) \text{ if } \mathbf{f} \in \{\lceil\cdot\rceil, \lfloor\cdot\rfloor, \mathtt{db}(\cdot)\}$$

We write $\rightarrow_\ell^P$ for labeled reduction restricted to contracting steps whose names verify the predicate P. SN for $\rightarrow_\ell^P$ relies on the abstract termination result[7]: WCR $\wedge$ WN $\wedge$ Inc $\implies$ SN. WCR follows from **Property 1**, and the local confluence diagram for a pair of coinitial steps R, S is closed with their relative residuals, which have the *same name* as their ancestors (**Property 2**). WN is attained by picking, at each step, a *non-duplicating* redex R. This implies that R itself has no residual, every other $\rightarrow_\ell^P$-step $\mathsf{S} \neq \mathsf{R}$ has exactly one residual with the same name (**Property 2**) and steps created by R will have height strictly greater than that of R since creation implies name contribution (**Property 3**). Finally, Inc is rather easy given that we have a bound on P.

▶ Proposition 5 (Bounded reduction is SN). Let P be a bounded predicate. Then $\rightarrow_\ell^P$ is SN.

We may now conclude with a proof of (FINITE): it follows from Prop. 5, the fact that all names in a redex family are identical, and that we only have a finite set of families. The axiom (COFINAL) follows from confluence of LLSC, a consequence of **Property 1**, FINITE and Newman's Lemma (WCR+SN $\Rightarrow$ CR). The axiom (EQUIVALENT) follows from the fact

[6] This operation is well-defined modulo associativity of juxtaposition.
[7] Due to Klop and Nederpelt (see for instance [26, Theorem 1.2.3 (iii)]).

that LLSC enjoys *algebraic confluence*: the confluence diagram for two coinitial derivations ρ and σ can be closed by tiling it with elementary permutation diagrams. This concludes the proof of FFD.

▶ Remark. We end the section with a remark on *stability*, stated as follows. Let $\mathsf{R} \neq \mathsf{S}$ be coinitial steps and let $\mathsf{T}_1, \mathsf{T}_2, \mathsf{T}_3$ be steps such that $\mathsf{T}_3 \in \mathsf{T}_1/(\mathsf{R}/\mathsf{S})$ and $\mathsf{T}_3 \in \mathsf{T}_2/(\mathsf{S}/\mathsf{R})$. Then there exists a step T_0 such that $\mathsf{T}_1 \in \mathsf{T}_0/\mathsf{R}$ and $\mathsf{T}_2 \in \mathsf{T}_0/\mathsf{S}$. Stability is known to fail if $\to_{\mathsf{gc}}$ is added to LSC (indeed, it suffices to consider the two ways in which a gc-step can be created in a term such as $x[y/z][z/t]$). Stability for LSC is an easy consequence[8] of the fact that residuals of steps have the same name as their ancestors (**Property 2**).

5 Optimal Reduction for LSC

An optimal reduction [19, 10] computes a value, assuming it exists, in the least number of steps. More precisely, if $\mathcal{A}$ and $\mathcal{B}$ are ARSs, we say that $\mathcal{B}$ is a **sub-ARS** of $\mathcal{A}$ if (1) they have the same objects, *i.e.* $Obj(\mathcal{A}) = Obj(\mathcal{B})$, (2) all the steps of $\mathcal{B}$ are also in $\mathcal{A}$, *i.e.* $Stp(\mathcal{B}) \subseteq Stp(\mathcal{A})$, and (3) the source (resp. target) of a step in $\mathcal{B}$ coincides with its source (resp. target) in $\mathcal{A}$. A **strategy** in an ARS $\mathcal{A}$ is a sub-ARS of $\mathcal{A}$ having the same set of objects and normal forms (*cf.* [26, Def. 9.1.1]). If $\mathcal{X}$ is a set of objects, a strategy is **$\mathcal{X}$-optimal** if for any object t the length of any reduction from t to an object $s \in \mathcal{X}$ is minimal among all the possible reductions from t to s. Strategies such as call-by-name and call-by-value are not optimal: the former duplicates arguments and the latter evaluates unnecessary arguments. Call-by-need evaluates only arguments that are needed and stores their value for subsequent lookup and is indeed optimal [11]. However, all these are strategies in the ARS of closed λ-terms with *weak* reduction, in the sense that β-steps are not performed under lambdas: the set of normal forms are the abstractions. It is relatively easy to implement call-by-need in this case since it suffices to share *subterms* by labeling them [11]. Optimal reduction for the ARS of λ-terms with *strong* (*i.e.* unrestricted) reduction is more complicated since it involves reducing *under* lambdas: the set of normal forms is the usual set of β-normal forms. As a consequence, it requires sharing *contexts*, which notably complicates its implementation. Here we concentrate on a characterization of which of these steps should be shared, leaving implementation concerns, such as how to share contexts, for future work.

In the case of LSC, $\mathcal{X}$-optimality is not very interesting when $\mathcal{X}$ is the set of normal forms: since LSC has no erasing rules, all steps are trivially $\mathcal{X}$-needed. *E.g.* the db-step in $x[y/II]$ is needed to get to the normal form $x[y/I[z/I]]$. However, II may be considered junk in that it is the body of a substitution whose target variable y has no occurrence in x. Therefore, we introduce a more refined notion of *result* as a candidate for our set $\mathcal{X}$. We are only interested in steps in a term t that are not junk in the sense that they have residuals in the gc-normal form. Let $\mathsf{nf}_{\mathsf{gc}}(t)$ stand for the gc-normal form of t. Our candidate $\mathcal{X}$ is the set of **reachable normal forms**, defined as $\mathsf{RNF} \stackrel{\text{def}}{=} \{t \mid \mathsf{nf}_{\mathsf{gc}}(t) \text{ is in } \to_{\mathsf{db}\cup\mathsf{ls}}\text{-normal form}\}$. Later we shall see that it has the properties required of a set of results (*cf.* notion of stable set of objects below).

5.1 An Abstract Framework for Optimal Reduction.

An abstract framework for obtaining optimal reduction results was developed by Khasidashvili and Glauert [16]. They introduce axioms on DRS that verse over steps, residuals and redex

[8] Also a consequence of LSC being a Deterministic Family Structure (*cf.* Sec. 5 and Lem. 4.1 of [16]).

families and show that if they are satisfied, then an optimal reduction result holds. These axioms are collected in a structure called *Deterministic Family Structures (DFS)*:

$$\text{ARS (Sec. 2)} \subseteq \text{DRS (Def. 2)} \subseteq \text{DFS (Def. 6)}.$$

▶ **Definition 6.** A **Deterministic Family Structure** is a triple $\langle R, \simeq, \hookrightarrow \rangle$, where R is a DRS, $\simeq$ is an equivalence relation between coinitial redexes with history whose equivalence classes are called *families*, and $\hookrightarrow$ is a binary relation of *contribution* between coinitial families. The family of a redex with history $\rho\mathsf{R}$ is written $\mathsf{Fam}_{\simeq}(\rho\mathsf{R})$. Two families are coinitial if their representatives are coinitial. Moreover, the following axioms hold:

1. Initial. If R, S are distinct coinitial steps, then $\mathsf{Fam}_{\simeq}(\mathsf{R}) \neq \mathsf{Fam}_{\simeq}(\mathsf{S})$.
2. Copy. $\leq \subseteq \simeq$. Recall that $\leq$ is the copy relation of DRS.
3. Finite Family Developments. Any derivation that contracts redexes of a finite number of families is finite.
4. Creation. If $\rho\mathsf{R}$ is a redex with history and R creates S, then $\mathsf{Fam}_{\simeq}(\rho\mathsf{R}) \hookrightarrow \mathsf{Fam}_{\simeq}(\rho\mathsf{RS})$.
5. Contribution. Given any two coinitial families $\phi_1, \phi_2 \in \mathsf{Hist}(t)/\simeq$, the relation $\phi_1 \hookrightarrow \phi_2$ holds, if and only if, for every redex with history $\sigma\mathsf{S} \in \phi_2$, there is a redex with history $\rho\mathsf{R} \in \phi_1$ such that $\rho\mathsf{R}$ is a prefix of σ (*i.e.* $\sigma = \rho\mathsf{R}\sigma'$).

A **family reduction** in a DFS is a multistep reduction $\mathcal{M}_1 \dots \mathcal{M}_\mathsf{n}$ such that for each $i \in 1..n$ all the steps in $\mathcal{M}_\mathsf{i}$ belong to the same redex family. More precisely, for all $i \in 1..n$ and any two $\mathsf{R}, \mathsf{S} \in \mathcal{M}_\mathsf{i}$, it holds that $\mathcal{M}_1 \dots \mathcal{M}_{\mathsf{i}-1}\mathsf{R} \simeq \mathcal{M}_1 \dots \mathcal{M}_{\mathsf{i}-1}\mathsf{S}$. A family reduction is **complete** if each $\mathcal{M}_\mathsf{i}$ is a maximal set of steps that have $\mathsf{src}(\mathcal{M}_\mathsf{i})$ as source and belong to the same family. Let $\mathcal{X}$ be a set of objects. A family reduction is **$\mathcal{X}$-needed** if each $\mathcal{M}_\mathsf{i}$ contains at least one $\mathcal{X}$-needed step (*cf.* Sec. 2).

For a set $\mathcal{X}$ of objects to be admitted as a set of results it has to satisfy the following property. A set $\mathcal{X}$ of objects is **stable** if: 1) $\mathcal{X}$ is closed under parallel moves, *i.e.* for any $t \notin \mathcal{X}$, any $\rho : t \twoheadrightarrow s \in \mathcal{X}$, and any $\sigma : t \twoheadrightarrow u$ which does not contain objects in $\mathcal{X}$, the final object of ρ/σ is in $\mathcal{X}$; and 2) $\mathcal{X}$ is closed under unneeded expansion, *i.e.* for any $t \xrightarrow{\mathsf{R}} s$ such that $t \notin \mathcal{X}$ and $s \in \mathcal{X}$, the step R is $\mathcal{X}$-needed. The set of LSC-normal forms and the set of abstractions are stable. Less obvious is the fact that RNF is a stable set. This is non-trivial. For item 1) we show that the set RNF is closed under reduction, which entails that it is closed under parallel moves. For item 2) we strengthen the notion of reachable steps to that of *strongly reachable steps* (reachable steps that are minimal w.r.t. the box order introduced in [4] for the purposes of studying standardisation).

▶ **Lemma 7.** *The set of reachable-normal forms* RNF *is stable.*

The main result of this section is the following theorem. It is a corollary of Prop. 9 and of Theorem 5.2 in [16]:

▶ **Theorem 8.** *Let $\mathcal{X}$ be a stable set of terms of* LSC*. Let t be a $\mathcal{X}$-normalising term. Then any $\mathcal{X}$-needed $\mathcal{X}$-normalizing complete family-reduction $\rho : t \twoheadrightarrow t' \in \mathcal{X}$ is $\mathcal{X}$-optimal,* i.e. *it has a minimal number of family-reduction steps.*

5.2 LSC is a Deterministic Family Structure.

Given two coinitial redexes with history $\rho\mathsf{R}$, $\sigma\mathsf{S}$, the binary relations of **family equivalence** $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$ and **family contribution** $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \sigma\mathsf{S}$ are defined as follows. Consider labeled variants $\rho^\ell\mathsf{R}^\ell$ and $\sigma^\ell\mathsf{S}^\ell$ of $\rho\mathsf{R}$ and $\sigma\mathsf{S}$ respectively, starting from the same initially labeled term. Let μ be the name of R^ℓ and let ν be the name of S^ℓ. We declare $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$ to hold if and only if $\mu = \nu$ and $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \sigma\mathsf{S}$ to hold if and only if $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$. Relation $\overset{\mathsf{Fam}}{\hookrightarrow}$ is also extended to coinitial families, declaring $\phi_1 \overset{\mathsf{Fam}}{\hookrightarrow} \phi_2$ to hold whenever for any $\rho\mathsf{R} \in \phi_1$ and $\sigma\mathsf{S} \in \phi_2$ we have $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \sigma\mathsf{S}$. It is straightforward to check that this is well-defined, regardless of the choice of representatives.

▶ Proposition 9. (LSC, $\overset{\mathsf{Fam}}{\simeq}$, $\overset{\mathsf{Fam}}{\hookrightarrow}$) is a Deterministic Family Structure.

The axioms Initial, Copy, and Creation can be checked by exhaustive case analysis. The Finite family developments axiom has already been established in Thm. 4. The Contribution axiom is more demanding and relies on a non-trivial application of FFD.

6 Standardisation by Selection for LSC

We introduce an abstract notion of *uniform multi-selection strategy*, show that repeated application of this strategy terminates using FFD in any DFS, and finally that two permutation equivalent derivations produce the same multiderivation. Then we instantiate our abstract result to LSC, obtaining an algorithm for standardizing LSC derivations by picking multisteps according to a given parametric partial order on its steps.

Uniform Multi-Selection Strategies. A step R **belongs** to a derivation ρ, written $\mathsf{R} \lhd \rho$, if and only if $\rho = \rho_1\mathsf{R}'\rho_2$ and $\mathsf{R}' \in \mathsf{R}/\rho_1$. Given a DRS $\mathcal{A}$, we write Stp^+ for the set of multisteps, *i.e.* non-empty finite sets of coinitial steps, and we let D, E, etc. range over multiderivations, *i.e.* derivations in the DRS whose steps are multisteps. A multistep $\mathcal{M}$ belongs to a derivation ρ, written $\mathcal{M} \lhd \rho$, if and only if $\mathsf{R} \lhd \rho$ for all $\mathsf{R} \in \mathcal{M}$. If $\mathsf{D} = \mathcal{M}_1 \ldots \mathcal{M}_\mathsf{n}$ is a multiderivation, we say that a derivation ρ is a complete development of D if $\rho = \rho_1 \ldots \rho_\mathsf{n}$, where each ρ_i is a complete development of the multistep $\mathcal{M}_\mathsf{i}$. By FD a complete development always exists and any two complete developments are permutation equivalent. We write $\partial\mathsf{D}$ to stand for some complete development of D, and ρ/D for $\rho/\partial\mathsf{D}$. A **multi-selection strategy** is a function $\mathbb{M}$ that maps every non-empty derivation ρ to a coinitial multistep $\mathcal{M} \in \mathit{Stp}^+$ such that $\mathcal{M} \lhd \rho$ and $\mathcal{M}/\rho = \emptyset$, *i.e.* residuals of every step appear somewhere in the sequence, and there are no residuals of any step left after the sequence. It is, moreover, **uniform** if $\rho \equiv \sigma$ implies $\mathbb{M}(\rho) = \mathbb{M}(\sigma)$ for any non-empty ρ, σ. *E.g.* $\mathbb{M}_{\mathrm{Triv}}(\mathsf{R}\rho) \overset{\mathrm{def}}{=} \{\mathsf{R}\}$ is a (trivial) multi-selection strategy, which is not uniform.

The **multiderivation induced by a multi-selection strategy** $\mathbb{M}$ **on a derivation** ρ, written $\mathbb{M}^\star(\rho)$, is a sequence of multisteps defined as follows:

$$\mathbb{M}^\star(\epsilon) = \epsilon \qquad\qquad \mathbb{M}^\star(\rho) = \mathbb{M}(\rho)\,\mathbb{M}^\star(\rho/\mathbb{M}(\rho)) \quad \text{if } \rho \neq \epsilon$$

Successively applying a multi-selection strategy $\mathbb{M}$ to build a reduction sequence $\mathbb{M}^\star(\rho)$ terminates, as long as the input ρ is finite, *i.e.* recursion is well-founded. This relies on FFD.

▶ **Lemma 10** (Induced multiderivations preserve finiteness). *Suppose that $\mathbb{M}$ is a multi-selection strategy in a DFS. If ρ is finite, then $\mathbb{M}^\star(\rho)$ is also finite.*

By definition, a uniform multi-selection strategy $\mathbb{M}$, when given two permutation equivalent derivations, always selects the same multistep. It, in fact, yields the same multiderivation.

▶ **Lemma 11.** *Let $\mathbb{M}$ be a uniform multi-selection strategy in a DFS, and let ρ, σ be finite derivations. If $\rho \equiv \sigma$ then $\mathbb{M}^\star(\rho) = \mathbb{M}^\star(\sigma)$.*

▶ **Lemma 12.** *Let $\mathbb{M}$ be a multi-selection strategy in a DFS, and ρ a finite derivation. Then $\rho \equiv \partial\mathbb{M}^\star(\rho)$.*

A multiderivation D is said to be $\mathbb{M}$-**compliant** if and only if $\mathbb{M}^\star(\partial\mathsf{D}) = \mathsf{D}$.

▶ Proposition 13. Let $\mathbb{M}$ be a uniform multi-selection strategy in a DFS. For any finite derivation ρ there exists a unique multiderivation D such that $\rho \equiv \partial\mathsf{D}$ and D is $\mathbb{M}$-compliant. Namely, $\mathsf{D} = \mathbb{M}^\star(\rho)$.

Standardisation Algorithm for LSC. For each term t let $\mathsf{Out}(t)$ be the set of steps whose source is t in LSC, and let $<_t$ be an arbitrary strict *partial order* on $\mathsf{Out}(t)$. The **arbitrary selector** $\mathbb{M}_<$ is defined as follows: $\mathbb{M}_<(\rho) \stackrel{\text{def}}{=} \{\mathsf{R} \mid \mathsf{R}/\rho = \emptyset \text{ and } \mathsf{R} \text{ is minimal}\}$. By minimal we mean that there is no step R' such that $\mathsf{R}'/\rho = \emptyset$ and $\mathsf{R}' <_{\mathsf{src}(\rho)} \mathsf{R}$. Note that $\mathbb{M}_<$ is a non-empty finite set, given that the set $\{\mathsf{R} \mid \mathsf{R}/\rho = \emptyset\}$ is non-empty and finite, so it has at least one minimal element.

▶ **Lemma 14.** *$\mathbb{M}_<$ is a uniform multi-selection strategy.*

▶ **Corollary 15** (Standardisation by arbitrary selection for LSC)**.** *For each finite sequence ρ in* LSC*, there is a unique multiderivation D such that $\rho \equiv \partial\mathsf{D}$ and D is $\mathbb{M}_<$-compliant. Moreover, if the order $<_t$ is computable, then D is computable from ρ, namely $\mathsf{D} = \mathbb{M}^\star_<(\rho)$.*

For example, let $\rho : x[x/t] \to x[x/t'] \to t'[x/t'] \to t''[x/t']$, where $t \to t' \to t''$.

1. If $<^1$ is the trivial partial order in which every step is incomparable, *i.e.* $\mathsf{R} <^1_t \mathsf{S}$ never holds, then $\mathbb{M}^\star_{<^1}(\rho) : x[x/t] \multimap\!\!\!\rightarrow t'[x/t'] \to t''[x/t']$. The first step is a proper multistep.
2. Let $<^2$ be the total left-to-right order, defined so that $\mathsf{R} <^2_t \mathsf{S}$ holds whenever R is to the left of S. Then $\mathbb{M}^\star_{<^2}(\rho) : x[x/t] \to t[x/t] \to t'[x/t] \to t'[x/t'] \to t''[x/t']$.
3. If $<^3$ is the total right-to-left order, defined so that $\mathsf{R} <^3_t \mathsf{S}$ holds if R is to the right of S. Then $\mathbb{M}^\star_{<^3}(\rho) = \rho : x[x/t] \to x[x/t'] \to t'[x/t'] \to t''[x/t']$.

7 Normalisation of the Linear Needed Strategy in LSC

Recall that a **strategy** in an ARS is a sub-ARS having the same objects and normal forms. We write $\mathsf{NF}(\mathcal{A})$ for the set of normal forms of an ARS $\mathcal{A}$, and $t \to_{\mathcal{A}} s$ to emphasize that a given step is in $\mathcal{A}$. If $\mathcal{X}$ is a superset of the normal forms of $\mathcal{A}$, a strategy $\mathbb{S}$ is said to be $\mathcal{X}$-**normalizing** if for every object t such that there exists a reduction $t \twoheadrightarrow_{\mathcal{A}} s \in \mathcal{X}$, every maximal reduction from t in the strategy $\mathbb{S}$ contains an object in $\mathcal{X}$. A sub-ARS $\mathcal{B}$ is **residual-invariant** if for any steps R and S such that $\mathsf{R} \in \mathcal{B}$ and $\mathsf{S} \neq \mathsf{R}$, there exists a step $\mathsf{R}' \in \mathbb{S}$ such that $\mathsf{R}' \in \mathsf{R}/\mathsf{S}$. A sub-ARS $\mathcal{B}$ is **closed** if the set $\mathsf{NF}(\mathcal{B})$ is closed by reduction, *i.e.* $t \to_{\mathcal{A}} s$ and $t \in \mathsf{NF}(\mathcal{B})$ imply $s \in \mathsf{NF}(\mathcal{B})$. Observe that any sub-ARS $\mathcal{B}$ can be extended to a strategy $\mathbb{S}_{\mathcal{B}}$ by adjoining the steps going out from normal forms, *i.e.* by setting $\mathit{Stp}(\mathbb{S}_{\mathcal{B}}) := \mathit{Stp}(\mathcal{B}) \cup \{\mathsf{R} \in \mathit{Stp}(\mathcal{A}) \mid \mathsf{src}(\mathsf{R}) \in \mathsf{NF}(\mathcal{B})\}$. We will instantiate the following normalisation result to the linear call-by-need strategy of LSC which we define below.

