Theoretical Computer Science 672 (2017) 36–63

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On abstract normalisation beyond neededness

Eduardo Bonelli [a,b], Delia Kesner [c], Carlos Lombardi [a,*], Alejandro Ríos [d]

[a] Departamento de Ciencia y Tecnología, Univ. Nacional de Quilmes, Roque Sáenz Peña 352 (1876), Bernal, Prov. de Buenos Aires, Argentina
[b] CONICET, Av. Rivadavia 1917 (1033), C.A. Buenos Aires, Argentina
[c] IRIF, CNRS and Univ. Paris-Diderot, Case 7014, 75205 Paris Cedex 13, France
[d] Univ. de Buenos Aires, Pabellón I, Ciudad Universitaria (1428), C.A. Buenos Aires, Argentina

ARTICLE INFO

Article history:
Received 31 December 2014
Received in revised form 30 December 2016
Accepted 30 January 2017
Available online 20 February 2017
Communicated by P.-A. Melliès

Keywords:
Rewriting
Normalisation
Neededness
Sequentiality
Pattern calculi

ABSTRACT

We study normalisation of multistep strategies, strategies that reduce a set of redexes at a time, focusing on the notion of *necessary sets*, those which contain at least one redex that cannot be avoided in order to reach a normal form. This is particularly appealing in the setting of non-sequential rewrite systems, in which terms that are not in normal form may not have any *needed* redex. We first prove a normalisation theorem for abstract rewrite systems (ARS), a general rewriting framework encompassing many rewriting systems developed by P-A. Melliès [20]. The theorem states that multistep strategies reducing so called *necessary* and *never-gripping* sets of redexes at a time are normalising in any ARS. Gripping refers to an abstract property reflecting the behaviour of higher-order substitution. We then apply this result to the particular case of **PPC**, a calculus of patterns and to the lambda-calculus with parallel-or.

1. Introduction

This paper is about computing normal forms in rewrite systems. Consider the λ-calculus. Let K stand for the term $\lambda x.\lambda y.x$, I for $\lambda x.x$ and Ω for $(\lambda x.x\,x)\,(\lambda x.x\,x)$. Then $s := K\,I\,\Omega$ admits an infinite reduction sequence of β-steps, namely the one obtained by repeatedly reducing Ω, and hence s, to itself. However, it also reduces in two β-steps to the normal form I by repeatedly reducing the leftmost–outermost redex:

$$K\,I\,\Omega \underset{\beta}{\to} (\lambda y.I)\,\Omega \underset{\beta}{\to} I \tag{1}$$

The reason this strategy normalises is that the redexes it selects are unavoidable or *needed* in any reduction to normal form. Indeed, leftmost–outermost redexes are needed in λ-calculus [6]. This paper studies normalisation for the broader case of rewriting systems where needed redexes may not exist. It does so by adapting Melliès' abstract rewriting framework [20] to encompass Sekar and Ramakrishnan's notion of *needed sets of redexes* [22]. In doing so, the relatively unfamiliar notion of *gripping*, used only marginally in the work of Melliès, is shown to play a crucial rôle, thus giving it an interest of its own.

Normalisation in TRS. Although in the λ-calculus the leftmost–outermost strategy does indeed attain a normal form (if it exists) [12], the same cannot be said for term rewriting systems (TRS). For example, consider the TRS:

* Corresponding author.
E-mail addresses: eabonelli@gmail.com (E. Bonelli), Delia.Kesner@pps.univ-paris-diderot.fr (D. Kesner), clombardi@unq.edu.ar (C. Lombardi), rios@dc.uba.ar (A. Ríos).

http://dx.doi.org/10.1016/j.tcs.2017.01.025
0304-3975/

$$\begin{aligned} a &\to b \\ c &\to c \\ f(x,b) &\to d \end{aligned}$$

and the term $t := f(c,a)$. The leftmost–outermost strategy selects redex c in t producing an infinite reduction sequence. Yet this term admits a normal form:

$$f(c,a) \to f(c,b) \to d \tag{2}$$

For *left-normal* TRS (those in which the constant and function symbols precede, in the linear term notation, the variable occurrences in the left-hand side of rewrite rules), the leftmost–outermost strategy does indeed normalise [21]; the same applies to left-normal higher-order rewrite systems [19]. Alternatively, one might decide to reduce all *outermost* redexes at once: parallel-outermost reduction is normalising for (almost) orthogonal TRS [21], where an *orthogonal* TRS is one whose rewrite rules are left-linear and non-overlapping, and an *almost orthogonal* TRS has trivial critical pairs at the root, if at all. Parallel-outermost reduction is also normalising for almost orthogonal systems in higher-order rewriting [24].

Needed redexes. There is indeed a deep connection between redexes reduced in (1) and (2): they are unavoidable in the sense that in any reduction sequence from s and t to normal form, they (or their residuals) must be reduced at some point. Such redexes are called *needed* and a theory of needed redexes was developed by Huet and Lévy [14] for orthogonal TRS. In [14] it is shown that in these TRS, terms that are not in normal form always have at least one needed redex and that reduction of needed redexes is normalising. They also showed that determining whether a redex is needed or not is undecidable in general; this led them to study restrictions of the class of orthogonal TRS (the *strongly sequential* TRS) in which needed redexes could be identified effectively.

A fundamental limitation of Huet and Lévy's work is the requirement of orthogonality. This requirement does have its reasons though: in non-orthogonal TRS, terms that are not in normal form may not have needed redexes. A paradigmatic example is the "parallel-or" TRS:

$$\begin{aligned} \mathtt{or}(x,\mathtt{tt}) &\to \mathtt{tt} \\ \mathtt{or}(\mathtt{tt},x) &\to \mathtt{tt} \end{aligned}$$

The term $u := \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{tt}),\mathtt{or}(\mathtt{tt},\mathtt{tt}))$ has four redexes: the occurrence of $\mathtt{or}(\mathtt{tt},\mathtt{tt})$ on the left is an instance of the first and second rules of the parallel-or TRS, and the one on the right is also an instance of both of these rules. None of these is needed since one can always obtain a normal form without reducing it. For example, the reduction sequence:

$$\mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{tt}),\mathtt{or}(\mathtt{tt},\mathtt{tt})) \to \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{tt}),\mathtt{tt}) \to \mathtt{tt} \tag{3}$$

never reduces any of the two redexes on the left. A similar argument applies to the two redexes on the right in u. In fact u seems to suggest that no sensible normalising *strategy*, picking one redex at a time by looking solely at the term, can be constructed. The impossibility to construct an effective needed strategy can even occur in orthogonal TRSs, a paradigmatic example being Gustave's TRS [7].

Any almost orthogonal TRS,[1] such as the parallel-$\mathtt{or}$ example above, does admit a normalising one-step reduction strategy [18,3]. There is a price to pay though, namely that such a strategy has to perform lookahead (in the form of cycle detection within terms of a given size).

Another example of the absence of needed redexes in non-orthogonal rewrite systems are *pattern calculi*. Let $\mathtt{p}$ be a data constructor representing a person including her/his name, gender and marital status. For example, $\mathtt{pjms}$ represents the person named $\mathtt{j}$ (for "Jack") who is male and single. A function such as $\lambda\mathtt{p}x\mathtt{ms}.x$ returns the name of any person that is male and single. It computes by matching the *pattern* $\mathtt{p}x\mathtt{ms}$ against its argument: reporting an appropriate substitution, if it is successful, or a distinguished constant $\mathfrak{f}$, if it fails (*cf.* Sec. 2). Consider the following term which results from applying the above mentioned function to a person called $\mathtt{a}$ (for "Alice") that is female and divorced (recall from above that I is the identity function $\lambda x.x$):

$$t_0 := (\lambda\mathtt{p}x\mathtt{ms}.x)(\mathtt{p}\,\mathtt{a}\,(I\,\mathtt{f})(I\,\mathtt{d})) \tag{4}$$

This term has two redexes, namely $I\,\mathtt{f}$ and $I\,\mathtt{d}$. Note that the term itself is not a redex since the success or failure of the match between pattern and argument cannot be determined. We have two possible reduction sequences to normal form:

$$\begin{aligned} (\lambda\mathtt{p}x\mathtt{ms}.x)(\mathtt{p}\,\mathtt{a}\,(I\,\mathtt{f})(I\,\mathtt{d})) &\to (\lambda\mathtt{p}x\mathtt{ms}.x)(\mathtt{p}\,\mathtt{a}\,(I\,\mathtt{f})\,\mathtt{d}) \to \mathfrak{f} \\ (\lambda\mathtt{p}x\mathtt{ms}.x)(\mathtt{p}\,\mathtt{a}\,(I\,\mathtt{f})(I\,\mathtt{d})) &\to (\lambda\mathtt{p}x\mathtt{ms}.x)(\mathtt{p}\,\mathtt{a}\,\mathtt{f}\,(I\,\mathtt{d})) \to \mathfrak{f} \end{aligned}$$

The first reduction sequence does not reduce $I\,\mathtt{f}$; the second does not reduce $I\,\mathtt{d}$. Therefore the term t_0 does not contain any needed redexes.

Beyond neededness. This prompts one to consider whether it is possible to obtain normalisation results for possibly overlapping, and more generally *non-sequential* ([14]) rewrite systems. The following avenues have been pursued in this direction:

[1] In fact, any almost orthogonal Combinatory Reduction Systems [19].

1. Boudol [11] studies the reduction space of possibly non-orthogonal TRS and defines needed reduction for these systems.
2. Melliès [20] extends the notion of needed redex to that of a needed derivation (actually *external* derivation, a generalization of the notion of neededness).
3. van Oostrom [23] proves that outermost-fair reduction is normalising for weakly orthogonal fully-extended higher-order pattern rewrite systems (PRS). An outermost fair strategy is one in which no outermost redex is ignored (*i.e.* not contracted) indefinitely.
4. Sekar and Ramakrishnan [22] extend the notion of a needed redex to that of a *set* of redexes, called a *necessary set*, in the context of first-order rewriting.

The results of the first item above are restricted to first-order rewriting and hence are not applicable to our pattern calculus example. The second item above suffers from two problems. The first is that it requires the calculus to verify a property (among others) called *stability* which fails for some pattern calculi such as the one of our example (*cf.* Sec. 4.2). Also, it does not seem obvious how to implement the proposed strategies. For example, in the case of $\mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{tt}),\mathtt{or}(\mathtt{tt},\mathtt{tt}))$, although there are no needed redexes, [20] declares the reduction sequence (3) *itself* to be external. It then goes on to show that composition of these external reduction sequences are normalising. So in order to normalise a term one would have to construct such reduction sequences, for which no effective tools are available. In [23] a number of normalisation results are proved for PRS, the most relevant being that outermost-fair strategies are normalising for weakly orthogonal PRS. There are a number of notable differences with our work however. The fundamental aspect that sets our paper apart from [23] is the axiomatic development that we pursue. In [23], the crucial notions of *contribution* and *copying* rely heavily on *positions* since it is terms that are rewritten. In contrast, we propose a number of axioms that are assumed to hold over *objects* and *steps* whose compliance guarantees normalisation. The nature of the objects that are rewritten is irrelevant.

We now focus our attention on [22] mentioned above, the starting point of this paper. As mentioned, terms such as $(\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,\mathtt{a}\,(I\mathtt{f})(I\mathtt{d}))$ do not contain needed redexes. However, at least one of the two redexes in each of those terms will need to be reduced in order to obtain a normal form. We thus declare the set$\{I\mathtt{f}, I\mathtt{d}\}$ to be *necessary* for this term. The intuition is that at least one redex in a necessary set must be reduced in order to obtain a normal form, assuming that a normal form exists. Of course, selecting all redexes in a term will indeed yield a necessary set; the point is whether some given subset of the set of all redexes is a necessary one. These ideas have been developed in [22] for almost orthogonal TRS where it is shown that repeated contraction of necessary sets of redexes is normalising. In this paper we extend the normalisation results for necessary sets to the setting of abstract rewriting described by means of *abstract rewrite systems* (ARS) [20]. This generalization encompasses the first-order case, the higher-order case (in particular, pattern calculi such as the Pure Pattern Calculus – **PPC** [16,17]) or any other system that complies with the appropriate axioms.

Towards an abstract proof of normalisation. In order to convey a more precise idea of the abstract nature of the setting in which we develop our proof, we provide a glimpse of **abstract rewriting systems (ARS)**. An ARS consists of a set $\mathcal{O}$ of *objects* that are rewritten, a set $\mathcal{R}$ of rewriting *steps* each having a corresponding source and target object, and the following three relations over rewriting steps:

residual relation	$[\![\cdot]\!] \subseteq \mathcal{R}\times\mathcal{R}\times\mathcal{R}$
embedding relation	$< \subseteq \mathcal{R}\times\mathcal{R}$
gripping relation	$\ll \subseteq \mathcal{R}\times\mathcal{R}$

For instance, $\mathcal{O}$ could be the set of terms of our pattern calculus example. A step would then be a pair consisting of a term and a position such that the subterm at that position may be reduced. For example, $\underline{(\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,\mathtt{a}\,\mathtt{f}\,(I\mathtt{d}))}$, where we have used underlining for denoting the position (the root position in this case). The source object of this step is $(\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,\mathtt{a}\,\mathtt{f}\,(I\mathtt{d}))$ and the target $\mathtt{f}$. The *residual relation* $[\![\cdot]\!]$ relates to the tracing of steps. A triple $(a,b,a') \in [\![\cdot]\!]$, often written $a[\![b]\!]a'$, indicates that after contracting step b, step a becomes a' (or, equivalently, a' is what is left of a). Here a and b are assumed to have the same source. For example, consider steps $c := \underline{(\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{b}.I\,x)(\mathtt{p}\,(I\,\mathtt{j})\,\mathtt{m}\,\mathtt{b})}$ and $d := (\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{b}.\underline{I\,x})(\mathtt{p}\,(I\,\mathtt{j})\,\mathtt{m}\,\mathtt{b})$ and $d' := \underline{I\,(I\,\mathtt{j})}$. Then $d[\![c]\!]d'$. The *embedding relation* is a partial order on steps with the same source. It is sometimes referred to as *nesting*. For example, $\underline{(\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,\mathtt{a}\,(I\mathtt{f})(I\mathtt{d}))}$ embeds $(\lambda\mathtt{p}\,x\,\mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,\mathtt{a}\,(\underline{I\mathtt{f}})(I\mathtt{d}))$ in the tree ordering given that the position of the former is a prefix of the position of the latter. The *gripping relation* is an additional partial order on steps that seeks to capture a typical property of higher-order rewrite systems in which a reduction step a may cause a step b to be embedded inside another one c. For this to happen, a must embed c and b. In addition, c must have occurrences of variables that are to be replaced by the substitution generated from a successful match arising from the reduction of a. In this case we say c grips a. For example, $c := (\lambda x.\underline{Ix})(Iy)$ grips $a := \underline{(\lambda x.Ix)(Iy)}$ since the former is embedded by the latter and the former has a free occurrence of the bound variable x. Note how reduction of a would embed $b := (\lambda x.Ix)\,(\underline{Iy})$ in the residual $c' := \underline{(I(Iy))}$ of c.

A number of *axioms* on ARS shall be used to formulate a proof of normalisation of necessary and never-gripping sets. These axioms verse over the above mentioned elements of an ARS and are drawn from [20], except for one of them which is new. They are developed in detail in Sec. 4. Our abstract proof is then applied to concrete cases, showing how one may obtain normalisation for **PPC** and the λ-calculus with parallel-or.

Contributions. The primary contributions may be summarised as follows:

- A gentle introduction to ARS and, in particular, to its axioms.
- An abstract proof of normalisation that applies to possibly non-orthogonal systems.
- A concrete normalisation strategy for a *non-sequential* higher-order rewrite system, namely **PPC**, and also for the λ-calculus with parallel-or.

Verification of compliance of a system with the axioms of an ARS, although in some cases tedious, provides valuable insight into its computation dynamics.

This document supersedes [8] by reformulating the normalisation technique, previously specific to **PPC**, into an axiomatic one (encompassed in Melliès' ARS), introducing a new axiom along the way. It then shows how it may be applied not only to **PPC**, but also to any other system satisfying the relevant axioms.

Structure of the paper. We begin by introducing, in Sec. 2, a simple pattern calculus that shall serve as study companion for the axiomatic development that follows. Sec. 3 defines the axiomatic framework in which we develop our results. The axioms themselves are presented in Sec. 4. The concept of multireduction and necessary multisteps are defined in Sec. 5. The axiomatic proof of normalisation is elaborated in Sec. 6. We instantiate our axiomatic proof in Sec. 7, to obtain normalisation strategies for the Pure Pattern Calculus and for the λ-calculus with parallel-or. Finally, we conclude and suggest further avenues to pursue.

For some results in Sec. 7, we include only sketches of the corresponding proofs. An extended version of this work, including the full details of all the proofs, can be found in [9].

2. A study companion: the simple pattern calculus

The simple pattern calculus (**SPC**), an extension of the lambda calculus, is presented for the sole purpose of serving as our running example in order to illustrate the various notions we shall be introducing. It is simple enough that we may be informal in our description below. Full definitions are later supplied in Sec. 7.1, where the more general Pure Pattern Calculus (**PPC**), of which **SPC** is just a fragment, is developed.

Terms ($\boldsymbol{T}$) in **SPC** are given by the following grammar:

$$t ::= x \mid \mathtt{c} \mid tt \mid \lambda p.t$$

where x ranges over some set of term variables, $\mathtt{c}$ over some set of constants, and p ranges over a set of algebraic patterns. We write $t_1 \dots t_n$ as an abbreviation for $((\dots(t_1\, t_2)\, \dots)t_n)$. An **algebraic pattern** is either a variable x or an expression of the form $\mathtt{c}\, p_1 \dots p_n$, *e.g.* $\mathtt{p}\, x\, \mathtt{mb}$. The term tu is called an **application** (t is the **function** and u the **argument**) and $\lambda p.t$ an **abstraction** (p is the **pattern** and t is the **body**). All variables in the body that also occur in the pattern of an abstraction are said to be *bound*. Application (resp. abstraction) is left (resp. right) associative. We consider terms up to **alpha-conversion**, *i.e.* up to renaming of bound variables. Positions in terms are extended to terms with patterns (*cf.* Sec. 7.1.1). $\mathtt{Pos}(t)$ is the set of positions of t; $<$ is the strict prefix relation over positions; ϵ denotes the root position. A term of the form $\mathtt{c}\, t_1 \dots t_n$ is called a **data-structure**, *e.g.* $\mathtt{p}\,(I\, \mathtt{j})\,\mathtt{mb}$.

The reduction semantics is given by the following rewrite rule:

$$(\lambda p.s)\, t \to \{\!|p \triangleright t|\!\}(s)$$

$\{\!|p \triangleright t|\!\}$ is the result of matching t against p and is called a **match**. The meaning of the expression $\{\!|p \triangleright t|\!\}(s)$ depends on this match. The match can be successful, in which case it denotes a substitution σ and $\sigma(s)$ is thus the application of the substitution to s. It can also be the special symbol $\mathtt{fail}$. The question here is what does $\mathtt{fail}(s)$ denote? Following our introduction, it would be the distinguished constant $\mathfrak{f}$. However, if $\mathfrak{f}$ is produced it could block subsequent computation unless some additional considerations on the behaviour of terms such as $\mathfrak{f}\, t$ are taken. In order to encourage other patterns to be tested and avoid overcomplicating the metatheory, it is natural to return the identity function I rather than $\mathfrak{f}$. So we set $\mathtt{fail}(s)$ to denote the identity function I. In any of these two cases, success or failure, we say that the match is **decided**. If it is not decided, in which case the match is the special symbol $\mathtt{wait}$, then the expression $(\lambda p.s)\, t$ is not a redex; *e.g.* $(\lambda \mathtt{c}.s)\, x$ or $(\lambda \mathtt{c}.s)\,(I \mathtt{c})$. A match $\{\!|p \triangleright t|\!\}$, denoted μ, is computed by applying the following equations in the order of appearance:

$$\begin{array}{rcll}
\{\!|x \triangleright t|\!\} & := & \{x \to t\} & \\
\{\!|\mathtt{c} \triangleright \mathtt{c}|\!\} & := & \{\} & \\
\{\!|\mathtt{c}\, p_1 \dots p_n \triangleright \mathtt{c}\, t_1 \dots t_n|\!\} & := & \{\!|p_1 \triangleright t_1|\!\} \uplus \dots \uplus \{\!|p_n \triangleright t_n|\!\} & n \geq 1 \\
\{\!|\mathtt{c}\, p_1 \dots p_n \triangleright \lambda q.t|\!\} & := & \mathtt{fail} & \\
\{\!|\mathtt{c}\, p_1 \dots p_n \triangleright \mathtt{d}\, t_1 \dots t_m|\!\} & := & \mathtt{fail} & \text{if } \mathtt{d} \neq \mathtt{c} \text{ or } m \neq n \\
\{\!|\mathtt{c}\, p_1 \dots p_n \triangleright t|\!\} & := & \mathtt{wait} & \text{otherwise}
\end{array}$$

The use of disjoint union in the third clause of this definition restricts successful matching of compound patterns to the linear ones,[2] which is necessary to guarantee confluence [19]. Indeed, disjoint union of two substitutions fails whenever

[2] A pattern p is linear if it has at most one occurrence of any variable.

their domains are not disjoint. Thus $\{\!|\mathtt{c}\,x\,x \triangleright \mathtt{c}\,v\,w|\!\}$ gives $\mathtt{fail}$. Other examples are: $\{\!|\mathtt{c}\,\mathtt{d} \triangleright \mathtt{c}\,(I\,\mathtt{d})|\!\}$ gives $\mathtt{wait}$, however $\{\!|\mathtt{d}\,\mathtt{d} \triangleright \mathtt{c}\,(I\,\mathtt{d})|\!\}$ gives $\mathtt{fail}$. Disjoint union of matches μ_1 and μ_2 is defined as: their union if both μ_i are substitutions and $\mathtt{dom}(\mu_1) \cap \mathtt{dom}(\mu_2) = \emptyset$; $\mathtt{wait}$ if either of the μ_i is $\mathtt{wait}$ and none is $\mathtt{fail}$; $\mathtt{fail}$ otherwise. Note that this definition of disjoint union of matches validates the following equations:

$$\mathtt{fail} \uplus \mathtt{wait} = \mathtt{wait} \uplus \mathtt{fail} = \mathtt{fail}$$

These equations reflect the non-sequential nature of reduction in **SPC**. For example, in $\{\!|\mathtt{c}\,\mathtt{d}\,\mathtt{e} \triangleright \mathtt{c}\,s\,t|\!\}$ it is unclear whether we should pick s or t in order to obtain a decided match since either may not normalise while the other may help decide the match (towards $\mathtt{fail}$).

3. Abstract rewriting systems

This section revisits the definition of *abstract rewriting systems* given in the introduction supplying further details and introduces the axioms that such systems must enjoy in order for the abstract proof of normalisation to be applicable.

3.1. Basic components

Recall from the introduction that an ARS consists of a set of *objects* $\mathcal{O}$ that are rewritten and a set of *steps*[3] $\mathcal{R}$. Each step have a source and target object given by functions $\mathsf{src}, \mathsf{tgt} : \mathcal{R} \to \mathcal{O}$. If $t \in \mathcal{O}$, then we write $\mathcal{R}ed(t)$ for the set $\{a \in \mathcal{R} \text{ s.t. } \mathsf{src}(a) = t\}$. Two steps with the same source are said to be **coinitial**. We often write $t \xrightarrow{a} u$ for a step a s.t. $\mathsf{src}(a) = t$ and $\mathsf{tgt}(a) = u$.