▶ Proposition 16. Let $\mathcal{B}$ be a closed residual-invariant sub-ARS in a DFS. Then the corresponding strategy $\mathbb{S}_{\mathcal{B}}$ is $\mathsf{NF}(\mathcal{B})$-normalizing.

Needed linear reduction for LSC is the sub-ARS $\mathbb{NL}$ of LSC defined as follows. Need contexts are defined by the grammar $\mathtt{N} ::= \Box \mid \mathtt{N}\,t \mid \mathtt{N}[x/t] \mid \mathtt{N}\langle\!\langle x\rangle\!\rangle[x/\mathtt{N}]$. The reduction rule $\to_{\mathbb{NL}}$ is the union of the usual db rule, and the lsnl rule $\mathtt{N}\langle\!\langle x\rangle\!\rangle[x/\mathtt{vL}] \mapsto_{\mathsf{lsnl}} \mathtt{N}\langle\mathtt{vL}\rangle[x/\mathtt{vL}]$, both closed by need contexts, where $\mathtt{v}$ stands for a *value*, *i.e.* a term of the form $\lambda y.t$. Note that it is in fact a sub-ARS for LSC, *i.e.* the lsnl rule is a particular case of the ls rule, and closure by need contexts is a particular case of closure by general contexts. A similar, albeit slightly different call-by-need calculus based on LSC has been studied in [2] to relate the execution model of abstract machines with reduction in calculi with ES. In [18] it is shown, via intersection types, that it is a sound and complete implementation of call-by-name.

The set of **needed linear normal forms** NLNF is defined by the grammar $A ::= (\lambda x.t)\mathtt{L} \mid \mathtt{N}\langle\langle x\rangle\rangle$. Terms of the form $(\lambda x.t)\mathtt{L}$ are called **answers**, and $\mathtt{N}\langle\langle x\rangle\rangle$ are called **structures**. In structures, $\mathtt{N}$ does not bind x, the latter called its **needed variable**.

▶ **Corollary 17.** *The strategy $\mathbb{S}_{\mathbb{NL}}$ associated to the sub-ARS $\mathbb{NL}$ is NLNF-normalizing.*

The proof consists in first showing that $\mathsf{NF}(\mathbb{NL})$ coincides with the set NLNF, and then that the sub-ARS $\mathbb{NL}$ is closed residual-invariant. These items rely on a number of lemmata such as the fact that the needed variable in a structure is unique and that answers cannot be written as of the form $\mathtt{N}\langle\Delta\rangle$ where Δ is a redex or a variable not bound by $\mathtt{N}$.

8 Conclusions

The *Linear Substitution Calculus* sits between calculi with ES and the λ-calculus: it has ES but admits a theory of residuals. We devise a theory of optimal reduction for LSC. We start from the theory of residuals developed in [4] and use it to prove a Finite Family Developments result. This is achieved by introducing a Lévy labeling and associated notion of redex family which supports the two distinctive features of LSC, namely its use of *context rules* that allow substitutions to act "at a distance" and also the set of equations modulo which it rewrites which allow substitutions to "float" in a term. We then apply FFD to prove a number of novel results for LSC including: an optimal reduction result, an algorithm for standardisation by selection, and normalization of a linear call-by-need reduction strategy.

Perhaps the most relevant future work is devising an appropriate notion of extraction and showing that all three characterizations (labeling, zig-zag and extraction) of redex family coincide. This is non-trivial and has elided us for some time. Also, there is the topic of graph based implementations, labels and virtual redexes (*cf.* notion of paths in Ch.6 of [10]).

Ackn. To Thibaut Balabonski and Beniamino Accattoli for helpful discussions.

References

1 Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *J. Funct. Program.*, 1(4):375–416, 1991.

2 Beniamino Accattoli, Pablo Barenbaum, and Damiano Mazza. Distilling abstract machines. In J. Jeuring and M. Chakravarty, editors, *Proceedings of ICFP*, pages 363–376. ACM, 2014.

3 Beniamino Accattoli, Pablo Barenbaum, and Damiano Mazza. A strong distillery. In Xinyu Feng and Sungwoo Park, editors, *APLAS 2015, Pohang, South Korea, November 30 - December 2, 2015, Proceedings*, volume 9458 of *LNCS*, pages 231–250. Springer, 2015.

4 Beniamino Accattoli, Eduardo Bonelli, Delia Kesner, and Carlos Lombardi. A nonstandard standardization theorem. In Suresh Jagannathan and Peter Sewell, editors, *POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 659–670. ACM, 2014.

5 Beniamino Accattoli and Claudio Sacerdoti Coen. On the relative usefulness of fireballs. In *LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 141–155. IEEE Computer Society, 2015.

6 Beniamino Accattoli and Delia Kesner. The structural *lambda*-calculus. In Anuj Dawar and Helmut Veith, editors, *CSL 2010*, volume 6247 of *LNCS*, pages 381–395. Springer, 2010.

7 Beniamino Accattoli and Ugo Dal Lago. Beta reduction is invariant, indeed. In Thomas A. Henzinger and Dale Miller, editors, *CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 8:1–8:10. ACM, 2014.

8 Beniamino Accattoli and Ugo Dal Lago. (leftmost-outermost) beta reduction is invariant, indeed. *Logical Methods in Computer Science*, 12(1), 2016.

9 Beniamino Accattoli and Luca Paolini. Call-by-value solvability, revisited. In Tom Schrijvers and Peter Thiemann, editors, *FLOPS 2012, Kobe, Japan, May 23-25, 2012. Proceedings*, volume 7294 of *LNCS*, pages 4–16. Springer, 2012.

10 Andrea Asperti and Stefano Guerrini. *The Optimal Implementation of Functional Programming Languages*. Cambridge Tracts in Theoretical Computer Science. CUP, 1999.

11 Thibaut Balabonski. Weak optimality, and the meaning of sharing. In Greg Morrisett and Tarmo Uustalu, editors, *ICFP'13, Boston, MA, USA - September 25 - 27, 2013*, pages 263–274. ACM, 2013.

12 Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103. Elsevier, 1984.

13 Zine-El-Abidine Benaissa, Pierre Lescanne, and Kristoffer Høgsbro Rose. Modeling sharing and recursion for weak reduction strategies using explicit substitution. In Herbert Kuchen and S. Doaitse Swierstra, editors, *PLILP'96, Aachen, Germany, September 24-27, 1996, Proceedings*, volume 1140 of *LNCS*, pages 393–407. Springer, 1996.

14 Gérard Boudol. Computational semantics of term rewriting systems. In M. Nivat and J.C. Reynolds, editors, *Algebraic Methods in Semantics*, page 169–236. CUP, 1985.

15 H. J. Sander Bruggink. A proof of finite family developments for higher-order rewriting using a prefix property. In Frank Pfenning, editor, *RTA 2006*, volume 4098 of *LNCS*, pages 372–386. Springer, 2006.

16 John R. W. Glauert and Zurab Khasidashvili. Relative normalization in deterministic residual structures. In Hélène Kirchner, editor, *CAAP'96, Linköping, Sweden, April, 22-24, 1996, Proceedings*, volume 1059 of *LNCS*, pages 180–195. Springer, 1996.

17 Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In Ravi Sethi, editor, *POPL'92, Albuquerque, New Mexico, USA, January 19-22, 1992*, pages 15–26. ACM Press, 1992.

18 Delia Kesner. Reasoning about call-by-need by means of types. In *FoSSaCS 2016*, pages 424–441. Springer-Verlag, 2016.

19 Jean-Jacques Lévy. *Réductions correctes et optimales dans le lambda-calcul*. PhD thesis, Université de Paris 7, 1978.

20 Jean-Jacques Lévy. Generalized finite developments. In *Essays in Honour of Gilles Kahn*. CUP, 2007.

21 Luc Maranget. Optimal derivations in weak lambda-calculi and in orthogonal terms rewriting systems. In David S. Wise, editor, *POPL '91, Orlando, Florida, USA, January 21-23, 1991*, pages 255–269. ACM Press, 1991.

22 Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture*. PhD thesis, Université Paris 7, december 1996.

23 Paul-André Melliès. Axiomatic rewriting theory II: the $\lambda\sigma$-calculus enjoys finite normalisation cones. *J. Log. Comput.*, 10(3):461–487, 2000.

24 Paul-André Melliès. Axiomatic rewriting theory VI residual theory revisited. In Sophie Tison, editor, *RTA 2002, Copenhagen, Denmark, July 22-24, 2002, Proceedings*, volume 2378 of *LNCS*, pages 24–50. Springer, 2002.

25 Michael J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *LNCS*. Springer, 1977.

26 Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. CUP, 2003.

27 Vincent van Oostrom. Finite family developments. In Hubert Comon, editor, *RTA-97, Sitges, Spain, June 2-5, 1997, Proceedings*, volume 1232 of *LNCS*, pages 308–322. Springer, 1997.

28 Marijn Zwitserlood Vincent van Oostrom, Kees-Jan van de Looij.] (lambdascope). Workshop on Algebra and Logic on Programming Systems (ALPS), April 2004.

A Appendix

The appendix is organized as follows:

- Sec. A.1 corresponds to Sec. 3 in the main body. It includes representative examples of redex creation in the labeled LSC (Sec. A.1.1), and the proof that the labeled LSC is well-defined modulo structural equivalence (Sec. A.1.2).
- Sec. A.2 corresponds to Sec. 4 in the main body. It proves various properties of the labeled LSC, the Finite Family Developments theorem (FFD), and strong versions of confluence.
  More precisely, this section includes the proof of a strong version of the WCR property for the labeled calculus (Sec. A.2.1), the proof that redex creation implies name contribution (Sec. A.2.2), the proof that residuals may be characterized using the labeled calculus (Sec. A.2.3), the proof that reduction in the labeled calculus is WN (Sec .A.2.4) and SN (Sec .A.2.5) when restricted to bounded families (which in particular implies FFD), and a strong version of confluence, namely *algebraic confluence* (Sec. A.2.6).
- Sec. A.3 corresponds to Sec. 5 in the main body, and it contains the proof of the optimality result for the LSC. It includes the proof that LSC forms a Deterministic Family Structure (Sec. A.3.1), and that reachable normal forms are a stable set (Sec. A.3.2). These entail optimality, as has been discussed in Sec. 5.
- Sec. A.4 corresponds to Sec. 6 in the main body. It includes a proof of an abstract result of standardisation in DFSs based on *multiselection strategies* (Sec. A.5). Then it includes an application of this result for the LSC (Sec. A.6).
- Sec. A.7 corresponds to Sec. 7 in the main body. It includes a proof of an abstract result of normalisation in DFSs (Sec. A.7.1). Then it includes an application of this result for the *linear call-by-need strategy* (Sec. A.7.2).
- Finally, Sec. A.8 describes an extraction procedure which we conjecture to be correct and complete.

The appendix includes the outlines and interesting cases of most of the proofs, but some auxiliary lemmas and details have been omitted in order to keep it reasonably succint. For a detailed version of all the material please refer to the technical report.

A.1 The Labeled LSC

A.1.1 Redex Creation in LSC

Some examples of how redex creation in LSC is reflected as name contribution in the labeled calculus.

▶ **Example 18.** The following are representative examples of the three redex creation cases in LSC.

A db redex creates a db redex:

$$\begin{array}{rl} & @^{\mathbf{a}}(@^{\mathbf{b}}(\lambda^{\mathbf{c}}x.\lambda^{\mathbf{d}}y.x^{\mathbf{e}}, z^{\mathbf{f}}), z^{\mathbf{g}}) \\ \xrightarrow{\mathtt{db}(\mathbf{c})}_{\ell} & @^{\mathbf{a}}((\lambda^{\mathbf{b}\lceil\mathtt{db}(\mathbf{c})\rceil\mathbf{d}}y.x^{\mathbf{e}})[x/z^{\lfloor\mathtt{db}(\mathbf{b}\lceil\mathtt{db}(\mathbf{c})\rceil\mathbf{d})\rfloor\mathbf{f}}], z^{\mathbf{g}}) \\ \xrightarrow{\mathtt{db}(\mathbf{b}\lceil\mathtt{db}(\mathbf{c})\rceil\mathbf{d})}_{\ell} & x^{\mathbf{a}\lceil\mathtt{db}(\mathbf{b}\lceil\mathtt{db}(\mathbf{c})\rceil\mathbf{d})\rceil}[y/z^{\lfloor\mathtt{db}(\mathbf{b}\lceil\mathtt{db}(\mathbf{c})\rceil\mathbf{d})\rfloor\mathbf{g}}][x/z^{\lfloor\mathtt{db}(\mathbf{b}\lceil\mathtt{db}(\mathbf{c})\rceil\mathbf{d})\rfloor\mathbf{f}}] \end{array}$$

A db redex creates an ls redex:

$$\begin{array}{rl} & @^{\mathbf{a}}(\lambda^{\mathbf{b}}x.x^{\mathbf{c}}, y^{\mathbf{d}}) \\ \xrightarrow{\mathtt{db}(\mathbf{b})}_{\ell} & x^{\mathbf{a}\lceil\mathtt{db}(\mathbf{b})\rceil\mathbf{c}}[x/y^{\lfloor\mathtt{db}(\mathbf{b})\rfloor\mathbf{d}}] \\ \xrightarrow{\mathbf{c}\bullet\lfloor\mathtt{db}(\mathbf{b})\rfloor}_{\ell} & y^{\mathbf{a}\lceil\mathtt{db}(\mathbf{b})\rceil\mathbf{c}\bullet\lfloor\mathtt{db}(\mathbf{b})\rfloor\mathbf{d}}[x/y^{\lfloor\mathtt{db}(\mathbf{b})\rfloor\mathbf{d}}] \end{array}$$

Two ls redexes contribute towards the creation of a db redex:

$$\begin{array}{rl} & @^{\mathbf{a}}(x^{\mathbf{b}}, t)[x/y^{\mathbf{d}}][y/\lambda^{\mathbf{e}}z.z^{\mathbf{f}}] \\ \xrightarrow{\mathbf{b}\bullet\mathbf{d}}_{\ell} & @^{\mathbf{a}}(y^{\mathbf{b}\bullet\mathbf{d}}, t)[x/y^{\mathbf{d}}][y/\lambda^{\mathbf{e}}z.z^{\mathbf{f}}] \\ \xrightarrow{\mathbf{d}\bullet\mathbf{e}}_{\ell} & @^{\mathbf{a}}(\lambda^{\mathbf{b}\bullet\mathbf{d}\bullet\mathbf{e}}z.z^{\mathbf{f}}, t)[x/y^{\mathbf{d}}][y/\lambda^{\mathbf{e}}z.z^{\mathbf{f}}] \\ \xrightarrow{\mathsf{db}(\mathbf{b}\bullet\mathbf{d}\bullet\mathbf{e})}_{\ell} & z^{\mathbf{a}\lceil\mathsf{db}(\mathbf{b}\bullet\mathbf{d}\bullet\mathbf{e})\rceil\mathbf{f}}[z/\lfloor\mathsf{db}(\mathbf{b}\bullet\mathbf{d}\bullet\mathbf{e})\rfloor : t][x/y^{\mathbf{d}}][y/\lambda^{\mathbf{e}}z.z^{\mathbf{f}}] \end{array}$$

Note that, in each case, the name of the created redex contains the names of the redexes that have contributed towards its creation.

A.1.2 Structural Equivalence and LLSC

This section shows that LLSC is well-defined with respect to the structural equivalence $\sim$.

▶ **Definition 19** (Structural equivalence)**.** Structural equivalence for LLSC is defined as the reflexive, symmetric, transitive, and contextual closure of the following axioms:

$$\begin{array}{rcll} (\lambda^{\alpha}x.t)[y/s] & \sim_{\lambda} & \lambda^{\alpha}x.t[y/s] & \text{if } x \neq y \text{ and } x \notin \mathsf{fv}(s) \\ @^{\alpha}(t, s)[x/u] & \sim_{@} & @^{\alpha}(t[x/u], s) & \text{if } x \notin \mathsf{fv}(s) \\ t[x/s][y/u] & \sim_{com} & t[y/u][x/s] & \text{if } x \neq y,\ x \notin \mathsf{fv}(u), \text{ and } y \notin \mathsf{fv}(s) \end{array}$$

Moreover, we write $\Leftrightarrow$ to stand for the symmetric and contextual closure of the axioms $\sim_{\lambda} \cup \sim_{@} \cup \sim_{com}$. Note that $\sim$ is the reflexive and transitive closure of $\Leftrightarrow$.

▶ **Definition 20.** A context $\mathtt{C}$ is said to be a x^{α}-context of t if and only if $t = \mathtt{C}\langle\!\langle x^{\alpha}\rangle\!\rangle$.

▶ **Lemma 21** (Properties of the structural equivalence)**.** *Let $t_1 \sim t_2$. Then the following properties hold:*

1. **Adding a label.** $\alpha : t_1 \sim \alpha : t_2$.
2. **First label.** $\uparrow(t_1) = \uparrow(t_2)$.
3. **Free variables.** $\mathsf{fv}(t_1) = \mathsf{fv}(t_2)$.
4. **Correspondence of variables.** *If x is a variable and α a label, the x^{α}-contexts of t_1 and the x^{α}-contexts of t_2 are in 1–1 correspondence.*

Proof. All are straightforward by induction on the derivation that $t_1 \sim t_2$. ◀

▶ Proposition 22 (Structural equivalence is a strong bisimulation). The relation $\sim$ is a strong bisimulation with respect to the labelled reduction relation $\to_{\ell}$. Furthermore, given two equivalent terms $t_1 \sim t_2$ there is a bijection $f : \mathsf{Red}(t_1) \to \mathsf{Red}(t_2)$ between the set of redexes of t_1 and the set of redexes of t_2 such that $\mathsf{tgt}(\mathsf{R}) \sim \mathsf{tgt}(f(\mathsf{R}))$, and such that the name of the step R coincides with the name of the step $f(\mathsf{R})$.

Proof. If $t_1 \sim t_2$ we have that $t_1 \Leftrightarrow \ldots \Leftrightarrow t_2$ with n steps of $\Leftrightarrow$. By induction on n, it suffices to show that the property holds for the case $n = 1$ *i.e.* when $t_1 \Leftrightarrow t_2$.