The following relations are given over steps:

- The **residual** relation $[\![\cdot]\!] \subseteq \mathcal{R} \times \mathcal{R} \times \mathcal{R}$.
  This relation reflects how a step may be traced after some other *coinitial* step is reduced. Whenever $b[\![a]\!]b'$ we require a and b to be coinitial, and $\mathsf{src}(b') = \mathsf{tgt}(a)$. When $b[\![a]\!]b'$ we say that b' is a residual of b after a. By $b[\![a]\!]$ we denote the set $\{b' \text{ s.t. } b[\![a]\!]b'\}$ and similarly for $[\![a]\!]b$. Accordingly, we define $[\![a]\!]$ as the relation $\{(b, b') \text{ s.t. } b[\![a]\!]b'\}$. A step b is said to be **created** by a step a, with $\mathsf{src}(b) = \mathsf{tgt}(a)$, if $[\![a]\!]b = \emptyset$.
- The **embedding** relation $< \subseteq \mathcal{R} \times \mathcal{R}$.
  This well-founded relation[4] is a strict order[5] on coinitial steps.
  For each pair $a < b$, the steps a and b must be coinitial. A step a is said to be **outermost** iff there is no b such that $b < a$. A step a is **disjoint** from b, written $a \parallel b$, when a and b are coinitial, $a \neq b$ and $b \neq a$.
- The **gripping** relation $\ll \subseteq \mathcal{R} \times \mathcal{R}$.
  Just like for embedding, gripping is a well-founded and strict order on coinitial steps.
  For each pair $a \ll b$, the steps a and b must be coinitial. This relation seeks to capture a typical property of higher-order rewrite systems, in which steps disjoint with b in the source object of a, may become embedded by b in the target object of a.

An example of an ARS is the **SPC**. Its objects $\mathcal{O}$ are just the terms $\boldsymbol{T}$. A step is a pair consisting of a term and a position in the term s.t. the subterm at this position is of the form $(\lambda p.s)u$, and $\{\!|p \triangleright u|\!\}$ is decided. For example, $a := (\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,a\,\underline{\mathtt{f}\,(I\,\mathtt{d})})$ is a step, where we have underlined the relevant position. Then $\mathsf{src}(a)$ is the term $(\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,a\,\mathtt{f}\,(I\,\mathtt{d}))$ and $\mathsf{tgt}(a)$ is I (since matching fails and hence the identity function is produced). We could also write $(\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,a\,\mathtt{f}\,(I\,\mathtt{d})) \xrightarrow{a} I$. Also, $b := (\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,a\,\mathtt{f}\,(\underline{I\,\mathtt{d}}))$ is a step. It has the same source as a but the target is $(\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,a\,\mathtt{f}\,\mathtt{d})$.

For an example of steps related by the residual relation, consider the step $c := (\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,\underline{(I\,\mathtt{j})}\,\mathtt{mb})$ and $d := (\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,(\underline{I\,\mathtt{j}})\,\mathtt{mb})$ and $d' := \underline{I\,\mathtt{j}}$. Then $d[\![c]\!]d'$. Steps may be erased by other steps. For example, $\underline{(\lambda\mathtt{p}\,x\,y\,z.\mathtt{c})\,(\mathtt{p}\,(I\,u)\,\mathtt{mb})}$ erases the coinitial step $(\lambda\mathtt{p}\,x\,y\,z.\mathtt{c})\,(\mathtt{p}\,\underline{(I\,u)}\,\mathtt{mb})$. It may also duplicate a coinitial step. For example, $\underline{(\lambda\mathtt{p}\,x\,y\,z.x\,x)\,(\mathtt{p}\,(I\,u)\,\mathtt{mb})}$ duplicates $(\lambda\mathtt{p}\,x\,y\,z.x\,x)\,(\mathtt{p}\,\underline{(I\,u)}\,\mathtt{mb})$ yielding two residuals $\underline{(I\,u)}\,(I\,u)$ and $(I\,u)\,\underline{(I\,u)}$.

In **SPC** a step a embeds another step b iff the position of a is a prefix of the position of b. For example, c described above embeds d. However, the two steps $(\lambda\mathtt{p}\,x\,\mathtt{mb}.x)(\mathtt{p}\,a\,(\underline{I\,\mathtt{f}})\,(\underline{I\,\mathtt{d}}))$ are not related by embedding and are hence disjoint.

An example of gripping was given in the introduction. We revisit gripping in Sec. 4.3.

3.2. Reduction sequences, multisteps and developments

A **reduction sequence** (or **derivation**) is either $\mathtt{nil}_t$, *i.e.* an *empty sequence* indexed by the object t, or a (possibly infinite) sequence $a_1; a_2, \ldots; a_n; \ldots$ of steps verifying $\mathsf{tgt}(a_k) = \mathsf{src}(a_{k+1})$ for all $k \geq 1$. In the former case, we define the

[3] Called *redexes* ("radicaux") in [20], hence the reason why we use the letter $\mathcal{R}$.
[4] Notice that if $\mathcal{R}ed(t)$ is finite, then any relation on coinitial steps is necessarily well-founded.
[5] The embedding relation $<$ is assumed to be irreflexive and transitive.

source as t and in the latter case as the source of the first step in the sequence. We define the **target** of a finite reduction sequence as follows: $\mathsf{tgt}(\mathtt{nil}_t) := t$, $\mathsf{tgt}(a_1; \ldots; a_n) := \mathsf{tgt}(a_n)$. The **length** of a reduction sequence, denoted by $|\cdot|$, is defined as follows: $|\mathtt{nil}_t| := 0$, $|a_1; \ldots; a_n| := n$. The target and length of an infinite sequence are undefined. We write $\mathcal{RS}$ for the set of reduction sequences. In the following, reduction sequences are given the names δ, δ', δ_1, γ, ξ, etc. We write $t \overset{\delta}{\twoheadrightarrow} u$ to indicate that $\mathsf{src}(\delta) = t$ and $\mathsf{tgt}(\delta) = u$. Also, if $\delta = a_1; \ldots; a_n$, we denote with $\delta[k]$ the step a_k, and write $\delta[i..j]$ for the subsequence $a_i; \ldots; a_j$, if $i \leq j$, and $\mathtt{nil}_{\mathsf{src}(a_i)}$, if $i > j$. We use the symbol ; to denote the concatenation of reduction sequences, allowing to concatenate steps and sequences freely, *e.g.* $a; \delta$ or $a; b$ or $\delta; a$ or $\delta; \gamma$, as long as the concatenation yields a valid reduction sequence. If $\mathcal{R}ed(t) = \emptyset$ then we say that t is a **normal form**. An object t is **normalising** iff there exists a reduction sequence δ such that $t \overset{\delta}{\twoheadrightarrow} u$ and u is a normal form.

A **multistep** is a set of coinitial steps, *i.e.* a subset of $\mathcal{R}ed(t)$ for a certain object t. We denote such sets by the letters $\mathcal{A}$, $\mathcal{A}'$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, etc. Two multisteps are **coinitial** if their union is a multistep. **Residuals of coinitial steps $\mathcal{B}$ after a** are defined by $\mathcal{B}[\![a]\!]b'$ iff $b[\![a]\!]b'$ for some $b \in \mathcal{B}$. We also use the notation $\mathcal{B}[\![a]\!]$, defined analogously to $b[\![a]\!]$. Notice that for any a and b, $b[\![a]\!]$ is a set of coinitial steps; the same happens with $\mathcal{B}[\![a]\!]$ for any $\mathcal{B}$.

Residuals after reduction sequences $[\![\cdot]\!] \subseteq \mathcal{R} \times \mathcal{RS} \times \mathcal{R}$ are defined as follows: $b[\![\mathtt{nil}_t]\!]b$ for all $b \in \mathcal{R}ed(t)$, and $b[\![a; \delta]\!]b'$ whenever $b[\![a]\!]b''$ and $b''[\![\delta]\!]b'$ hold for some b''. We sometimes use the notation $b[\![\delta]\!]$ for the set of residuals of b after δ, and $[\![\delta]\!]$ to denote the relation $\{(b, b') \text{ s.t. } b[\![\delta]\!]b'\}$. We also write $\mathcal{B}[\![\delta]\!]b'$ and $\mathcal{B}[\![\delta]\!]$ for the obvious extension of residuals of steps after a reduction sequence to *multisteps*. Observe that $\mathcal{B}[\![a; \delta]\!] = \mathcal{B}[\![a]\!][\![\delta]\!]$.

Next we consider contraction of multisteps. Since, in principle, the order in which the steps comprising a multistep $\mathcal{A}$ are contracted could affect the target object of $\mathcal{A}$ and/or its residual relation, it becomes necessary to lay out precise definitions on the meaning of contraction. This is achieved through the concept of *development*. Let $\mathcal{A} \subseteq \mathcal{R}ed(t)$ for some object t. The reduction sequence δ is a **development** of $\mathcal{A}$ iff $\delta[i] \in \mathcal{A}[\![\delta[1..i-1]]\!]$ for all $i \leq |\delta|$. *E.g.* a development of the multistep $\mathcal{A} := \{a, b\}$ where a is $\underline{(\lambda x.Ix)(Iy)}$ and b is $(\lambda x.Ix)(\underline{Iy})$ is the reduction sequence $\underline{(\lambda x.Ix)(Iy)} \overset{a}{\to} I(\underline{Iy}) \overset{b'}{\to} Iy$, since $a \in \mathcal{A}[\![\mathtt{nil}_{(\lambda x.Ix)(Iy)}]\!] = \mathcal{A}$ and $b' \in \mathcal{A}[\![a]\!]$ given that $b[\![a]\!]b'$. The reduction sequence $(\lambda x.Ix)(\underline{Iy}) \overset{b}{\to} \underline{(\lambda x.Ix)y} \overset{a'}{\to} Iy$, where $a[\![b]\!]a'$, is also a development of $\mathcal{A}$. Note also that the reduction sequence consisting solely of the step a (or the step b) is a development of $\mathcal{A}$ too. A development δ of $\mathcal{A}$ is **complete** (written $\delta \Vdash \mathcal{A}$) iff δ is finite and $\mathcal{A}[\![\delta]\!] = \emptyset$.

The **depth** of a multistep $\mathcal{A}$, written $\nu(\mathcal{A})$, is the length of its longest complete development. If $a \in \mathcal{A}$ and $\delta \Vdash \mathcal{A}[\![a]\!]$, then $a; \delta \Vdash \mathcal{A}$. Consequently, $\nu(\mathcal{A}) > \nu(\mathcal{A}[\![a]\!])$, yielding a convenient induction principle for multisteps. Both the notion of depth and the derived induction principle are important tools in several proofs of this work.

Note that it is not a priori clear that a development terminates, nor that the residual relation is finitely branching. Moreover, since there may be more than one development of a multistep, it is natural to wonder whether they all have the same target and induce the same residual relation. These topics are discussed in the next section (*cf.* finite residuals, finite developments and semantic orthogonality axioms). Suffice it to say, for now, that complete developments are a valid means of defining contraction of multisteps since the latter do not depend on the complete development chosen (Proposition 4.1).

Let $\mathcal{A} \subseteq \mathcal{R}ed(t)$ be a multistep. Define $\mathsf{src}(\mathcal{A}) := t$, $\mathsf{tgt}(\mathcal{A}) := \mathsf{tgt}(\delta)$, and $b[\![\mathcal{A}]\!]b'$ iff $b[\![\delta]\!]b'$ where δ is an arbitrary complete development of $\mathcal{A}$. In order for these definitions to be coherent, the empty multistep should be indexed by an object, *i.e.* $\emptyset_t$, so that $\mathsf{src}(\emptyset_t) := \mathsf{tgt}(\emptyset_t) := t$. We will use the notations $b[\![\mathcal{A}]\!]$, $\mathcal{B}[\![\mathcal{A}]\!]b'$, $\mathcal{B}[\![\mathcal{A}]\!]$ with meanings analogous to those described for steps. Notice that for any $a \in \mathcal{A}$, the reduction sequence $a; \delta'$ is a (complete) development of $\mathcal{A}$ iff δ' is a (complete) development of $\mathcal{A}[\![a]\!]$. As a consequence, $\mathcal{B}[\![\mathcal{A}]\!] = \mathcal{B}[\![a]\!][\![\mathcal{A}[\![a]\!]]\!]$. **Multistep contraction** of $\mathcal{A} \subseteq \mathcal{R}ed(t)$, written $t \overset{\mathcal{A}}{\twoheadrightarrow} u$, where $\mathsf{src}(\mathcal{A}) = t$ and $\mathsf{tgt}(\mathcal{A}) = u$, denotes an arbitrary complete development $\delta \Vdash \mathcal{A}$, where $t \overset{\delta}{\twoheadrightarrow} u$.

As a closing comment to this section, it should be mentioned that the analysis of contraction of multisteps for higher-order rewriting is non-trivial even for sets of *pairwise disjoint* steps, since residuals of such sets are not necessarily pairwise disjoint. Conversely, in first-order rewriting, residuals of pairwise disjoint sets of steps are always pairwise disjoint again. This difference yields the need of a subtle analysis of the behaviour of multisteps, which is not required for the first-order case (*cf.* [22]), in order to obtain normalisation results applicable to higher-order rewrite systems.

4. Axioms for ARS

We next introduce the axioms for ARS. These are presented in three groups (Fig. 1), together with their associated concepts. The *fundamental axioms* deal with basic properties of the residual relation; the *embedding axioms* deal with the interaction between residuals and embedding; and the *gripping axioms* deal with the basic properties of the gripping relation on redexes. In what follows, free variables in the statement of an axiom are assumed implicitly universally quantified. For example, "$a[\![a]\!] = \emptyset$" should be read as "For all $a \in \mathcal{R}$, $a[\![a]\!] = \emptyset$". Finally, bear in mind that in an expression such as "$a[\![b]\!]a'$", steps a and b are assumed coinitial.

4.1. Fundamental axioms

The fundamental axioms of an ARS have to do with the properties of the residual relation over redexes and derivations. The embedding and gripping relations do not participate in these axioms. The first is Self Reduction and states, quite reasonably, that nothing is left of a step a if it is contracted.

Axiom group	Axioms	Reference
Fundamental	Self Reduction, Finite Residuals, Ancestor Uniqueness, FD, SO	Sec. 4.1
Embedding	Linearity, Context-Freeness, Enclave–Creation, Enclave–Embedding, Pivot	Sec. 4.2
Gripping	Grip–Instantiation, Grip–Density, Grip–Convexity	Sec. 4.3

Fig. 1. Axioms for ARS presented in three groups.

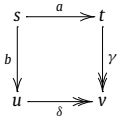


Fig. 2. The semantic orthogonality axiom.

Self Reduction $\quad a[\![a]\!] = \emptyset.$

The second is Finite Residuals and states that the residuals of a step b after contraction of a coinitial (and possibly the same) one a is a finite set. In other words, a step may erase ($b[\![a]\!] = \emptyset$) or copy other coinitial steps, however only a finite number of copies can be produced.

Finite Residuals $\quad b[\![a]\!]$ is a finite set.

The third one, namely Ancestor Uniqueness, states that a step a cannot "fuse" two different steps b_1 and b_2, coinitial with a, into one. In other words, if we use the term "ancestor" to refer to the inverse of the residual relation, then any step can have at most one ancestor.

Ancestor Uniqueness $\quad b_1[\![a]\!]b' \wedge b_2[\![a]\!]b' \Rightarrow b_1 = b_2.$

As discussed in Sec. 3.2, a multistep $\mathcal{A}$ is contracted by performing any complete development of $\mathcal{A}$. However, as already mentioned, developments may a priori not terminate and, since there may be more than one development of a multistep, they may not all have the same target or induce the same residual relation. Any of these situations would render the purported notion of multistep contraction senseless. The following two axioms FD and SO ensure exactly that these three properties are met. The first states that any development of a multistep $\mathcal{A}$ necessarily terminates.

Finite developments (FD) $\quad$ All developments of $\mathcal{A}$ are finite.

Note that, together with Finite Residuals, FD implies (by König's Lemma) that the notion of *depth* of a multistep $\mathcal{A}$ (*i.e.* the length of the longest complete development of $\mathcal{A}$) is well-defined. As already mentioned, this provides us with a convenient measure for proving properties involving multisteps.

The second axiom states that complete developments of a multistep $\{a, b\}$, consisting of two coinitial steps, are joinable and induce the same residual relation (Fig. 2). It is called PERM in [20].

Semantic orthogonality (SO) $\quad \exists \delta, \gamma$ s.t. $\delta \Vdash a[\![b]\!] \wedge \gamma \Vdash b[\![a]\!] \wedge \mathsf{tgt}(a;\gamma) = \mathsf{tgt}(b;\delta) \wedge$ the relations $[\![a;\gamma]\!]$ and $[\![b;\delta]\!]$ coincide.

Developments of an *arbitrary* multistep of an ARS are also joinable and induce the same residual relation. This is reflected in the following result (Lem. 2.18 and Lem. 2.19 in [20]) which is proved by resorting to all of the above axioms (except for Ancestor Uniqueness):

Proposition 4.1. *Consider an ARS enjoying the fundamental axioms. Suppose $\delta \Vdash \mathcal{A}$ and $\gamma \Vdash \mathcal{A}$. Then $tgt(\delta) = tgt(\gamma)$ and the relations $[\![\delta]\!]$ and $[\![\gamma]\!]$ coincide.*

4.2. Embedding axioms

The embedding axioms establish coherence conditions between the embedding relation $<$ and the residual relation $[\![\cdot]\!]$. In reading these axioms it helps to think of $a < b$ in the setting of **SPC** as indicating that the position of the step a is a prefix of the position of b. Bear in mind however, that an ARS does not assume the existence of terms nor positions; this reading is solely for the purpose of aiding the interpretation of the axioms.

The first axiom, Linearity, states that the only way in which a step a can either erase or produce multiple (two or more) copies of a coinitial step, is if it embeds it.

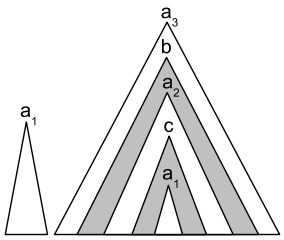


Fig. 3. Three redexes a, b, c such that $b < c$ and $a \neq b$.

Linearity $\quad a \not< b \Rightarrow \exists! b'$ s.t. $b[\![a]\!]b'$.

The second axiom pertains to the invariance of the embedding relation w.r.t. contraction of steps. Consider three coinitial steps a, b and c. Suppose that $b[\![a]\!]b'$ and $c[\![a]\!]c'$, for some steps b' and c' (this implies $a \neq c$ and $a \neq b$). If a does not embed c ($a \not< c$), then a cannot grant the ability to b of embedding c ($b \not< c \ \Rightarrow b' < c'$ cannot happen) or revoke it from b ($b < c \ \Rightarrow b' \not< c'$ cannot happen).

Context-Freeness $\quad b[\![a]\!]b' \wedge c[\![a]\!]c' \Rightarrow a < c \vee (b < c \ \Leftrightarrow b' < c')$.

The next two axioms assume that two steps a and b are given such that $b < a$. It considers under what conditions b', the unique residual of b after a ($b[\![a]\!]b'$), embeds other steps c' in the target of a. Two cases are considered, first when c' is created by a (Enclave–Creation) and then when it is not (Enclave–Embedding).

Enclave–Creation $\quad b < a \wedge b[\![a]\!]b' \wedge \emptyset[\![a]\!]c' \Rightarrow b' < c'$.
Enclave–Embedding $\quad b[\![a]\!]b' \wedge c[\![a]\!]c' \wedge b < a < c \Rightarrow b' < c'$.

Finally, the axiom Pivot is new, in the sense that it does not appear in [20] since it is not required for the results that are proved there. It reads as follows:

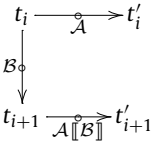
Pivot $\quad a < c \wedge b < c \wedge b \not< a \wedge c[\![a]\!]c' \Rightarrow \exists b'$ s.t. $b[\![a]\!]b' \wedge b' < c'$.

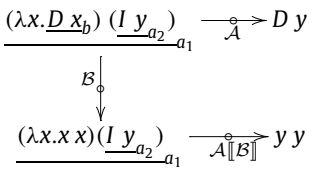
To motivate this axiom we illustrate an important property that we shall need to prove for our normalisation result (Lemma 6.4). We assume given $b < c$ and a step $a \neq b$ s.t. $c[\![a]\!]c'$ for some c' (*cf.* shaded triangles in the figure). We would like to deduce the existence of b' s.t. (i) $b[\![a]\!]b'$ and (ii) $b' < c'$. For that we proceed to consider all possible embedding relations between a, on the one hand, and b and c, on the other (see Fig. 3):

- $a \not< c$. This is represented with the two occurrences of a subscripted with 1. We conclude (i) and (ii) using Linearity and Context-Freeness.
- $a < c$.
  - $b < a$ (hence $b < a < c$). This is represented with the occurrence of a subscripted with 2. We conclude (i) and (ii) using Linearity and Enclave–Embedding.
  - $b \not< a$. This is represented with the occurrence of a subscripted with 3. We conclude (i) and (ii) using Pivot.

4.3. Gripping axioms

In order to motivate our need for the gripping relation [20] we provide a brief glimpse of the approach we take for our abstract proof of normalisation. We shall show that, starting from a normalising object t and by repeatedly contracting multisteps enjoying certain properties (let us call such multisteps *judicious*), a normal form will be reached. That this process does not continue indefinitely, shall be guaranteed by providing an appropriate measure that decreases with each such judicious multistep. The elements that are measured are certain "multireductions", sequences of multisteps, that originate from each of the sources and targets of judicious multisteps. In the particular case that a multireduction consists of a sole multistep $\mathcal{A}$, our measure computes its *depth* (the length of the longest complete development). This is depicted in the figure where $\mathcal{B}$ is the judicious multistep, $\mathcal{A}$ is the multireduction consisting of just one multistep that is measured and $\mathcal{A}[\![\mathcal{B}]\!]$ is what remains of multistep $\mathcal{A}$ after $\mathcal{B}$ which will also be measured and compared with the measure of $\mathcal{A}$. We are interested in showing that the depth of $\mathcal{A}$ is greater than that of $\mathcal{A}[\![\mathcal{B}]\!]$. In general, this is not true as the following example in λ-calculus (suggested by V. van Oostrom and also applicable to higher-order rewrite systems in general) illustrates, where $\mathcal{A} := \{a_1, a_2\}$, $\mathcal{B} := \{b\}$ and $D = \lambda z.z\,z$. Note that indeed we have $\nu(\mathcal{A}) = 2 < 3 = \nu(\mathcal{A}[\![\mathcal{B}]\!])$.





The problem is that a_1 embeds the step b that duplicates a_1's bound variable; if this variable is substituted by some other step (in this example, a_2) then, after contracting b more copies of a_2 have to be contracted in the development of $\mathcal{A}[\![\mathcal{B}]\!]$. When a step a embeds another step b that has a free occurrence of a variable bound by a we say that b **grips** a and write:

$$a \ll b$$

We shall avoid the above situation by requiring our judicious multisteps to be *never-gripping* (*cf.* Lemma 6.2), in other words, that this situation never occurs. Since our ARS are over abstract objects (hence there is no notion of term, nor variable, nor bound variable) we must put forward appropriate axioms that capture gripping in an abstract way. These axioms were presented in [20] for the purpose of providing an abstract proof of finite developments for ARS (see remark at the end of this section). We next present these axioms.

The first one, Grip–Instantiation, states the role gripping plays in the creation of new embeddings. Consider three coinitial steps a, b, c and steps b', c' s.t. $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Suppose b' embeds c' (*i.e.* $b' < c'$). Two situations are possible. If $a \neq c$, then by Context-Freeness, we already know that $b < c$. However, if $a < c$ (and $b \neq c$), then this axiom may be seen to provide further information. It informs us that b grips a: this is the only way in which a can place (the residual of) c under (the residual of) b. As an example, consider $\underline{(\lambda x.\underline{I\, x}_b)\overline{(I\, I)}^{c}}_a$; after contracting a we obtain $\underline{I\, \overline{(I\, I)}^{c'}}_{b'}$; notice how a has nested c under b.