Let $t_1 \Leftrightarrow t_2$ and let us construct a bijection f as in the statement. Let $\mathsf{R} \in \mathsf{Red}(t_1)$ be a redex of t_1 Let us construct a redex $f(\mathsf{R}) \in \mathsf{Red}(t_2)$. More precisely, we will construct diagrams of the form:

$$\begin{tikzcd} t_1 \arrow[r, "\mathsf{R}"] \arrow[d, dash, "\Leftrightarrow"'] & t_2 \arrow[d, dotted, dash, "\sim"] \\ t_1' \arrow[r, dotted, "f(\mathsf{R})"'] & t_2' \end{tikzcd}$$

We proceed by induction on the context $\mathtt{C}$ on which the step contracting R takes place and then by case analysis on the position where the structural equation is applied. There are many uninteresting overlappings such as when the source is a $\mathtt{db}$ step $@^{\alpha}((\lambda x.t)\mathtt{L}, s) \to \alpha\lceil\mathtt{db}(\beta)\rceil : t[x/\lfloor\mathtt{db}(\beta)\rfloor : s]\mathtt{L}$ and the structural equation is applied inside t, or inside s, or inside $\mathtt{L}$. These cases are straightforward, resorting to the properties listed in Lem. 21 when necessary. Following we deal with the interesting overlappings:

- **db vs. $\sim_{\lambda}$:**

$$\begin{array}{ccc} @^{\alpha}((\lambda^{\beta}x.t)[y/u]\mathtt{L}', s) & \xrightarrow{\mathtt{db}(\beta)} & \alpha\lceil\mathtt{db}(\beta)\rceil : t[x/\lfloor\mathtt{db}(\beta)\rfloor : s][y/u]\mathtt{L}' \\ \Leftrightarrow | & & \vdots \sim_{com} \\ @^{\alpha}((\lambda^{\beta}x.t[y/u])\mathtt{L}', s) & \xrightarrow[\mathtt{db}(\beta)]{} & \alpha\lceil\mathtt{db}(\beta)\rceil : t[y/u][x/\lfloor\mathtt{db}(\beta)\rfloor : s]\mathtt{L}' \end{array}$$

- **db vs. $\sim_{@}$:**

$$\begin{array}{ccc} @^{\alpha}((\lambda^{\beta}x.t)\mathtt{L}'[y/u], s) & \xrightarrow{\mathtt{db}(\beta)} & \alpha\lceil\mathtt{db}(\beta)\rceil : t[x/\lfloor\mathtt{db}(\beta)\rfloor : s]\mathtt{L}'[y/u] \\ \Leftrightarrow | & & \vdots = \\ @^{\alpha}((\lambda^{\beta}x.t)\mathtt{L}', s)[y/u] & \xrightarrow[\mathtt{db}(\beta)]{} & \alpha\lceil\mathtt{db}(\beta)\rceil : t[x/\lfloor\mathtt{db}(\beta)\rfloor : s]\mathtt{L}'[y/u] \end{array}$$

- **ls vs. structural rule on the body:** Let $\mathtt{C}_1\langle\!\langle x^{\alpha}\rangle\!\rangle[x/s]$ be the source of an ls step and suppose that $\mathtt{C}_1\langle\!\langle x^{\alpha}\rangle\!\rangle \Leftrightarrow t'$. By Lem. 21, $\sim$ establishes a 1–1 correspondence between x^{α}-contexts, so t' is of the form $t' = \mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle$. Then:

$$\begin{array}{ccc} \mathtt{C}_1\langle\!\langle x^{\alpha}\rangle\!\rangle[x/s] & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(s)} & \mathtt{C}_1\langle\alpha : s\rangle[x/s] \\ \Leftrightarrow | & & \vdots \sim \\ \mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle[x/s] & \xrightarrow[\downarrow(\alpha)\bullet\uparrow(s)]{} & \mathtt{C}_2\langle\alpha : s\rangle[x/s] \end{array}$$

- **ls vs. $\sim_{\lambda}$:**

$$\begin{array}{ccc} (\lambda^{\beta}y.\mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle)[x/s] & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(s)} & (\lambda^{\beta}y.\mathtt{C}_2\langle\alpha : s\rangle)[x/s] \\ \sim_{\lambda} | & & \vdots \sim_{\lambda} \\ \lambda^{\beta}y.\mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle[x/s] & \xrightarrow[\downarrow(\alpha)\bullet\uparrow(s)]{} & \lambda^{\beta}y.\mathtt{C}_2\langle\alpha : s\rangle[x/s] \end{array}$$

- **ls vs. $\sim_{@}$:**

$$\begin{array}{ccc} @^{\beta}(\mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle, u)[x/s] & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(s)} & @^{\beta}(\mathtt{C}_2\langle\alpha : s\rangle, u)[x/s] \\ \sim_{@} | & & \vdots \sim_{@} \\ @^{\beta}(\mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle[x/s], u) & \xrightarrow[\downarrow(\alpha)\bullet\uparrow(s)]{} & @^{\beta}(\mathtt{C}_2\langle\alpha : s\rangle[x/s], u) \end{array}$$

- **ls vs. $\sim_{com}$:**

$$\begin{array}{ccc} \mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle[y/u][x/s] & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(s)} & \mathtt{C}_2\langle\alpha : s\rangle[y/u][x/s] \\ \sim_{com} | & & \vdots \sim_{com} \\ \mathtt{C}_2\langle\!\langle x^{\alpha}\rangle\!\rangle[x/s][y/u] & \xrightarrow[\downarrow(\alpha)\bullet\uparrow(s)]{} & \mathtt{C}_2\langle\alpha : s\rangle[x/s][y/u] \end{array}$$

◀

A.2 Finite Family Developments

We address the proof of the main properties required for FFD: (1) Labeled permutation (Prop. 24); (2) Creation implies contribution (Prop. 26); (3) Residuals of a redex have the same name (Prop. 27); (4) Bounded reduction is SN (Prop. 44).

To prove (4), we first show that bounded reduction is WN (Prop. 41). As an easy corollary of (1) and (4) we obtain that reduction in LLSC is CR (Coro. 45). Moreover, we discuss a stronger result: *algebraic confluence* (Prop. 46).

A.2.1 Labeled permutation

We prove a strong version of WCR: every local peak may be closed, and each residual has the same redex name as its ancestor in the labeled calculus.

▶ **Lemma 23** (Properties of labels, contexts, and reduction). *The following properties hold:*

1. **Adding labels I.** *If* $\mathtt{C}$ *is a substitution context,* $\alpha : \mathtt{C}\langle t\rangle = \mathtt{C}\langle \alpha : t\rangle$.
2. **Adding labels II.** *If* $\mathtt{C}$ *is not a substitution context,* $\alpha : \mathtt{C}\langle t\rangle = (\alpha : \mathtt{C})\langle t\rangle$.
3. **Adding labels III.** $\alpha : \mathtt{C}\langle x^{\beta}\rangle = \mathtt{C}\langle x^{\beta'}\rangle$ *where* $\downarrow(\beta) = \downarrow(\beta')$.
4. **Adding labels is functorial.** *If* $t \xrightarrow{\mu}_{\ell} s$ *then* $\alpha : t \xrightarrow{\mu}_{\ell} \alpha : s$.
5. **First label I.** *If* $t \to_{\ell} s$ *then* $\uparrow(t) = \uparrow(s)$.
6. **First label II.** $\uparrow(\alpha : t) = \uparrow(\alpha)$.
7. **First label III.** $\uparrow(\mathtt{C}\langle x^{\alpha}\rangle) = \uparrow(\alpha\bullet : t)$.

Proof. The proofs are straightforward by induction; 1–3 are by induction on $\mathtt{C}$; 4 and 5 are by induction on the context under which the step takes place; 6 and 7 are by induction on the term. ◀

▶ Proposition 24 (Labeled permutation). Let R, S be coinitial redexes in LSC, and let $\mathsf{R}^{\ell} : t \xrightarrow{\mu}_{\ell} s$ and $\mathsf{S}^{\ell} : t \xrightarrow{\nu}_{\ell} u$ be coinitial labeled variants in LLSC. Then the local peak can be closed with their relative residuals, that is, if ρ is a complete development of R/S and σ is a complete development of S/R then there exists a labeled term r such that $\sigma^{\ell} : s \xrightarrow{\nu}\!\!\!\twoheadrightarrow_{\ell} r$ and $\rho^{\ell} : u \xrightarrow{\mu}\!\!\!\twoheadrightarrow_{\ell} r$ are labeled variants of σ and ρ respectively. Moreover, if $\mathsf{R} \neq \mathsf{S}$ then closing the diagram requires at least one step on each side.

Proof. We check all the critical pairs. If R and S lie at disjoint positions, it is straightforward to close the diagram. So without loss of generality we may suppose that the position of R is a prefix of the position of the redex occurrence S.

The proof goes by induction on the context $\mathtt{C}$ under which the redex occurrence R is contracted. There are many uninteresting overlappings which can be closed by *i.h.* or simply by closing the diagram if the steps are orthogonal, resorting to Lem. 23 whenever needed. Following we deal with the interesting overlappings:

- **db vs. ls in the body:** the body of the abstraction contracted by the $\mathtt{db}$ step is of the form $\mathtt{C}_2\langle y^{\gamma}\rangle$ and y is bound by some substitution to a term u, which might be in $\mathtt{C}_1$, in $\mathtt{C}_2$, or in $\mathtt{L}$. Let $u' := \gamma\bullet : u$. Then:

$$\begin{array}{ccc}
\mathtt{C}_1\langle @^{\alpha}((\lambda^{\beta}x.\mathtt{C}_2\langle y^{\gamma}\rangle)\mathtt{L}, s)\rangle & \xrightarrow{\mathtt{db}(\beta)} & \mathtt{C}_1\langle \alpha\lceil \mathtt{db}(\beta)\rceil : \mathtt{C}_2\langle y^{\gamma}\rangle[x/\lfloor \mathtt{db}(\beta)\rfloor : s]\mathtt{L}\rangle \\
\downarrow(\gamma)\bullet\uparrow(u)\downarrow & & \downarrow(\gamma)\bullet\uparrow(u)\downarrow \\
\mathtt{C}_1\langle @^{\alpha}((\lambda^{\beta}x.\mathtt{C}_2\langle u'\rangle)\mathtt{L}, s)\rangle & \xrightarrow{\mathtt{db}(\beta)} & \mathtt{C}_1\langle \alpha\lceil \mathtt{db}(\beta)\rceil : \mathtt{C}_2\langle u'\rangle[x/\lfloor \mathtt{db}(\beta)\rfloor : s]\mathtt{L}\rangle
\end{array}$$

- **db vs. ls in the substitution:** the list of substitutions $\mathtt{L}$ involved in the $\mathtt{db}$ step is of the form $\mathtt{L}_1[y/\mathtt{C}_2\langle z^{\gamma}\rangle]\mathtt{L}_2$ and z is bound by some substitution to a term u, which might be in $\mathtt{C}_1$, in $\mathtt{C}_2$, or in $\mathtt{L}_2$. Let $u' := \gamma\bullet : u$. Then: Then:

$$\begin{array}{ccc}
\mathtt{C}_1\langle @^{\alpha}((\lambda^{\beta}x.t)\mathtt{L}_1[y/\mathtt{C}_2\langle z^{\gamma}\rangle]\mathtt{L}_2, s)\rangle & \xrightarrow{\mathtt{db}(\beta)} & \mathtt{C}_1\langle \alpha\lceil \mathtt{db}(\beta)\rceil : t[x/\lfloor \mathtt{db}(\beta)\rfloor : s]\mathtt{L}_1[y/\mathtt{C}_2\langle z^{\gamma}\rangle]\mathtt{L}_2\rangle \\
\downarrow(\gamma)\bullet\uparrow(u)\downarrow & & \downarrow(\gamma)\bullet\uparrow(u)\downarrow \\
\mathtt{C}_1\langle @^{\alpha}((\lambda^{\beta}x.t)\mathtt{L}_1[y/\mathtt{C}_2\langle u'\rangle]\mathtt{L}_2, s)\rangle & \xrightarrow{\mathtt{db}(\beta)} & \mathtt{C}_1\langle \alpha\lceil \mathtt{db}(\beta)\rceil : t[x/\lfloor \mathtt{db}(\beta)\rfloor : s]\mathtt{L}_1[y/\mathtt{C}_2\langle u'\rangle]\mathtt{L}_2\rangle
\end{array}$$

- **ls vs. step in the argument:** let $\mathtt{C}_1\langle t\rangle \xrightarrow{\nu}_\ell \mathtt{C}_1\langle t'\rangle$ be a step. Then:

$$\begin{array}{ccc} \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle x^\alpha\rangle\!\rangle[x/t]\rangle & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(t)} & \mathtt{C}_1\langle\mathtt{C}_2\langle\alpha\bullet : t\rangle[x/t]\rangle \\ \nu\downarrow & & \nu\downarrow\star \\ \mathtt{C}_1\langle\mathtt{C}_2\langle x^\alpha\rangle[x/t']\rangle & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(t')} & \mathtt{C}_1\langle\mathtt{C}_2\langle\alpha\bullet : t'\rangle[x/t']\rangle \end{array}$$

Note that the names of the two steps marked with $\star$ are both ν, by the fact that adding labels preserves redex names (Lem. 23). To close this diagram, note also that $\uparrow(t') = \uparrow(t)$ by the fact that reduction preserves the first label of a term (Lem. 23).

- **ls vs. step in the body not duplicating x^α:** let $\mathtt{C}_2\langle x^\alpha\rangle$ be a term with an occurrence of x^α and consider a step $\mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle x^\alpha\rangle\!\rangle\rangle \xrightarrow{\nu} \mathtt{C}_1\langle\mathtt{C}_3\langle\!\langle x^\alpha\rangle\!\rangle\rangle$ that does not duplicate x^α. Then:

$$\begin{array}{ccc} \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle x^\alpha\rangle\!\rangle[x/t]\rangle & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(t)} & \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle\alpha\bullet : t\rangle\!\rangle[x/t]\rangle \\ \nu\downarrow & & \nu\downarrow\star \\ \mathtt{C}_1\langle\mathtt{C}_3\langle x^\alpha\rangle[x/t]\rangle & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(t')} & \mathtt{C}_1\langle\mathtt{C}_3\langle\alpha\bullet : t\rangle[x/t]\rangle \end{array}$$

We omit a more detailed analysis in which all possible forms of steps $\mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle x^\alpha\rangle\!\rangle\rangle \xrightarrow{\nu} \mathtt{C}_1\langle\mathtt{C}_3\langle\!\langle x^\alpha\rangle\!\rangle\rangle$ are considered.

- **ls vs. step in the body duplicating x^α:**

$$\begin{array}{ccc} \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle\mathtt{C}_3\langle y^\beta\rangle[y/\mathtt{C}_4\langle\!\langle x^\alpha\rangle\!\rangle]\rangle\!\rangle\rangle[x/t] & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(t)} & \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle\mathtt{C}_3\langle y^\beta\rangle[y/\mathtt{C}_4\langle\alpha : t\rangle]\rangle\!\rangle\rangle[x/t] \\ \downarrow(\beta)\bullet\uparrow(x^\alpha)\downarrow & & \downarrow(\beta)\bullet\uparrow(\alpha:t)\downarrow \\ \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle\mathtt{C}_3\langle\beta : \mathtt{C}_4\langle\!\langle x^\alpha\rangle\!\rangle\rangle[y/\mathtt{C}_4\langle\!\langle x^\alpha\rangle\!\rangle]\rangle\!\rangle\rangle[x/t] & \xrightarrow{\downarrow(\alpha)\bullet\uparrow(t)}\!\!\!\!\rightarrow & \mathtt{C}_1\langle\mathtt{C}_2\langle\!\langle\mathtt{C}_3\langle\beta : \mathtt{C}_4\langle\alpha : t\rangle\rangle[y/\mathtt{C}_4\langle\alpha : t\rangle]\rangle\!\rangle\rangle[x/t] \end{array}$$

Note that $\uparrow(x^\alpha) = \uparrow(\alpha : t)$ by Lem. 23.

◀

▶ Proposition 25 (Multiplicity of residuals). Let $\mathsf{R} \neq \mathsf{S}$ be coinitial redexes. Then:

1. If R is a db step, S/R is a single redex.
2. If R is an ls step, S/R consists of either one or two redexes.

Proof. By analysis on the relative positions of R and S. The proof goes by inspecting all the diagrams in the strong permutation proof (Prop. 24). ◀

A.2.2 Creation implies name contribution

In the following lemma it is shown that whenever a redex R creates a redex S, the name of R contributes to the name of S.

▶ Proposition 26 (Creation implies name contribution). Let R and S be redexes in LLSC such that R creates S. If μ denotes the name of R and ν denotes the name of S, then $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$.

Proof. By case analysis on the creation cases, according to [4].

1. **db creates db.**

$$\begin{array}{ll} & \mathtt{C}\langle @^\alpha(@^\beta((\lambda^\gamma x.(\lambda^\delta y.t)\mathtt{L}_1)\mathtt{L}_2, s)\mathtt{L}_3, u)\rangle \\ \xrightarrow{\mathtt{db}(\gamma)}_\ell & \mathtt{C}\langle @^\alpha((\lambda^{\beta\lceil\mathtt{db}(\gamma)\rceil\delta} y.t)\mathtt{L}_1[x/\lfloor\mathtt{db}(\gamma)\rfloor : s]\mathtt{L}_2\mathtt{L}_3, u)\rangle \\ \xrightarrow{\mathtt{db}(\beta\lceil\mathtt{db}(\gamma)\rceil\delta)}_\ell & \mathtt{C}\langle\alpha\lceil\mathtt{db}(\beta\lceil\mathtt{db}(\gamma)\rceil\delta)\rceil : t\mathtt{L}_1[x/\lfloor\mathtt{db}(\gamma)\rfloor : s]\mathtt{L}_2[y/\lfloor\mathtt{db}(\beta\lceil\mathtt{db}(\gamma)\rceil\delta)\rfloor : u]\mathtt{L}_3\rangle \end{array}$$

Note that $\mathtt{db}(\gamma) \overset{\mathsf{Name}}{\hookrightarrow}_1 \mathtt{db}(\beta\lceil\mathtt{db}(\gamma)\rceil\delta)$, as required.

2. **db creates ls.**

$$\begin{array}{ll} & \mathtt{C}_1\langle @^{\alpha}((\lambda^{\beta}x.\mathtt{C}_2\langle\!\langle x^{\gamma}\rangle\!\rangle)\mathtt{L},t)\rangle \\ \xrightarrow{\mathbf{db}(\beta)}_{\ell} & \mathtt{C}_1\langle\alpha\lceil\mathbf{db}(\beta)\rceil : \mathtt{C}_2\langle\!\langle x^{\gamma}\rangle\!\rangle[x/\lfloor\mathbf{db}(\beta)\rfloor : t]\mathtt{L}\rangle \\ \xrightarrow{\downarrow(\gamma)\bullet\lfloor\mathbf{db}(\beta)\rfloor}_{\ell} & \mathtt{C}_1\langle\alpha\lceil\mathbf{db}(\beta)\rceil : \mathtt{C}_2\langle\gamma\bullet\lfloor\mathbf{db}(\beta)\rfloor : t\rangle[x/\lfloor\mathbf{db}(\beta)\rfloor : t]\mathtt{L}\rangle \end{array}$$

Observe that, according to Lem. 23, if $\mathtt{C}_2$ is a substitution context then $\alpha\lceil\mathbf{db}(\beta)\rceil$: $\mathtt{C}_2\langle\!\langle x^{\gamma}\rangle\!\rangle = \mathtt{C}_2\langle\!\langle x^{\alpha\lceil\mathbf{db}(\beta)\rceil\gamma}\rangle\!\rangle$ whereas, if $\mathtt{C}_2$ involves at least one application or abstraction node, then $\alpha\lceil\mathbf{db}(\beta)\rceil : \mathtt{C}_2\langle\!\langle x^{\gamma}\rangle\!\rangle = (\alpha\lceil\mathbf{db}(\beta)\rceil : \mathtt{C}_2)\langle\!\langle x^{\gamma}\rangle\!\rangle$.
In any case, the name of the created ls step is $\downarrow(\gamma)\bullet\lfloor\mathbf{db}(\beta)\rfloor$.
Finally note that $\mathbf{db}(\beta) \overset{\mathsf{Name}}{\hookrightarrow}_1 \downarrow(\gamma)\bullet\lfloor\mathbf{db}(\beta)\rfloor$, as required.

3. **ls creates db upwards.**

$$\begin{array}{ll} & \mathtt{C}\langle @^{\alpha}(x^{\beta}\mathtt{L}_1[x/(\lambda^{\gamma}y.t)\mathtt{L}_2]\mathtt{L}_3,s)\rangle \\ \xrightarrow{\downarrow(\beta)\bullet\uparrow(\gamma)}_{\ell} & \mathtt{C}\langle @^{\alpha}((\lambda^{\beta\bullet\gamma}y.t)\mathtt{L}_2\mathtt{L}_1[x/(\lambda^{\gamma}y.t)\mathtt{L}_2]\mathtt{L}_3,s)\rangle \\ \xrightarrow{\mathbf{db}(\beta\bullet\gamma)}_{\ell} & \mathtt{C}\langle\alpha\lceil\mathbf{db}(\beta\bullet\gamma)\rceil : t[y/\lfloor\mathbf{db}(\beta\bullet\gamma)\rfloor : s]\mathtt{L}_2\mathtt{L}_1[x/(\lambda^{\gamma}y.t)\mathtt{L}_2]\mathtt{L}_3\rangle \end{array}$$

Note that $\downarrow(\beta)\bullet\uparrow(\gamma) \overset{\mathsf{Name}}{\hookrightarrow}_1 \mathbf{db}(\beta\bullet\gamma)$, as required.

4. **ls creates db downwards.** Similar to the previous case.