Grip–Instantiation $\qquad b[\![a]\!]b' \wedge c[\![a]\!]c' \wedge b' < c' \Rightarrow b < c \vee (a \ll b \wedge a < c).$

The second axiom, Grip–Density, states that at any moment a step grips some other step, this can be traced back to a "chain" of grippings over the ancestors of these steps. An example in **SPC** that shows the $b \ll a \ll c$ case is $\underline{(\lambda y.((\lambda x.\underline{I\, x}_c)\, y)_a)u}_b$. The other case, namely when $b' \ll c'$ holds because already $b \ll c$, may be illustrated by $\underline{I\underline{((\lambda x.\overline{I\, x}^{c})\, y)}_b}_a$.

Grip–Density $\qquad b[\![a]\!]b' \wedge c[\![a]\!]c' \wedge b' \ll c' \Rightarrow b \ll c \vee b \ll a \ll c.$

The third axiom, Grip–Convexity, states if a step b grips another step a (*i.e.* $a \ll b$), then any step that embeds b either grips a or embeds a. Examples for each case are $\underline{(\lambda x.I\, \underline{(I\, \overline{x}^{b})}_c)\, y}_a$ and $\underline{I\, \underline{((\lambda x.I\, \overline{x}^{b})\, y)}_a}_c$ respectively.

Grip–Convexity $\qquad a \ll b \wedge c < b \Rightarrow a \ll c \vee c \leq a.$

Note that the embedding and gripping relations are defined independently in the abstract framework. Moreover, the gripping axioms do not imply neither $< \subseteq \ll$ nor $\ll \subseteq <$. That being said, in our concrete **PPC** framework, the gripping relation between **PPC**-redexes (on page 56) is included in the embedding relation.

As remarked above, the gripping axioms entail FD, Thm. 3.2. in [20].

4.4. An additional axiom: stability

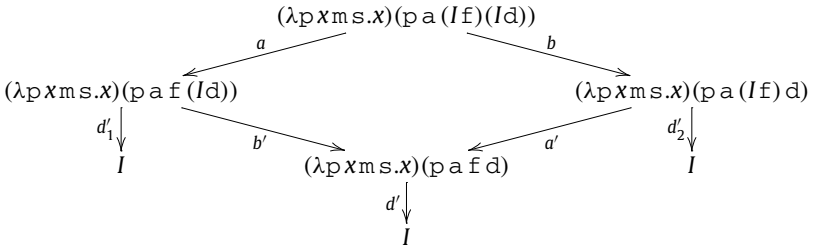
One final axiom which, although not required for our abstract normalisation proof, is worthy of mention given the key rôle that it plays in the axiomatic standardisation proof of [20], is briefly discussed here. An ARS satisfying the fundamental axioms and the embedding axioms (disregarding both enclave axioms and the axiom Pivot), enjoys the property of *existence* of standard derivations [20]. A standard derivation is one in which contraction takes place outside-in. The additional property of *uniqueness* of such derivations can also be proved in this axiomatic framework. For that the ARS must satisfy the fundamental axioms, the embedding axioms (disregarding Pivot, which is not required) and an additional axiom called Stability. This last axiom states that steps can be created in a unique way.

Stability $\qquad$ Assume $a \parallel b$, $a[\![b]\!]a'$, $b[\![a]\!]b'$, and there exists some d' such that $d'_1[\![b']\!]d'$ and $d'_2[\![a']\!]d'$. Then there exists d such that $d[\![a]\!]d'_1$, $d[\![b]\!]d'_2$, and either $a \not\leq d$ or $b \not\leq d$.

As mentioned, Stability is not required for our abstract normalisation proof. This is quite fortunate since neither the parallel-or TRS nor the **SPC** of the introduction, enjoy stability. Let us look at the case of **SPC**.

Fundamental axioms	
Self Reduction	$a[\![a]\!] = \emptyset$.
Finite Residuals	$b[\![a]\!]$ is a finite set.
Ancestor Uniqueness	$b_1[\![a]\!]b' \wedge b_2[\![a]\!]b' \Rightarrow b_1 = b_2$.
Finite developments	All developments of a multistep $\mathcal{A}$ are finite.
Semantic orthogonality (SO)	$\exists \delta, \gamma$ s.t. $\delta \Vdash a[\![b]\!] \wedge \gamma \Vdash b[\![a]\!] \wedge \mathsf{tgt}(a;\gamma) = \mathsf{tgt}(b;\delta) \wedge$ the relations $[\![a;\gamma]\!]$ and $[\![b;\delta]\!]$ coincide.
Embedding axioms	
Linearity	$a \not\leq b \Rightarrow \exists! b'$ s.t. $b[\![a]\!]b'$.
Context-Freeness	$b[\![a]\!]b' \wedge c[\![a]\!]c' \Rightarrow a < c \vee (b < c \Leftrightarrow b' < c')$.
Enclave–Creation	$b < a \wedge b[\![a]\!]b' \wedge \emptyset[\![a]\!]c' \Rightarrow b' < c'$.
Enclave–Embedding	$b[\![a]\!]b' \wedge c[\![a]\!]c' \wedge b < a < c \Rightarrow b' < c'$.
Pivot	$a < c \wedge b < c \wedge b \not\leq a \wedge c[\![a]\!]c' \Rightarrow \exists b'$ s.t. $b[\![a]\!]b' \wedge b' < c'$.
Gripping axioms	
Grip–Instantiation	$b[\![a]\!]b' \wedge c[\![a]\!]c' \wedge b' < c' \Rightarrow b < c \vee (a \ll b \wedge a < c)$.
Grip–Density	$b[\![a]\!]b' \wedge c[\![a]\!]c' \wedge b' \ll c' \Rightarrow b \ll c \vee b \ll a \ll c$.
Grip–Convexity	$a \ll b \wedge c < b \Rightarrow a \ll c \vee c \leq a$.

Fig. 4. The three groups of axioms of an ARS.



The steps depicted above meet the antecedent of the statement of the stability axiom. However, the conclusion is not satisfied since both steps d'_1 and d'_2 are created, by a and b, respectively. In other words, there is no d as in the conclusion of the Stability axiom.

This concludes our presentation of the axioms of an ARS. A summary of all three groups is given in Fig. 4.

5. Multireductions over an ARS

Our normalisation result states conditions under which an object can be reduced to a normal form by repeatedly contracting multisteps, thus requiring a precise meaning for sequences of such multisteps. Also, we must introduce some qualifiers for multisteps that enjoy properties that are useful for the development that shall follow.

5.1. Multireductions

The concept of reduction sequence introduced earlier for steps, makes sense for multisteps as well. A **multireduction sequence**, or just **multireduction**, is either $\mathtt{nil}_t$, an *empty multireduction* indexed by the object t, or a sequence of multisteps $\mathcal{A}_1; \ldots; \mathcal{A}_n; \ldots$ where $\mathsf{tgt}(\mathcal{A}_{k+1}) = \mathsf{src}(\mathcal{A}_k)$ for all $k \geq 1$. We use Δ, Γ, Π, Ψ to denote multireductions and $\Delta[k]$ and $\Delta[i..j]$ with the same meanings given for reduction sequences. Source, target and length of multireductions are defined analogously as done before for reduction sequences. We write $t \overset{\Delta}{\multimap\!\!\!\!\rightarrow\!\!\!\!\rightarrow} u$ to denote that $\mathsf{src}(\Delta) = t$ and $\mathsf{tgt}(\Delta) = u$. Some comments:

- The **length of a multireduction** is the number of its multisteps, it is not related to the size of the sets.
- An element of a multireduction can be an empty multistep, so that the only corresponding development is the empty reduction sequence indexed by its source.
- A multireduction consisting of one or more occurrences of $\emptyset_t$, and $\mathtt{nil}_t$, are different multireductions. In particular, $|\emptyset_t| = 1$ while $|\mathtt{nil}_t| = 0$. We will say that a multireduction is **trivial** iff all its elements are empty multisteps. Empty multireductions are trivial.

The residual relation is extended from multisteps to multireductions, exactly as we have extended it from steps to reduction sequences. We use the notations $b[\![\Delta]\!]b'$, $b[\![\Delta]\!]$, $\mathcal{B}[\![\Delta]\!]b'$, $\mathcal{B}[\![\Delta]\!]$ and $[\![\Delta]\!]$ as expected.

Let $\mathcal{MR}$ be the set of multisteps associated to an ARS. Notice that, in contrast to the notion of residuals for steps, residuals can be considered as a function on multisteps, *i.e.* $[\![\cdot]\!] : \mathcal{MR} \times \mathcal{MR} \to \mathcal{MR}$, since $\mathcal{A}[\![\mathcal{B}]\!]$ is again a multistep for any $\mathcal{A}, \mathcal{B}$. This distinguishing feature of multisteps leads to the definition of the **residual of a multireduction after a**

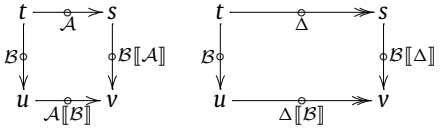
multistep, for which we will (ab)use the notation $[\![\cdot]\!]$. If $\mathsf{src}(\mathcal{B}) = t$ then $\mathtt{nil}_t[\![\mathcal{B}]\!] := \mathtt{nil}_{\mathsf{tgt}(\mathcal{B})}$; if $\mathcal{A}$ and $\mathcal{B}$ are coinitial then $(\mathcal{A}; \Delta)[\![\mathcal{B}]\!] := \mathcal{A}[\![\mathcal{B}]\!]; (\Delta[\![\mathcal{B}[\![\mathcal{A}]\!]]\!])$. Observe that, in spite of the name "residual" and the notation $[\![\cdot]\!]$, the above definition corresponds to a partial function $\mathcal{MRS} \times \mathcal{MR} \to \mathcal{MRS}$, where $\mathcal{MRS}$ stands for the set of multireductions. Notice also that $|\Delta[\![\mathcal{B}]\!]| = |\Delta|$.

A **(multistep) reduction strategy** for an ARS $\mathfrak{A}$ is any function $\mathcal{S} : (\mathcal{O} \setminus NF) \to \mathcal{P}(\mathcal{R})$ such that $\mathcal{S}(t) \neq \emptyset$ and $\mathcal{S}(t) \subseteq \mathcal{R}ed(t)$ for all t; here NF stands for the set of normal forms of $\mathfrak{A}$. A multistep reduction strategy determines, for each object, a *multireduction*: if $t \in NF$, then the associated multireduction is $\mathtt{nil}_t$, otherwise it is $\mathcal{S}(t_0); \mathcal{S}(t_1); \ldots; \mathcal{S}(t_n); \ldots$ where $t_0 := t$ and $t_{n+1} := \mathsf{tgt}(\mathcal{S}(t_n))$. A reduction strategy is **normalising** iff for any object t, the determined multireduction ends in a normal form for all normalising objects. A **single-step reduction strategy** is a multistep reduction strategy $\mathcal{S}$ s.t. $\mathcal{S}(t)$ is a singleton for every t in the domain of $\mathcal{S}$. In this case, the multireduction sequence determined by $\mathcal{S}$ is in fact a *reduction sequence*.

The independence of order of contraction of steps, formalised in Proposition 4.1, extends to multisteps [20, Lem. 2.24] and to multireductions. The former is a consequence of Proposition 4.1 and the latter then follows by induction on Δ.

Proposition 5.1. *Consider an ARS enjoying the group of fundamental axioms.*

1. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{R}ed(t)$. The target and residual relation of $\mathcal{A}; \mathcal{B}[\![\mathcal{A}]\!]$ and $\mathcal{B}; \mathcal{A}[\![\mathcal{B}]\!]$ coincide.*
2. *Let Δ be a multireduction, and $\mathcal{B} \subseteq \mathcal{R}ed(t)$. The target and residual relation associated to $\Delta; \mathcal{B}[\![\Delta]\!]$ and $\mathcal{B}; \Delta[\![\mathcal{B}]\!]$ coincide.*



5.2. Key concepts

This section introduces several notions that are crucial in the development of our abstract normalisation proof. More precisely, our normalisation result holds for strategies choosing never-gripping and necessary multisteps, which are introduced as follows. First of all, starting from the embedding relation on redexes, we define two key relations on multisteps: *free-from* and *embedded by*. Secondly, starting from the gripping notion on redexes, we define a gripping notion on multisteps, together with the associated concept of *never-gripping* multisteps. Last but not least, we define the *uses* relation which defines what it means for a multistep to be *necessary*.

Free-from and embedded multisteps

Two notions related with embedding and involving multisteps are crucial to define the main elements of the abstract normalisation proof. In order to introduce these notions, we discuss different ways to extend the notion of embedding to multisteps.

Two different meanings could be assigned to the notation $a > \mathcal{B}$: either that there exists some $b \in \mathcal{B}$ that verifies $a > b$, or else that $a > b$ for all $b \in \mathcal{B}$. Since we are going to apply this general framework to *terms*, it seems natural to adopt the former interpretation. Conversely, when considering $\mathcal{A} > \mathcal{B}$, we take a "forall" meaning on the $\mathcal{A}$ side: to consider that $\mathcal{B}$ embeds $\mathcal{A}$, it must embed each of its elements. In the sequel, we use the notations $a > \mathcal{B}$ and $\mathcal{A} > \mathcal{B}$ with the just given meanings, and we say that a (or $\mathcal{A}$) is **embedded by** $\mathcal{B}$.

On the other hand, we say that a step is **free from** a coinitial multistep, if it is neither equal to nor embedded by a step in $\mathcal{B}$. In turn, a multistep $\mathcal{A}$ is free from another, coinitial multistep $\mathcal{B}$, if a is free from $\mathcal{B}$ *for all*[6] $a \in \mathcal{A}$. The notion also extends to multireductions, as defined below. Formally, given $a, \mathcal{A}, \Delta$ coinitial with $\mathcal{B}$:

- a is **free from** $\mathcal{B}$, written $a \not\trianglerighteq \mathcal{B}$, iff $a \not> b$ for all $b \in \mathcal{B}$.
- $\mathcal{A}$ is **free from** $\mathcal{B}$, written $\mathcal{A} \not\trianglerighteq \mathcal{B}$, iff $a \not\trianglerighteq \mathcal{B}$ for all $a \in \mathcal{A}$.
- Δ is **free from** $\mathcal{B}$, written $\Delta \not\trianglerighteq \mathcal{B}$, iff either $\Delta = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$ or $\Delta = \mathcal{A}; \Delta'$, $\mathcal{A} \not\trianglerighteq \mathcal{B}$ and $\Delta' \not\trianglerighteq \mathcal{B}[\![\mathcal{A}]\!]$.
- a is **embedded by** $\mathcal{B}$, written $a > \mathcal{B}$, iff $a \notin \mathcal{B}$ and $\exists b \in \mathcal{B}$ s.t. $a > b$.
- $\mathcal{A}$ is **embedded by** $\mathcal{B}$, written $\mathcal{A} > \mathcal{B}$, iff $a > \mathcal{B}$ for all $a \in \mathcal{A}$.

Notice that being free from and embedded by $\mathcal{B}$ are complementary for a single (coinitial) step a, unless $a \in \mathcal{B}$, *i.e.* exactly one of $a \in \mathcal{B}$, $a \not\trianglerighteq \mathcal{B}$ and $a > \mathcal{B}$ holds. This need not be the case for a multistep $\mathcal{A}$: even if $\mathcal{A} \cap \mathcal{B} = \emptyset$, it could well be the case that neither $\mathcal{A} \not\trianglerighteq \mathcal{B}$ nor $\mathcal{A} > \mathcal{B}$ hold, if some elements of $\mathcal{A}$ are free from $\mathcal{B}$ while others are embedded by it.

On the other hand, any $\mathcal{A}$ verifying $\mathcal{A} \cap \mathcal{B} = \emptyset$ can be **partitioned** into a free subset $\mathcal{A}^F$ and an embedded subset $\mathcal{A}^E$ w.r.t. $\mathcal{B}$, *i.e.* $\mathcal{A} = \mathcal{A}^F \uplus \mathcal{A}^E$, $\mathcal{A}^F \not\trianglerighteq \mathcal{B}$, and $\mathcal{A}^E > \mathcal{B}$. The partition of a multistep into a free and an embedded part w.r.t. another, coinitial multistep, is a relevant notion for the development of the abstract proof we describe in Sec. 6.

[6] Observe that given the just discussed meanings, "$\mathcal{A} \not> \mathcal{B}$" and "$\mathcal{A}$ free from $\mathcal{B}$" are different predicates.

Consider the following multireduction Δ in the λ-calculus:

$$\overbrace{(\lambda x.x(\underbrace{I5}_{a}))(\underbrace{I3}_{b})}^{d}\ \overbrace{(I(\underbrace{I4}_{c})}^{e} \xrightarrow{\{e\}} \overbrace{(\lambda x.x(\underbrace{I5}_{a'}))(\underbrace{I3}_{b'})}^{d'}\ (\underbrace{I4}_{c'}) \xrightarrow{\{d',c'\}} (\underbrace{I3}_{b''})(\underbrace{I5}_{a''})\ 4$$

In this case, we have $\{c,d,e\} \not\supset \{a,b\}$, $\{a,b\} \not\supset \{c,e\}$, $\{a,b,c\} > \{d,e\}$. Also, we have $\Delta \not\supset \{a,b\}$, because $\{e\} \not\supset \{a,b\}$ and $\{d',c'\} \not\supset \{a',b'\}$. If we define $\mathcal{A} = \{b,c,e\}$ and $\mathcal{B} = \{a,d\}$, we observe that neither $\mathcal{A} \not\supset \mathcal{B}$ nor $\mathcal{A} > \mathcal{B}$ hold. The partition of $\mathcal{A}$ w.r.t. $\mathcal{B}$ gives $\mathcal{A}^F = \{c,e\}$ and $\mathcal{A}^E = \{b\}$.

Observe also that being free from a multistep extends to parts of a multireduction, namely[7]:

Lemma 5.2. *Assume $\Delta_1; \Delta_2; \Delta_3 \not\supset \mathcal{B}$. Then $\Delta_2 \not\supset \mathcal{B}[\![\Delta_1]\!]$.*

Proof. We proceed by induction on $\langle |\Delta_1|, |\Delta_2| \rangle$. Let Δ be $\Delta_1; \Delta_2; \Delta_3$.

The base case is when $\Delta_1 = \Delta_2 = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$. In this case $\mathcal{B}[\![\Delta_1]\!] = \mathcal{B}$. Then the definition of $\not\supset$ suffices to conclude.

Suppose that $\Delta_1 = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$ and $\Delta_2 = \mathcal{A}; \Delta_2'$. In this case, $\Delta = \mathcal{A}; \Delta_2'; \Delta_3$, so that $\Delta \not\supset \mathcal{B}$ implies $\mathcal{A} \not\supset \mathcal{B}$ and $\Delta_2'; \Delta_3 = \mathtt{nil}_{\mathsf{tgt}(\mathcal{A})}; \Delta_2'; \Delta_3 \not\supset \mathcal{B}[\![\mathcal{A}]\!]$. We observe that $\langle |\mathtt{nil}_{\mathsf{tgt}(\mathcal{A})}|, |\Delta_2'| \rangle < \langle |\Delta_1|, |\Delta_2| \rangle$, therefore we can apply *i.h.*, obtaining that $\Delta_2' \not\supset \mathcal{B}[\![\mathcal{A}]\!][\![\mathtt{nil}_{\mathsf{tgt}(\mathcal{A})}]\!] = \mathcal{B}[\![\mathcal{A}]\!]$. Recalling that $\mathcal{A} \not\supset \mathcal{B}$, we get $\Delta_2 \not\supset \mathcal{B} = \mathcal{B}[\![\Delta_1]\!]$.

If $\Delta_1 = \mathcal{A}; \Delta_1'$, then $\Delta \not\supset \mathcal{B}$ implies $\mathcal{A} \not\supset \mathcal{B}$ and $\Delta_1'; \Delta_2; \Delta_3 \not\supset \mathcal{B}[\![\mathcal{A}]\!]$. Observe $\langle |\Delta_1'|, |\Delta_2| \rangle < \langle |\Delta_1|, |\Delta_2| \rangle$, then *i.h.* yields $\Delta_2 \not\supset \mathcal{B}[\![\mathcal{A}]\!][\![\Delta_1']\!] = \mathcal{B}[\![\Delta_1]\!]$. □

The axiom Linearity can be extended to the residuals of a step *after a multistep* from which it is free from, as proved in the following Lemma. This fact is used in Sec. 6.

Lemma 5.3 *(Linearity after a multistep). Consider an ARS enjoying the fundamental axioms and Linearity, and $a, \mathcal{B}$ such that $a \not\supset \mathcal{B}$. Then $a[\![\mathcal{B}]\!]$ is a singleton.*

Proof. By induction on $\nu(\mathcal{B})$. If $\mathcal{B} = \emptyset$, then we conclude by observing that $a[\![\emptyset]\!] = \{a\}$. Otherwise assume some $b \in \mathcal{B}$. Then $a \not\supset \mathcal{B}$ implies $b \not\leq a$, thus Linearity yields $a[\![b]\!] = \{a'\}$. Let us show that $a' \not\supset \mathcal{B}[\![b]\!]$. Take b_0' such that $b_0[\![b]\!]b_0'$ for some $b_0 \in \mathcal{B}$. Assume $b_0' < a'$. Then $b \not\leq a$ and Context-Freeness imply $b_0 < a$ thus contradicting $a \not\supset \mathcal{B}$. On the other hand, $b_0' = a'$ would contradict Ancestor Uniqueness. Consequently, $a' \not\supset \mathcal{B}[\![b]\!]$. The *i.h.* can then be applied to obtain that $a'[\![\mathcal{B}[\![b]\!]]\!]$ is a singleton. We conclude by observing that $a[\![\mathcal{B}]\!] = a[\![b]\!][\![\mathcal{B}[\![b]\!]]\!] = a'[\![\mathcal{B}[\![b]\!]]\!]$. □

Gripping for multisteps

Now we discuss the extension of the gripping relation to multisteps. We say that:

- $\mathcal{B}$ **grips** a, written $a \ll \mathcal{B}$, iff $a \ll b$ for some $b \in \mathcal{B}$.
- $\mathcal{B}$ **grips** $\mathcal{A}$, written $\mathcal{A} \ll \mathcal{B}$, iff $a \ll \mathcal{B}$ for at least one $a \in \mathcal{A}$.

Informally, we say that a multistep $\mathcal{B}$ is *never-gripping* iff no residual of $\mathcal{B}$ (including $\mathcal{B}$ itself) grips any coinitial redex. Formally, we define $\mathcal{B}$ to be **never-gripping** (ng) iff for any finite multireduction Ψ, if Ψ is coinitial with $\mathcal{B}$, then $\mathcal{R}ed(\mathsf{tgt}(\Psi)) \not\ll \mathcal{B}[\![\Psi]\!]$. Notice that $\mathcal{B}$ being never-gripping implies that every residual of $\mathcal{B}$ (after a coinitial step, multistep or multiderivation) is.

It is worth noticing that an alternative definition of never-gripping can be given *coinductively* as follows: a multistep $\mathcal{B}$ is **coinductively never-gripping** (cng) iff $\mathcal{R}ed(\mathsf{src}(\mathcal{B})) \not\ll \mathcal{B}$, and for any multistep $\mathcal{A}$ coinitial with $\mathcal{B}$, the set $\mathcal{B}[\![\mathcal{A}]\!]$ is never-gripping. It is not difficult to show that both definitions are equivalent.

Lemma 5.4. *A multistep $\mathcal{B}$ is $\mathtt{ng}$ iff $\mathcal{B}$ is $\mathtt{cng}$.*

Proof.

$\Rightarrow$) The proof is by coinduction.

Take $\Psi = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$. Then $\mathcal{R}ed(\mathsf{src}(\mathcal{B})) = \mathcal{R}ed(\mathsf{src}(\mathtt{nil}_{\mathsf{src}(\mathcal{B})})) = \mathcal{R}ed(\mathsf{tgt}(\mathtt{nil}_{\mathsf{src}(\mathcal{B})})) \not\ll \mathcal{B}[\![\mathtt{nil}_{\mathsf{src}(\mathcal{B})}]\!] = \mathcal{B}$.

Take any multistep $\mathcal{A}$ which is coinitial with $\mathcal{B}$. By hypothesis $\mathcal{B}$ is ng so that $\mathcal{B}[\![\mathcal{A}]\!]$ is also ng. The coinductive hypothesis gives $\mathcal{B}[\![\mathcal{A}]\!]$ cng and we can thus conclude $\mathcal{B}$ cng.

$\Leftarrow$) Let Ψ be a finite multiderivation which is coinitial with $\mathcal{B}$. We want to show $\mathcal{R}ed(\mathsf{tgt}(\Psi)) \not\ll \mathcal{B}[\![\Psi]\!]$. We proceed by induction on Ψ.

[7] The statement of Lemma 5.2 could be seen as more general than needed. However, it is applied twice in the proof of Lemma 6.6, once with empty Δ_1 and once with empty Δ_3.

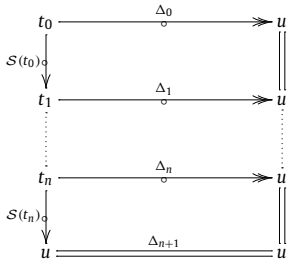


Fig. 5. Proof idea.

If $\Psi = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$, then $\mathcal{R}ed(\mathsf{tgt}(\mathtt{nil}_{\mathsf{src}(\mathcal{B})})) = \mathcal{R}ed(\mathsf{src}(\mathtt{nil}_{\mathsf{src}(\mathcal{B})})) = \mathcal{R}ed(\mathsf{src}(\mathcal{B})) \not\ll \mathcal{B} = \mathcal{B}[\![\mathtt{nil}_{\mathsf{src}(\mathcal{B})}]\!]$.
If $\Psi = \mathcal{A}; \Psi'$, then we want to show $\mathcal{R}ed(\mathsf{tgt}(\mathcal{A}; \Psi')) \not\ll \mathcal{B}[\![\mathcal{A}; \Psi']\!]$.
Since $\mathcal{B}$ is cng, then $\mathcal{B}[\![\mathcal{A}]\!]$ is cng by definition, so that by *i.h.* we have $\mathcal{R}ed(\mathsf{tgt}(\Psi')) \not\ll \mathcal{B}[\![\mathcal{A}]\!][\![\Psi']\!]$. We conclude since $\mathcal{R}ed(\mathsf{tgt}(\Psi')) = \mathcal{R}ed(\mathsf{tgt}(\mathcal{A}; \Psi'))$ and $\mathcal{B}[\![\mathcal{A}]\!][\![\Psi']\!] = \mathcal{B}[\![\mathcal{A}; \Psi']\!]$. □

The choice between the use of the predicate ng or cng remains a matter of taste, as the (forthcoming) proofs relying on ng are not simplified in any notable way by adopting instead cng. Still, since *never-gripping* plays a central rôle in this paper, explicitly spelling out its coinductive nature provides one more way of understanding it.

Another interesting remark concerns the relation between our *never-gripping* predicate and that of *universally* $<$*-external* [1]: a redex a is said to be **universally** $<$**-external**, *i.e.* external with respect to any reduction step (and thus w.r.t. any derivation) if a is $<$-minimal in $\mathcal{R}ed(\mathsf{src}(a))$, and $a[\![b]\!]a'$ implies a' universally $<$-external for all b coinitial with a. A redex which is universally $<$-external is in particular *never-gripping*, but the converse does not necessarily hold: a *never-gripping* redex is not always universally $<$-external. For example, in the case of the λ-calculus, the redex b in $\underline{I(I(\underline{II_c})_b)}_a$ is *never-gripping*, but is not universally $<$-external as b is not $<$-minimal. Another example can be found in Section 7.1.3, where we define a reduction strategy for **PPC**, which selects *never-gripping* redexes which are not necessarily universally $<$-external.

Uses, needed step and necessary multisteps
Given a multireduction and some coinitial multistep, a further property the abstract normalisation proof is interested in is whether the multistep is at least partially contracted along the multireduction, or if it is otherwise completely ignored. We will say that a multistep is **used** in a multireduction, iff at least one residual of the former is included (*i.e.* contracted) in the latter. Formally, let b be a step, $\mathcal{A}$ and $\mathcal{B}$ two multisteps, and Δ a multireduction, such that all of them are coinitial.

- $\mathcal{A}$ **uses** b iff $b \in \mathcal{A}$;
- Δ **uses** b iff $\Delta[k] \cap (b[\![\Delta[1..k-1]]\!]) \neq \emptyset$ for at least one k; and
- $\mathcal{A}$ (resp. Δ) **uses** $\mathcal{B}$ iff it uses at least one $b \in \mathcal{B}$.

A step a is **needed** iff for every multireduction $\mathsf{src}(a) \overset{\Delta}{\multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow} u$ such that u is a normal form, Δ uses a. A multistep $\mathcal{A}$ is **necessary**, iff for every multireduction $\mathsf{src}(\mathcal{A}) \overset{\Delta}{\multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow} u$ such that u is a normal form, Δ uses $\mathcal{A}$. The notion of necessary multistep generalises that of needed redex (notice that any singleton whose only element is a needed redex is a necessary set). As mentioned in the introduction, there is an important difference: while not all terms admit a needed redex, any term admits at least one necessary set, *i.e.* the set of *all* its redexes.

6. Necessary normalisation for ARS

We prove in this section that, for any ARS verifying the fundamental axioms, the embedding axioms, and the gripping axioms, the systematic contraction of *necessary* and *never-gripping* multisteps is normalising.

The overall structure of the proof is inspired by the work on first-order term rewriting systems by Sekar and Ramakrishnan in [22]. Assume that $\mathcal{S}$ is a reduction strategy selecting always necessary and never-gripping multisteps. Consider an initial multireduction $t_0 \overset{\Delta_0}{\multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow} u \in \boldsymbol{NF}$, and t_1 the target term of the multistep selected by $\mathcal{S}$ for t_0, *i.e.* $t_0 \overset{\mathcal{S}(t_0)}{\multimap\!\!\!\!\circ\!\!\!\!\rightarrow} t_1$. We construct a multireduction $t_1 \overset{\Delta_1}{\multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow} u$, such that the multireduction Δ_1 is strictly smaller than the original one w.r.t. a convenient well-founded ordering $<$ on multireductions. We have thus transformed the original $t_0 \overset{\Delta_0}{\multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow} u$ in $t_0 \overset{\mathcal{S}(t_0)}{\multimap\!\!\!\!\circ\!\!\!\!\rightarrow} t_1 \overset{\Delta_1}{\multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow} u$. Well-foundedness of $<$ entails that by iterating this procedure one may deduce that repeated contraction of the multisteps selected by the strategy $\mathcal{S}$ yields the normal form u. This is depicted in Fig. 5 where Δ_{k+1} is strictly smaller than Δ_k for all k and Δ_{n+1} is a trivial multireduction. The original multireduction Δ_0 is first transformed into $\mathcal{S}(t_0); \Delta_1$, then successively into $\mathcal{S}(t_0); \ldots; \mathcal{S}(t_k); \Delta_{k+1}$; and finally into $\mathcal{S}(t_0); \ldots; \mathcal{S}(t_n)$.

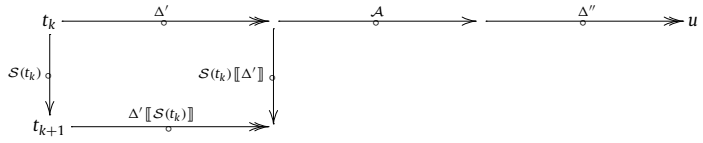


Fig. 6. Construction of Δ_{k+1}, starting point.

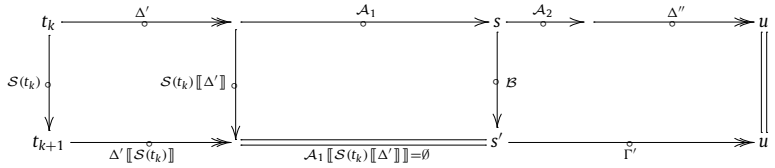


Fig. 7. Construction of Δ_{k+1}, finished.

Several notions contribute to this proof. We define a **measure** inspired from [22,23], based on the *depths* of the multisteps composing a multireduction. More precisely, the measure of a multireduction is defined as the sequence of the depths of its elements (see definition of depth on page 41), *taken in reverse order*, *i.e.* given a multireduction $\Delta = \Delta[1..n]$, the measure of Δ, written $\chi(\Delta)$, is the n-tuple $\langle \nu(\Delta[n]), \ldots, \nu(\Delta[1])\rangle$. Then, the **lexicographic order** $<_{lex}$ is used to compare (measures of) multireductions, where $<_{lex}$ is defined on n-tuples of natural numbers as follows:

$$\langle x_1, \ldots, x_n\rangle <_{lex} \langle y_1, \ldots, y_n\rangle \text{ iff } \exists 1 \leq j \leq n \,.\, x_j < y_j \text{ and } \forall 1 \leq i < j \,.\, x_i = y_i$$

Therefore, $\chi(\Delta) < \chi(\Gamma)$ implies $\chi(\Pi;\Delta) < \chi(\Psi;\Gamma)$ for any Π, Γ that verify $\mathsf{tgt}(\Pi) = \mathsf{src}(\Delta)$, $\mathsf{tgt}(\Psi) = \mathsf{src}(\Gamma)$, and $|\Pi| = |\Psi|$, because the elements of a multireduction are taken in reversed order. Notice that multireductions comparable by $<_{lex}$ may not be coinitial.

This (well-founded) ordering allows only to compare multireductions having the same length; the minimal elements are the n-tuples of the form $\langle 0, \ldots, 0\rangle$ which corresponds exactly to the trivial multireductions. As remarked in [23], the measure used in [22], based on sizes of multisteps rather than depths, is not well-suited for a higher-order setting.

To construct Δ_{k+1}, we observe that the fact that $\mathcal{S}(t_k)$ is a *necessary* set, implies that it is used along Δ_k at least once. Therefore, we can consider the last element of Δ_k that includes (some residual of) an element of $\mathcal{S}(t_k)$. Let us call this element $\mathcal{A}$. We build the diagram shown in Fig. 6, where $\Delta_k = \Delta'; \mathcal{A}; \Delta''$, $\mathcal{A} \cap \mathcal{S}(t_k)[\![\Delta']\!] \neq \emptyset$, and Δ'' does not use $\mathcal{S}(t_k)[\![\Delta'; \mathcal{A}]\!]$.

By setting $\mathcal{A}_1 = \mathcal{A} \cap \mathcal{S}(t_k)[\![\Delta']\!] \neq \emptyset$, $\mathcal{A}_2 = (\mathcal{A} \setminus \mathcal{A}_1)[\![\mathcal{A}_1]\!]$, and $\mathcal{B} = \mathcal{S}(t_k)[\![\Delta'; \mathcal{A}_1]\!]$, we can refine the previous diagram as depicted in Fig. 7. Now $\mathcal{A}_2; \Delta''$ does not use $\mathcal{B}$. Notice that $\mathcal{A}_1 \neq \emptyset$ implies $\nu(\mathcal{A}_2) < \nu(\mathcal{A})$. Observe also that $\mathcal{A}_1 \subseteq \mathcal{S}(t_k)[\![\Delta']\!]$, implying $\mathcal{A}_1[\![\mathcal{S}(t_k)[\![\Delta']\!]]\!] = \emptyset$.

To conclude the construction of Δ_{k+1}, it suffices to obtain a multireduction Γ' such that $s' \xrightarrow{\Gamma'} u$ and $\chi(\Gamma') \leq_{lex} \chi(\mathcal{A}_2; \Delta'') <_{lex} \chi(\mathcal{A}; \Delta'')$; taking the multisteps of a multireduction in *reversed* order in the measure allows to assert $\chi(\Delta_{k+1}) <_{lex} \chi(\Delta_k)$. Building such a Γ' is the most demanding part of the proof. Following [22], this construction is based on the following observations:

- **Partition of each multistep in free and embedded parts.** Each multistep comprising $\mathcal{A}_2; \Delta''$ can be partitioned into a free and an embedded part w.r.t. $\mathcal{B}$, as remarked in Sec. 5.2 after the definition of free and embedded multisteps.
- **Postponement of embedded parts.** We prove that each embedded part can be postponed, *i.e.* permuted with a subsequent free part, preserving the free and embedded nature of the permuted multisteps, and the depth of the free part (*cf.* Lemmas 6.4, 6.5 and 6.6). We describe this phenomenon in more detail at the beginning of Sec. 6.2.
- **Irrelevance of postponed embedded parts.** Since $\mathcal{B}$ is not used and $u \in \boldsymbol{NF}$ implies $\mathcal{B}[\![\mathcal{A}_2; \Delta'']\!] = \emptyset$, we prove that the (postponed) embedded part can be simply ignored when defining Γ' (*cf.* Lemmas 6.7 and 6.8).
- **Measure of free multireduction does not increase in projection.** Since $\mathcal{S}(t_k)$ is never-gripping, hence also $\mathcal{B}$ is never-gripping, the depth of the free part of each multistep can be proven greater or equal than that of its residual after (the corresponding residual of) $\mathcal{B}$ (*cf.* Lemmas 6.2 and 6.3). This is the reason for the introduction of gripping, which then allows to apply the general structure of the proof in [22] in the abstract setting of this work.

We describe the details in the remainder of this section.

6.1. Relevance of gripping

This section develops the key properties that show the relevance of gripping in the abstract normalisation proof, as described in Section 4.3. To this end, we introduce a strengthened variant of the free from relation. Given two coinitial multisteps $\mathcal{A}$ and $\mathcal{B}$, we say that $\mathcal{A}$ is **independent** from $\mathcal{B}$, iff $\mathcal{A} \not\sqsupset \mathcal{B}$ and $\mathcal{A} \not\ll \mathcal{B}$. Analogously, if Δ and $\mathcal{B}$ are coinitial, we say that Δ is **independent** from $\mathcal{B}$, iff $\Delta \not\sqsupset \mathcal{B}$ and $\Delta[k] \not\ll \mathcal{B}[\![\Delta[1..k-1]]\!]$ for all k.

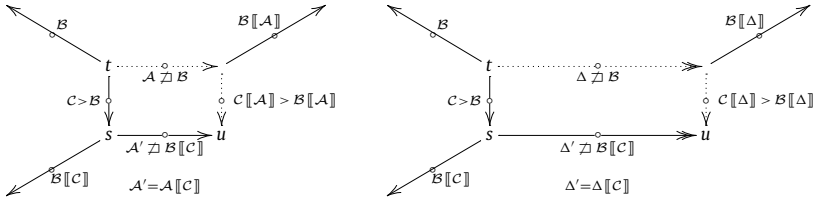


Fig. 8. Postponement of embedded multisteps: the one step and multiple step cases; Lemma 6.5 and Lemma 6.6 respectively.

We prove the preservation of the depth of a multistep $\mathcal{A}$ after the contraction of a multistep $\mathcal{B}$, if $\mathcal{A}$ is independent from $\mathcal{B}$ (*cf.* Lemma 6.2), and the extension of such preservation to the measure of multireductions after the contraction of a never-gripping set (*cf.* Lemma 6.3).

Lemma 6.1 *(Independence preservation). Consider $\mathcal{A}$ independent from $\mathcal{B}$, and $d \in \mathcal{A}$. Then $\mathcal{A}[\![d]\!]$ is independent from $\mathcal{B}[\![d]\!]$.*

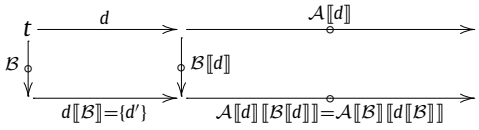
Proof. If $\mathcal{B} = \emptyset$, then the result holds trivially since also $\mathcal{B}[\![d]\!] = \emptyset$. So assume $b \in \mathcal{B}$. Next, we may assume some $a \in \mathcal{A}$ s.t. $a \neq d$. Otherwise $\mathcal{A}[\![d]\!] = \emptyset$ and the result also holds trivially. For the same reason, we may assume $a[\![d]\!]a'$ for some a'. Similarly, we may assume there exists b' s.t. $b[\![d]\!]b'$.

The hypotheses implies the following: $b \not\leq a$, $b \not\leq d$, $a \not\ll b$, and $d \not\ll b$.

Observe $b' = a'$ would contradict Ancestor Uniqueness. On the other hand, $b' < a'$ would imply $b < a \vee (d \ll b \wedge d < a)$ by Grip–Instantiation, while $a' \ll b'$ would imply $a \ll b \vee a \ll d \ll b$ by Grip–Density. Therefore, either case would contradict the hypotheses. Thus we conclude. □

Lemma 6.2 *(Depth preservation). Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{R}ed(t)$ such that $\mathcal{A}$ is independent from $\mathcal{B}$. Then $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![\mathcal{B}]\!])$.*

Proof. By induction on $\nu(\mathcal{A})$. If $\mathcal{A} = \emptyset$, then $\mathcal{A}[\![\mathcal{B}]\!] = \emptyset$ and we conclude. Otherwise, let $\delta = d; \delta'$ such that $\delta \Vdash \mathcal{A}$ and $\nu(\mathcal{A}) = |\delta|$. Observe that $\delta' \Vdash \mathcal{A}[\![d]\!]$, implying $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![d]\!]) + 1$. Lemma 6.1 allows to apply the *i.h.*, obtaining $\nu(\mathcal{A}[\![d]\!]) = \nu(\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!])$, so that Proposition 5.1:(1) yields $\nu(\mathcal{A}[\![d]\!]) = \nu(\mathcal{A}[\![\mathcal{B}]\!][\![d[\![\mathcal{B}]\!]]\!])$. In turn, Lemma 5.3 implies $d[\![\mathcal{B}]\!] = \{d'\}$ for some step d'.



Therefore, for any γ such that $\gamma \Vdash \mathcal{A}[\![\mathcal{B}]\!][\![d[\![\mathcal{B}]\!]]\!]$, we have $d'; \gamma \Vdash \mathcal{A}[\![\mathcal{B}]\!]$. Consequently, $\nu(\mathcal{A}) \leq \nu(\mathcal{A}[\![d]\!]) + 1 = \nu(\mathcal{A}[\![\mathcal{B}]\!][\![d[\![\mathcal{B}]\!]]\!]) + 1 \leq \nu(\mathcal{A}[\![\mathcal{B}]\!])$.

Conversely, consider $\gamma = e'; \gamma'$ such that $\gamma \Vdash \mathcal{A}[\![\mathcal{B}]\!]$ and $\nu(\mathcal{A}[\![\mathcal{B}]\!]) = |\gamma|$; observe that $\gamma' \Vdash \mathcal{A}[\![\mathcal{B}]\!][\![e']\!]$. Let $e \in \mathcal{A}$ such that $e[\![\mathcal{B}]\!]e'$. Lemma 5.3 implies $e[\![\mathcal{B}]\!] = \{e'\}$, implying that $\gamma' \Vdash \mathcal{A}[\![\mathcal{B}]\!][\![e[\![\mathcal{B}]\!]]\!]$, so that Proposition 5.1:(1) yields $\gamma' \Vdash \mathcal{A}[\![e]\!][\![\mathcal{B}[\![e]\!]]\!]$. Therefore, $\nu(\mathcal{A}[\![\mathcal{B}]\!]) \leq \nu(\mathcal{A}[\![e]\!][\![\mathcal{B}[\![e]\!]]\!]) + 1$. Again, Lemma 6.1 allows to apply the *i.h.*, to obtain $\nu(\mathcal{A}[\![e]\!]) = \nu(\mathcal{A}[\![e]\!][\![\mathcal{B}[\![e]\!]]\!])$. In turn, for any δ such that $\delta \Vdash \mathcal{A}[\![e]\!]$, we get $e; \delta \Vdash \mathcal{A}$. Consequently, $\nu(\mathcal{A}[\![\mathcal{B}]\!]) \leq \nu(\mathcal{A}[\![e]\!][\![\mathcal{B}[\![e]\!]]\!]) + 1 = \nu(\mathcal{A}[\![e]\!]) + 1 \leq \nu(\mathcal{A})$. Thus we conclude. □

The following result lifts Lemma 6.2 to multireductions. By replacing local absence of gripping to the hereditary never-gripping property, we enforce independence of the multireduction from the given multistep. Hence, invariance of depth for a multistep can be lifted to invariance of measure for a multireduction.

Lemma 6.3 *(Measure preservation). Let Δ be a multireduction and $\mathcal{B}$ a multistep, such that Δ and $\mathcal{B}$ are coinitial, $\mathcal{B}$ is never-gripping and $\Delta \not\sqsupset \mathcal{B}$. Then $\chi(\Delta) = \chi(\Delta[\![\mathcal{B}]\!])$.*

Proof. We proceed by induction on $|\Delta|$. If $\Delta = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$, then $\Delta[\![\mathcal{B}]\!] = \mathtt{nil}_{\mathsf{tgt}(\mathcal{B})}$, so we conclude immediately. Assume, therefore, $\Delta = \mathcal{A}; \Delta'$, so that $\Delta[\![\mathcal{B}]\!] = \mathcal{A}[\![\mathcal{B}]\!]; \Delta'[\![\mathcal{B}[\![\mathcal{A}]\!]]\!]$. Observe $\mathcal{A} \not\sqsupset \mathcal{B}$, $\mathcal{A} \not\ll \mathcal{B}$, $\Delta' \not\sqsupset \mathcal{B}[\![\mathcal{A}]\!]$ and $\mathcal{B}[\![\mathcal{A}]\!]$ is never-gripping. Then Lemma 6.2 implies $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![\mathcal{B}]\!])$, and the *i.h.* on Δ' yields $\chi(\Delta') = \chi(\Delta'[\![\mathcal{B}[\![\mathcal{A}]\!]]\!])$. Thus we conclude. □

6.2. Normalisation proof

The next ingredient in the normalisation proof is the ability to *postpone* an embedded multistep after a free multistep or multireduction. The situation is described in Fig. 8. The diagram on the left shows that an embedded (by $\mathcal{B}$) multistep can be *postponed* after a free (from $\mathcal{B}[\![\mathcal{C}]\!]$) one, yielding a multireduction in which the free multistep precedes the embedded one

(Lemma 6.5). Moreover, the *depth of the free multistep is preserved* by the postponement. To enforce this we show, resorting to Grip–Convexity, that the embedded multistep does not grip the (ancestor of the) free one, so that Lemma 6.2 can be applied. This is the only rôle of Grip–Convexity in the normalisation proof.

The diagram on the right shows that a embedded multistep can be postponed after a free *multireduction* as well (Lemma 6.6).

We observe that the only rôle of the added Pivot axiom in the normalisation proof, is to verify that $\mathcal{C}[\![\mathcal{A}]\!] > \mathcal{B}[\![\mathcal{A}]\!]$ in the left-hand side diagram.

Lemma 6.4 *(Embedding preservation). Let $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \mathcal{R}ed(t)$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$ and $\mathcal{C} > \mathcal{B}$. Then $\mathcal{C}[\![\mathcal{A}]\!] > \mathcal{B}[\![\mathcal{A}]\!]$.*

Proof. By induction on $\nu(\mathcal{A})$. If $\mathcal{A} = \emptyset$, then $\mathcal{C}[\![\mathcal{A}]\!] = \mathcal{C}$ and $\mathcal{B}[\![\mathcal{A}]\!] = \mathcal{B}$, so that we conclude immediately. Otherwise, consider $a \in \mathcal{A}$ and $c' \in \mathcal{C}[\![a]\!]$ (if $\mathcal{C}[\![a]\!] = \emptyset$, then $\mathcal{C}[\![\mathcal{A}]\!] = \emptyset$ and $\mathcal{C}[\![\mathcal{A}]\!] > \mathcal{B}[\![\mathcal{A}]\!]$ holds trivially). Let $c \in \mathcal{C}$ such that $c' \in c[\![a]\!]$. Note that $a \neq c$ for otherwise $c[\![a]\!] = \emptyset$. We will verify the existence of some $b' \in \mathcal{B}[\![a]\!]$ such that $b' < c'$, so that $\mathcal{C}[\![a]\!] > \mathcal{B}[\![a]\!]$. Let $b \in \mathcal{B}$ be such that $b < c$, as follows from the hypothesis. Observe that $a = b$ or $a = c$ would contradict, respectively, the hypotheses of this lemma or our observation above on the existence of c'. Therefore $a \neq b$ and $a \neq c$. We consider two cases.