◀

A.2.3 Equivalence of residuals and names

In this section we study the relation between the residual relation and labeling. An easy but essential result is that redexes in the labeled calculus have the same name as their ancestors. Moreover, we develop some technical tools, namely *label morphisms* and show a few results that will be used later.

▶ Proposition 27 (Residuals have the same name). Let ρ be a derivation and let $\mathsf{S}' \in \mathsf{S}/\rho$. Then S and S' have the same name in any labeling of ρ.

Proof. By exhaustive case analysis. It suffices to observe that this holds for residuals of S after a single step, and this is an immediate consequence of labeled permutation Prop. 24. ◀

A term t is **initially reachable** if there is an initially labeled term t_0 such that $t_0 \twoheadrightarrow_{\ell} t$. A term is **correctly labeled** if the labels decorating its nodes are generated by the following grammar: $\alpha ::= \mathbf{a} \mid \lceil\alpha\rceil \mid \lfloor\alpha\rfloor \mid \mathbf{db}(\alpha) \mid \alpha\,\alpha \mid \alpha\bullet\alpha$ where $\mathbf{a} \neq \bullet$. We also say that a derivation ρ is initially labeled (resp. correctly labeled) whenever the source $\mathsf{src}(\rho)$ is initially labeled (resp. correctly labeled). The name is quite appropriate since such derivations may be shown to be initial in a precise way. Define a **label morphism** f as a function $f : \mathcal{L} \to \mathcal{L}$ homomorphic on label constructors except for initial labels:

$$f(\bullet) = \bullet \quad f(\lceil\alpha\rceil) = \lceil f(\alpha)\rceil \quad f(\lfloor\alpha\rfloor) = \lfloor f(\alpha)\rfloor \quad f(\mathbf{db}(\alpha)) = \mathbf{db}(f(\alpha)) \quad f(\alpha\,\beta) = f(\alpha)\,f(\beta)$$

Label morphisms are extended to redex names as follows:

$$f(\mathbf{db}(\alpha)) = \mathbf{db}(f(\alpha)) \quad f(\alpha\bullet\beta) = \downarrow(f(\alpha))\bullet\uparrow(f(\beta))$$

Initially reachable terms are of course not necessarily initially labeled, but they are correctly labeled:

▶ **Lemma 28** (Labeling invariant). *Initially reachable terms are correctly labeled.*

Proof. Straightforward, observing that the invariant holds for initially labeled terms, and that it is preserved by labeled reduction. ◀

The converse property is not true in general, *i.e.* that two redexes have the same name does not imply that one is a residual of the other, unless one starts from an initially labeled term.

▶ **Lemma 29** (Label morphisms are functorial)**.** *If* $\rho = (t_0 \xrightarrow{\mu_1}_\ell t_1 \ldots \xrightarrow{\mu_n}_\ell t_n)$ *is a labeled derivation, then* $f(\rho) := (f(t_0) \xrightarrow{f(\mu_1)}_\ell f(t_1) \ldots \xrightarrow{f(\mu_n)}_\ell f(t_n))$ *is a variant of* ρ*.*

Proof. By induction on n. The interesting case is when $n = 1$, which can be checked by straightforward case analysis on the kind of redex that is contracted. ◀

▶ Proposition 30 (Initial labeled reduction). Let ρ and σ be two variants of the same derivation, such that ρ is initially labeled and σ is correctly labeled. Then there is a unique label morphism f such that $f(\rho) = \sigma$. Moreover, f is determined by $\mathsf{src}(\rho)$ and $\mathsf{src}(\sigma)$ only.

Proof. Let t be the source of ρ and s the source of σ. Consider the label morphism f that maps an initial label $\mathbf{a}$ to a label α whenever $\mathbf{a}$ decorates a subterm of t and α decorates the corresponding subterm of s. Then $f(t) = s$. The labeling of a derivation depends only on the labeling of its source, so we obtain that $f(\rho) = \sigma$. ◀

Given a fixed unlabeled derivation ρ, consider the category whose objects are correctly labeled variants of ρ and there is an arrow $\sigma \to f(\sigma)$ for each label morphism f. The previous proposition proves that any initially labeled variant ρ_0 is an initial object. In particular, any two initially labeled variants ρ_0, ρ_1 of ρ are isomorphic.

▶ Proposition 31 (Initially labelled redexes have unique names). Let R and S be two different coinitial redexes in the labeled calculus such that the source is initially labelled. Then the names of R and S are different.

Proof. Two different db redexes must involve two different abstractions, hence they must have different names. Similarly, two different ls redexes must involve two different variables, so their names must also be different. ◀

▶ **Definition 32** (Residuals defined via the labeled calculus)**.** Let $M \xrightarrow{\mathsf{S}} N$. If R is a redex from M and R' is a redex from N, we define $\mathsf{R}//\mathsf{S}$ as the following set:

- Let $t^\ell \to_\ell s^\ell$ be a variant of the step $M \xrightarrow{\mathsf{S}} N$, with t^ℓ initially labelled.
- Let μ be the name of R in t^ℓ.
- Then $\mathsf{R}' \in \mathsf{R}//\mathsf{S}$ if and only if the name of R' is μ.

Note that this is well-defined, in the sense that it does not depend on the initial labelling chosen for t^ℓ, as can be seen by taking the label isomorphism f induced by the obvious bijection between initial labels, and resorting to the fact that label morphisms lift reductions (Lem. 29).

▶ Proposition 33 (Equivalence of residual notions). Let $\mathsf{R}, \mathsf{S} \in \mathsf{Red}(M)$ for M some unlabelled term. Then $\mathsf{R}//\mathsf{S} = \mathsf{R}/\mathsf{S}$, where $\mathsf{R}//\mathsf{S}$ denotes the set of residuals according to Def. 32 and R/S is the usual notion of residuals.

Proof. We omit the technical, but straightforward proof. ◀

A.2.4 Bounded reduction is WN

In this section it is shown that reduction in LLSC restricted to labels of bounded height is WN.

Throughout this section we work with auxiliary calculi LLSC_P and $\mathrm{LLSC}^\star_P$, in which we only allow contraction of redexes whose names verify a given predicate P.

▶ **Definition 34** (The P-restricted LLSC). Let P be a predicate on redex names. LLSC_P is given by the set of terms $\mathcal{T}^\ell$, with the reduction relation $\to_P$ being defined as in LLSC, but restricted to contracting only db and ls-redexes whose names verify the predicate P. As in LLSC, we write $t \xrightarrow{\mu}_P s$ if $t \to_P s$ and the redex name is μ.

▶ **Definition 35** (Bounded predicate). A predicate P on redex names is said to be *bounded* iff there exists a bound $H \in \mathbb{N}$ such that $P(\mu)$ implies $h(\mu) < H$ for every redex name μ.

▶ Proposition 36 (Created redexes are taller than the contracted redex). Let $t \xrightarrow{\alpha}_P s$, and let α' be the name of a created redex in s. Then $h(\alpha) < h(\alpha')$.

Proof. Recall that creation implies name contribution (Prop. 26), so $\alpha \overset{\mathsf{Name}}{\hookrightarrow} \alpha'$. Recall also that $\overset{\mathsf{Name}}{\hookrightarrow}$ is the transitive closure of $\overset{\mathsf{Name}}{\hookrightarrow}_1$. To conclude it suffices to show that if $\alpha \overset{\mathsf{Name}}{\hookrightarrow}_1 \beta$ then $h(\alpha) < h(\beta)$, which is immediate by inspecting the definition of $\overset{\mathsf{Name}}{\hookrightarrow}_1$ given in Sec. 4. ◀

▶ **Definition 37** (P-redex, P-normal term). A P-redex is a redex verifying the predicate P. A term is P-normal if it is in normal form for $\to_P$.

▶ **Definition 38** (Non-duplicating P-redex). A P-redex $\mathsf{R} : t \to s$ is said to be *non-duplicating* if S/R is a singleton for every P-redex S such that $\mathsf{S} \neq \mathsf{R}$. Otherwise we say that R is duplicating.

▶ **Lemma 39** (Every non-P-normal term has a non-duplicating P-redex). *Let t be a term not in $\to_P$-normal form. Then t has a non-duplicating redex.*

Proof. We resort to the following definition of *anchor*. Given any db redex R, of the form $(\lambda \underline{x}.t)\mathtt{L}\, s$, we call *anchor* to the underlined (binding) occurrence of x. Given any ls redex R, of the form $\mathtt{C}\langle\langle \underline{x} \rangle\rangle[x/t]$, we call *anchor* to the underlined (bound) occurrence of x.

Let R be the P-redex from t whose anchor is rightmost. Then we claim that R is non-duplicating. In fact, if R is a db redex, then it is immediate, since db redexes do not duplicate redexes (*i.e.* S/R is a singleton for every redex $\mathsf{S} \neq \mathsf{R}$). If R is an ls redex $\mathtt{C}\langle\langle x \rangle\rangle[x/t] \to \mathtt{C}\langle t \rangle[x/t]$, then R is duplicating if there is another P-redex whose anchor lies inside the term t. But this is impossible since the anchor of R is the rightmost one among the anchors of P-redexes. ◀

Let $\mathfrak{m}, \mathfrak{n}, \ldots$ stand for multisets, and let $\uplus$ stand for the union of multisets (adding occurrences). We use the following standard extension of a well-founded ordering $(A, >)$ to a well-founded ordering over multi-sets of elements of A:

▶ **Definition 40** (Well-founded ordering for multisets). Let $(A, >)$ be a well-founded set, and $\mathfrak{m}$, $\mathfrak{n}$ two multisets of elements of A. Then:

$$\mathfrak{m} > \mathfrak{n} \quad \overset{\text{def}}{\iff} \quad \mathfrak{m} \neq \mathfrak{n} \ \wedge\ \forall x \in A, (\mathfrak{n}(x) > \mathfrak{m}(x) \implies \exists y \in A,\ y > x \ \wedge\ \mathfrak{m}(y) > \mathfrak{n}(y))$$

▶ Proposition 41 (Bounded reduction is WN). If P is a bounded predicate, $\to_P$ is WN.

Proof. We show that repeatedly contracting a non-duplicating P-redex always reaches the $\to_P$-normal form. Let $H \in \mathbb{N}$ be a bound for the bounded predicate P. Consider the following measure, which takes a term and yields a multiset of integers:

$$\#(t) \stackrel{\text{def}}{=} \{H - h(\alpha) \mid \alpha \text{ is the name of a } P\text{-redex whose source is } t\}$$

If t is not P-normal it has at least one non-duplicating P-redex $\mathsf{R} : t \xrightarrow{\alpha}_\ell s$ (by Lem. 39). The multiset of names of P-redexes of t is of the form $\mathfrak{m} \uplus \{\alpha\}$ where α is the name of R and $\mathfrak{m}$ are the names of all the other P-redexes. Since R is non-duplicating, there is exactly one residual S/R for each P-redex $\mathsf{S} \neq \mathsf{R}$. So the multiset of names of redexes of s is of the form $\mathfrak{m} \uplus \mathfrak{n}$ where $\mathfrak{m}$ are the names of the residuals of all the P-redexes $\mathsf{S} \neq \mathsf{R}$, and $\mathfrak{n}$ are the P-redexes created by R. Recall that residuals of redexes always have the same name as their ancestors (Prop. 27) so these are in fact the only P-redexes in s.

Note that the heights of $\mathfrak{n}$ are all greater than the height of α, since the names of created redexes are always taller than the name of their creator (Prop. 36). Hence $\#(t) = \mathfrak{m} \uplus \{\alpha\} > \mathfrak{m} \uplus \mathfrak{n} = \#(s)$. The order on multisets is well-founded, so this proves that the normal form is eventually reached. ◀

A.2.5 Bounded reduction is SN

In this section we prove that the labelled calculus LLSC restricted to bounded families of labels is SN. It is a well-known fact that, in the LSC without gc, WN $\iff$ SN, so this is in fact a corollary of Prop. 41. Here we give an alternative proof that can also be extended to the LSC with gc.

The structure of the proof is as follows: first we prove that $\to_P$ is increasing. Then we note that if P is bounded then $\to_P$ is WCR, WN, and increasing which allows us to conclude that it is SN, resorting to an abstract result due to Klop and Nederpelt.

▶ **Definition 42** (Measure of labels, terms, and contexts)**.** The measure $\|\alpha\|$ of a label α is defined as follows:

$$\|\mathbf{a}\| \stackrel{\text{def}}{=} 1 \qquad \|\lceil\alpha\rceil\| = \|\lfloor\alpha\rfloor\| = \|\mathtt{db}(\alpha)\| \stackrel{\text{def}}{=} 1 + \|\alpha\| \qquad \|\alpha\,\beta\| \stackrel{\text{def}}{=} \|\alpha\| + \|\beta\|$$

The measure $\|t\|$ of a term t is the sum of the measures of all its labels:

$$\begin{array}{rclcrcl} \|x^\alpha\| & \stackrel{\text{def}}{=} & \|\alpha\| & \qquad & \|\lambda^\alpha x.t\| & \stackrel{\text{def}}{=} & \|\alpha\| + \|t\| \\ \|@^\alpha(t,s)\| & \stackrel{\text{def}}{=} & \|\alpha\| + \|t\| + \|s\| & & \|t[x/s]\| & \stackrel{\text{def}}{=} & \|t\| + \|s\| \end{array}$$

The measure of a context $\|\mathtt{C}\|$ is defined similarly, taking $\|\Box\| \stackrel{\text{def}}{=} 0$.

▶ **Lemma 43** (Labeled reduction is increasing)**.** *$t \to_P s$ implies $\|t\| < \|s\|$.*

Proof. By induction on the context $\mathtt{C}$ under which the $\to_P$ redex in t is contracted. It is straightforward to check that this holds by inspection of the reduction rules. ◀

The following proposition is **Prop. 5** in the main body:

▶ Proposition 44 (Bounded reduction is SN). Let P be a bounded predicate. Then $\to_P$ is SN.

Proof. We already know that $\to_P$ is WCR (by Prop. 24), WN (by Prop. 41) and increasing (by Lem. 43). By Klop-Nederpelt's Lemma (see e.g. Theorem 1.2.3 (iii) in [26]), we obtain that WCR $\wedge$ WN $\wedge$ Inc $\implies$ SN. ◀

As a corollary we obtain that LLSC_P is confluent, by Newman's Lemma, since it is SN and WCR. Finally:

▶ **Corollary 45** (LLSC is confluent)**.** *Reduction in the labelled calculus is* CR.

Proof. This is also immediate. If $\rho : t \twoheadrightarrow_\ell s$ and $\sigma : t \twoheadrightarrow_\ell u$, define $P(\mu)$ to hold iff μ is the name of redex contracted in ρ or σ. Since the number of such labels is finite, P is bounded. Since LLSC_P is confluent, we conclude. ◀

A.2.6 Algebraic confluence

▶ Proposition 46 (Algebraic confluence). The confluence diagram for two coinitial derivations ρ and σ can be closed with their relative residuals ρ/σ and σ/ρ.

Proof. It suffices to check that the labeled LSC verifies the four basic axioms for orthogonal abstract rewriting systems proposed by Melliès in [22]: **Autoerasure**, **Finite residuals**, **Finite Developments**, and **Semantic orthogonality**[9]. Checking the first three axioms for the labeled LSC is immediate by lifting them from the unlabeled LSC, which is already known to verify all these axioms ([4]). **Semantic orthogonality** for the labeled calculus has already been shown in Prop. 24. Algebraic confluence is then a corollary of Theorem 2.4 in [22]. ◀

A.3 Applications of FFD – Optimal Reduction

We address the proof of the properties required for optimal reduction: (1) LSC forms a Deterministic Family Structure (Prop. 54); (2) the set of reachable normal forms RNFis stable. In this section sometimes we write $t^\ell, s^\ell, \ldots$ rather than $t, s, \ldots$, to emphasize that we are speaking of labeled terms, and similarly for steps and derivatios.

A.3.1 LSC is a DFS

In this subsection we will show that the LSC forms a Deterministic Family Structure according to [16]. We also follow the presentation in [11].

We begin by recalling that LSC forms a Deterministic Residual Structure (DRS), by virtue of the properties that have already been shown in [4]. To endow the LSC with the structure of a DFS, we need a few auxiliary definitions.

▶ **Definition 47** (Inclusion of labels)**.** The order relation of *inclusion* between labels α, β, written $\alpha \subseteq \beta$, is given by the reflexive and transitive closure of the following rules:

$$\alpha \subseteq \lceil \alpha \rceil \quad \alpha \subseteq \lfloor \alpha \rfloor \quad \alpha \subseteq \mathtt{db}(\alpha) \quad \alpha \subseteq \alpha\beta \quad \alpha \subseteq \beta\alpha$$

▶ Remark. If $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$ then $h(\mu) < h(\nu)$.

▶ **Lemma 48** (Name contribution is preserved by morphisms)**.** *Let f be a label morphism and let μ and ν be (non-*gc*) redex names. Then $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$ implies $f(\mu) \overset{\mathsf{Name}}{\hookrightarrow} f(\nu)$.*

[9] We follow the nomenclature in [4]. In the original work of Melliès [22] **Autoerasure** is called **A**, **Finite residuals** is called **B**, and **Semantic orthogonality** is called **PERM**.

Proof. Recall that $\overset{\mathsf{Name}}{\hookrightarrow}$ is the transitive closure of $\overset{\mathsf{Name}}{\hookrightarrow}_1$, so it suffices to check that the property holds for one step of $\overset{\mathsf{Name}}{\hookrightarrow}_1$. This is straighforward by case analysis on the rules defining the relation $\overset{\mathsf{Name}}{\hookrightarrow}_1$. ◀

▶ Proposition 49 (Permutation equivalent derivations yield the same labellings). Let ρ_1 and ρ_2 be permutation equivalent derivations. Let ρ_1^ℓ and ρ_2^ℓ be labelled variants of ρ_1 and ρ_2 respectively such that $\mathsf{src}(\rho_1^\ell) = \mathsf{src}(\rho_2^\ell)$. Then $\mathsf{tgt}(\rho_1^\ell) = \mathsf{tgt}(\rho_2^\ell)$.

Proof. Recall that $\equiv$ is the reflexive, symmetric and transitive closure of the one-step permutation axiom $\equiv^1$. We proceed by induction on the derivation that $\rho_1 \equiv \rho_2$. The reflexivity, symmetry and transitivity cases are immediate. The only interesting case is the axiom, *i.e.* when: $\rho_1 = \tau_1 \,\mathsf{R}\, \sigma\, \tau_2 \equiv^1 \tau_1 \,\mathsf{S}\, \rho\, \tau_2 = \rho_2$. This is a consequence of the strong permutation property (Prop. 24). ◀

▶ **Definition 50** (Family relation and contribution relation). Given two coinitial steps with history $\rho\mathsf{R}$, $\sigma\mathsf{S}$, we define two binary relations, written $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$ ("$\rho\mathsf{R}$ and $\sigma\mathsf{S}$ are in the same family") and $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \sigma\mathsf{S}$ ("$\rho\mathsf{R}$ contributes to $\sigma\mathsf{S}$") according to the following conditions:

1. Consider the labelled variants $\rho^\ell\mathsf{R}^\ell$ and $\sigma^\ell\mathsf{S}^\ell$ of $\rho\mathsf{R}$ and $\sigma\mathsf{S}$ respectively, whose source is an initially labelled variant t^ℓ of t. Let μ be the name of R^ℓ and let ν be the name of S^ℓ.
2. We declare $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$ to hold if and only if $\mu = \nu$.
3. We declare $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \sigma\mathsf{S}$ to hold if and only if $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$.

It is straighforward to check that the relations $\overset{\mathsf{Fam}}{\simeq}$ and $\overset{\mathsf{Fam}}{\hookrightarrow}$ are well-defined, *i.e.* that they do not depend on the initial labelling t^ℓ, and that $\overset{\mathsf{Fam}}{\simeq}$ is an equivalence relation. Equivalence clases of $\overset{\mathsf{Fam}}{\simeq}$ are called *families*. Note that $\overset{\mathsf{Fam}}{\hookrightarrow}$ is a binary relation between steps with history. We also extend $\overset{\mathsf{Fam}}{\hookrightarrow}$ to families as follows: $\phi_1 \overset{\mathsf{Fam}}{\hookrightarrow} \phi_2$ whenever $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \sigma\mathsf{S}$ for some $\rho\mathsf{R} \in \phi_1$ and $\sigma\mathsf{S} \in \phi_2$. It is also straighforward to check that this definition does not depend upon the choice of representatives.