1. **Case** $a \not< c$**.** Then $b < c$ implies $a \not< b$, so that Linearity implies $b[\![a]\!] = \{b'\}$, and then Context-Freeness applies to obtain $b' < c'$.
2. **Case** $a < c$**.** If $b < a$, *i.e.* $b < a < c$, then Linearity implies $b[\![a]\!] = \{b'\}$ (since $a \not< b$), and therefore Enclave–Embedding applies to obtain $b' < c'$. Otherwise, we have $a < c$, $b < c$ and $b \not< a$, then Pivot applies to obtain $b[\![a]\!]b'$ and $b' < c'$ for some b'.

Hence, we have verified $\mathcal{C}[\![a]\!] > \mathcal{B}[\![a]\!]$. Moreover, Ancestor Uniqueness yields $\mathcal{A}[\![a]\!] \cap \mathcal{B}[\![a]\!] = \emptyset$. Consequently, we can apply the *i.h.* on $\mathcal{A}[\![a]\!]$, obtaining $\mathcal{C}[\![a]\!][\![\mathcal{A}[\![a]\!]]\!] > \mathcal{B}[\![a]\!][\![\mathcal{A}[\![a]\!]]\!]$. Thus we conclude. □

Lemma 6.5 *(Postponement after a multistep). Let $\mathcal{B} \subseteq \mathcal{R}ed(t)$, $t \xrightarrow[\mathcal{O}]{\mathcal{C}} s \xrightarrow[\mathcal{O}]{\mathcal{A}'} u$, such that $\mathcal{C} > \mathcal{B}$, $\mathcal{A}' \not\supset \mathcal{B}[\![\mathcal{C}]\!]$ and $\mathcal{B}$ is never-gripping. Then there exists $\mathcal{A} \subseteq \mathcal{R}ed(t)$ s.t. $\mathcal{A}' = \mathcal{A}[\![\mathcal{C}]\!]$, $\mathcal{A} \not\supset \mathcal{B}$ and $\nu(\mathcal{A}) = \nu(\mathcal{A}')$.*

Proof. If $\mathcal{A}' = \emptyset_s$, then taking $\mathcal{A} = \emptyset_t$ suffices to conclude. So we assume $\mathcal{A}' \neq \emptyset_s$ and proceed by induction on $\nu(\mathcal{C})$. If $\mathcal{C} = \emptyset$, *i.e.* $s = t$, then we conclude by taking $\mathcal{A}' := \mathcal{A}$; observe that in this case $\mathcal{B}[\![\mathcal{C}]\!] = \mathcal{B}$.

Consider $c \in \mathcal{C}$ and $t \xrightarrow{c} t_0 \xrightarrow[\mathcal{O}]{\mathcal{C}[\![c]\!]} s$. Since $\mathcal{C} > \mathcal{B}$, $c \notin \mathcal{B}$ and hence $\{c\} \cap \mathcal{B} = \emptyset$; we can apply Lemma 6.4 to obtain $\mathcal{C}[\![c]\!] > \mathcal{B}[\![c]\!]$. Moreover $\mathcal{B}[\![\mathcal{C}]\!] = \mathcal{B}[\![c]\!][\![\mathcal{C}[\![c]\!]]\!]$, and $\mathcal{B}$ never-gripping implies $\mathcal{B}[\![c]\!]$ never-gripping. Therefore, the *i.h.* on $\mathcal{C}[\![c]\!]$ yields the existence of some $\mathcal{A}'' \subseteq \mathcal{R}ed(t_0)$ such that $\mathcal{A}' = \mathcal{A}''[\![\mathcal{C}[\![c]\!]]\!]$, $\mathcal{A}'' \not\supset \mathcal{B}[\![c]\!]$ and $\nu(\mathcal{A}'') = \nu(\mathcal{A}')$. Hence, to conclude the proof, it suffices to verify the existence of some $\mathcal{A} \subseteq \mathcal{R}ed(t)$ verifying (1) $\mathcal{A}'' = \mathcal{A}[\![c]\!]$, (2) $\mathcal{A} \not\supset \mathcal{B}$ and (3) $\nu(\mathcal{A}) = \nu(\mathcal{A}'')$. Observe that $\mathcal{A}' \neq \emptyset_s$ and $\nu(\mathcal{A}'') = \nu(\mathcal{A}')$ imply $\mathcal{A}'' \neq \emptyset_{t_0}$.

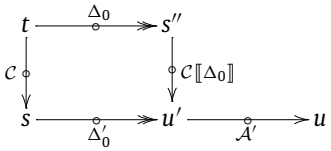
1. Let $b_0 \in \mathcal{B}$ such that $b_0 < c$, so that Linearity implies $b_0[\![c]\!] = \{b_0''\}$. Let $a'' \in \mathcal{A}''$. Then a'' being created by c would imply $b_0'' < a''$ by Enclave–Creation, contradicting $\mathcal{A}'' \not\supset \mathcal{B}[\![c]\!]$. Therefore, $a[\![c]\!]a''$ for some a. Let $\mathcal{A} := \{a \in \mathcal{R}ed(t) \text{ s.t. } \exists a'' \in \mathcal{A}''.\ a[\![c]\!]a''\}$. Observe that $\mathcal{A}'' \subseteq \mathcal{A}[\![c]\!]$ and let us show that also $\mathcal{A}[\![c]\!] \subseteq \mathcal{A}''$.
   Let $a_0 \in \mathcal{A}[\![c]\!]$, $a \in \mathcal{A}$ such that $a[\![c]\!]a_0$, $a'' \in \mathcal{A}''$ such that $a[\![c]\!]a''$. Observe that $c < a$ would imply $b_0 < c < a$, and then $b_0'' < a''$ by Enclave–Embedding, contradicting $\mathcal{A}'' \not\supset \mathcal{B}[\![c]\!]$. Moreover, $c = a$ would contradict $a[\![c]\!]a''$, *cf.* Self Reduction. Therefore $c \not< a$, so that Linearity applies yielding that $a[\![c]\!]$ is a singleton, hence $a_0 = a'' \in \mathcal{A}''$. Consequently, $\mathcal{A}[\![c]\!] \subseteq \mathcal{A}''$, and then $\mathcal{A}[\![c]\!] = \mathcal{A}''$.
2. Let $a \in \mathcal{A}$ and $b \in \mathcal{B}$. If b is minimal in $\mathcal{B}$ w.r.t. $<$, then $\mathcal{C} > \mathcal{B}$ implies $b_0 < c$ for some $b_0 \in \mathcal{B}$, therefore $c \not< b$ (since $c \leq b$ would contradict minimality of b), hence $b[\![c]\!] = \{b''\}$ by Linearity. Let $a'' \in \mathcal{A}''$ such that $a[\![c]\!]a''$. Observe that we have already verified that $c \neq a$. Then $b < a$ would imply $b'' < a''$ by Context-Freeness, contradicting $\mathcal{A}'' \not\supset \mathcal{B}[\![c]\!]$; hence, $b \neq a$. In turn, if b is not minimal in $\mathcal{B}$ w.r.t. $<$, then well-foundedness of $<$ implies the existence of some b_0 such that $b_0 < b$ and b_0 is minimal in $\mathcal{B}$ w.r.t. $<$, so that $b_0 \neq a$ as we have just shown, and therefore $b \neq a$. Consequently, $\mathcal{A} \not\supset \mathcal{B}$.
3. Consider $b_0 \in \mathcal{B}$ such that $b_0 < c$ and $a \in \mathcal{A}$. Observe that $a \ll c$ would imply either $a \ll b_0$ or $b_0 \leq a$ by Grip–Convexity, contradicting $\mathcal{B}$ being never-gripping and $\mathcal{A} \not\supset \mathcal{B}$ respectively. Therefore $\mathcal{A} \not\ll c$, and moreover $\mathcal{A} \not\supset c$ (recall $c \not< a$ for any $a \in \mathcal{A}$). Hence we can apply Lemma 6.2 to obtain $\nu(\mathcal{A}) = \nu(\mathcal{A}'')$. Thus we conclude. □

The next result extends Lemma 6.5 to multireductions: a multistep embedded by $\mathcal{B}$ may be *postponed* after a multireduction free from the same $\mathcal{B}$, without affecting neither the free-from and embedding relations w.r.t. (the corresponding residual of) $\mathcal{B}$, nor the measure of the "free" multireduction.

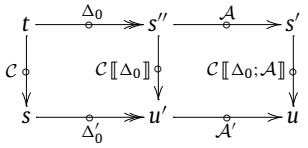
Lemma 6.6 *(Postponement after a multireduction). Let $t \xrightarrow{\mathcal{C}}_{\circ} s \xrightarrow{\Delta'}_{\circ\!\!\rightarrow} u$ and $\mathcal{B} \subseteq \mathcal{R}ed(t)$ such that $\mathcal{B}$ is never-gripping, $\mathcal{C} > \mathcal{B}$, and $\Delta' \not\supset \mathcal{B}[\![\mathcal{C}]\!]$. Then there exists some multireduction Δ verifying $\Delta' = \Delta[\![\mathcal{C}]\!]$, so that Proposition 5.1:(2) yields $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} s' \xrightarrow{\mathcal{C}[\![\Delta]\!]}_{\circ} u$ for some object s'; and moreover $\Delta \not\supset \mathcal{B}$, $\mathcal{C}[\![\Delta]\!] > \mathcal{B}[\![\Delta]\!]$, and $\chi(\Delta) = \chi(\Delta')$.*

Proof. By induction on $|\Delta'|$. If $\Delta' = \mathtt{nil}_s$, *i.e.* $u = s$, then it suffices to take $\Delta := \mathtt{nil}_t$, so that $s' = t$.

Assume $\Delta' = \Delta'_0; \mathcal{A}'$, so that $t \xrightarrow{\mathcal{C}}_{\circ\!\!\rightarrow} s \xrightarrow{\Delta'_0}_{\circ\!\!\rightarrow} u' \xrightarrow{\mathcal{A}'}_{\circ} u$. Lemma 5.2 on $\mathtt{nil}_s; \Delta'_0; \mathcal{A}'$ implies $\Delta'_0 \not\supset \mathcal{B}[\![\mathcal{C}]\!]$. Then we can apply the *i.h.* on Δ'_0 obtaining $\Delta'_0 = \Delta_0[\![\mathcal{C}]\!]$ for some multireduction Δ_0, such that $t \xrightarrow{\Delta_0}_{\circ\!\!\rightarrow} s'' \xrightarrow{\mathcal{C}[\![\Delta_0]\!]}_{\circ} u'$ for some object s'', and moreover $\Delta_0 \not\supset \mathcal{B}$, $\mathcal{C}[\![\Delta_0]\!] > \mathcal{B}[\![\Delta_0]\!]$, and $\chi(\Delta_0) = \chi(\Delta'_0)$. We can thus build the following diagram:



On the other hand, $\Delta' \not\supset \mathcal{B}[\![\mathcal{C}]\!]$ implies $\mathcal{A}' \not\supset \mathcal{B}[\![\mathcal{C}; \Delta'_0]\!]$ (*cf.* Lemma 5.2, now on $\Delta'_0; \mathcal{A}'; \mathtt{nil}_u$), therefore Proposition 5.1:(2) yields $\mathcal{A}' \not\supset \mathcal{B}[\![\Delta_0; \mathcal{C}[\![\Delta_0]\!]]\!] = \mathcal{B}[\![\Delta_0]\!][\![\mathcal{C}[\![\Delta_0]\!]]\!]$. Moreover, $\mathcal{B}$ never-gripping implies $\mathcal{B}[\![\Delta_0]\!]$ never-gripping. Hence we can apply Lemma 6.5 to $s'' \xrightarrow{\mathcal{C}[\![\Delta_0]\!]}_{\circ} u'$, obtaining that $\mathcal{A}' = \mathcal{A}[\![\mathcal{C}[\![\Delta_0]\!]]\!]$ for some $\mathcal{A} \subseteq \mathcal{R}ed(s'')$ verifying $\mathcal{A} \not\supset \mathcal{B}[\![\Delta_0]\!]$ and $\nu(\mathcal{A}) = \nu(\mathcal{A}')$. Consequently, we can complete the previous diagram as follows.



We define $\Delta := \Delta_0; \mathcal{A}$. Given $\mathcal{C}[\![\Delta_0]\!] > \mathcal{B}[\![\Delta_0]\!]$ and $\mathcal{A} \not\supset \mathcal{B}[\![\Delta_0]\!]$, so that $\mathcal{A} \cap \mathcal{B}[\![\Delta_0]\!] = \emptyset$, Lemma 6.4 applied on $\mathcal{A}$ implies $\mathcal{C}[\![\Delta]\!] > \mathcal{B}[\![\Delta]\!]$.

Moreover, given $\Delta_0 \not\supset \mathcal{B}$ and $\mathcal{A} \not\supset \mathcal{B}[\![\Delta_0]\!]$, a simple induction on $|\Delta_0|$ yields $\Delta \not\supset \mathcal{B}$. Finally, $\chi(\Delta) = \chi(\Delta')$ is immediate. Thus we conclude. □

The postponement result is used to show that, whenever $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} u$ and $\mathcal{B} \subseteq \mathcal{R}ed(t)$ is never-gripping and not used in Δ, and $\mathcal{B}[\![\Delta]\!] = \emptyset$, all activity embedded by (the successive residuals of) $\mathcal{B}$ is irrelevant, *i.e.* it can be omitted without compromising the target object u, and moreover without increasing the measure. Therefore, the embedded part of each multistep in $\mathcal{A}_2; \Delta''$ can just be discarded in the construction of Δ_{k+1}, *cf.* Fig. 7 on page 49.

Lemma 6.7 *(Irrelevance of one multistep). Let $t \xrightarrow{\mathcal{C}}_{\circ} s \xrightarrow{\Delta'}_{\circ\!\!\rightarrow} u$ and $\mathcal{B} \subseteq \mathcal{R}ed(t)$, such that $\mathcal{B}$ is never-gripping, $\mathcal{C} > \mathcal{B}$, $\Delta' \not\supset \mathcal{B}[\![\mathcal{C}]\!]$, and $\mathcal{B}[\![\mathcal{C}; \Delta']\!] = \emptyset$. Then there is a multireduction Δ such that $\Delta' = \Delta[\![\mathcal{C}]\!]$, $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} u$, $\Delta \not\supset \mathcal{B}$, $\mathcal{B}[\![\Delta]\!] = \emptyset$ and $\chi(\Delta) = \chi(\Delta')$.*

Proof. Lemma 6.6 implies the existence of Δ such that $\Delta' = \Delta[\![\mathcal{C}]\!]$, $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} s' \xrightarrow{\mathcal{C}[\![\Delta]\!]}_{\circ} u$, $\Delta \not\supset \mathcal{B}$, $\mathcal{C}[\![\Delta]\!] > \mathcal{B}[\![\Delta]\!]$, and $\chi(\Delta) = \chi(\Delta')$. Then $\mathcal{B}[\![\Delta]\!][\![\mathcal{C}[\![\Delta]\!]]\!] = \mathcal{B}[\![\Delta; \mathcal{C}[\![\Delta]\!]]\!] = \mathcal{B}[\![\mathcal{C}; \Delta']\!] = \emptyset$; *cf.* Proposition 5.1:(2). We will show that $\mathcal{B}[\![\Delta]\!] = \emptyset$, and also that $\mathcal{C}[\![\Delta]\!] = \emptyset$ implying $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} u$.

Assume for contradiction the existence of some $b \in \mathcal{B}[\![\Delta]\!]$, we assume wlog that b is minimal in $\mathcal{B}[\![\Delta]\!]$ w.r.t. $<$ (recall that $<$ is assumed well-founded). Then $\mathcal{C}[\![\Delta]\!] > \mathcal{B}[\![\Delta]\!]$ implies $b \not\supset \mathcal{C}[\![\Delta]\!]$, so that Lemma 5.3 yields $b[\![\mathcal{C}[\![\Delta]\!]]\!] = \{b'\}$, contradicting $(\mathcal{B}[\![\Delta]\!])[\![\mathcal{C}[\![\Delta]\!]]\!] = \emptyset$. Therefore $\mathcal{B}[\![\Delta]\!] = \emptyset$. In turn, the existence of some $c \in \mathcal{C}[\![\Delta]\!]$ would imply the existence of some $b \in \mathcal{B}[\![\Delta]\!]$ such that $b < c$, contradicting $\mathcal{B}[\![\Delta]\!] = \emptyset$. Therefore $\mathcal{C}[\![\Delta]\!] = \emptyset$, implying $u = s'$ so that $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} u$. Thus we conclude. □

Lemma 6.8 *(Irrelevance of many multisteps). Let $t \xrightarrow{\Delta}_{\circ\!\!\rightarrow} u$ and $\mathcal{B} \subseteq \mathcal{R}ed(t)$, such that $\mathcal{B}$ is never-gripping, Δ does not use $\mathcal{B}$, and $\mathcal{B}[\![\Delta]\!] = \emptyset$. Then there exists a multireduction Γ such that $t \xrightarrow{\Gamma}_{\circ\!\!\rightarrow} u$, $\Gamma \not\supset \mathcal{B}$, $\mathcal{B}[\![\Gamma]\!] = \emptyset$ and $\chi(\Gamma) \leq_{lex} \chi(\Delta)$.*

Proof. By induction on $|\Delta|$. If $\Delta = \mathtt{nil}_t$, then it suffices to take $\Gamma := \Delta$.

Assume $\Delta = \mathcal{A}; \Delta_0$, so that $t \xrightarrow{\mathcal{A}}_{\circ} s \xrightarrow{\Delta_0}_{\circ\!\!\rightarrow} u$ for some object s. Observe $\mathcal{B}[\![\mathcal{A}]\!]$ is never-gripping, Δ_0 does not use $\mathcal{B}[\![\mathcal{A}]\!]$ and $\mathcal{B}[\![\mathcal{A}]\!][\![\Delta_0]\!] = \mathcal{B}[\![\Delta]\!] = \emptyset$. Then we can apply the *i.h.* on $s \xrightarrow{\Delta_0}_{\circ\!\!\rightarrow} u$, thus obtaining $s \xrightarrow{\Gamma'_0}_{\circ\!\!\rightarrow} u$ for some Γ'_0 verifying $\Gamma'_0 \not\supset \mathcal{B}[\![\mathcal{A}]\!]$, $\mathcal{B}[\![\mathcal{A}]\!][\![\Gamma'_0]\!] = \emptyset$ and $\chi(\Gamma'_0) \leq_{lex} \chi(\Delta_0)$.

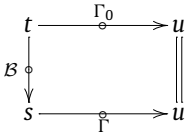
We partition $\mathcal{A}$ in its free and embedded parts w.r.t. $\mathcal{B}$, according to the idea described in Sec. 5.2 after the definition of free and embedded multisteps. Formally, we define $\mathcal{A}^F := \{a \in \mathcal{A} \text{ s.t. } a \not\supset \mathcal{B}\}$ and $\mathcal{A}^E := (\mathcal{A} \setminus \mathcal{A}^F)[\![\mathcal{A}^F]\!]$, so that $t \xrightarrow{\mathcal{A}^F}_{\circ} t' \xrightarrow{\mathcal{A}^E}_{\circ}$

$s \xrightarrow[\circ]{\Gamma_0'} u$ for some object t'. It is easy to check $\mathcal{A}^F \not\supset \mathcal{B}$ and $(\mathcal{A} \setminus \mathcal{A}^F) > \mathcal{B}$; since $\mathcal{A}$ does not use $\mathcal{B}$ and then $\mathcal{A} \cap \mathcal{B} = \emptyset$. As moreover $\mathcal{A}^F \cap \mathcal{B} = \emptyset$, then Lemma 6.4 yields $\mathcal{A}^E > \mathcal{B}[\![\mathcal{A}^F]\!]$. Observe that $\mathcal{B}$ never-gripping implies $\mathcal{B}[\![\mathcal{A}^F]\!]$ never-gripping, $\Gamma_0' \not\supset \mathcal{B}[\![\mathcal{A}]\!] = \mathcal{B}[\![\mathcal{A}^F]\!][\![\mathcal{A}^E]\!]$, and $\mathcal{B}[\![\mathcal{A}^F]\!][\![\mathcal{A}^E; \Gamma_0']\!] = \mathcal{B}[\![\mathcal{A}]\!][\![\Gamma_0']\!] = \emptyset$; *cf.* Proposition 4.1. Therefore Lemma 6.7 applies to $t' \xrightarrow[\circ]{\mathcal{A}^E} s \xrightarrow[\circ]{\Gamma_0'} u$, implying the existence of some Γ_0 verifying $t' \xrightarrow[\circ]{\Gamma_0} u$, $\Gamma_0 \not\supset \mathcal{B}[\![\mathcal{A}^F]\!]$, $\mathcal{B}[\![\mathcal{A}^F]\!][\![\Gamma_0]\!] = \emptyset$ and $\chi(\Gamma_0) = \chi(\Gamma_0') \leq_{lex} \chi(\Delta_0)$. Hence we conclude by taking $\Gamma := \mathcal{A}^F; \Gamma_0$ since $\mathcal{A}^F \subseteq \mathcal{A}$ implies in particular that $\nu(\mathcal{A}^F) \leq \nu(\mathcal{A})$. □

The following propositions describe the construction of the multireduction Δ_{k+1} (*cf.* Fig. 7 on page 49). In terms of the general proof structure described at the beginning of Sec. 6, we can consider t_k, t_{k+1} and $\mathcal{S}(t_k)$ as t, s and $\mathcal{B}$ respectively in the statement of Proposition 6.9 and Proposition 6.10. and Δ_k as Δ in the latter Proposition.

Proposition 6.9 *(Projection over non-used multistep). Let $t \xrightarrow[\circ]{\Delta} u$ and $\mathcal{B} \subseteq \mathcal{R}ed(t)$ s.t. $\mathcal{B}$ is never-gripping, Δ does not use $\mathcal{B}$, $\mathcal{B}[\![\Delta]\!] = \emptyset$ and $t \xrightarrow[\circ]{\mathcal{B}} s$. Then there exists a multireduction Γ s.t. $s \xrightarrow[\circ]{\Gamma} u$ and $\chi(\Gamma) \leq_{lex} \chi(\Delta)$.*

Proof. Lemma 6.8 implies the existence of some Γ_0 such that $t \xrightarrow[\circ]{\Gamma_0} u$, $\Gamma_0 \not\supset \mathcal{B}$, $\mathcal{B}[\![\Gamma_0]\!] = \emptyset$ and $\chi(\Gamma_0) \leq_{lex} \chi(\Delta)$. We define $\Gamma := \Gamma_0[\![\mathcal{B}]\!]$. Then we can build the following diagram; *cf.* Proposition 5.1(2).



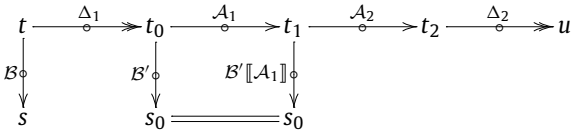
Lemma 6.3 implies $\chi(\Gamma) = \chi(\Gamma_0) \leq_{lex} \chi(\Delta)$. Thus we conclude. □

Proposition 6.10 *(Projection over used multistep). Let $t \xrightarrow[\circ]{\Delta} u$ and $\mathcal{B} \subseteq \mathcal{R}ed(t)$, s.t. $\mathcal{B}$ is never-gripping, Δ uses $\mathcal{B}$, $\mathcal{B}[\![\Delta]\!] = \emptyset$ and $t \xrightarrow[\circ]{\mathcal{B}} s$. Then there exists a multireduction Γ such that $s \xrightarrow[\circ]{\Gamma} u$ and $\chi(\Gamma) <_{lex} \chi(\Delta)$.*

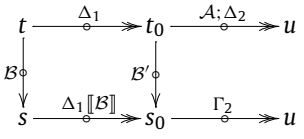
Proof. The hypotheses indicate Δ uses $\mathcal{B}$, therefore the "last" element of Δ which uses the corresponding residual of $\mathcal{B}$ can be determined, *i.e.* Δ can be written as $\Delta_1; \mathcal{A}; \Delta_2$, such that $\mathcal{A}$ uses $\mathcal{B}[\![\Delta_1]\!]$ (*i.e.* $\mathcal{A} \cap \mathcal{B}[\![\Delta_1]\!] \neq \emptyset$) and Δ_2 does not use $\mathcal{B}[\![\Delta_1; \mathcal{A}]\!]$. Observe $|\Delta| = |\Delta_1| + |\Delta_2| + 1$.