▶ **Definition 51** (All labels in a term). Given a labelled term t^ℓ, the set of all labels decorating nodes in t^ℓ is written $\mathit{labels}(t^\ell)$. This definition is also extended to contexts.

The following lemma collects various syntactical properties of reduction in the labelled LSC. We omit the detailed proofs. They are all straightforward case analyses by inspection of the reduction rules:

▶ **Lemma 52** (Properties of redex names). *The following hold:*

1. **Redex names that contribute to a redex must occur in the source.**
   If $\mathsf{R}^\ell : t_0^\ell \to t_1^\ell$ is a labelled step, ν is the name of R^ℓ, and μ is another redex name such that $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$, then there exists a label $\alpha \in \mathit{labels}(t_0^\ell)$ such that $\mu \subseteq \alpha$.
2. **All redex names that occur in a term result from contracting a redex.**
   Let $\rho^\ell : t_0^\ell \twoheadrightarrow_\ell t_1^\ell$ be a derivation in the labelled calculus, where t_0^ℓ is an initially labelled term. Let μ be a redex name such that $\mu \subseteq \alpha$ for some label $\alpha \in \mathit{labels}(t_1^\ell)$. Then ρ^ℓ has a step whose name is μ, i.e. *ρ^ℓ can be written as of the form $\rho_1^\ell\mathsf{R}^\ell\rho_2^\ell$, where the name of R^ℓ is μ.*
3. **Any redex has a residual after a derivation not including its name.**
   Let R_0 be a step and ρ be a coinitial derivation. Let t^ℓ be an initially reachable variant of the source, and consider the labelled variants R_0^ℓ and ρ^ℓ of R_0 and ρ respectively, whose source is t^ℓ. Let μ be the name of R_0^ℓ, and suppose that μ is not among the names of the

redexes contracted by ρ^ℓ*. Then there exists a step* $\mathsf{R}_1 \in \mathsf{R}_0/\rho$*. Moreover, the name of its labelled variant* R_1^ℓ *is also* μ*.*

4. **Any redex has an ancestor before a derivation not contributing to its name.** *Let* ρ *be a derivation and let* R_1 *be a composable step,* i.e. $\mathsf{tgt}(\rho) = \mathsf{src}(\mathsf{R}_1)$*. Let* t^ℓ *be an initially reachable variant of the source of* ρ*, consider the labelled variant* ρ^ℓ *of* ρ *whose source is* t^ℓ*, and the labelled variant of* R_1^ℓ *of* R_1 *whose source if* $\mathsf{tgt}(\rho^\ell)$*. Let* μ *be the name of* R_1^ℓ*, and suppose that the names of the redexes contracted by* ρ^ℓ *do not contribute to* μ*,* i.e. *every step* S^ℓ *in* ρ^ℓ *has a name* ν *such that* $\nu \overset{\mathsf{Name}}{\hookrightarrow} \mu$ *does not hold. Then there exists a step* R_0 *such that* $\mathsf{R}_1 \in \mathsf{R}_0/\rho$*. Moreover, the name of its labelled variant* R_0^ℓ *is also* μ*.*

Proof. We omit the technical proofs, which are by induction/case analysis, and require some auxiliary lemmas. ◀

▶ Proposition 53 (Contribution axiom for LSC). Let $\phi_1, \phi_2 \in \mathsf{Hist}(t)/\overset{\mathsf{Fam}}{\simeq}$ be coinitial families in LLSC. Then the following are equivalent:

- $\phi_1 \overset{\mathsf{Fam}}{\hookrightarrow} \phi_2$
- For every step with history $\sigma\mathsf{S} \in \phi_2$, there is a step with history $\rho\mathsf{R} \in \phi_1$ such that $\rho\mathsf{R}$ is a prefix of σ.

Proof. Let us show each direction of the implication.

($\Rightarrow$) Let $\sigma\mathsf{S} \in \phi_2$ be a step with history. Consider an initially labelled variant t_0^ℓ of t, and the labelled variant $\sigma^\ell\mathsf{S}^\ell$ of $\sigma\mathsf{S}$ whose source is t_0^ℓ. Let $t_1^\ell = \mathsf{tgt}(\sigma^\ell) = \mathsf{src}(\mathsf{S}^\ell)$. Moreover, let $\tau\mathsf{T} \in \phi_1$, and consider the labelled variant $\tau^\ell\mathsf{T}^\ell$ of $\tau\mathsf{T}$ whose source is t_0^ℓ.
Let ν be the name of S^ℓ, and let μ be the name of T^ℓ. Since $\phi_1 \overset{\mathsf{Fam}}{\hookrightarrow} \phi_2$ we have, by definition, that $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$. By the fact that names contributing to a step must occur in the source (Lem. 52), there exists a label $\alpha \in \mathit{labels}(t_1^\ell)$ such that $\mu \subseteq \alpha$. By Lem. 52, there must exist a step in σ^ℓ whose name is μ. This means that $\sigma^\ell = \rho^\ell\mathsf{R}^\ell\upsilon^\ell$ where the name of R^ℓ is μ, hence $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \tau\mathsf{T}$ and so $\sigma = \rho\mathsf{R}\upsilon$ where $\rho\mathsf{R} \in \phi_1$, as wanted.

($\Leftarrow$) Let us show that $\phi_1 \overset{\mathsf{Fam}}{\hookrightarrow} \phi_2$. Let $\sigma\mathsf{S} \in \phi_2$ be a step with history. Consider an initially labelled variant t_0^ℓ of t, and consider the labelled variant $\sigma^\ell\mathsf{S}^\ell$ whose source is t_0^ℓ. Let ν be the name of S^ℓ. Let P be the predicate on redex names such that $P(\mu)$ holds if and only if $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$. Observe that P is a bounded predicate, since by Rem. A.3.1 we have that $h(\mu) < h(\nu)$ for every μ such that $P(\mu)$ holds. Hence labelled reduction in the calculus restricted to P is strongly normalizing (Prop. 44). Consider a maximal derivation ρ^ℓ starting from t_0^ℓ and contracting redexes whose names verify the predicate P; then ρ^ℓ must be finite as we have just argued. By algebraic confluence for the labelled calculus (Prop. 46) we may close the diagram formed by ρ^ℓ and σ^ℓ with labelled variants of the relative projections ρ/σ and σ/ρ. Note that, by definition of the residual relation, any step contracted along ρ/σ must be the residual of some step in ρ. Moreover, we know that residuals of redexes have the same name as their ancestor (Prop. 27), so given any step T^ℓ that is contracted along $(\rho/\sigma)^\ell$ its name ξ is also the name of a step T_0^ℓ that is contracted along ρ^ℓ. Hence ξ must verify the predicate P, which means that $\xi \overset{\mathsf{Name}}{\hookrightarrow} \nu$. In particular $\xi \neq \nu$, since the relation $\overset{\mathsf{Name}}{\hookrightarrow}$ is a strict partial order. Then by Lem. 52 there is a residual $\mathsf{S}_1 \in \mathsf{S}/(\rho/\sigma)$ and the name of its corresponding labelled variant S_1^ℓ is also ν.
We need an auxiliary claim:
Claim: the names of the redexes contracted along $(\sigma/\rho)^\ell$ do not contribute to ν.
Proof of the claim. By contradiction, suppose that $(\sigma/\rho)^\ell$ is of the form $\tau_1^\ell\mathsf{T}^\ell\tau_2^\ell$ where

the name of T^ℓ is ξ and it contributes to ν, that is $\xi \overset{\mathsf{Name}}{\hookrightarrow} \nu$. Without loss of generality, let T^ℓ be the first such step. Then the names of the redexes contracted along τ_1^ℓ do not contribute to ξ, because if τ_1^ℓ contracts a redex T'^ℓ whose name is $\xi' \overset{\mathsf{Name}}{\hookrightarrow} \xi$, then by transitivity of $\overset{\mathsf{Name}}{\hookrightarrow}$ we have $\xi' \overset{\mathsf{Name}}{\hookrightarrow} \nu$, contradicting the hypothesis that T is the first redex with that property. By Lem. 52 this means that T^ℓ must have an ancestor T_0^ℓ, that is a step T_0 such that $\mathsf{T} \in \mathsf{T}_0/(\sigma/\rho)$ and such that the name of T_0^ℓ is also ξ. Thus we obtain a derivation $\rho^\ell \mathsf{T}_0^\ell$ where the name of T_0 verifies P. This contradicts the hypothesis that ρ^ℓ was a maximal derivation contracting only redexes that verify P. *(End of the proof of the claim).*

Now since redexes contracted along $(\sigma/\rho)^\ell$ do not contribute to the name of S_1^ℓ, we may apply Lem. 52 and obtain that there exists an ancestor S_2^ℓ, *i.e.* a step S_2 such that $\mathsf{S}_1 \in \mathsf{S}_2/(\sigma/\rho)$ and such that the name of S_2^ℓ is also ν. The situation is as follows:

$$\begin{tikzcd}
t_0^\ell \arrow[r, two heads, "\sigma^\ell"] \arrow[d, two heads, "\rho^\ell"'] & {} \arrow[r, "\mathsf{S}^\ell"] \arrow[d, two heads, "(\rho/\sigma)^\ell"] & {} \\
{} \arrow[r, two heads, "(\sigma/\rho)^\ell"'] \arrow[d, dotted, "\mathsf{S}_2^\ell"'] & {} \arrow[dr, dotted, "\mathsf{S}_1^\ell"] & \\
{} & & {}
\end{tikzcd}$$

To conclude the proof, note that $\rho\mathsf{S}_2 \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$ since S_2^ℓ and S^ℓ have the same name, namely ν. So $\rho\mathsf{S}_2 \in \phi_2$ since $\sigma\mathsf{S} \in \phi_2$. By hypothesis, this implies that there exists a step with history $\rho_1\mathsf{R} \in \phi_1$ such that ρ can be written as of the form $\rho_1\mathsf{R}\rho_2$. Consider the labelled variant $\rho_1^\ell\mathsf{R}^\ell$ of $\rho_1\mathsf{R}$ whose source is t_0^ℓ. The step R^ℓ is one of the redexes in ρ^ℓ. By construction, the names of all the steps contracted along ρ^ℓ verify the predicate P. In particular, if we let μ stand for the name of R^ℓ, we have that $P(\mu)$ holds, *i.e.* that $\mu \overset{\mathsf{Name}}{\hookrightarrow} \nu$. This, by definition, means that $\rho_1\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \rho\mathsf{S}_2$, and this in turn means that $\phi_1 \overset{\mathsf{Fam}}{\hookrightarrow} \phi_2$, as required.

◀

The following proposition is **Prop. 9** in the main body.

▶ Proposition 54 (The LSC with families induced by labels is a DFS). The triple $(\mathrm{LSC}, \overset{\mathsf{Fam}}{\simeq}, \overset{\mathsf{Fam}}{\hookrightarrow})$ forms a DFS, where LSC stands, by abuse of notation, for the corresponding DRS, $\overset{\mathsf{Fam}}{\simeq}$ is the family relation between steps with history, and $\overset{\mathsf{Fam}}{\hookrightarrow}$ is the contribution relation between families.

Proof. Let us check each of the axioms:

1. **Initial.** Let R and S be distinct coinitial steps. Then we claim that $\mathsf{R} \simeq \mathsf{S}$ does not hold. Indeed, let t^ℓ be an initially labelled variant of the source of R and S, and let R^ℓ and S^ℓ be their respective labelled variants. Then Lem. 31 ensures that, since R^ℓ and S^ℓ are different coinitial steps whose source is an initially labelled term, they must have different names. We conclude that $\mathsf{R} \in \mathsf{Fam}_\simeq(\mathsf{R})$ but $\mathsf{R} \notin \mathsf{Fam}_\simeq(\mathsf{S})$, which implies $\mathsf{Fam}_\simeq(\mathsf{R}) \neq \mathsf{Fam}_\simeq(\mathsf{S})$.
2. **Copy.** Let $\rho\mathsf{R} \leq \sigma\mathsf{S}$, and let us show that $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$. By definition of $\leq$, there exists a derivation τ such that $\mathsf{S} \in \mathsf{R}/\tau$ and $\rho\tau \equiv \sigma$. Let t be the source of the derivations ρ and σ, let t^ℓ be an initially labelled variant of the term t, and let ρ^ℓ, σ^ℓ, τ^ℓ, R^ℓ, S^ℓ, $\mathsf{S}^{\ell\ell}$ denote labelled variants of ρ, σ, τ, R, S, and S respectively, in such a way that:
   - $\rho^\ell\tau^\ell\mathsf{S}^\ell$ is a labelled variant of $\rho\tau\mathsf{S}$ whose source is t^ℓ,

- $\rho^\ell \mathsf{R}^\ell$ is a labelled variant of $\rho\mathsf{R}$ whose source is t^ℓ,
- $\sigma^\ell \mathsf{S}^{\ell\ell}$ is a labelled variant of $\sigma\mathsf{S}$ whose source is t^ℓ.

To see that $\rho\mathsf{R} \overset{\mathsf{Fam}}{\simeq} \sigma\mathsf{S}$. it suffices to check that R^ℓ and $\mathsf{S}^{\ell\ell}$ have the same name. Recall that coinitial labelled variants of permutation equivalent derivations must be cofinal (Prop. 49). This implies that $\mathsf{tgt}(\rho^\ell\tau^\ell) = \mathsf{tgt}(\sigma^\ell)$, so $\mathsf{S}^\ell = \mathsf{S}^{\ell\ell}$. Moreover, residuals of redexes have the same name (Prop. 27), and $\mathsf{S} \in \mathsf{R}/\tau$ so the names of R^ℓ and $\mathsf{S}^\ell = \mathsf{S}^{\ell\ell}$ coincide, as required.

3. **Finite family developments.** Let ρ be a potentially infinite derivation that contracts redexes in a finite number of families. Let t^ℓ be an initially labelled variant of the source of ρ, and let ρ^ℓ be a labelled variant of ρ starting from t^ℓ. Let P be the predicate on redex names such that $P(\mu)$ holds if and only if μ is one of the names of the redexes contracted along ρ^ℓ. Then P is bounded, since only a finite number of families are contracted by ρ^ℓ, so by Prop. 44 ρ^ℓ must be finite. Hence ρ is also finite.
4. **Creation.** Let $\rho\mathsf{R}$ be a step with history such that R creates S, and let us check that $\mathsf{Fam}_{\simeq}(\rho\mathsf{R}) \overset{\mathsf{Fam}}{\hookrightarrow} \mathsf{Fam}_{\simeq}(\rho\mathsf{RS})$. By definition, it suffices to check that $\rho\mathsf{R} \overset{\mathsf{Fam}}{\hookrightarrow} \rho\mathsf{RS}$.
Consider an initially labelled variant t^ℓ of the source of ρ, and labelled variants ρ^ℓ, R^ℓ, and S^ℓ of ρ, R, and S respectively, such that $\rho^\ell\,\mathsf{R}^\ell\,\mathsf{S}^\ell$ is a labelled variant of $\rho\mathsf{RS}$ whose source is t^ℓ. Let μ be the name of R^ℓ and let ν be the name of S^ℓ. By applying Lem. 26, we conclude that $\mu \overset{\mathsf{Name}}{\hookrightarrow}_1 \nu$, as required.
5. **Contribution.** This has been shown in Prop. 53.

◀

A.3.2 RNF is a stable set

The following definitions of X-needed redex and stable set are taken from [16].

▶ **Definition 55** (X-needed redex)**.** Let X be a set of terms and let R be a redex. Let $t = \mathsf{src}(\mathsf{R})$. Then R is X-needed if any reduction $\rho : t \twoheadrightarrow s \in X$ contracts at least one residual of R.

▶ **Definition 56** (Stable set)**.** A set of terms X is *stable* if:

1. **X is closed under parallel moves:** for any $t \notin X$, any $\rho : t \twoheadrightarrow s \in X$, and any $\sigma : t \twoheadrightarrow u$ which does not contain terms in X, the final term of ρ/σ is in X.
2. **X is closed under unneeded expansion:** for any $\mathsf{R} : t \to s$ such that $t \notin X$ and $s \in X$, the step R is needed.

▶ **Definition 57** (Reachable contexts)**.** Reachable contexts are defined by the following grammar:

$$\mathtt{R} ::= \Box \mid \mathtt{R}t \mid t\mathtt{R} \mid \lambda x.\mathtt{R} \mid \mathtt{R}[x/t] \mid \mathtt{R}\langle\!\langle x\rangle\!\rangle[x/\mathtt{R}]$$

A variable x is *reachable* in a term t if it occurs free under a reachable context, *i.e.* $t = \mathtt{R}\langle\!\langle x\rangle\!\rangle$ such that $\mathtt{R}$ does not bind x. We write $\mathsf{rv}(t)$ for the set of reachable variables of t. Given a term t, a *reachable redex* is either a db redex whose application node is under a reachable context, or an ls redex contracting a variable under a reachable context. A term t is a *reachable-normal form* if it has no reachable redexes. The set of reachable-normal forms is written RNF. If a context (resp. variable, redex) is not reachable we say that it is *unreachable*.

Our aim is to prove that the set of reachable normal forms is a stable set. The proof depends on a number of technical definitions and lemmas. We state the lemmas below but we omit their proofs.

▶ **Definition 58** (Nesting). We follow the definition of nesting given in [4]. Namely R *immediately nests* S (written $\mathsf{R} \prec_{\mathsf{B}}^{1} \mathsf{S}$) if the *anchor* of S lies inside the *box* of R. Moreover, $\mathsf{R} \prec_{\mathsf{B}} \mathsf{S}$ is defined as the transitive closure of $\prec_{\mathsf{B}}^{1}$, and then we say that R *nests* S.

▶ **Definition 59** (Strongly reachable redex). A redex $\mathsf{R} : t \to_{\mathsf{db} \cup \mathsf{ls}} s$ is said to be *strongly reachable* if and only if R is reachable and it is not nested by any other redex, *i.e.* R is minimal with respect to $\prec_{\mathsf{B}}$.

▶ **Lemma 60** (Characterization of terms in RNF). *A term t is a reachable normal form if and only if $\mathsf{nf}_{\mathsf{gc}}(t)$ is in $\to_{\mathsf{db} \cup \mathsf{ls}}$-normal form.*

▶ **Lemma 61** (Unreachable redexes have no residual after gc normalization). *Let $\mathsf{R} : t_1 \to_{\mathsf{db} \cup \mathsf{ls}} t_2$ be an unreachable redex. Let $\sigma : t_1 \twoheadrightarrow_{\mathsf{gc}} \mathsf{nf}_{\mathsf{gc}}(t_1)$ be a reduction to gc-normal form. Then R/σ is empty.*

▶ **Lemma 62** (Strongly reachable redexes have reachable residuals). *Let R be a strongly reachable redex and let $\mathsf{S} \neq \mathsf{R}$ be any other redex coinitial to R. Then:*
- *The set of residuals R/S is a singleton and it is reachable.*
- *If $\mathsf{tgt}(\mathsf{R})$ is in RNF, then R/S is strongly reachable.*

Finally, the lemma below proves the result we are aiming for, corresponding to **Lem. 7** in the main body.