Let $\mathcal{B}' := \mathcal{B}[\![\Delta_1]\!]$, $\mathcal{A}_1 := \mathcal{A} \cap \mathcal{B}'$, and $\mathcal{A}_2 := (\mathcal{A} \setminus \mathcal{A}_1)[\![\mathcal{A}_1]\!]$. Observe that $\mathcal{A}_1 \neq \emptyset$. To verify that $\nu(\mathcal{A}_2) < \nu(\mathcal{A})$, let δ, γ such that $\delta \Vdash \mathcal{A}_1$, $\gamma \Vdash \mathcal{A}_2$, and particularly $|\gamma| = \nu(\mathcal{A}_2)$. Observe $\delta; \gamma \Vdash \mathcal{A}$. We obtain $|\delta| > 0$ since $\mathcal{A}_1 \neq \emptyset$. Then $\nu(\mathcal{A}) \geq |\delta; \gamma| > \nu(\mathcal{A}_2)$.

Therefore, $\chi(\mathcal{A}_2; \Delta_2) <_{lex} \chi(\mathcal{A}; \Delta_2)$. Moreover $\mathcal{A}_1[\![\mathcal{B}']\!] = \emptyset$. We can build the following diagram:



Suppose $\mathcal{A}_2$ uses $\mathcal{B}'[\![\mathcal{A}_1]\!]$. Notice that the existence of some $b' \in \mathcal{A}_2 \cap \mathcal{B}'[\![\mathcal{A}_1]\!]$ would in turn imply the existence of some $b_1 \in \mathcal{B}'$ s.t. $b_1[\![\mathcal{A}_1]\!]b'$ and also the existence of some $b_2 \in \mathcal{A} \setminus \mathcal{A}_1$ s.t. $b_2[\![\mathcal{A}_1]\!]b'$. Consider an arbitrary $\delta \Vdash \mathcal{A}_1$. Then, by a simple induction on $|\delta|$ and resorting to Ancestor Uniqueness, one deduces $b_1 = b_2$. Therefore $b_1 = b_2 \in \mathcal{B}' \cap (\mathcal{A} \setminus \mathcal{A}_1)$. But then, by definition of $\mathcal{A}_1$, $b_1 = b_2 \in \mathcal{A}_1$, which is absurd. Therefore $\mathcal{A}_2$ does not use $\mathcal{B}'[\![\mathcal{A}_1]\!]$ and hence, since Δ_2 does not use $\mathcal{B}[\![\Delta_1; \mathcal{A}]\!]$, we obtain that $\mathcal{A}_2; \Delta_2$ does not use $\mathcal{B}'[\![\mathcal{A}_1]\!]$. Moreover, $\mathcal{B}$ never-gripping implies $\mathcal{B}'[\![\mathcal{A}_1]\!]$ never-gripping. Hence Proposition 6.9 yields the existence of some Γ_2 verifying $s_0 \xrightarrow[\circ]{\Gamma_2} u$ and $\chi(\Gamma_2) \leq_{lex} \chi(\mathcal{A}_2; \Delta_2) <_{lex} \chi(\mathcal{A}; \Delta_2)$. Remark that, by definition of χ, $|\Gamma_2| = |\mathcal{A}; \Delta_2| = |\Delta_2| + 1$.



Thus if we define $\Gamma := \Delta_1[\![\mathcal{B}]\!]; \Gamma_2$, then $|\Gamma| = |\Delta_1| + |\Delta_2| + 1 = |\Delta|$, and $\chi(\Gamma_2) <_{lex} \chi(\mathcal{A}; \Delta_2)$ implies $\chi(\Gamma) <_{lex} \chi(\Delta)$ independently of the relative measures of $\Delta_1[\![\mathcal{B}]\!]$ and Δ_1, since elements of multireductions are considered in *reversed order* when building measures. Thus we conclude. □

As we already remarked, Proposition 6.10 shows the existence of an adequate Δ_{k+1} following the general proof structure described at the beginning of Sec. 6. Therefore, we can prove the main result of this work.

Theorem 6.11 *(The abstract normalisation result). Let $\mathfrak{A}$ be an ARS $\langle \mathcal{O}, \mathcal{R}, src, tgt, [\![\cdot]\!], <, \ll \rangle$ enjoying all the axioms listed in Fig. 4. Repeated contraction of necessary and never-gripping multisteps on $\mathfrak{A}$ normalises.*

Proof. Let $t_0 \in \mathcal{O}$ a normalising object in $\mathfrak{A}$. Then there exists some multireduction Δ_0 such that $t_0 \xrightarrow[\circ\!\!\rightarrow\!\!\!\!\rightarrow]{\Delta_0} u$ where u is a normal form. We proceed by induction on $\chi(\Delta_0)$, *i.e.* using the well-founded ordering defined at the beginning of Sec. 6. If $\chi(\Delta_0)$ is minimal, *i.e.* either $\Delta_0 = \mathtt{nil}_{t_0}$ or $\Delta_0 = \langle \emptyset_{t_0}, \ldots, \emptyset_{t_0} \rangle$, then t_0 is a normal form, and therefore there is nothing to prove. Otherwise, let $\mathcal{B}$ be a necessary and never-gripping multistep such that $t_0 \xrightarrow[\circ\!\!\rightarrow]{\mathcal{B}} t_1$. Then Δ_0 uses $\mathcal{B}$, and u being a normal form implies $\mathcal{B}[\![\Delta_0]\!] = \emptyset$. Therefore Proposition 6.10 implies the existence of a multireduction Δ_1 such that $t_1 \xrightarrow[\circ\!\!\rightarrow\!\!\!\!\rightarrow]{\Delta_1} u$ and $\chi(\Delta_1) <_{lex} \chi(\Delta_0)$. The *i.h.* on Δ_1 suffices to conclude. □

7. Applications

7.1. The pure pattern calculus (and the simple pattern calculus)

PPC is a pattern calculus which extends **SPC** and stands out for the novel forms of polymorphism it supports. Since arbitrary terms may be used as patterns and hence reduction inside patterns is allowed, **PPC** models *pattern polymorphism* where functions over patterns that are computed at runtime may be defined. Another language feature is *path polymorphism*, which permits functions that are generic in the sense that they operate over arbitrary data structures.

This section has four parts. We first present a brief overview of **PPC** following [17]. Then we show that **PPC** fits the ARS framework, including all the axioms. The third part formulates a multistep strategy $\mathcal{S}$. The final part shows that $\mathcal{S}$ computes necessary and never-gripping multisteps. In view of the results of the previous section, *cf.* Theorem 6.11, these last three parts – taken together – imply that $\mathcal{S}$ is normalising for this calculus.

7.1.1. Overview of ***PPC***

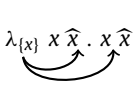
Consider a countable set of **symbols** $f, g, \ldots, x, y, z$. Sets of symbols are denoted by meta-variables θ, ϕ, ... The syntax of **PPC** is summarised by the following grammar:

Terms	$(\boldsymbol{T})$	$t ::= x \mid \widehat{x} \mid tt \mid \lambda_\theta\, t.t$
Data-Structures	$(\boldsymbol{DS})$	$D ::= \widehat{x} \mid Dt$
Abstractions	$(\boldsymbol{ABS})$	$A ::= \lambda_\theta\, t.t$
Matchable-forms	$(\boldsymbol{MF})$	$F ::= D \mid A$

The term x is called a **variable**, $\widehat{x}$ a **matchable**, tu an **application** (t is the **function** and u the **argument**) and $\lambda_\theta\ p.u$ an **abstraction** (θ is the set of **binding symbols**, p is the **pattern** and u is the **body**). Application (resp. abstraction) is left (resp. right) associative.

A λ-abstraction $\lambda x.t$ can be defined by $\lambda_{\{x\}}\ \widehat{x}.t$. The **identity function** $\lambda_{\{x\}}\ \widehat{x}.x$ is abbreviated I.

A binding symbol $x \in \theta$ of an abstraction $\lambda_\theta\ p.s$ *binds* matchable occurrences of x in p and variable occurrences of x in s.



The derived notions of **free variables** and **free matchables** are respectively denoted by $\mathtt{fv}(\_)$ and $\mathtt{fm}(\_)$. This is illustrated in the figure on the left. Formally, **free variables** and **free matchables** of terms are defined by:

$$\begin{array}{rcl@{\qquad}rcl}
\mathtt{fv}(x) & := & \{x\} & \mathtt{fm}(x) & := & \emptyset \\
\mathtt{fv}(\widehat{x}) & := & \emptyset & \mathtt{fm}(\widehat{x}) & := & \{x\} \\
\mathtt{fv}(tu) & := & \mathtt{fv}(t) \cup \mathtt{fv}(u) & \mathtt{fm}(tu) & := & \mathtt{fm}(t) \cup \mathtt{fm}(u) \\
\mathtt{fv}(\lambda_\theta\ p.u) & := & (\mathtt{fv}(u) \setminus \theta) \cup \mathtt{fv}(p) & \mathtt{fm}(\lambda_\theta\ p.u) & := & (\mathtt{fm}(p) \setminus \theta) \cup \mathtt{fm}(u)
\end{array}$$

As usual, we consider terms up to **alpha-conversion**, *i.e.* up to renaming of bound matchables and variables. **Constructors** are matchables which are not bound and, to ease the presentation, they are often denoted in typewriter fonts $\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \ldots$, thus for example $\lambda_{\{x,y\}}\ \widehat{x}\ y\ \mathtt{a}.y$ denotes $\lambda_{\{x,y\}}\ \widehat{x}\ y\ \widehat{z}.y$. The distinction between matchables and variables is unnecessary for standard (static) patterns which do not contain free variables.

A **position** is either ϵ (the empty position), or na, where $n \in \{1, 2\}$ and a is a position. We use $a, b, \ldots$ to denote positions. The **set** $\mathtt{Pos}(t)$ **of positions** of t is defined as expected, provided that for abstractions $\lambda_\theta\ p.s$ positions inside both p and s are considered. Here is an example $\mathtt{Pos}(\lambda_{\{x\}}\ \mathtt{a}\ \widehat{x}.\mathtt{a}\ x\ x) = \{\epsilon, 1, 2, 11, 12, 21, 22, 211, 212\}$. We write $a \leq b$ (resp. $a \parallel b$) when the position a is a **prefix** of (resp. **disjoint** from) the position b. Notice that $a \parallel b$ and $a \leq c$ imply $c \parallel b$. All these notions are defined as expected [10] and extended to sets of positions as well. In particular, given a position a and a set of positions $\mathcal{B}$, we will say that $a \leq \mathcal{B}$ iff $a \leq b$ for all $b \in \mathcal{B}$, and analogously for $<$, $\parallel$, etc.

We write $t|_a$ for the **subterm of** t **at position** a and $t[s]_a$ for the **replacement** of the subterm at position a in t by s. Finally, we write $s \subseteq t$ if s is a subterm of t (note in particular $s \subseteq s$). Notice that replacement may capture variables. An

occurrence of a term s in a term t is any position $p \in \mathtt{Pos}(t)$ verifying $t|_p = s$. Particularly, *variable occurrences* are defined this way.

Substitution and matching A **substitution** σ is a mapping from variables to terms with finite domain $\mathtt{dom}(\sigma)$. We write $\{x_1 \to t_1, \ldots, x_n \to t_n\}$ for a substitution with domain $\{x_1, \ldots, x_n\}$. A **match** μ is either a substitution or a special constant in the set $\{\mathtt{fail}, \mathtt{wait}\}$. A **match** is **positive** if it is a substitution; it is **decided** if it is either positive or $\mathtt{fail}$. The set of free variables of a match μ are defined as follows: $\mathtt{fv}(\sigma) = \bigcup_{x \in \mathtt{dom}(\sigma)} \mathtt{fv}(\sigma x)$, $\mathtt{fv}(\mathtt{fail}) = \emptyset$ and $\mathtt{fv}(\mathtt{wait})$ is undefined. Similarly for $\mathtt{fm}(\mu)$. We also define $\mathtt{dom}(\mathtt{fail}) = \emptyset$, whereas $\mathtt{dom}(\mathtt{wait})$ is undefined. The **symbols** of μ are $\mathtt{sym}(\mu) := \mathtt{dom}(\mu) \cup \mathtt{fv}(\mu) \cup \mathtt{fm}(\mu)$. A set of symbols θ **avoids** a match μ, written $\theta \# \mu$, iff $\forall x \in \theta, x \notin \mathtt{sym}(\sigma)$. The **application of a substitution** σ to a term is written and defined as usual on alpha-equivalence classes; in particular $\sigma(\lambda_\theta\ p.s) := \lambda_\theta\ \sigma(p).\sigma(s)$, if $\theta \# \sigma$. Notice that data structures and matchable forms are stable by substitution. The **application of a match** μ to a term t, written μt, is defined as follows: if μ is a substitution, then it is applied as explained above; if $\mu = \mathtt{wait}$, then μt is undefined; if $\mu = \mathtt{fail}$, then μt is the identity function I. Other *closed terms in normal form* could be taken to define the last case, this one allows in particular to encode pattern-matching definitions given by alternatives [17].

The **restriction** of a substitution σ to a set of variables $\{x_1, \ldots, x_n\} \subseteq \mathtt{dom}(\sigma)$ is written $\sigma|_{\{x_1,\ldots,x_n\}}$. This notion is extended to matchings by defining $\mathtt{wait}|_{\{x_1,\ldots,x_n\}} = \mathtt{wait}$ and $\mathtt{fail}|_{\{x_1,\ldots,x_n\}} = \mathtt{fail}$, for any set of variables $\{x_1, \ldots, x_n\}$. The **composition** $\sigma \circ \eta$ of two substitutions σ and η is defined by $(\sigma \circ \eta)x = \sigma(\eta x)$. Furthermore, if μ_1 and μ_2 are matches of which at least one is $\mathtt{fail}$, then $\mu_2 \circ \mu_1$ is defined to be $\mathtt{fail}$. Otherwise, if μ_1 and μ_2 are matches of which at least one is $\mathtt{wait}$, then $\mu_2 \circ \mu_1$ is defined to be $\mathtt{wait}$. Thus, in particular, $\mathtt{fail} \circ \mathtt{wait}$ is $\mathtt{fail}$.

The **disjoint union** of two matches μ_1 and μ_2 is as in **SPC**. In particular, the equation from **SPC** also holds

$$\mathtt{fail} \uplus \mathtt{wait} = \mathtt{wait} \uplus \mathtt{fail} = \mathtt{fail}$$

and is the culprit for the non-sequential nature of **PPC** (just as in **SPC**).[8]

The **compound matching operation** takes a term, a set of binding symbols and a pattern and returns a match, it is defined by applying the following equations in order:

$$\begin{array}{rcll}
\{\!|\widehat{x} \rhd_\theta t|\!\} & := & \{x \to t\} & \text{if } x \in \theta \\
\{\!|\widehat{x} \rhd_\theta \widehat{x}|\!\} & := & \{\} & \text{if } x \notin \theta \\
\{\!|\widehat{x}t_1 \ldots t_n \rhd_\theta \widehat{x}p_1 \ldots p_n|\!\} & := & \{\!|p_1 \rhd_\theta t_1|\!\} \uplus \ldots \uplus \{\!|p_n \rhd_\theta t_n|\!\} & \text{if } x \notin \theta \\
\{\!|\widehat{y}t_1 \ldots t_m \rhd_\theta \widehat{x}p_1 \ldots p_n|\!\} & := & \mathtt{fail} & \text{if } x, y \notin \theta \text{ and } (x \neq y \text{ or } (x = y \text{ and } m \neq n)) \\
\{\!|\lambda_\theta\ p.u \rhd_\theta \widehat{x}p_1 \ldots p_n|\!\} & := & \mathtt{fail} & \\
\{\!|\lambda_\theta\ p_2.s_2 \rhd_\theta \lambda_\theta\ p_1.s_1|\!\} & := & \mathtt{fail} & \\
\{\!|p \rhd_\theta t|\!\} & := & \mathtt{wait} & \text{otherwise}
\end{array}$$

The use of disjoint union in the third case of the previous definition restricts compound matching to linear patterns, as in **SPC**. The result of the **matching operation**[9] $\{p/_\theta\ t\}$ is defined to be the *check* of $\{\!|p \rhd_\theta t|\!\}$ on θ; where the **check** of a match μ on θ is $\mathtt{fail}$ if μ is a substitution whose domain is not θ, μ otherwise. Notice that $\{p/_\theta\ t\}$ is never positive if p is not linear with respect to θ. We now give some examples: $\{\widehat{xx}/_{\{x\}}\ uv\}$ gives $\mathtt{fail}$ because $\widehat{xx}$ is not linear; $\{\widehat{xy}/_{\{x,y,z\}}\ uv\}$ gives $\mathtt{fail}$ because $\{x, y, z\} \neq \{x, y\}$, $\{\widehat{x}/_\emptyset\ u\}$ gives $\mathtt{fail}$ because $\emptyset \neq \{x\}$; $\{\widehat{y}/_{\{x\}}\ \widehat{y}\}$ gives $\mathtt{fail}$ because $\{x\} \neq \emptyset$; $\{\widehat{xy}/_{\{x\}}\ u\widehat{z}\}$ gives $\mathtt{fail}$ because $\{\!|\widehat{y} \rhd_{\{x\}} \widehat{z}|\!\}$ is $\mathtt{fail}$; $\{\widehat{xy}/_\emptyset\ u\widehat{z}\}$ gives $\mathtt{fail}$ since both $\{\widehat{x}/_\emptyset\ u\}$ and $\{\!|\widehat{y} \rhd_\emptyset \widehat{z}|\!\}$ are $\mathtt{fail}$.

7.1.2. **PPC** as an ARS

PPC can be described as an ARS. Its objects $\mathcal{O}$ are the *terms* of **PPC**. The *steps* are the pairs $\langle t, a\rangle$ where t is a term, $a \in \mathtt{Pos}(t)$, $t|_a = (\lambda_\theta p.s)u$, and $\{p/_\theta\ u\}$ is decided. In this case $\mathrm{src}(\langle t, a\rangle) := t$ and $\mathrm{tgt}(\langle t, a\rangle) := t[\{p/_\theta\ u\}s]_a$. If $\{p/_\theta\ u\} = \mathtt{fail}$, then we say that the step is a **matching failure**. We will often denote by $\mathfrak{a}$ a given step $\langle t, a\rangle$; analogously, we will often denote by $\mathfrak{D}$ the set $\{\langle t, d\rangle \mid d \in D\}$ where $D \subseteq \mathtt{Pos}(t)$. Conversely, whenever $\mathfrak{a}$ is a step, we often refer to its position as a, even without specifying explicitly that $\mathfrak{a} = \langle t, a\rangle$ for some term t, and similarly, whenever $\mathfrak{D}$ is a set of steps, we refer to the corresponding set of positions as D. This notation shall prove convenient when we address the compliance of **PPC** w.r.t. the axioms of an ARS. Regarding the relations over objects and steps:

- **Residual relation**. If $\mathfrak{a} = \langle t, a\rangle$, $\mathfrak{b} = \langle t', b\rangle$ and $\mathfrak{b}' = \langle u, b'\rangle$ are steps, then $\mathfrak{b}[\![\mathfrak{a}]\!]\mathfrak{b}'$ iff $t' = t$, $u = \mathrm{tgt}(\mathfrak{a})$, and one of the following cases apply, where $t|_a = (\lambda_\theta p.s)u$:
  - $a \not\leq b$ and $b' = b$.
  - $b = a12n$, $b' = an$ and $\{p/_\theta\ u\} \neq \mathtt{fail}$.
  - $b = a2mn$, $b' = akn$, $\{p/_\theta\ u\} \neq \mathtt{fail}$, and there is a variable $x \in \theta$ such that $t|_{a11m} = p|_m = \widehat{x}$ and $t|_{a12k} = s|_k = x$.

[8] Sequentiality can be recovered (see *e.g.* [15,4,5]) by simplifying the equations of disjoint union, however, some meaningful terms will no longer be normalising. *E.g.* if in particular $\mathtt{wait} \uplus \mathtt{fail} = \mathtt{wait}$, then $(\lambda_\emptyset\ \mathtt{a}\ \mathtt{b}\ \mathtt{b}\ .\widehat{y})(\mathtt{a}\ \Omega\ \mathtt{c})$, where Ω is a non-terminating term, would never fail as we expect.

[9] Note that the notation for (compound) matching we have just given differs from [16] and [17]: the pattern and argument appear in reversed order there.

- **Embedding relation**. We define the embedding relation between redexes as the *tree order* [20]. Namely, $\mathfrak{a} < \mathfrak{b}$ iff $\mathfrak{a} = \langle t, a\rangle$, $\mathfrak{b} = \langle t, b\rangle$, and $a < b$. Notice that whenever $\mathfrak{a} < \mathfrak{c}$ and $\mathfrak{b} < \mathfrak{c}$, then $\mathfrak{a}$ and $\mathfrak{b}$ are comparable w.r.t. the embedding, *i.e.* either $\mathfrak{a} = \mathfrak{b}$, $\mathfrak{a} < \mathfrak{b}$ or $\mathfrak{b} < \mathfrak{a}$.
- **Gripping relation**. Let $\mathfrak{a} = \langle t, a\rangle$ and $\mathfrak{b} = \langle t, b\rangle$ be steps and let $t|_a = (\lambda_\theta p.s)u$. Then $\mathfrak{a} \ll \mathfrak{b}$ iff $\{p/_\theta\ u\} \neq \mathtt{fail}$, $b = a12n$, and $\theta \cap \mathtt{fv}(s|_n) \neq \emptyset$.

We now address the axioms of Fig. 4. A word on notation: if t and θ are a term and a set of symbols respectively, then we will write $bm(t, \theta)$ when $t = \widehat{x}$ for some $x \in \theta$.

Fundamental axioms.
Self Reduction is immediate from the definition of residuals for **PPC**: none of the cases there applies for $\mathfrak{a}[\![\mathfrak{a}]\!]$. Finite Residuals follows from the fact that terms are finite. Axiom Ancestor Uniqueness is proved below.

Lemma 7.1 *(Ancestor uniqueness). Let $\mathfrak{b}_1, \mathfrak{b}_2, \mathfrak{a}, \mathfrak{b}'$ be steps verifying $\mathfrak{b}_1[\![\mathfrak{a}]\!]\mathfrak{b}'$ and $\mathfrak{b}_2[\![\mathfrak{a}]\!]\mathfrak{b}'$. Then $\mathfrak{b}_1 = \mathfrak{b}_2$.*

Proof. Let $\mathfrak{b}_1 = \langle t, b_1\rangle$, $\mathfrak{b}_2 = \langle t, b_2\rangle$ and $\mathfrak{b}' = \langle t', b'\rangle$, where $t \xrightarrow{\mathfrak{a}} t'$. We prove that $b_1 = b_2$. Let $t|_a = (\lambda_\theta p.s)u$. We consider three cases according to the definition of $\mathfrak{b}_1[\![\mathfrak{a}]\!]\mathfrak{b}'$.