▶ **Lemma 63.** *The set of reachable-normal forms RNF is stable.*

Proof. The proof goes by checking items 1. and 2. in the definition of stable set:

1. **RNF is closed under parallel moves.** It suffices to check that RNF is closed under reduction. Let $\mathsf{R} : t \to_{\mathsf{db} \cup \mathsf{ls}} s$ with $t \in \mathsf{RNF}$, and let us check that $s \in \mathsf{RNF}$. It can be proved as a lemma that RNFs given in Lem. 60, that $t \in \mathsf{RNF}$ if and only if $\mathsf{nf}_{\mathsf{gc}}(t)$ is in $\to_{\mathsf{db} \cup \mathsf{ls}}$-normal form, and similarly for s.
   Let $\sigma : t \twoheadrightarrow_{\mathsf{gc}} \mathsf{nf}_{\mathsf{gc}}(t)$ be a sequence of gc steps to normal form. Since $t \in \mathsf{RNF}$, by Lem. 60, we have that $\mathsf{nf}_{\mathsf{gc}}(t)$ is in $\to_{\mathsf{db} \cup \mathsf{ls}}$-normal form. Consider the relative projections σ/R and R/σ. Since σ/R is the projection of a sequence of gc steps, it is also sequence of gc steps. Let $\sigma/\mathsf{R} : s \twoheadrightarrow_{\mathsf{gc}} s'$. The situation is:

$$\begin{tikzcd} t \arrow[r, "\mathsf{R}"] \arrow[d, twoheadrightarrow, "\sigma"'] & s \arrow[d, twoheadrightarrow, dotted, "\sigma/\mathsf{R}"] \\ \mathsf{nf}_{\mathsf{gc}}(t) \arrow[r, twoheadrightarrow, dotted, "\mathsf{R}/\sigma"'] & s' \end{tikzcd}$$

   Since $\mathsf{nf}_{\mathsf{gc}}(t)$ is in $\to_{\mathsf{db} \cup \mathsf{ls}}$-normal form, R/σ must be empty, so $s' = \mathsf{nf}_{\mathsf{gc}}(t)$. In particular, s' is a gc normal form, so by confluence s' is the gc normal form of s, *i.e.* $\mathsf{nf}_{\mathsf{gc}}(s) = s' = \mathsf{nf}_{\mathsf{gc}}(t)$. Therefore $\mathsf{nf}_{\mathsf{gc}}(s)$ is in $\to_{\mathsf{db} \cup \mathsf{ls}}$-normal form which means, by Lem. 60, that $s \in \mathsf{RNF}$ as required.
2. **RNF is closed under unneeded expansion.**. Let $\mathsf{R} : t \to_{\mathsf{db} \cup \mathsf{ls}} s$ with $t \notin \mathsf{RNF}$ and $s \in \mathsf{RNF}$, and let us show that R is RNF-needed. In fact, it suffices to show that R is a strongly reachable redex. First we prove that R is reachable.
   Claim: R **is a reachable redex.** By contradiction, suppose that R is unreachable, consider a reduction from t to gc-normal form $\sigma : t \twoheadrightarrow_{\mathsf{gc}} \mathsf{nf}_{\mathsf{gc}}(t)$, and the relative projections $\mathsf{R}/\sigma : \mathsf{nf}_{\mathsf{gc}}(t) \twoheadrightarrow_{\mathsf{db} \cup \mathsf{ls}} s'$ and $\sigma/\mathsf{R} : s \twoheadrightarrow_{\mathsf{db} \cup \mathsf{ls}} s'$. By the fact that unreachable redexes have no residual after going to gc-normal form (Lem. 61) we know that R has

no residual after σ, so R/σ is empty. Hence $\mathsf{nf}_{\mathsf{gc}}(t) = s'$, so s' is in gc-normal form and by confluence we obtain that $\mathsf{nf}_{\mathsf{gc}}(s) = s' = \mathsf{nf}_{\mathsf{gc}}(t)$. The situation is:

$$\begin{tikzcd} t \arrow[r, "\mathsf{R}"] \arrow[d, two heads, "\sigma"'] & s \arrow[d, two heads, dotted, "\sigma/\mathsf{R}"] \\ \mathsf{nf}_{\mathsf{gc}}(t) \arrow[r, two heads, dotted, "\mathsf{R}/\sigma"'] & s' \end{tikzcd}$$

Since $t \in \mathsf{RNF}$, by the characterization of RNFs given in Lem. 60, we have that $\mathsf{nf}_{\mathsf{gc}}(t)$ is not a $\to_{\mathsf{db}\,\cup\,\mathsf{ls}}$-normal form. On the other hand, since $s \in \mathsf{RNF}$, by Lem. 60, we have that $\mathsf{nf}_{\mathsf{gc}}(s) = \mathsf{nf}_{\mathsf{gc}}(t)$ is a $\to_{\mathsf{db}\,\cup\,\mathsf{ls}}$-normal form. This is a contradiction, which concludes the proof of the claim.

To see that R is a strongly reachable redex, we are left to check that R is minimal, among the reachable redexes, with respect to the nesting order $\prec_{\mathsf{B}}$. Indeed, by contradiction, suppose that R is not minimal. Then since the order $\prec_{\mathsf{B}}$ is well-founded (as there are finitely many redexes in any given term) there is a reachable redex such that $\mathsf{S} \prec_{\mathsf{B}} \mathsf{R}$ and such that S is minimal among the reachable redexes. That is, S is a strongly reachable redex. Then by the fact that strongly reachable redexes have reachable residuals (Lem. 62) the redex S/R is reachable. This contradicts the fact that s is in RNF. So R must be minimal with respect to the nesting order $\prec_{\mathsf{B}}$, as required.

◀

A.4 Applications of FFD – Standardisation

A.5 Standardisation by multi-selection strategies

▶ **Definition 64** (Multisteps and multiderivations)**.** Given an ARS $\mathcal{A}$, we define Stp^+ as the set:

$$\mathit{Stp}^+ \stackrel{\text{def}}{=} \{\mathcal{M} \mid \mathcal{M} \text{ is a non-empty finite set of coinitial steps in } \mathit{Stp}\}$$

the source (resp. target) of a multistep $\mathcal{M} \in \mathit{Stp}^+$ is defined as the source (resp. target) of any complete development of $\mathcal{M}$. We know (*e.g.* by the work of Melliès, [22]) that the multisteps of $\mathcal{A}$ for an orthogonal ARS $\mathcal{A}^M$. We write D, E, etc. for *multiderivations i.e.* derivations in $\mathcal{A}^M$. If $\mathsf{D} = \mathcal{M}_1 \ldots \mathcal{M}_\mathsf{n}$ is a multiderivation we say that ρ is a *complete development* of D if $\rho = \rho_1 \ldots \rho_\mathsf{n}$ where ρ_i is a complete development of the multistep $\mathcal{M}_\mathsf{i}$. By FD a complete development always exists and any two complete developments are permutation equivalent. We write $\partial\mathsf{D}$ to stand for some complete development of D in such a way that $\partial(\mathsf{DE}) = (\partial\mathsf{D})(\partial\mathsf{E})$, and we write ρ/D for $\rho/\partial\mathsf{D}$. We write $\mathcal{M} \lhd \sigma$ if for all $\mathsf{R} \in \mathcal{M}$ we have that $\mathsf{R} \lhd \sigma$.

▶ **Definition 65** (Multi-selection strategy)**.** Given an orthogonal ARS $\mathcal{A}$, a *multi-selection strategy* is a function $\mathbb{M}$ that associates every non-empty derivation ρ to a multistep $\mathcal{M}$ such that for $\mathcal{M} \lhd \rho$ and $\mathcal{M}/\rho = \emptyset$, *i.e.* every redex in the multistep belongs to the derivation, and there are no residuals of the multistep left after the derivation.

▶ **Definition 66** (Uniform multi-selection strategy)**.** A multi-selection strategy $\mathbb{M}$ is *uniform* if it is well-defined for permutation equivalence classes, *i.e.* for any two non-empty and permutation equivalent derivations $\rho \equiv \sigma$ we have that $\mathbb{M}(\rho) = \mathbb{M}(\sigma)$.

▶ **Definition 67** (Reduction sequence induced by a multi-selection strategy)**.** The *multiderivation induced by a multi-selection strategy* $\mathbb{M}$ *on a derivation* ρ, written $\mathbb{M}^\star(\rho)$, is a potentially infinite multiderivation defined as $\mathbb{M}^\star(\rho) \stackrel{\text{def}}{=} \mathcal{M}_1\mathcal{M}_2\ldots\mathcal{M}_\mathsf{n}\ldots$. The elements $\mathcal{M}_\mathsf{n} \in \mathit{Stp}^+ \cup \{\bot\}$ are defined by induction on n for every $n \in \mathbb{N}$ as follows:

$$\mathcal{M}_\mathsf{n} \stackrel{\text{def}}{=} \begin{cases} \mathbb{M}(\rho/\mathcal{M}_1\ldots\mathcal{M}_{\mathsf{n}-1}) & \text{if } \mathcal{M}_\mathsf{i} \neq \bot \text{ for all } i < n \text{ and } \rho/\mathcal{M}_1\ldots\mathcal{M}_{\mathsf{n}-1} \neq \epsilon \\ \bot & \text{otherwise} \end{cases}$$

Intuitively, $\mathbb{M}^\star(\rho)$ is given by successively applying $\mathbb{M}$. The next result clarifies this intuition.

▶ Proposition 68 (Properties of the sequence induced by a multi-selection strategy). If $\mathbb{M}$ is a multi-selection strategy in any orthogonal ARS, the following property holds:

$$\mathbb{M}^\star(\rho) = \begin{cases} \epsilon & \text{if } \rho = \epsilon \\ \mathbb{M}(\rho)\,\mathbb{M}^\star(\rho/\mathbb{M}(\rho)) & \text{otherwise} \end{cases}$$

Proof. We omit this proof which is technical but straightforward. The proof consists in checking, by complete induction on n, that the n-th step of the derivation at the left-hand side of the equation coincides with the n-th step of the derivation at the right-hand side. ◀

▶ **Lemma 69** (Projection does not create families in a DFS)**.** *Let F be a DFS, let $\phi : t \twoheadrightarrow t'$ be a derivation in F, and let ρ and σ be coinitial derivations in F starting from t'. Then the set of families of redexes contracted along ρ/σ is contained in the set of families of redexes contracted along ρ, relatively to the history ϕ. More precisely, if ρ/σ can be written as $\tau_1\mathsf{T}\tau_2$ then ρ can be written as $\upsilon_1\mathsf{U}\upsilon_2$ such that $\mathsf{Fam}_\simeq(\phi\upsilon_1\mathsf{U}) = \mathsf{Fam}_\simeq(\phi\sigma\tau_1\mathsf{T})$.*

Proof. By induction on the length of ρ. The base case is trivial. If $\rho = \mathsf{R}\rho'$ we have that $\rho/\sigma = (\mathsf{R}/\sigma)(\rho'/(\sigma/\mathsf{R}))$ by definition. Let ρ/σ be written as $\tau_1\mathsf{T}\tau_2$. We consider two subcases, depending on whether τ_1 is a proper prefix of R/σ or not:

1. **If τ_1 is a proper prefix of R/σ.** Then $\mathsf{R}/\sigma = \tau_1\mathsf{T}\tau_2'$ and $\tau_2 = \tau_2'(\rho'/(\sigma/\mathsf{R}))$. Note that $\mathsf{T} \in (\mathsf{R}/\sigma)/\tau_1$ so $\mathsf{R}\ \langle\sigma\tau_1\rangle\ \mathsf{T}$. Then by taking $\upsilon_1 := \epsilon$, $\mathsf{U} := \mathsf{R}$ and $\upsilon_2 := \rho'$ we have that $\mathsf{Fam}_\simeq(\phi\mathsf{R}) = \mathsf{Fam}_\simeq(\phi\sigma\tau_1\mathsf{T})$ since T is a copy of R, and as a consequence of the Copy axiom.
2. **If τ_1 is not a proper prefix of R/σ.** Then $\rho'/(\sigma/\mathsf{R}) = \tau_1'\mathsf{T}\tau_2$ and $\tau_1 = (\mathsf{R}/\sigma)\tau_1'$. By *i.h.* on the derivation ρ' (using $\phi\mathsf{R}$ as the new history), we conclude that ρ' can be written as $\upsilon_1'\mathsf{U}\upsilon_2$ in such a way that:

$$\begin{array}{rcll} \mathsf{Fam}_\simeq(\phi\mathsf{R}\upsilon_1'\mathsf{U}) & = & \mathsf{Fam}_\simeq(\phi\mathsf{R}(\sigma/\mathsf{R})\tau_1'\mathsf{T}) & \\ & = & \mathsf{Fam}_\simeq(\phi\sigma(\mathsf{R}/\sigma)\tau_1'\mathsf{T}) & \text{by the Copy axiom,} \\ & & & \text{since } \phi\mathsf{R}(\sigma/\mathsf{R})\tau_1'\mathsf{T} \leq \phi\sigma(\mathsf{R}/\sigma)\tau_1'\mathsf{T} \\ & & & \text{since } \phi\mathsf{R}(\sigma/\mathsf{R})\tau_1' \equiv \phi\sigma(\mathsf{R}/\sigma)\tau_1' \end{array}$$

   Hence by taking $\upsilon_1 := \mathsf{R}\upsilon_1'$ we conclude.

◀

The following lemma shows that the process of successively applying a multi-selection strategy $\mathbb{M}$ to build an induced sequence $\mathbb{M}^\star(\rho)$ terminates, as long as the input ρ is finite.

▶ **Lemma 70** (Induced multiderivations preserve finiteness)**.** *Suppose that $\mathbb{M}$ is a multi-selection strategy in a DFS. If ρ is finite, then $\mathbb{M}^\star(\rho)$ is also finite.*

Proof. Let ρ be a finite derivation, let $\mathsf{D} = \mathbb{M}^\star(\rho)$ be the multiderivation induced by $\mathbb{M}$ on ρ, and let $\mathcal{F}$ be the set of redex families that are contracted along ρ, more precisely:

$$\mathcal{F} \stackrel{\text{def}}{=} \{\mathsf{Fam}_{\simeq}(\rho_1\mathsf{R}) \mid \exists\rho_2.\ \rho = \rho_1\mathsf{R}\rho_2\}$$

Claim: Let us write D as a potentially infinite sequence of multisteps $\mathsf{D} = \mathcal{M}_1 \ldots \mathcal{M}_n \ldots$. Suppose that $\sigma = \sigma_1 \ldots \sigma_n$ is any complete development of a prefix $\mathcal{M}_1 \ldots \mathcal{M}_n$ of D, where each σ_i is a complete development of $\mathcal{M}_i$. Then the set of families of the redexes contracted along σ is contained in $\mathcal{F}$.

Proof of the claim. Let $\sigma = \sigma_1 \ldots \sigma_n$ and let $\sigma_i = \mathsf{S}^i_1 \ldots \mathsf{S}^i_{m_i}$ for each $1 \leq i \leq n$. An arbitrary step of σ is one of the steps S^i_j with $1 \leq i \leq n$ and $1 \leq j \leq m_i$. It suffices to show that the family of each S^i_j is in $\mathcal{F}$. More precisely, we aim to show that $\mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\mathsf{S}^i_1 \ldots \mathsf{S}^i_{j-1}\mathsf{S}^i_j) \in \mathcal{F}$ holds for every i, j.

Let $1 \leq i \leq n$ and $1 \leq j \leq m_i$ be arbitrary indices. Note that S^i_j is a redex in σ_i and σ_i is a complete deveopment of $\mathcal{M}_i$, so S^i_j has an ancestor $\mathsf{S}^\star\ \langle \mathsf{S}^i_1 \ldots \mathsf{S}^i_{j-1}\rangle\ \mathsf{S}^i_j$ with $\mathsf{S}^\star \in \mathcal{M}_i$. This means that S^i_j is a copy of $\mathsf{S}^\star$, hence they are in the same family:

$$\mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\mathsf{S}^i_1 \ldots \mathsf{S}^i_j) = \mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\mathsf{S}^\star)$$

Moreover, note that, by construction, $\mathcal{M}_i = \mathbb{M}(\rho/\mathcal{M}_1 \ldots \mathcal{M}_{i-1})$. Since $\mathbb{M}$ is a multi-selection strategy, we have that $\mathsf{S}^\star \lhd \rho/\mathcal{M}_1 \ldots \mathcal{M}_{i-1}$. This means that $\rho/\mathcal{M}_1 \ldots \mathcal{M}_{i-1}$ can be written as $\rho_1\mathsf{S}^{\star\star}\rho_2$ where $\mathsf{S}^\star\ \langle\rho_1\rangle\ \mathsf{S}^{\star\star}$. This means that $\mathsf{S}^{\star\star}$ is a copy of $\mathsf{S}^\star$, hence they are in the same family: $\mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\mathsf{S}^\star) = \mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\rho_1\mathsf{S}^{\star\star})$. Moreover, since projection does not create families in a DFS (Lem. 69) and $\rho/\mathcal{M}_1 \ldots \mathcal{M}_{i-1} = \rho/\sigma_1 \ldots \sigma_{i-1} = \rho_1\mathsf{S}^{\star\star}\rho_2$ we have that $\mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\rho_1\mathsf{S}^{\star\star}) \in \mathcal{F}$. To conclude the proof of the claim, collecting all the facts we have already established above: $\mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\mathsf{S}^i_1 \ldots \mathsf{S}^i_j) = \mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\mathsf{S}^\star) = \mathsf{Fam}_{\simeq}(\sigma_1 \ldots \sigma_{i-1}\rho_1\mathsf{S}^{\star\star}) \in \mathcal{F}$.

To conclude the proof, note that the set $\mathcal{F}$ is finite since ρ is finite. By FFD, this implies that there cannot be infinite derivations contracting redexes whose family is in $\mathcal{F}$. Therefore D must be finite. Suppose otherwise, *i.e.* suppose that D is an infinite multiderivation of the form $\mathcal{M}_1 \ldots \mathcal{M}_n \ldots$. Construct an infinite derivation $\sigma = \sigma_1 \ldots \sigma_n \ldots$ where σ_i is a complete development of $\mathcal{F}$. By applying the previous claim on each finite prefix of D, we obtain that the families of all the redexes in σ are in $\mathcal{F}$, which is a contradiction. ◀

▶ Remark. If $\rho \equiv \epsilon$ then $\rho = \epsilon$.

By definition, a **uniform** multi-selection strategy $\mathbb{M}$, when given two permutation equivalent derivations, always selects the same multistep. In the following lemma, we show that this property can be strengthened to show that this also holds for the **multiderivations** induced by the strategy.

The following lemma corresponds to **Lem. 11** in the main body.

▶ **Lemma 71** (Induced multiderivations modulo permutations). *Let $\mathbb{M}$ be a uniform multi-selection strategy in a DFS, and let ρ, σ be finite derivations. If $\rho \equiv \sigma$, then $\mathbb{M}^\star(\rho) = \mathbb{M}^\star(\sigma)$.*

Proof. By the fact that the process of building an induced sequence terminates (*i.e.* multi-selection strategies preserve finiteness, as shown in Lem. 70) we know that $\mathbb{M}^\star(\rho)$ must be finite. Moreover, we will use the equation Prop. 68 characterizing the shape of $\mathbb{M}^\star(\rho)$, as if it were given as a case-by-case definition. We proceed by induction on the length of $\mathbb{M}^\star(\rho)$:

1. **Empty, $\mathbb{M}^\star(\rho) = \epsilon$.** Then by Prop. 68, we have that $\rho = \epsilon$. Since $\rho \equiv \sigma$, by the Rem. A.5, it must be the case that $\sigma = \epsilon$. So $\mathbb{M}^\star(\rho) = \epsilon = \mathbb{M}^\star(\sigma)$.

2. **Non-empty.** Then by Prop. 68, we have that ρ is non-empty. Since $\rho \equiv \sigma$, by the Rem. A.5, σ must be also non-empty. By Prop. 68 we have:

$$\mathbb{M}^\star(\rho) = \mathbb{M}(\rho)\,\mathbb{M}^\star(\rho/\mathbb{M}(\rho)) \quad \mathbb{M}^\star(\sigma) = \mathbb{M}(\sigma)\,\mathbb{M}^\star(\sigma/\mathbb{M}(\sigma))$$

Recall that we aim to show that $\mathbb{M}^\star(\rho) = \mathbb{M}^\star(\sigma)$. First note that, since $\rho \equiv \sigma$ and $\mathbb{S}$ is a uniform selection strategy, we have $\mathbb{M}(\rho) = \mathbb{M}(\sigma)$.
Note that the tail of $\mathbb{M}^\star(\rho)$ is of the form $\mathbb{M}^\star(\rho/\mathbb{M}(\rho))$, and it is strictly shorter than $\mathbb{M}^\star(\rho)$. So we can apply the *i.h.* on the tails of $\mathbb{M}^\star(\rho)$ and $\mathbb{M}^\star(\sigma)$. The *i.h.* states:

$$\rho/\mathbb{M}(\rho) \equiv \sigma/\mathbb{M}(\sigma) \implies \mathbb{M}^\star(\rho/\mathbb{M}(\rho)) = \mathbb{M}^\star(\sigma/\mathbb{M}(\sigma))$$

To conclude, we are left to show that $\rho/\mathbb{M}(\rho) \equiv \sigma/\mathbb{M}(\sigma)$ holds. This is an immediate consequence of the fact that $\rho \equiv \sigma$, which is known by hypothesis, as we have already argued that $\mathbb{M}(\rho) = \mathbb{M}(\sigma)$, and the residuals of equivalent derivations are again equivalent (*cf.* [22]).

◀

▶ **Definition 72** ($\mathbb{M}$ applied to a multiderivation)**.** If $\mathbb{M}$ is a uniform multi-selection strategy, given a multi-derivation D we write $\mathbb{M}^\star(\mathsf{D})$ to stand for $\mathbb{M}^\star(\rho)$, where ρ is any complete development of D. This is well-defined by virtue of the previous lemma (Lem. 70) since any two complete developments ρ, ρ' of D are permutation equivalent, hence $\mathbb{M}^\star(\rho) = \mathbb{M}^\star(\rho')$.