- If $a \not\leq b_1$, then $b_1 = b'$ so that $a \not\leq b'$. A straightforward case analysis on the definition of residuals yields $a \not\leq b_2$, therefore $b_1 = b_2 = b'$.
- If $b_1 = a2mn$ and $b' = akn$, then $s|_k = x$ and $p|_m = \widehat{x}$ for some $x \in \theta$. Observe that $a < b'$ implies $a < b_2$. We consider two cases. If $b_2 = a12n'$ and $b' = an'$, then $kn = n'$. This would imply $t|_{b_2} = s|_{kn}$ has the form $(\lambda_{\theta'} p'.s')u'$, contradicting $s|_k$ being a variable. Therefore, $akn = b' = ak'n'$ and $b_2 = a2m'n'$, where $s|_{k'} = y$ and $p|_{m'} = \widehat{y}$ for some $y \in \theta$. Observe that $k < k'$, *i.e.* $k' = kc$ where $c \neq \epsilon$, would imply $kc \in \mathtt{Pos}(s)$, contradicting the fact that $s|_k$ is a variable; so that $k \not< k'$. We obtain $k' \not< k$ analogously. On the other hand, $k \parallel k'$ would contradict $kn = k'n'$. Hence $k = k'$, implying $n = n'$ and also $y = x$. In turn, $\{p/_\theta\ u\}$ being positive implies that p is linear, and then $m = m'$. Thus we conclude.
- If $b_1 = a12n$ and $b' = an$, then we have again that $a < b'$ implies $a < b_2$. On the other hand, assuming $b_2 = a2m'n'$, so that $an = b' = akn'$, would yield a contradiction as already stated. Therefore $b_2 = a12n'$ and $an = b' = an'$, implying $n = n'$ and consequently $b_1 = b_2$. □

Finally, FD and SO are left for the end of this section.

The Enclave–Creation axiom.
To verify Enclave–Creation involves a rather long technical development, including some preliminary lemmas, particularly a creation lemma indicating the creation cases for **PPC**.

Lemma 7.2. *Let $p \twoheadrightarrow p'$ and $u \twoheadrightarrow u'$. Then,*

(i) $\{\!\{p \triangleright_\theta u\}\!\}$ *positive implies* $\{\!\{p' \triangleright_\theta u'\}\!\}$ *positive,*
(ii) $\{\!\{p \triangleright_\theta u\}\!\} = \mathtt{fail}$ *implies* $\{\!\{p' \triangleright_\theta u'\}\!\} = \mathtt{fail}$.
(iii) $\{p/_\theta\ u\}$ *positive implies* $\{p'/_\theta\ u'\}$ *positive,*
(iv) $\{p/_\theta\ u\} = \mathtt{fail}$ *implies* $\{p'/_\theta\ u'\} = \mathtt{fail}$.

Proof. By induction and case analysis on p. See [9] for further details. □

Lemma 7.3 *(Creation cases). Let $t \xrightarrow{\mathfrak{a}} t'$, and $\emptyset[\![\mathfrak{a}]\!]\mathfrak{b}$, i.e. $\mathfrak{b}$ is* created *by (the contraction of) $\mathfrak{a}$. Say $t|_a = (\lambda_\theta p.s)u$ and $t'|_b = (\lambda_{\theta'} p'.s')u'$. Then one of the following holds.*

Case I. *The contraction of $\mathfrak{a}$ contributes to the creation of $\mathfrak{b}$ from below, i.e., $b \in \mathtt{Pos}(t)$, $a = b1$ implying $t|_b = (\lambda_\theta p.s)uu'$, and either*
(i) *$s = x$ where $x \in \theta$ and $\widehat{x}$ occurs in p, $\{p/_\theta\ u\} = \sigma$, $\sigma x = (\lambda_{\theta'} p'.s')$.*
(ii) *$s = \lambda_{\theta'} p''.s''$, $\{p/_\theta\ u\} = \sigma$, $p' = \sigma p''$, $s' = \sigma s''$.*
(iii) *$\{p/_\theta\ u\} = \mathtt{fail}$, $\lambda_{\theta'} p'.s' = I$.*

Case II. *The contraction of $\mathfrak{a}$ contributes to the creation of $\mathfrak{b}$ from above, i.e., $b = an$, $s|_n = xu''$, $\{p/_\theta\ u\} = \sigma$, $\sigma x = (\lambda_{\theta'} p'.s')$, $u' = \sigma u''$.*

Case III. *The argument of a redex pattern becomes decided. We have three such situations:*
(i) *$b = an$, $s|_n = (\lambda_{\theta'} p''.s'')u''$, $\{p''/_{\theta'}\ u''\} = \mathtt{wait}$, $\{p/_\theta\ u\} = \sigma$, $p' = \sigma p''$, $s' = \sigma s''$ and $u' = \sigma u''$.*
(ii) *$a = b2n$, $t|_b = (\lambda_{\theta'} p'.s')u''$ and $\{p'/_{\theta'}\ u''\} = \mathtt{wait}$.*
(iii) *$a = b11n$, $t|_b = (\lambda_{\theta'} p''.s')u'$ and $\{p''/_{\theta'}\ u'\} = \mathtt{wait}$.*

Proof. We proceed by comparing a with b.

- If $a \parallel b$, then $t|_b = t'|_b$ so that $\langle t, b\rangle \llbracket \mathfrak{a} \rrbracket \mathfrak{b}$, contradicting the hypotheses.
- Assume $a \leq b$, *i.e.* $b = ac$.
  In this case, $\{p/_\theta\ u\} = \mathtt{fail}$ would imply $t'|_a = I$, contradicting $t'|_b$ being a redex. Then $\{p/_\theta\ u\} = \sigma$, implying $t'|_b = \sigma s|_c$. Now the redex at position c of σs is either entirely contained in σ or otherwise it occurs at a non-variable position of s. Observe that $c = kn$, $s|_k = x$ and $t'|_b = \sigma x|_n$ for some variable x would imply $\langle t, a2mn\rangle \llbracket \mathfrak{a} \rrbracket \mathfrak{b}$ where $p|_m = \widehat{x}$. This is not possible since $\mathfrak{b}$ is created. Therefore $s|_c = t_1 u''$ and $t'|_b = (\lambda_{\theta'} p'.s')u' = (\sigma t_1)\sigma u''$. If t_1 is a variable, so that $\sigma t_1 = \lambda_{\theta'} p'.s'$, then case II applies, otherwise case III.(i) applies.
- Assume $b < a$.
  If $a = b1$, *i.e.* $t|_b = (\lambda_\theta p.s)uu'$, then observe $\{p/_\theta\ u\}s = t'|_a = \lambda_{\theta'} p'.s'$. If $\{p/_\theta\ u\} = \mathtt{fail}$, then case I.(iii) applies. If s is a variable, then case I.(i) applies. Otherwise, s is an abstraction, so that case I.(ii) applies.
  If $b11 \leq a$, *i.e.* $t|_b = (\lambda_{\theta'} p''.s')u'$, then observe $\emptyset \llbracket \mathfrak{a} \rrbracket \mathfrak{b}$ implies $\{p''/_{\theta'}\ u'\} = \mathtt{wait}$. Then case III.(iii) applies. If $b2 \leq a$, a similar argument yields that case III.(ii) applies.
  Finally, $b12 \leq a$ implies $t|_b = (\lambda_{\theta'} p'.s'')u'$, and $t'|_b$ being a redex implies $\{p'/_{\theta'}\ u'\}$ decided so that $\langle t, b\rangle \llbracket \mathfrak{a} \rrbracket \mathfrak{b}$, contradicting the hypothesis. □

Lemma 7.4.

1. *Let* $t \xrightarrow{\mathfrak{a}} t'$ *such that* $t \notin \boldsymbol{MF}$ *and* $t' \in \boldsymbol{MF}$. *Then* $\mathfrak{a}$ *is outermost.*
2. *Let* $t \xrightarrow{\mathfrak{a}} t'$ *such that* $\{\!\{p \rhd_\theta t\}\!\} = \mathtt{wait}$ *and* $\{\!\{p \rhd_\theta t'\}\!\}$ *is decided for some* θ, p. *Then* $\mathfrak{a}$ *is outermost.*
3. *Let* $p \xrightarrow{\mathfrak{a}} p'$ *such that* $\{\!\{p \rhd_\theta t\}\!\} = \mathtt{wait}$ *and* $\{\!\{p' \rhd_\theta t\}\!\}$ *is decided for some* θ, t. *Then* $\mathfrak{a}$ *is outermost.*

Proof. By induction on t', t and p, respectively. We use item 1 in the proof of the other items. See [9] for further details. □

Lemma 7.5 *(Enclave–Creation). Let* $\mathfrak{a}$, $\mathfrak{b}$ *be steps such that* $\mathfrak{b} < \mathfrak{a}$, $\mathfrak{b} \llbracket \mathfrak{a} \rrbracket \mathfrak{b}'$, *and* $\emptyset \llbracket \mathfrak{a} \rrbracket \mathfrak{c}'$. *Then* $\mathfrak{b}' < \mathfrak{c}'$.

Proof. Observe that $a \not\leq b$ implying $b' = b$. Say $t \xrightarrow{\mathfrak{a}} t'$, $t|_a = (\lambda_\theta p.s)u$, and $t'|_{c'} = (\lambda_{\theta'} p'.s')u'$. We proceed by case analysis w.r.t. Lemma 7.3.

Case I In this case $c' \in \mathtt{Pos}(t)$ and $a = c'1$, so that $t|_{c'} = (\lambda_\theta p.s)uu'$. Therefore, it suffices to observe that $b = c'$ would contradict $\mathfrak{b}$ to be a step, then $b < a$ implies $b < c'$.

Cases II or III.(i) In either case $c' = an$, thus $b < a$ implies $b < c'$.

Case III.(ii) In this case $a = c'2n$ and $t|_{c'} = (\lambda_{\theta'} p'.s')u''$. Then, $b < a$ implies either $b < c'$, $b = c'$ or $b = c'2n'$ where $n' < n$. We conclude by observing that the second and third cases would contradict $\emptyset \llbracket \mathfrak{a} \rrbracket \mathfrak{c}'$ and Lemma 7.4:(2) respectively.

Case III.(iii) In this case $a = c'11n$ and $t|_{c'} = (\lambda_{\theta'} p''.s')u'$. A similar analysis applies, resorting to Lemma 7.4:(3) instead of Lemma 7.4:(2). □

The other embedding and gripping axioms.

Linearity is immediate from the definition of residuals. The remaining embedding axioms, and also Grip–Instantiation, are related with the invariance of embedding w.r.t. residuals. The following result characterises those situations in which the embedding relation between two steps fails to be preserved w.r.t. the contraction of a third one.

Lemma 7.6. *Suppose* $\mathfrak{b} \llbracket \mathfrak{a} \rrbracket \mathfrak{b}'$ *and* $\mathfrak{c} \llbracket \mathfrak{a} \rrbracket \mathfrak{c}'$, *such that* $\neg(\mathfrak{b} < \mathfrak{c} \Leftrightarrow \mathfrak{b}' < \mathfrak{c}')$. *Then:*

- $(\mathfrak{a} < \mathfrak{b}) \wedge (\mathfrak{a} < \mathfrak{c})$, *and moreover;*
- *either* $(\mathfrak{b} < \mathfrak{c}) \wedge (\mathfrak{b}' \parallel \mathfrak{c}')$, *or* $(\mathfrak{a} \ll \mathfrak{b}) \wedge (\mathfrak{b} \parallel \mathfrak{c}) \wedge (\mathfrak{b}' < \mathfrak{c}') \wedge (a2 \leq c)$.

Proof. By case analysis of a, b and c. The most challenging case is when $a < b$ and $a < c$. In this case, a detailed analysis of the possible cases w.r.t. the residual relation is required. See [9] for further details. □

It is easy to obtain Context-Freeness, Enclave–Embedding and Grip–Instantiation as corollaries of Lemma 7.6.

Lemma 7.7 *(Pivot). Let* $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{c}'$ *steps verifying* $\mathfrak{a} < \mathfrak{c}$, $\mathfrak{b} < \mathfrak{c}$, $\mathfrak{b} \not\leq \mathfrak{a}$, *and* $\mathfrak{c} \llbracket \mathfrak{a} \rrbracket \mathfrak{c}'$. *Then there exists a step* $\mathfrak{b}'$ *such that* $\mathfrak{b} \llbracket \mathfrak{a} \rrbracket \mathfrak{b}'$ *and* $\mathfrak{b}' < \mathfrak{c}'$.

Proof. Observe that $a < c$, $b < c$ and $b \not\leq a$ implies $a < b < c$. A case analysis w.r.t. the definition of residuals, considering $\mathfrak{a} < \mathfrak{c}$, allows to conclude. See [9] for further details. □

Lemma 7.8. *Suppose* $t = (\lambda_\theta p.s)u \xrightarrow{\mathfrak{a}} t'$, $\mathfrak{c} \llbracket \mathfrak{a} \rrbracket \mathfrak{c}'$, *and* $x \in \mathtt{fv}(t'|_{c'})$. *Then* $x \in \mathtt{fv}(t|_c)$, *or* $\mathfrak{a} \ll \mathfrak{c}$ *and* $x \in \mathtt{fv}(u)$.

Proof. By case analysis on the definition of residuals. See [9] for further details. □

Lemma 7.9 *(Grip–Density). Consider steps* $\mathfrak{a}, \mathfrak{b}, \mathfrak{b}', \mathfrak{c}, \mathfrak{c}'$ *verifying* $\mathfrak{b}[\![\mathfrak{a}]\!]\mathfrak{b}'$, $\mathfrak{c}[\![\mathfrak{a}]\!]\mathfrak{c}'$, *and* $\mathfrak{b}' \ll \mathfrak{c}'$. *Then* $\mathfrak{b} \ll \mathfrak{c} \vee \mathfrak{b} \ll \mathfrak{a} \ll \mathfrak{c}$.

Proof. Let $t \xrightarrow{\mathfrak{a}} t'$, and say $t|_a = (\lambda_\theta p.s)u$, $t'|_{b'} = (\lambda_{\theta'} p'.s')u'$, and $t|_b = (\lambda_{\theta'} p''.s'')u''$; notice that the set θ' is invariant w.r.t. the contraction of $\mathfrak{a}$. Recall that $\mathfrak{b}' \ll \mathfrak{c}'$ implies $\{p''/_{\theta'}\ u''\}$ positive, $b'12 \leq c'$ and $\theta' \cap \mathtt{fv}(t'|_{c'}) \neq \emptyset$. Observe that $\{p''/_{\theta'}\ u''\}$ positive and $\{p'/_{\theta'}\ u'\}$ decided imply $\{p'/_{\theta'}\ u'\}$ positive; *cf.* Lemma 7.2. Let $x \in \theta' \cap \mathtt{fv}(t'|_{c'})$. Then Lemma 7.8 implies $x \in \mathtt{fv}(t|_c) \vee (\mathfrak{a} \ll \mathfrak{c} \wedge x \in \mathtt{fv}(u))$.

Given $b' < c'$, Lemma 7.6 implies $b < c$ or $(b \parallel c \wedge a2 \leq c)$. The latter case implies $\mathfrak{a} \not\ll \mathfrak{c}$ (since $a2 \leq c$) and $\theta' \cap \mathtt{fv}(t|_c) = \emptyset$ (since $b \parallel c$ and $t|_b = (\lambda_{\theta'} p''.s'')u''$), contradicting $x \in \mathtt{fv}(t|_c) \vee \mathfrak{a} \ll \mathfrak{c}$. Hence $b < c$. An analysis of the three possible cases w.r.t. a, *i.e.* $a < b < c$, $b < a < c$ and $b < c < a$, allows to conclude. See [9] for further details. □

Lemma 7.10 *(Grip–Convexity). Let* $\mathfrak{a}, \mathfrak{b}, \mathfrak{c} \in \mathcal{R}ed(t)$ *such that* $\mathfrak{a} \ll \mathfrak{b}$ *and* $\mathfrak{c} < \mathfrak{b}$. *Then* $\mathfrak{a} \ll \mathfrak{c} \vee \mathfrak{c} \leq \mathfrak{a}$.

Proof. Observe that $a < b$ and $c < b$ implies that either $c \leq a$ or $a < c$. In the former case we immediately conclude. Otherwise, it suffices to notice that $a < c < b$, $a12 \leq b$ and $t|_c$ being a redex imply $a12 \leq c$, and that $c < b$, along with the variable convention, implies $\theta \cap \mathtt{fv}(t|_b) \subseteq \theta \cap \mathtt{fv}(t|_c)$, where $t|_a = (\lambda_\theta p.s)u$. Therefore $\emptyset \neq \theta \cap \mathtt{fv}(t|_c)$ so that we conclude $\mathfrak{a} \ll \mathfrak{c}$. □

The axioms FD and SO.

FD is a consequence of the gripping axioms. Thm. 3.2. in [20] states that an ARS satisfying the gripping axioms along with Self Reduction, Finite Residuals and Linearity, and whose embedding and gripping relations are acyclic, also enjoys FD. For the ARS modelling **PPC**, we have verified all the required axioms. The embedding relation being an order, and the gripping relation being included in the former, imply immediately that both are acyclic. Hence we obtain FD.

For the axiom SO, the interesting case is when the steps are nested, *i.e.* $\mathfrak{a} < \mathfrak{b}$. Let $t|_a = (\lambda_\theta p.s)u$, $t \xrightarrow{\mathfrak{b}} t'$, and $t'|_a = (\lambda_\theta p'.s')u'$. If $\{p/_\theta\ u\} = \mathtt{fail}$ is a matching failure, it suffices to observe that $\{p'/_\theta\ u'\} = \mathtt{fail}$, *cf.* Lemma 7.2. If $\{p/_\theta\ u\}$ is positive, then a simple, yet extensive, analysis resorting to various properties related to substitutions (*e.g.* that reduction steps, their targets, and residuals, are preserved by substitutions), suffices to conclude.

7.1.3. A reduction strategy for ***PPC***

This section introduces a normalising strategy $\mathcal{S}$ for **PPC**. A **prestep** is a term of the form $(\lambda_\theta\ p.t)u$, regardless of whether the match $\{p/_\theta\ u\}$ is decided or not. The rationale behind the definition of $\mathcal{S}$ can be described through two observations. First, it focuses on the leftmost–outermost (LO) prestep of t, entailing that when **PPC** is restricted to the λ-calculus it behaves exactly as the LO strategy for the λ-calculus. Second, if the match corresponding to the LO occurrence of a prestep is not decided, then the strategy selects only the (outermost) step, or steps, in that subterm which should be contracted to get it "closer" to a decided match. *E.g.* in the term $(\lambda_{\{x,y\}}\ \mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y}).y\ x)\ (\mathtt{a}\ r_1\ r_2)$, where all the r_i's are steps, the match $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ r_1\ r_2\}$ is not decided and the rôle played by r_1 is different from that of r_2 in obtaining a decided match. Replacing r_1 by an arbitrary term t_1 does not yield a decided match, *i.e.* $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ t_1\ r_2\}$ is not decided. However, replacing r_2 by $\mathtt{c}\ s_2$ (resp. by $\mathtt{d}\ s_2$) does: $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ r_1\ (\mathtt{c}\ s_2)\} = \{x \to r_1, y \to s_2\}$ (resp. $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ r_1\ (\mathtt{d}\ s_2)\} = \mathtt{fail}$). Hence, contraction of r_2 can contribute towards obtaining a decided match, while contraction of r_1 does not. A different example, in which multiple steps (including one in the pattern) are selected, is $(\lambda_{\{x,y\}}\ \mathtt{a}\ (\mathtt{b}\ \widehat{x})\ r_1.r_2)\ (\mathtt{a}\ r_3\ (\mathtt{d}\ r_4))$, where all the r_i's are steps. The strategy selects r_1 and r_3. Moreover, notice that contraction of r_4 is delayed since the match operation is not decided when the pattern is a redex (if the contractum of r_1 were *e.g.* either $\mathtt{d}\ \widehat{y}$ or $\mathtt{a}$, then the match w.r.t. $\mathtt{d}\ r_4$ would be decided without the need of reducing r_4). We note that in both examples, the decision made by the strategy (namely, to select r_2 in the first case and $\{r_1, r_3\}$ in the second one) coincides for any term having the indicated form. This decision is based solely on the structure of the term, in order to avoid the need for history or lookahead.

Formally, we define the **reduction strategy** $\mathcal{S}$ as a function from terms to sets of steps by means of an auxiliary function $\mathcal{S}_\pi$. This auxiliary function gives the *positions* of the steps to be selected: $\mathcal{S}(t) := \{\langle t, p\rangle \ \text{ s.t. } p \in \mathcal{S}_\pi(t)\}$.

In turn, the definition of $\mathcal{S}_\pi$ resorts to an additional auxiliary function, called $\mathcal{SM}$, that formulates the simultaneous structural analysis of the argument and pattern of a prestep. The arguments of $\mathcal{SM}$ are the pattern and the argument of a prestep. Its outcome is a pair of sets of positions, corresponding to steps inside the pattern and argument respectively, which could contribute to turning a non-decided match into a decided one.

The formal definition of $\mathcal{S}_\pi$ and $\mathcal{SM}$ follows. Recall that we write $bm(t, \theta)$ when $t = \widehat{x}$ for some $x \in \theta$.

$$
\begin{array}{rcll}
\mathcal{S}_\pi(x) & := & \emptyset & \\
\mathcal{S}_\pi(\widehat{x}) & := & \emptyset & \\
\mathcal{S}_\pi(\lambda_\theta\ p.t) & := & 1\mathcal{S}_\pi(p) & \text{if } p \notin \mathbf{NF} \\
\mathcal{S}_\pi(\lambda_\theta\ p.t) & := & 2\mathcal{S}_\pi(t) & \text{if } p \in \mathbf{NF} \\
\mathcal{S}_\pi((\lambda_\theta\ p.t)u) & := & \{\epsilon\} & \text{if } \{p/_\theta\ u\} \text{ decided} \\
\mathcal{S}_\pi((\lambda_\theta\ p.t)u) & := & 11G \cup 2D & \text{if } \{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle G, D\rangle \neq \langle\emptyset,\emptyset\rangle, \\
\mathcal{S}_\pi((\lambda_\theta\ p.t)u) & := & 11\mathcal{S}_\pi(p) & \text{if } \{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle\emptyset,\emptyset\rangle,\ p \notin \mathbf{NF} \\
\mathcal{S}_\pi((\lambda_\theta\ p.t)u) & := & 12\mathcal{S}_\pi(t) & \text{if } \{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle\emptyset,\emptyset\rangle,\ p \in \mathbf{NF}, t \notin \mathbf{NF} \\
\mathcal{S}_\pi((\lambda_\theta\ p.t)u) & := & 2\mathcal{S}_\pi(u) & \text{if } \{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle\emptyset,\emptyset\rangle,\ p \in \mathbf{NF}, t \in \mathbf{NF} \\
\mathcal{S}_\pi(tu) & := & 1\mathcal{S}_\pi(t) & \text{if } t \text{ is not an abstraction and } t \notin \mathbf{NF} \\
\mathcal{S}_\pi(tu) & := & 2\mathcal{S}_\pi(u) & \text{if } t \text{ is not an abstraction and } t \in \mathbf{NF} \\
\mathcal{SM}_\theta(\widehat{x}, t) & := & \langle\emptyset,\emptyset\rangle & \text{if } x \in \theta \\
\mathcal{SM}_\theta(\widehat{x}, \widehat{x}) & := & \langle\emptyset,\emptyset\rangle & \text{if } x \notin \theta \\
\mathcal{SM}_\theta(p_1p_2, t_1t_2) & := & \langle 1G_1 \cup 2G_2, 1D_1 \cup 2D_2\rangle & \text{if } t_1t_2,\ p_1p_2 \in \mathbf{MF}, \mathcal{SM}_\theta(p_i,t_i) = \langle G_i, D_i\rangle \\
\mathcal{SM}_\theta(p,t) & := & \langle \mathcal{S}_\pi(p), \emptyset\rangle & \text{if } p \notin \mathbf{MF} \\
\mathcal{SM}_\theta(p,t) & := & \langle \emptyset, \mathcal{S}_\pi(t)\rangle & \text{if } p \in \mathbf{MF}\ \&\ t \notin \mathbf{MF}\ \&\ \neg bm(p,\theta)
\end{array}
$$

Notice the similarities between the first three clauses in the definition of $\mathcal{SM}$ and those of the definition of the matching operation (*cf.* Sec. 7.1.1). Also notice that whenever a non-decided match can be turned into a decided one, the function $\mathcal{SM}$ chooses at least one (contributing) step. Formally, it can be proved that, given p and u such that $\{p/_\theta\ u\} = \mathtt{wait}$, if p' and u' exist such that $p \multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow p'$, $u \multimap\!\!\!\!\circ\!\!\!\!\twoheadrightarrow u'$ and $\{p'/_\theta\ u'\}$ is decided, then $\mathcal{SM}_\theta(p,u) \neq \langle\emptyset,\emptyset\rangle$.