▶ **Definition 73** ($\mathbb{M}$-compliant derivations)**.** Let $\mathbb{M}$ be a multi-selection strategy. A multiderivation D is said to be $\mathbb{M}$-compliant if and only if $\mathbb{M}^\star(\partial\mathsf{D}) = \mathsf{D}$.

The following lemma corresponds to **Lem. 12** in the main body.

▶ **Lemma 74** (Induced sequences are permutations of the sequences)**.** *Let $\mathbb{M}$ be a multi-selection strategy in a DFS, and let ρ be a finite derivation. Then $\rho \equiv \partial\mathbb{M}^\star(\rho)$.*

Proof. By Lem. 70, we have that $\mathbb{M}^\star(\rho)$ must be finite. We proceed by induction on the length of the multiderivation $\mathbb{M}^\star(\rho)$.

1. **Empty,** $\mathbb{M}^\star(\rho) = \epsilon$**.** Then by Prop. 68, we have that $\rho = \epsilon$, so $\rho = \partial\epsilon = \partial\mathbb{M}^\star(\rho)$.
2. **Non-empty.** Let $\mathcal{M} = \mathbb{M}(\rho)$ be the first multistep selected by the strategy. By Prop. 68, we have that $\mathbb{M}^\star(\rho) = \mathcal{M}\,\mathbb{M}^\star(\rho/\mathcal{M})$. To show that $\rho \equiv \partial\mathbb{M}^\star(\rho)$, let us check that they are projection equivalent, *i.e.* that $\rho/\partial\mathbb{M}^\star(\rho) = \epsilon$ and $\partial\mathbb{M}^\star(\rho)/\rho = \epsilon$.
   a. **Check that** $\rho \sqsubseteq \partial\mathbb{M}^\star(\rho)$**.**

$$\begin{array}{cll} & \rho/\partial\mathbb{M}^\star(\rho) & \\ = & \rho/\partial(\mathcal{M}\,\mathbb{M}^\star(\rho/\mathcal{M})) & \text{by Prop. 68} \\ = & \rho/(\partial\mathcal{M})\,(\partial\mathbb{M}^\star(\rho/\mathcal{M})) & \\ = & (\rho/\partial\mathcal{M})/\partial\mathbb{M}^\star(\rho/\mathcal{M}) & \text{since in general } \alpha/\beta\gamma = (\alpha/\beta)/\gamma \\ = & \epsilon & \text{since by } \textit{i.h. } \rho/\mathcal{M} \equiv \partial\mathbb{M}^\star(\rho/\mathcal{M}) \end{array}$$

   b. **Check that** $\partial\mathbb{M}^\star(\rho) \sqsubseteq \rho$**.** Since $\mathbb{M}$ is a multi-selection strategy, we have that $\mathcal{M}/\rho = \emptyset$. Hence:

$$\begin{array}{cll} & (\partial\mathbb{M}^\star(\rho))/\rho & \\ = & (\partial(\mathcal{M}\,\mathbb{M}^\star(\rho/\mathcal{M})))/\rho & \text{by Prop. 68} \\ = & (\partial\mathcal{M})\,(\partial\mathbb{M}^\star(\rho/\mathcal{M}))/\rho & \\ = & ((\partial\mathcal{M})/\rho)\,((\partial\mathbb{M}^\star(\rho/\mathcal{M}))/(\rho/\partial\mathcal{M})) & \text{since in general } \alpha\beta/\gamma = (\alpha/\beta)(\gamma/(\beta/\alpha)) \\ = & (\partial\mathbb{M}^\star(\rho/\mathcal{M}))/(\rho/\partial\mathcal{M}) & \text{since } \mathcal{M}/\rho = \emptyset\text{, so } (\partial\mathcal{M})/\rho = \epsilon \\ = & (\partial\mathbb{M}^\star(\rho/\mathcal{M}))/(\rho/\mathcal{M}) & \text{since } \rho/\mathcal{M} \text{ stands for } \rho/\partial\mathcal{M} \\ = & \epsilon & \text{since by } \textit{i.h. } \rho/\mathcal{M} \equiv \partial\mathbb{M}^\star(\rho/\mathcal{M}) \end{array}$$

◀

The following proposition corresponds to **Prop. 13** in the main body:

▶ Proposition 75 (Uniform selection defines a standardisation procedure). Let $\mathbb{M}$ be a uniform multi-selection strategy in a DFS. For any finite derivation ρ there exists a unique multiderivation D such that $\rho \equiv \partial\mathsf{D}$ and D is $\mathbb{M}$-compliant. Namely, $\mathsf{D} = \mathbb{M}^\star(\rho)$.

Proof. We prove the result in two parts:

Existence. First note that $\rho \equiv \partial\mathbb{M}^\star(\rho)$ by Lem. 74. To see that $\mathbb{M}^\star(\rho)$ is $\mathbb{M}$-compliant, apply Lem. 71 on the fact that $\partial\mathbb{M}^\star(\rho) \equiv \rho$ to conclude that $\mathbb{M}^\star(\partial\mathbb{M}^\star(\rho)) = \mathbb{M}^\star(\rho)$. This is exactly the definition of $\mathbb{M}^\star(\rho)$ being $\mathbb{M}$-compliant.

Uniqueness. Suppose that there is a multiderivation E such that $\rho \equiv \partial\mathsf{E}$ and E is $\mathbb{M}$-compliant. We claim that $\mathsf{E} = \mathbb{M}^\star(\rho)$. By applying Lem. 71 on the fact that $\partial\mathsf{E} \equiv \rho$, we obtain that $\mathbb{M}^\star(\partial\mathsf{E}) = \mathbb{M}^\star(\rho)$. Finally, since E is $\mathbb{M}$-compliant, $\mathsf{E} = \mathbb{M}^\star(\partial\mathsf{E}) = \mathbb{M}^\star(\rho)$ and we conclude. ◀

A.6 Standardisation by arbitrary selection in LSC

In this subsection we apply the previous standardisation result for LSC.

▶ **Definition 76** (The arbitrary selector). For each term t let $\mathsf{Out}(t) = \{\mathsf{R} \mid \mathsf{src}(\mathsf{R}) = t\}$ be the set of steps outgoing from t in the LSC (without gc), and let $<_t$ be an arbitrary strict partial order on $\mathsf{Out}(t)$. The *arbitrary selector* $\mathbb{M}_<$ is a function that takes a non-empty sequence ρ and yields a non-empty set of coinitial steps, defined as follows:

$$\mathbb{M}_<(\rho) \stackrel{\text{def}}{=} \{\mathsf{R} \mid \mathsf{R}/\rho = \emptyset \text{ and } \mathsf{R} \text{ is minimal with respect to } <_t\}$$

By *minimal* we mean that there is no R' such that $\mathsf{R}'/\rho = \emptyset$ and $\mathsf{R}' <_t \mathsf{R}$.

To see that the function $\mathbb{M}_<$ is well defined, we must check that the set $\mathbb{M}_<(\rho)$ is non-empty. Consider an arbitrary reduction non-empty sequence ρ, and note that the set $A_\rho = \{\mathsf{R} \mid \mathsf{R}/\rho = \emptyset\}$ is non-empty and finite:

1. **The set A_ρ is non-empty.** The sequence ρ is of the form $\mathsf{R}\rho'$ since it is non-empty, so $\mathsf{R}/\rho = \mathsf{R}/(\mathsf{R}\rho') = \emptyset$, hence $\mathsf{R} \in A_\rho$.
2. **The set A_ρ is finite.** Note that A_ρ is contained in $\mathsf{Out}(\mathsf{src}(\rho))$, which is a finite set by the fact that the LSC is finitely-branching.

Hence A_ρ has at least one minimal element.

▶ Remark. If the order $<_t$ is computable, the arbitrary selector function $\mathbb{M}_<$ is also computable. This is because there are finitely many steps R in the set $\mathsf{Out}(\mathsf{src}(\rho))$, and checking whether $\mathsf{R}/\rho = \emptyset$ is a decidable property.

▶ **Lemma 77.** *Let $\mathcal{M}$ be a multistep in the LSC without* gc.
If $\mathcal{M}/\rho = \emptyset$ then $\mathcal{M} \lhd \rho$.

Proof. By induction on ρ:

1. **Empty, $\rho = \epsilon$.** Immediate, since $\mathcal{M}/\rho = \mathcal{M} = \emptyset$, hence $\mathcal{M} \lhd \rho$.
2. **Non-empty, $\rho = \mathsf{S}\sigma$.** Let $\mathsf{R} \in \mathcal{M}$ and let us show that $\mathsf{R} \lhd \rho$. Two cases, depending on whether $\mathsf{R} = \mathsf{S}$:
   a. **If $\mathsf{R} = \mathsf{S}$.** Then $\mathsf{R} \lhd \mathsf{S}\sigma$, so we are done.

b. **If** $\mathsf{R} \neq \mathsf{S}$**.** Then there must be a residual $\mathsf{R}' \in \mathcal{M}/\mathsf{S}$, since we are in the calculus without gc and S cannot erase R (this is a consequence of Prop. 25). Since $(\mathcal{M}/\mathsf{S})/\rho = \emptyset$ we have by *i.h.* that $\mathcal{M}/\mathsf{S} \lhd \rho$ which means that $\mathsf{R}' \lhd \rho$. This in turn implies that $\mathsf{R} \lhd \mathsf{S}\sigma$, as required.

◀

The following lemma corresponds to **Lem. 14** in the main body.

▶ **Lemma 78** (The arbitrary selector is a uniform multi-selection strategy)**.** *Let* $<_t$ *be an arbitrary strict partial order on the set* $\mathsf{Out}(t)$ *of outgoing steps for each term* t*.*
Then $\mathbb{M}_<$ *is a uniform selection strategy.*

Proof. Let us check that $\mathbb{M}_<$ is a multi-selection strategy and that it is uniform:

1. $\mathbb{M}_<$ **is a multi-selection strategy.** Let ρ be a non-empty reduction sequence. Recall that a function $\mathbb{M}$ is a selection strategy if $\mathbb{M}(\rho)$ is a non-empty multistep $\mathcal{M}$ coinitial to ρ such that $\mathcal{M}/\rho = \emptyset$ and $\mathcal{M} \lhd \rho$.
   In our case, we have constructed $\mathbb{M}_<(\rho)$ to be a non-empty multistep coinitial to ρ (Def. 76). Moreover, also by construction, any step $\mathsf{R} \in \mathbb{M}_<(\rho)$ verifies $\mathsf{R}/\rho = \emptyset$, so indeed $\mathbb{M}_<(\rho)/\rho = \emptyset$. Moreover, since we are in the LSC without gc, by Lem. 77 we have that $\mathbb{M}_<(\rho) \lhd \rho$, as required.
2. $\mathbb{M}_<$ **is uniform.** Let $\rho \equiv \sigma$, and let us check that $\mathbb{M}_<(\rho) = \mathbb{M}_<(\sigma)$. It suffices to show that the set $A_\rho = \{\mathsf{R} \mid \mathsf{R}/\rho = \emptyset\}$ coincides with the set $A_\sigma = \{\mathsf{R} \mid \mathsf{R}/\sigma = \emptyset\}$, since $\mathbb{M}_<(\rho)$ is the subset of the minimal elements of A_ρ and $\mathbb{M}_<(\sigma)$ is the subset of the minimal elements of A_σ.
   Note that:
   $$\begin{array}{rcll} \mathsf{R} \in A_\rho & \iff & \mathsf{R}/\rho = \emptyset & \\ & \iff & \mathsf{R}/\sigma = \emptyset & \text{since } \rho \equiv \sigma \\ & \iff & \mathsf{R} \in A_\sigma & \end{array}$$
   So $A_\rho = A_\sigma$, as wanted.

◀

The following corollary corresponds to **Coro. 15** in the main body.

▶ **Corollary 79** (Standardisation by arbitrary selection for the LSC without gc)**.** *Let* $<_t$ *be an arbitrary strict partial order on the set* $\mathsf{Out}(t)$ *of outgoing steps for each term* t*.*
Then for each finite sequence ρ *in the LSC without* gc*, there is a unique finite sequence* σ *such that* $\rho \equiv \sigma$ *and* σ *is* $\mathbb{M}_<$*-compliant.*
Moreover, if the order $<_t$ *is computable, then* σ *is computable from* ρ*, namely* $\sigma = \mathbb{M}^\star_<(\rho)$*.*

Proof. A direct consequence of the general fact that uniform selection strategies define a standardisation procedure for finite sequences (Prop. 75) and the particular fact that the arbitrary selector function $\mathbb{M}_<$ is a uniform selection strategy for the LSC without gc (Lem. 78).

See Rem. A.6 for the remark on the computability of $\mathbb{M}_<$. ◀

A.7 Applications of FFD – Normalisation of strategies

A.7.1 Normalisation in DFSs

▶ **Definition 80** (X-normalizing strategy)**.** Let X be a superset of the normal forms of a DRS $\mathcal{A}$. A strategy $\mathbb{S}$ is said to be X*-normalizing* if for every term t such that there exists a reduction $t \twoheadrightarrow_\mathcal{A} s \in X$, every maximal reduction from t in the strategy $\mathbb{S}$ contains a term in X.

▶ **Lemma 81** (Steps of residual-invariant sub-ARSs are preserved in DFSs)**.** *Let $F = (\mathcal{A}, \simeq, \hookrightarrow)$ be a DFS and suppose that $\mathcal{B}$ is a residual-invariant sub-ARS of $\mathcal{A}$. Let $\rho\mathsf{R}$ be a redex with history such that R is in $\mathcal{B}$, and let σ be any finite reduction coinitial to R. Let us also suppose that σ does not contract redexes in the family of $\rho\mathsf{R}$. More precisely, let us suppose that whenever σ can be written as $\sigma_1\mathsf{S}\sigma_2$ then $\rho\mathsf{R} \not\simeq \rho\sigma_1\mathsf{S}$. Then R has a residual $\mathsf{R}' \in \mathsf{R}/\sigma$ in $\mathcal{B}$.*

Proof. By induction on σ. If σ is empty it is immediate. If $\sigma = \mathsf{T}\,\tau$, we know that $\rho\mathsf{R} \not\simeq \rho\mathsf{T}$ by hypothesis, so in particular $\mathsf{R} \neq \mathsf{T}$. Since $\mathcal{B}$ is residual-invariant this means that there exists a step $\mathsf{R}' \in \mathcal{B}$ such that $\mathsf{R}\,\langle\mathsf{T}\rangle\,\mathsf{R}'$. Moreover, by the Copy axiom $\rho\mathsf{R} \simeq \rho\mathsf{T}\mathsf{R}'$. By *i.h.* on the derivation τ and the redex with history $\rho\mathsf{T}\mathsf{R}'$ we conclude that there is a step $\mathsf{R}'' \in \mathcal{B}$ such that $\mathsf{R}'\,\langle\tau\rangle\,\mathsf{R}''$. So $\mathsf{R}\,\langle\mathsf{T}\tau\rangle\,\mathsf{R}''$ and we are done. ◀

The following proposition corresponds to **Prop. 16** in the main body.

▶ Proposition 82 (Closed residual-invariant sub-ARSs are X-normalizing in a DFS). Let $\mathcal{B}$ be a closed residual-invariant sub-ARS in a DFS $F = (\mathcal{A}, \simeq, \hookrightarrow)$. Then the corresponding strategy $\mathbb{S}_{\mathcal{B}}$ is $\mathsf{NF}(\mathcal{B})$-normalizing.

Proof. Let ρ_1 be a derivation $t \twoheadrightarrow_{\mathcal{A}} s \in \mathsf{NF}(\mathcal{B})$ and consider a maximal derivation σ starting from t in the strategy $\mathbb{S}_{\mathcal{B}}$. We must show that σ contains a term in $\mathsf{NF}(\mathcal{B})$. Let $\mathcal{F}$ be the set of families of all the redexes contracted along ρ_1. The set $\mathcal{F}$ is finite, so by the FFD axiom, the derivation ρ_1 can be extended to a complete family development $\rho_1\rho_2$ of $\mathcal{F}$.

By contradiction, suppose that the reduction sequence σ has no terms in $\mathsf{NF}(\mathcal{B})$. Then σ is contained in the sub-ARS $\mathcal{B}$, and it is infinite. By the FFD axiom, σ cannot be a family development of $\mathcal{F}$, so there must be at least one redex whose family is not in $\mathcal{F}$. Let S be the first such step, *i.e.* let us write σ as $\sigma_1\mathsf{S}\sigma_2$ where σ_1 is a family development of $\mathcal{F}$ and $\mathsf{Fam}_{\simeq}(\sigma) \notin \mathcal{F}$. The situation is as follows, closing the square with the derivations $\rho_1\rho_2/\sigma_1$ and $\sigma_1/\rho_1\rho_2$:

$$\begin{tikzcd}
{} \arrow[r, two heads, "\rho_1"] \arrow[d, two heads, "\sigma_1"'] & {} \arrow[r, two heads, "\rho_2"] & {} \arrow[d, dotted, two heads, "\sigma_1/\rho_1\rho_2 \;=\; \tau"] \\
{} \arrow[rr, dotted, two heads, "\rho_1\rho_2/\sigma_1"'] \arrow[d, "\mathsf{S}"'] & & {} \\
{} \arrow[d, two heads, "\sigma_2"'] & & \\
{} & &
\end{tikzcd}$$

First we claim that the derivation $\tau = \sigma_1/\rho_1\rho_2$ is actually empty. Indeed, by the Copy axiom the families of all the redexes contracted along $\sigma_1/\rho_1\rho_2$ are containted in the families of all the redexes contracted along σ_1. In particular, ${}_{\rho_1\rho_2}|\tau$ is a family development of $\mathcal{F}$. If τ were not empty, it would mean that $\tau = \mathsf{T}\tau'$, where $\mathsf{Fam}_{\simeq}(\rho_1\rho_2\mathsf{T}) \in \mathcal{F}$. This contradict the fact that $\rho_1\rho_2$ is a complete family development, as it can be extended with T, so T is indeed empty. This means that the diagram is as follows:

$$\begin{tikzcd}
{} \arrow[r, two heads, "\rho_1"] \arrow[d, two heads, "\sigma_1"'] & {} \arrow[r, two heads, "\rho_2"] & {} \\
{} \arrow[urr, dotted, two heads, "\rho_1\rho_2/\sigma_1"'] \arrow[d, "\mathsf{S}"'] & & \\
{} \arrow[d, two heads, "\sigma_2"'] & & \\
{} & &
\end{tikzcd}$$

By the Copy axiom, we know that the families of all the redexes contracted along $\rho_1\rho_2/\sigma_1$ are contained in the families of all the redexes contracted along $\rho_1\rho_2$. In particular, ${}_{\sigma_1}|\rho_1\rho_2/\sigma_1$ is a family development of $\mathcal{F}$.

By Lem. 81, we obtain that there must exist a step $\mathsf{S}' \in \mathcal{B}$ such that $\mathsf{S}\ \langle \rho_1\rho_2/\sigma_1 \rangle\ \mathsf{S}'$. To be able to apply Lem. 81 note that:

- S is a step in the sub-ARS $\mathcal{B}$;
- by hypothesis, the sub-ARS $\mathcal{B}$ is residual-invariant;
- the derivation $\rho_1\rho_2/\sigma_1$ does not contract redexes in the family of $\sigma_1\mathsf{S}$ since $\mathsf{Fam}_{\simeq}(\sigma_1\mathsf{S}) \notin \mathcal{F}$ while ${}_{\sigma_1}|\rho_1\rho_2/\sigma_1$ is a family development of $\mathcal{F}$.

So the situation is:

$$\begin{tikzcd}
{} \arrow[r, two heads, "\rho_1"] \arrow[d, two heads, "\sigma_1"'] & {} \arrow[r, two heads, "\rho_2"] & {} \arrow[d, "\mathsf{S}'"] \\
{} \arrow[urr, two heads, dotted, "\rho_1\rho_2/\sigma_1"'] \arrow[d, "\mathsf{S}"'] & & {} \\
{} \arrow[d, two heads, "\sigma_2"'] & & \\
{} & &
\end{tikzcd}$$

Finally, recall that $\mathsf{tgt}(\rho_1) \in \mathsf{NF}(\mathcal{B})$, and that, by hypothesis, $\mathcal{B}$ is *closed* residual-invariant, which means that the set $\mathsf{NF}(\mathcal{B})$ is closed by reduction. So $\mathsf{tgt}(\rho_2) \in \mathsf{NF}(\mathcal{B})$, contradicting the fact that there is an outgoing step S' in the sub-ARS $\mathcal{B}$. We conclude that σ must be have a term in $\mathsf{NF}(\mathcal{B})$, as required. ◀

A.7.2 Normalisation for linear call-by-need

▶ **Definition 83** (Needed linear reduction and needed linear normal forms). *Needed linear reduction* the LSC is the sub-ARS $\mathbb{NL}$ defined as follows. Need contexts are defined by the grammar:

$$\mathtt{N} ::= \Box \mid \mathtt{N}\,t \mid \mathtt{N}[x/t] \mid \mathtt{N}\langle\!\langle x\rangle\!\rangle[x/\mathtt{N}]$$

The reduction rule $\to_{\mathbb{NL}}$ is defined as the union of the usual db rule, and the lsnl rule introduced below, both closed by need contexts:

$$(\lambda x.t)\mathtt{L}\,s \to_{\mathsf{db}} t[x/s]\mathtt{L} \qquad\qquad \mathtt{N}\langle\!\langle x\rangle\!\rangle[x/\mathtt{vL}] \to_{\mathsf{lsnl}} \mathtt{N}\langle \mathtt{vL}\rangle[x/\mathtt{vL}]$$

where $\mathtt{v}$ stands for a *value*, *i.e.* a term of the form $\lambda y.t$. Note that it is in fact a sub-ARS of LSC, *i.e.* the lsnl rule is a particular case of the ls rule, and closure by need contexts is a particular case of closure by general contexts.

The set of needed linear normal forms NLNF is defined as the set of terms generated by the grammar:

$$A \quad ::= \quad (\lambda x.t)\mathtt{L} \quad \mid \quad \mathtt{N}\langle\!\langle x\rangle\!\rangle \quad \text{where } \mathtt{N} \text{ does not bind } x$$

Terms of the form $(\lambda x.t)\mathtt{L}$ are called *answers*, and terms of the form $\mathtt{N}\langle\!\langle x\rangle\!\rangle$ are called *structures*. The variable x is called the *needed variable* of a structure $\mathtt{N}\langle\!\langle x\rangle\!\rangle$.

▶ **Lemma 84** (Properties of needed contexts). *The following hold:*

1. **Answers have no redexes or variables under need contexts.**
   If $(\lambda x.s)\mathtt{L} = \mathtt{N}\langle\Delta\rangle$ then Δ is not a redex nor a free occurrence of a variable.
2. **Unique needed variable.**
   If $\mathtt{N}_1\langle\!\langle x\rangle\!\rangle = \mathtt{N}_2\langle\!\langle y\rangle\!\rangle$ then $\mathtt{N}_1 = \mathtt{N}_2$.
3. **Erasing a substitution in a need context.**
   If $\mathtt{N}_1\langle \mathtt{N}_2[x/t]\rangle$ is a need context, then $\mathtt{N}_1\langle\mathtt{N}_2\rangle$ is also a need context.
4. **Replacing a term in a need context.**
   If $\widehat{\mathtt{C}}$ is a two-hole context, $\widehat{\mathtt{C}}\langle\Box, t\rangle$ is a need context, and t has no variables bound by $\widehat{\mathtt{C}}$, then $\widehat{\mathtt{C}}\langle\Box, s\rangle$ is also a need context (where s is an arbitrary term).

Proof. Item 1 is by induction on $\mathtt{L}$. Items 2 and 3 are by induction on $\mathtt{N}_1$. Item 4 is by induction on the formation of the need context $\widehat{\mathtt{C}}\langle\Box, t\rangle$. ◀

The following corollary corresponds to **Coro. 17** in the appendix.

▶ **Corollary 85** (Needed linear reduction is NLNF-normalizing)**.** *The strategy $\mathbb{S}_{\mathbb{NL}}$ associated to the sub-ARS $\mathbb{NL}$ is* NLNF*-normalizing.*

Proof. To show that $\mathbb{S}_{\mathbb{NL}}$ is NLNF-normalizing, we will apply Prop. 82 to conclude that $\mathbb{S}_{\mathbb{NL}}$ is NF($\mathbb{NL}$)-normalizing. We must show that:

1. The set NF($\mathbb{NL}$) coincides with the set NLNF, so being NF($\mathbb{NL}$)-normalizing is equivalent to being NLNF-normalizing. For this we will show the two inclusions, **(1a)** NF($\mathbb{NL}$) $\subseteq$ NLNF and **(1b)** NLNF $\subseteq$ NF($\mathbb{NL}$).
2. The sub-ARS $\mathbb{NL}$ is closed residual-invariant, to be able to apply Prop. 82. For this we will show that **(2a)** the set NF($\mathbb{NL}$) is closed by reduction, and **(2b)** the sub-ARS $\mathbb{NL}$ is residual-invariant.

Part 1a: every $\mathbb{NL}$-normal form is a NLNF.
By induction on t it is straightforward to check that if $t \in$ NF($\mathbb{NL}$) then $t \in$ NLNF.

Part 1b: every NLNF is a $\mathbb{NL}$-normal form.
Given $t \in$ NLNF it can be shown that it is a $\mathbb{NL}$-normal form. There are two cases, depending on the shape of t. If t is an answer it is a direct consequence of Lem. 84. If it is a structure, $t = \mathtt{N}\langle\!\langle x\rangle\!\rangle$, then it is straightforward by induction on $\mathtt{N}$.

Part 2a: the set NF($\mathbb{NL}$) is closed by reduction.
By items **(1a)** and **(1b)**, we know that NF($\mathbb{NL}$) = NLNF. Let $t_1 \in$ NLNF and let $t_1 \to t_2$ be an arbitrary step (not necessarily in the strategy). We claim that $t_2 \in$ NLNF. There are two cases, depending on the shape of t_1: if t_1 is an answer $(\lambda x.t)\mathtt{L}$, then by induction on $\mathtt{L}$ it can be seen that t_2 is also an answer. If t_1 is a structure $\mathtt{N}\langle\!\langle x\rangle\!\rangle$, then by induction on $\mathtt{N}$ it can be seen that t_2 is also of the form $\mathtt{N}'\langle\!\langle x\rangle\!\rangle$.

Part 2b: the sub-ARS $\mathbb{NL}$ is residual-invariant.
Let $\mathsf{R} \in \mathbb{NL}$ and consider $\mathsf{R} \neq \mathsf{S}$. Let us show that there is a residual $\mathsf{R}' \in \mathbb{NL} \cap \mathsf{R}/\mathsf{S}$. By induction on the need context $\mathtt{N}$ under which the step R takes place. Most overlappings between redexes R and S are uninteresting, and it is immediate to show that there is a residual $\mathsf{R}' \in \mathsf{R}/\mathsf{S}$ in the strategy, resorting to Lem. 84 when required. Below we deal with the interesting cases:

- **lsnl vs. ls at the root:** let $\widehat{\mathtt{C}}$ be a two-hole context such that $\widehat{\mathtt{C}}\langle\Box, x\rangle$ is a need context. Then:

$$\begin{array}{ccc} \mathtt{N}\langle\widehat{\mathtt{C}}\langle x, x\rangle[x/\mathtt{vL}]\rangle & \xrightarrow{\mathsf{R}} & \mathtt{N}\langle\widehat{\mathtt{C}}\langle \mathtt{vL}, x\rangle[x/\mathtt{vL}]\rangle \\ \mathsf{S}\downarrow & & \\ \mathtt{N}\langle\widehat{\mathtt{C}}\langle x, \mathtt{vL}\rangle[x/\mathtt{vL}]\rangle & \xrightarrow{\mathsf{R}/\mathsf{S}} & \mathtt{N}\langle\widehat{\mathtt{C}}\langle \mathtt{vL}, \mathtt{vL}\rangle[x/\mathtt{vL}]\rangle \end{array}$$

  To conclude that $\mathsf{R}/\mathsf{S} \in \mathbb{NL}$ it suffices to observe that $\widehat{\mathtt{C}}\langle\Box, t\rangle$ is a need context as a consequence of Lem. 84.

- **lsnl vs. db above the variable:** that is, $\mathsf{R} : \mathtt{N}_1\langle\mathtt{N}_2\langle\!\langle x\rangle\!\rangle[x/\mathtt{vL}']\rangle \to \mathtt{N}_1\langle\mathtt{N}_2\langle\mathtt{vL}'\rangle[x/\mathtt{vL}']\rangle$ and $\mathsf{S} : \mathtt{N}_1\langle\mathtt{N}_2'\langle(\lambda y.s)\mathtt{L}\,u\rangle[x/\mathtt{vL}']\rangle \to \mathtt{N}_1\langle\mathtt{N}_2'\langle s[y/u]\mathtt{L}\rangle[x/\mathtt{vL}']\rangle$ such that the context $\mathtt{N}_2'$ is a prefix of the context $\mathtt{N}_2$, *i.e.* $\mathtt{N}_2 = \mathtt{N}_2'\langle\mathtt{N}_2''\rangle$. The variable x must lie somewhere inside the db-redex $(\lambda y.s)\mathtt{L}\,u$, below the need context $\mathtt{N}_2''$. But need contexts do not go below abstractions or to the right of applications, so this case is impossible.

- **lsnl vs. ls above the variable:** let $\widehat{\mathtt{C}}$ be a two-hole context such that $\widehat{\mathtt{C}}\langle\Box, y\rangle$ is a need context. Then:

$$\begin{tikzcd}
\mathtt{N}_1\langle\mathtt{N}_2\langle\widehat{\mathtt{C}}\langle x, y\rangle[y/s]\rangle[x/\mathtt{vL}]\rangle \arrow[r, "\mathsf{R}"] \arrow[d, "\mathsf{S}"'] & \mathtt{N}_1\langle\mathtt{N}_2\langle\widehat{\mathtt{C}}\langle \mathtt{vL}, y\rangle[y/s]\rangle[x/\mathtt{vL}]\rangle \\
\mathtt{N}_1\langle\mathtt{N}_2\langle\widehat{\mathtt{C}}\langle x, s\rangle[y/s]\rangle[x/\mathtt{vL}]\rangle \arrow[r, "\mathsf{R}/\mathsf{S}"] & \mathtt{N}_1\langle\mathtt{N}_2\langle\widehat{\mathtt{C}}\langle \mathtt{vL}, s\rangle[y/s]\rangle[x/\mathtt{vL}]\rangle
\end{tikzcd}$$

To conclude that $\mathsf{R}/\mathsf{S} \in \mathbb{NL}$ it suffices to observe that $\mathtt{N}_2\langle\widehat{\mathtt{C}}\langle\Box, s\rangle[y/s]\rangle$ is a need context as a consequence of Lem. 84.

- **lsnl vs. ls duplicating R on the needed position:** let x be bound to an answer $\mathtt{vL}$ either in $\mathtt{N}_1$ or in $\mathtt{N}_3$. Then:

$$\begin{tikzcd}
\mathtt{N}_1\langle\mathtt{N}_2\langle\!\langle y\rangle\!\rangle[y/\mathtt{N}_3\langle x\rangle]\rangle \arrow[r, "\mathsf{R}"] \arrow[d, "\mathsf{S}"'] & \mathtt{N}_1\langle\mathtt{N}_2\langle\!\langle y\rangle\!\rangle[y/\mathtt{N}_3\langle \mathtt{vL}\rangle]\rangle \\
\mathtt{N}_1\langle\mathtt{N}_2\langle\mathtt{N}_3\langle x\rangle\rangle[y/\mathtt{N}_3\langle x\rangle]\rangle \arrow[r, "\mathsf{R}_1"] & \mathtt{N}_1\langle\mathtt{N}_2\langle\mathtt{N}_3\langle \mathtt{vL}\rangle\rangle[y/\mathtt{N}_3\langle x\rangle]\rangle
\end{tikzcd}$$

Note that R_1 is one of the two residuals of R, and $\mathsf{R}_1 \in \mathbb{NL}$.

- **lsnl vs. ls duplicating R on a non-needed position:** let x be bound to an answer $\mathtt{vL}$ either in $\mathtt{N}_1$ or in $\mathtt{N}_2$, and let $\widehat{\mathtt{C}}$ be a two-hole context such that $\widehat{\mathtt{C}}\langle\Box, y\rangle$ is a need context. Then:

$$\begin{tikzcd}
\mathtt{N}_1\langle\widehat{\mathtt{C}}\langle y, y\rangle[y/\mathtt{N}_2\langle x\rangle]\rangle \arrow[r, "\mathsf{R}"] \arrow[d, "\mathsf{S}"'] & \mathtt{N}_1\langle\widehat{\mathtt{C}}\langle y, y\rangle[y/\mathtt{N}_2\langle \mathtt{vL}\rangle]\rangle \\
\mathtt{N}_1\langle\widehat{\mathtt{C}}\langle y, \mathtt{N}_2\langle x\rangle\rangle[y/\mathtt{N}_2\langle x\rangle]\rangle \arrow[r, "\mathsf{R}_1"] & \mathtt{N}_1\langle\widehat{\mathtt{C}}\langle y, \mathtt{N}_2\langle x\rangle\rangle[y/\mathtt{N}_2\langle \mathtt{vL}\rangle]\rangle
\end{tikzcd}$$

To conclude that $\mathsf{R}_1 \in \mathbb{NL}$ it suffices to observe that $\widehat{\mathtt{C}}\langle\Box, \mathtt{N}_2\langle x\rangle\rangle$ is a need context as a consequence of Lem. 84.

- **lsnl vs. step internal to the argument:** Let $\mathtt{vL} \to t$ be a step. By Part 2a, the set of $\mathbb{S}_{NL}$-normal forms is closed by reduction and, more specifically, the set of answers is closed by reduction. So $t = \mathtt{v'L'}$. Then:

$$\begin{tikzcd}
\mathtt{N}_1\langle\mathtt{N}_2\langle\!\langle x\rangle\!\rangle[x/\mathtt{vL}]\rangle \arrow[r, "\mathsf{R}"] \arrow[d, "\mathsf{S}"'] & \mathtt{N}_1\langle\mathtt{N}_2\langle t\rangle[x/\mathtt{vL}]\rangle \\
\mathtt{N}_1\langle\mathtt{N}_2\langle\!\langle x\rangle\!\rangle[x/\mathtt{v'L'}]\rangle \arrow[r, "\mathsf{R}/\mathsf{S}"] & \mathtt{N}_1\langle\mathtt{N}_2\langle \mathtt{v'L'}\rangle[x/\mathtt{v'L'}]\rangle
\end{tikzcd}$$

◀

A.8 Extraction

Lévy provided three equivalent characterisations of redex families: via labels, zig-zag and extraction. Zig-zag is easy to define since it relies on permutation equivalence. Extraction consists in erasing redexes from the history of a redex with history that do not contribute to it. Defining an appropriate notion of extraction for LSC is non-trivial and has elided our attempts to fully capture it. More precisely, it is confluence of the extraction procedure which seems non-trivial. We next briefly discuss our proposed notion of extraction.

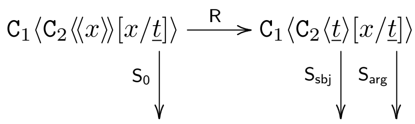
We say ρ **does not duplicate** a coinitial redex S ($\rho \# \sigma$), whenever $\#(\mathsf{S}/\rho) = 1$. We also say ρ does not duplicate a coinitial reduction sequence σ ($\rho \# \sigma$), defined by induction on σ where ϵ denotes the empty reduction: $\rho \# \epsilon$ and $\rho \# \mathsf{S}\sigma$ whenever $\rho \# \mathsf{S}$ and $\rho/\mathsf{S} \# \sigma$. Note that if $\rho \# \sigma$ then σ/ρ has the same length as σ.

Define R to be **internal to a context** $\mathtt{C}$ ($\mathtt{C} \prec \mathsf{R}$) whenever the source of R is of the form $\mathtt{C}\langle t \rangle$ and, moreover:

- If R is a db redex, the position of the hole of $\mathtt{C}$ must be a prefix of the position where the db redex takes place.
- If R is an ls redex, the position of the hole of $\mathtt{C}$ must be a prefix of the position where the variable contracted by the ls redex occurs.

A **derivation** ρ **is internal to a context** $\mathtt{C}$, according to the following inductive definition: $\mathtt{C} \prec \epsilon$ and $\mathtt{C} \prec \mathsf{R}\rho$ whenever $\mathtt{C} \prec \mathsf{R}$ and $\mathtt{C} \prec \rho$. If R is an ls redex and σ is a composable derivation, *i.e.* $\mathsf{tgt}(\mathsf{R}) = \mathsf{src}(\sigma)$, the derivation σ is said to be **internal to the subject (resp. argument)** of R, written $\mathsf{R} \prec_{\mathsf{sbj}} \sigma$ (resp. $\mathsf{R} \prec_{\mathsf{arg}} \sigma$) whenever the redex R is of the form $\mathtt{C}_1\langle \mathtt{C}_2\langle\langle x \rangle\rangle[x/t] \rangle \to \mathtt{C}_1\langle \mathtt{C}_2\langle t \rangle[x/t] \rangle$ and $\mathtt{C}_1\langle \mathtt{C}_2\langle \Box \rangle[x/t] \rangle \prec \sigma$ (resp. $\mathtt{C}_1\langle \mathtt{C}_2\langle t \rangle[x/\Box] \rangle \prec \sigma$).

Note that if $\mathsf{R} \prec_i \mathsf{S}_\mathsf{i}$ for $i \in \{\mathsf{sbj}, \mathsf{arg}\}$, then S_i has an ancestor S_0, that is $\mathsf{S}_\mathsf{i} \in \mathsf{S}_0/\mathsf{R}$. Moreover, S_0/R consists of exactly two redexes, namely $\mathsf{S}_{\mathsf{sbj}}$ and $\mathsf{S}_{\mathsf{arg}}$ such that S_i is internal to i. Also note that S_0 does not duplicate R. We write $\mathsf{S}_\mathsf{i} \longleftarrow\!\!\prec \mathsf{R}$ for S_0, *i.e.* for the ancestor of S_i:

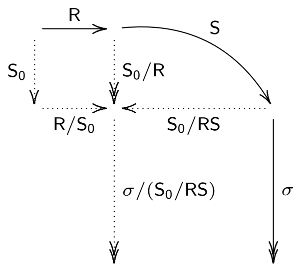


The definition of "$\longleftarrow\!\!\prec$" is also extended for derivations. If $\mathsf{R} \prec_i \sigma$ for $i \in \{\mathsf{sbj}, \mathsf{arg}\}$, the **retraction** of σ before R, written $\sigma \longleftarrow\!\!\prec \mathsf{R}$, is defined inductively as follows:

$$\begin{array}{rcll} \epsilon \longleftarrow\!\!\prec \mathsf{R} & \stackrel{\text{def}}{=} & \epsilon & \\ \mathsf{S}\sigma \longleftarrow\!\!\prec \mathsf{R} & \stackrel{\text{def}}{=} & \mathsf{S}_0\,(\sigma/(\mathsf{S}_0/\mathsf{RS}) \longleftarrow\!\!\prec \mathsf{R}/\mathsf{S}_0) & \text{where } \mathsf{S}_0 = \mathsf{S} \longleftarrow\!\!\prec \mathsf{R} \end{array}$$

This operation is well defined as R/S_0 is a single redex, since, as we have already discussed, S_0 does not duplicate R. To see that the inductive definition is in fact well-defined, it can be checked that $\mathsf{R}/\mathsf{S}_0 \prec_i \sigma/(\mathsf{S}_0/\mathsf{RS})$ and, moreover, that the length of $\sigma/(\mathsf{S}_0/\mathsf{RS})$ coincides with the length of σ, so recursion is well-founded.

The following diagram illustrates the situation:



Finally, we may define an **extraction procedure** as a binary relation $\triangleright$ between redexes with history by the two following rules. We conjecture that it is confluent:

$$\begin{array}{lcll} \rho\mathsf{R}(\sigma/\mathsf{R}) & \triangleright & \rho\sigma & \text{if } \sigma \neq \epsilon \text{ and } \mathsf{R} \# \sigma \\ \rho\mathsf{R}\sigma & \triangleright & \rho(\sigma \longleftarrow\!\!\prec \mathsf{R}) & \text{if } \sigma \neq \epsilon \text{ and } \mathsf{R} \prec_i \sigma \text{ for some } i \in \{\mathsf{sbj}, \mathsf{arg}\} \end{array}$$