Let us analyse briefly the clauses in the definition of $\mathcal{S}_\pi$. The focus on the LO prestep of a term is formalised in the first four and the last two clauses. If the LO prestep is in fact a step, then the strategy selects exactly that step; this is the meaning of the fifth clause. If the LO prestep is not a step, then $\mathcal{SM}$ is used. If it returns some steps which could contribute towards a decided match, then the strategy selects them (sixth clause). Otherwise, as we already remarked, the prestep will never turn into a step, so that the strategy looks for the LO prestep inside the components of the term (seventh, eighth and ninth clauses).

While the strategy focuses on the obtaining of a decided match for the LO prestep, it can select more steps than needed for that aim. *E.g.*, for the term $(\lambda_{\{y\}}\ \mathtt{a}\ \mathtt{b}\ \mathtt{c}\ \widehat{y}.y)\ (\mathtt{a}\ (I\ \mathtt{c})\ (I\ \mathtt{b})\ (I\ \mathtt{a}))$, the set selected by the strategy $\mathcal{S}$ is $\{I\ \mathtt{c}, I\ \mathtt{b}\}$, even if the contraction of just one step of the set suffices to make the head match decided.

Notice that $\mathcal{S}$ collapses to the LO-strategy when considering the subset of **PPC** terms given by the terms of the λ-calculus.

The reduction strategy $\mathcal{S}$ is complete, *i.e.*, if t is not a normal form, then $\mathcal{S}(t) \neq \emptyset$. Moreover, all steps in $\mathcal{S}(t)$ are *outermost*. On the other hand, notice that $\mathcal{S}$ is not *outermost fair* [25]. Indeed, given $(\lambda\, \mathtt{c}\, x.s)\ \Omega$, where Ω is a non-terminating term, $\mathcal{S}$ continuously contracts Ω, even when s contains a step.

Additionally, the steps in $\mathcal{S}(t)$ are not always hereditarily outermost, *i.e.*, universally $<$-external in the sense of [1] (*cf.* Sec. 5.2). Thus for example, given the term $t = (\lambda_x \mathtt{a}\mathtt{b}\widehat{x}.t_1)((I\mathtt{d})(I\mathtt{b})t_2)$, the strategy $\mathcal{S}$ selects the set of redexes $\{I\mathtt{d}, I\mathtt{b}\}$. By contracting only $I\mathtt{d}$, we get $t \to t' = (\lambda_x \mathtt{a}\mathtt{b}\widehat{x}.t_1)(\mathtt{d}(I\mathtt{b})t_2)$, where t' contains a (created) redex that embeds (the residual of the original) $I\mathtt{b}$. Note that the created, embedding redex is a *matching failure*. Such is always the case whenever a redex embeds a residual of $\mathcal{S}(t)$, observation which is used to prove that $\mathcal{S}(t)$ is never-gripping.

In fact, the term $t = (\lambda_x \mathtt{a}\mathtt{c}\widehat{x}.t_1)((I\mathtt{d})(I\mathtt{b})t_2)$ shows that there may be no external redexes at all. Indeed, contracting either of $I\mathtt{d}$ or $I\mathtt{b}$ create *matching failures*.

7.1.4. Properties of the reduction strategy $\mathcal{S}$

In this section we prove that $\mathcal{S}$ computes necessary (Proposition 7.14) and non-gripping (Proposition 7.16) sets. These proofs rely on the notion of *projection* of a multireduction *w.r.t. a position*. We describe briefly this notion in the following. See [9] for further details.

Let a be a position. Given $\mathfrak{b} = \langle t, b\rangle$, we say that $a \leq \mathfrak{b}$ iff $a \leq b$. This definition is extended to multisteps and reduction sequences: $a \leq \mathfrak{B}$ iff $a \leq \mathfrak{b}$ for all $\mathfrak{b} \in \mathfrak{B}$, $a \leq \delta$ is defined similarly.

If $a \leq \mathfrak{b} = \langle t, b\rangle$, implying $b = ab'$, then we define the **projection** of $\mathfrak{b}$ w.r.t. a, as follows: $\mathfrak{b}|_a = \langle t|_a, b'\rangle$. If $a \leq \mathfrak{B}$, then the projection $\mathfrak{B}|_a$ is defined as expected. We define similarly $\delta|_a$ if $a \leq \delta$. The targets of steps and reduction sequences, the residual relation, and the developments of a multistep, are compatible with these projections.

A multistep $\mathfrak{B}$ **preserves** a iff all $\mathfrak{b} \in \mathfrak{B}$ verify $b \neq a$ (or equivalently $a \leq b$ or $a \parallel b$). If $\mathfrak{B}$ preserves a, then this set can be partitioned[10] into two parts, say $\mathfrak{B}^F_a$ and $\mathfrak{B}^E_a$, such that $b \parallel a$ if $\mathfrak{b} \in \mathfrak{B}^F_a$, and $a \leq b$ if $\mathfrak{b} \in \mathfrak{B}^E_a$. Observe that $\mathfrak{B} = \mathfrak{B}^F_a \uplus \mathfrak{B}^E_a$.

[10] The relation *preserves* is similar to *free-from* (*cf.* Sec. 5.2). Moreover, the partition given by $\mathfrak{B} = \mathfrak{B}^F_a \uplus \mathfrak{B}^E_a$ bears some similarity to that described after the definition of free-from, albeit the former is restricted to multisteps that preserves some position, while the latter applies to any multistep. Additionally, free-from is a relation on abstract steps, multisteps and multireduction, while preserves is a relation between **PPC** multisteps and *positions* (not necessarily redexes).

If $t \xrightarrow{\mathfrak{B}} t'$ preserves a, then $t'|_a$ is determined by $\mathfrak{B}^F_a$, *i.e.* $t'|_a = t''|_a$ where $t \xrightarrow{\mathfrak{B}^F_a} t''$. Therefore we can extend the definition of the **projection** $\mathfrak{B}|_a$ to any $\mathfrak{B}$ preserving a: $\mathfrak{B}^F_a$ is simply ignored.

In turn, a *multireduction* Δ **preserves** a iff all its elements do. Suppose Δ preserves a, and let $t \xrightarrow{\Delta[1]} t_1 \xrightarrow{\Delta[2]} t_2 \xrightarrow{\Delta[3..]} t'$. Observe that $t|_a \xrightarrow{\Delta[1]|_a} t_1|_a \xrightarrow{\Delta[2]|_a} t_2|_a \ldots$. This observation leads to define $\Delta|_a$, the **projection** of Δ w.r.t.a, as expected.

Some notions related to multireductions are compatible with projections:

Lemma 7.11. *Let $t \xrightarrow{\Delta} t'$ and assume Δ preserves a. Then:*

(i) $t|_a \xrightarrow{\Delta|_a} t'|_a$.
(ii) *If $\mathfrak{a}\mathfrak{c} \in \mathcal{R}ed(t)$, then $\mathfrak{a}\mathfrak{c}[\![\Delta]\!]\mathfrak{d}$ iff $d = ad_1$ and $\mathfrak{c}[\![\Delta|_a]\!]\mathfrak{d}_1$.*
(iii) *If $\mathfrak{a}\mathfrak{c} \in \mathcal{R}ed(t)$, then Δ uses $\mathfrak{a}\mathfrak{c}$ iff $\Delta|_a$ uses $\mathfrak{c}$.*

Proof. This result admits a simple, though long, proof. See [9] for further details. □

In the remainder of this section, we show that $\mathcal{S}$ always selects *necessary* and *never-gripping* sets of redexes, along with the needed auxiliary results.

Lemma 7.12. *If $\{\!\{p \rhd_\theta u\}\!\}$ is positive, then $\mathcal{SM}_\theta(p,u) = \langle \emptyset, \emptyset \rangle$.*

Proof. Observe that $\{\!\{p \rhd_\theta u\}\!\}$ positive implies $p \in \boldsymbol{DS}$. Then a simple induction on p suffices. Particularly, if $p = p_1 p_2$, then $\{\!\{p \rhd_\theta u\}\!\}$ positive implies $u = u_1 u_2$ and both $\{\!\{p_i \rhd_\theta u_i\}\!\}$ positive, so that the *i.h.* on each p_i allows to conclude. □

Lemma 7.13. *Let t, u be terms and p be a pattern.*

(i) *Let $t \xrightarrow{\Delta} t'$ where $t \notin \boldsymbol{MF}$, $t' \in \boldsymbol{MF}$. Then Δ uses $\mathcal{S}(t)$ and $\mathcal{S}(t)[\![\Delta]\!] = \emptyset$.*
(ii) *Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$, where $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$ and $\{\!\{p' \rhd_\theta u'\}\!\}$ is decided. Let $\langle G, D \rangle = \mathcal{SM}_\theta(p,u)$. Then Γ uses $\mathfrak{G}$ or Π uses $\mathfrak{D}$. Moreover, $\{\!\{p' \rhd_\theta u'\}\!\}$ positive implies $\mathfrak{G}[\![\Gamma]\!] = \mathfrak{D}[\![\Pi]\!] = \emptyset$.*
(iii) *Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$, where $\{p/_\theta u\} = \mathtt{wait}$ and $\{p'/_\theta u'\}$ is decided. Let $\langle G, D \rangle = \mathcal{SM}_\theta(p,u)$. Then Γ uses $\mathfrak{G}$ or Π uses $\mathfrak{D}$. Moreover, $\{p'/_\theta u'\}$ positive implies $\mathfrak{G}[\![\Gamma]\!] = \mathfrak{D}[\![\Pi]\!] = \emptyset$.*

Proof. For items (i) and (ii), we proceed by simultaneous induction on $|t| + |u| + |p|$. The proof analyses the different cases in the definitions of $\mathcal{S}$ and $\mathcal{SM}$; it resorts to Lemma 7.11 and Lemma 7.12. Item (iii) follows easily from item (ii). See [9] for further details. □

Proposition 7.14. *Let $t \xrightarrow{\Delta} t'$ where $t \notin \boldsymbol{NF}$ and $t' \in \boldsymbol{NF}$. Then Δ uses $\mathcal{S}(t)$.*

Proof. We prove the proposition simultaneously with the following two statements.

Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$ where $p', u' \in \boldsymbol{NF}$, $\langle G, D \rangle = \mathcal{SM}_\theta(p,u) \neq \langle \emptyset, \emptyset \rangle$, and $\{\!\{p \rhd_\theta u\}\!\} = \{\!\{p' \rhd_\theta u'\}\!\} = \mathtt{wait}$ (resp. $\{p/_\theta u\} = \{p'/_\theta u'\} = \mathtt{wait}$). Then Γ uses $\mathfrak{G}$ or Π uses $\mathfrak{D}$.

The structure of the proof is similar to that of Lemma 7.13, resorting to Lemma 7.11, Lemma 7.13, and Lemma 7.12. See [9] for further details. □

Lemma 7.15. *Let $t \xrightarrow{\Delta} t'$, $\mathfrak{b} \in \mathcal{S}(t)[\![\Delta]\!]$, and $\mathfrak{a}$ verifying $\mathfrak{a} < \mathfrak{b}$. Then $\mathfrak{a}$ is a matching failure.*

Proof. The structure of the proof is similar to that of Proposition 7.14. The two statements which are proved simultaneously follow.

Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$ such that $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$ (resp. $\{p/_\theta u\} = \mathtt{wait}$), $\mathfrak{b} \in \mathfrak{G}[\![\Gamma]\!]$ or $\mathfrak{b} \in \mathfrak{D}[\![\Pi]\!]$ where $\mathcal{SM}_\theta(p,u) = \langle G, D \rangle$, and $\mathfrak{a}$ verifying $\mathfrak{a} < \mathfrak{b}$. Then $\mathfrak{a}$ is a matching failure.

See [9] for further details. □

Proposition 7.16. *Let t be a term not in normal form. Then $\mathcal{S}(t)$ is a non-gripping set.*

Proof. Let $t \xrightarrow{\Psi} u$, $\mathfrak{a} \in \mathcal{R}ed(u)$, $\mathfrak{b} \in \mathcal{S}(t)[\![\Psi]\!]$; it suffices to deduce that $\mathfrak{b}$ does not grip $\mathfrak{a}$. If $\mathfrak{a} \not< \mathfrak{b}$, then we immediately conclude. If $\mathfrak{a} < \mathfrak{b}$, then Lemma 7.15 entails that $\mathfrak{a}$ is a matching failure so $\mathfrak{b}$ cannot grip $\mathfrak{a}$. □

7.2. λ-Calculus with parallel-or

The lambda calculus extended with parallel-or also falls within the scope of our abstract proof. Its terms are given by the grammar:

$$t ::= x \mid \lambda x.t \mid t\,t \mid \mathtt{or}(t,t) \mid \mathtt{tt}$$

The reduction rules are

$$\begin{aligned}(\lambda x.s)u &\to s\{x \leftarrow u\}\\ \mathtt{or}(t,\mathtt{tt}) &\to \mathtt{tt}\\ \mathtt{or}(\mathtt{tt},t) &\to \mathtt{tt}\end{aligned}$$

It may be seen as an ARS under the standard reading of each of its elements. Two comments on this. First the notion of gripping. A step $\langle s,p\rangle$ grips a step $\langle s,q\rangle$ where $s|_q = (\lambda y.u')\,v'$, if $q1 \le p$ and $s|_p$ has a free occurrence of y (we assume the standard variable convention). Second, the fact that although this is an almost-orthogonal higher-order rewrite system, from the point of view of the underlying ARS it enjoys semantic orthogonality since the critical pair is trivial.

The reduction strategy $\mathcal{S}$ is defined by means of an auxiliary function $\mathcal{S}_\pi$ that gives the positions of the steps to be selected, as described for **PPC** in Sec. 7.1.3. In turn, $\mathcal{S}_\pi$ is defined as follows

$$\begin{aligned}\mathcal{S}_\pi((\lambda x.s)\,u) &:= \{\epsilon\}\\ \mathcal{S}_\pi(\mathtt{or}(\mathtt{tt},u)) &:= \{\epsilon\}\\ \mathcal{S}_\pi(\mathtt{or}(u,\mathtt{tt})) &:= \{\epsilon\}\\ \mathcal{S}_\pi(s\,u) &:= 1\mathcal{S}_\pi(s) && \text{if } s \neq \lambda x.s' \text{ and } s \notin \boldsymbol{NF}\\ \mathcal{S}_\pi(s\,u) &:= 2\mathcal{S}_\pi(u) && \text{if } s \neq \lambda x.s' \text{ and } s \in \boldsymbol{NF}\\ \mathcal{S}_\pi(\lambda x.t) &:= 1S(t)\\ \mathcal{S}_\pi(\mathtt{or}(s,u)) &:= 1\mathcal{S}_\pi(s) \cup 2\mathcal{S}_\pi(u) && \text{if } s \neq \mathtt{tt} \text{ and } u \neq \mathtt{tt}\\ \mathcal{S}_\pi(\mathtt{tt}) &:= \emptyset\end{aligned}$$

This strategy may be proved to produce necessary and never-gripping sets of redexes following the lines of the (more complicated) proofs developed for **PPC**. As a consequence, Theorem 6.11 is applicable and allows us to infer that $\mathcal{S}$ is normalising.

8. Conclusions

Relying on an axiomatic presentation of rewriting [20], we study normalisation for a wide class of rewriting systems. The main result of this paper states that multistep strategies that contract *sets* of *necessary* and *never-gripping* steps are normalising, *i.e.* they reach a normal form, if it exists.

This is particularly appealing for non-sequential rewrite systems, in which terms that are not in normal form may not have any needed redex, where strategies that contract only a *single step* rather than a *set of steps* and rely only on the term itself to decide which redex to reduce, cannot be normalising.

We give a concrete example of such a phenomenon by means of the pattern calculus **PPC**, that fails to be sequential, and hence includes reducible terms without any needed redex. More precisely, this behaviour is manifested by the failure mechanism of **PPC**. Consider for example the term $t = (\lambda_{\{x\}}\mathtt{abc.bd})(\mathtt{a}\,r_1\,r_2)$ where r_1 and r_2 are redexes. If r_1 rewrites to $\mathtt{d}$, then t can be reduced to $t' = (\lambda_{\{x\}}\mathtt{abc.bd})(\mathtt{a\,d}\,r_2)$ which rewrites to the normal-form I in one step, because the match of the pattern $\mathtt{abc}$ against the argument $\mathtt{a\,d}\,r_2$ yields $\mathtt{fail}$. A similar situation holds if r_2 rewrites to $\mathtt{d}$. Consequently, either r_1 or r_2 could be selected to yield a normal form from t. But choosing always r_1 would be a bad decision for other terms, as for example $u = (\lambda_{\{x\}}\mathtt{abc.bd})(\mathtt{a}\,r_1'\,r_2')$, where r_1' leads to an infinite reduction, whilst r_2' rewrites to $\mathtt{d}$. An analogous reasoning invalidates the selection of r_2.

Since the reduction strategy $\mathcal{S}$ (*cf.* Sec. 7.1.3) for **PPC** chooses a set of redexes (both steps r_1 and r_2 are selected in our example t), it is then a *multistep* reduction strategy. We prove that $\mathcal{S}$ computes necessary and never-gripping sets of steps. Following the above mentioned abstract normalisation result, this implies that the multistep strategy is normalising for **PPC**. This result can then be seen as an extension of needed normalising strategies to non-sequential rewrite systems. Moreover, our strategy $\mathcal{S}$ coincides with the leftmost–outermost strategy when restricting **PPC** to the λ-calculus.

Another interesting remark concerns the recent embedding [26] of **PPC** into higher-order pattern rewriting systems, which was motivated by the fact that one can understand some properties of **PPC** by just looking at the corresponding properties of the image of the embedding. However, as explained in Section 7.1.3, the strategy $\mathcal{S}$ is not outermost-fair, so that no available normalisation result for higher-order rewriting can be applied in our case. More importantly, the results developed in this paper can be applied to other higher-order rewriting systems for which outermost-fair strategies are, in general, difficult to compute or to express inductively.

This work also shows that the notion of *gripping* can be a useful tool to study fine properties of reduction in λ-calculi. We already noted, in Sec. 4.3, that gripping is used in an abstract proof of the finiteness of developments. We cite other links between gripping and λ-calculi.

1. Gripping explains the size-exploding phenomenon described in [2]. Let $t_0 = yxx$ and $t_{n+1} = (\lambda x.t_n)(yxx)$. The term t_n reduces in n steps to a term whose size is *exponential* in n, while the size of t_n is linear in n. We observe that the n redexes present in t_n are all linked by gripping. *E.g.*, in $t_3 = (\lambda x_3.(\lambda x_2.(\lambda x_1.yx_1x_1)(yx_2x_2))(yx_3x_3))(yxx)$ we have $a_3 \ll a_2 \ll a_1$, where a_i is the redex corresponding to the bound x_i. The successive gripping between redexes produces the multiplication of variable occurrences, and thus the explosion in the size of the normal form.
2. A link also exists between gripping and the box order on redexes in the linear substitution calculus [1]. In this calculus, the term $x[x/y][y/z]$ has two substitution redexes, corresponding to the bound occurrences of the variables x and y. In the box order, the x-redex precedes the y-redex. Beta-expansion of this term yields $(\lambda y.(\lambda x.x)y)z$, where the x-redex grips the y-redex.
3. Finally, we observe that a variant of gripping is used in [13] to characterise the cases in which α-conversion is unavoidable in calculi containing the rewrite rule $\mu x.M \to M[x := \mu x.M]$. *E.g.*, in the term $t = \mu x.F(y, \mu y.x)$, the inner redex $\mu y.x$ grips the outer one. Observe that the step $t \to F(y, \mu y'.(\mu x.F(y, \mu y.x)))$ forces the renaming of the bound variable associated to the (residual of the) gripping redex.

The scope of our work could be expanded in several ways. First, we believe that the main ideas underlying the definition of $\mathcal{S}$ for **PPC** can lead to reduction strategies for other abstract rewriting formats, such as HRS, CRS or ERS. These strategies could be proved to be normalising by resorting to the abstract normalisation proof given in this paper. This would give a powerful extension of the results in [22] to higher-order rewriting.

A second research direction is to broaden the scope of the normalisation proof presented in Section 6. More precisely, the abstract proof has been instantiated for **PPC** with the strategy $\mathcal{S}$, which always selects a subset of the *outermost* steps in a term. On the other hand, the proof does not apply to the *parallel-outermost* reduction strategy, which simultaneously selects *all* the outermost steps in a term. This is due to the fact that $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{A}$ never-gripping does not imply $\mathcal{B}$ never-gripping. For example, consider:

$$t = (\lambda_{\{x\}} \mathfrak{a}\, x.\ \underbrace{Dx}_{\mathfrak{b}})(\underbrace{I(\mathfrak{a}\,\mathfrak{b})}_{\mathfrak{a}})$$

whose only steps are $\mathfrak{a}$ and $\mathfrak{b}$. Let $\mathcal{S}(t) = \{\mathfrak{a}\} \subseteq \{\mathfrak{a}, \mathfrak{b}\} = \mathcal{O}(t)$. Remark that $\mathcal{S}(t)$ is the set of redexes selected by the strategy $\mathcal{S}$ and $\mathcal{O}(t)$ is the set of all outermost steps of t. The set $\mathcal{S}(t)$ is indeed never-gripping. However, the set $\mathcal{O}(t)$ does not satisfies the never-gripping property in the general case. Indeed, contracting $\mathfrak{a}$ results in

$$\overbrace{(\lambda_{\{x\}} \mathfrak{a}\, x.\ \underbrace{Dx}_{\mathfrak{b}'})(\mathfrak{a}\,\mathfrak{b})}^{\mathfrak{c}'}$$

where $\mathfrak{b} [\![\mathfrak{a}]\!] \mathfrak{b}'$ and $\mathfrak{c}' \ll \mathfrak{b}'$. Hence $\mathcal{O}(t)$ does not enjoy the never-gripping property.

We conjecture that some variation of the given proof could apply to the parallel-outermost strategy in some cases, for example for **PPC**. In this perspective, it could be possible that the property of always selecting *necessary* sets of steps could suffice to guarantee that a reduction strategy is normalising. A proof of this conjecture, or a counterexample falsifying it, would be an interesting result in this direction.

Acknowledgements

To Vincent van Oostrom for having pointed out a mistake in a previous version of this work. To Yann Régis-Gianas for discussions on coinduction. To Beniamino Accattoli who provided valuable comments. This work was partially supported by LIA INFINIS, the ECOS-Sud cooperation program between France and Argentina, and by the grants PUNQ of the Universidad Nacional de Quilmes and UBACyT of the Universidad de Buenos Aires, Argentina.

References

[1] B. Accattoli, E. Bonelli, D. Kesner, C. Lombardi, A nonstandard standardization theorem, in: S. Jagannathan, P. Sewell (Eds.), POPL '14, ACM, 2014, pp. 659–670.
[2] B. Accattoli, U. Dal Lago, Beta reduction is invariant, indeed, in: T. Henzinger, D. Miller (Eds.), CSL-LICS '14, ACM, 2014, pp. 8:1–8:10.
[3] S. Antoy, A. Middeldorp, A sequential reduction strategy, Theoret. Comput. Sci. 165 (1) (1996) 75–95.
[4] T. Balabonski, On the implementation of dynamic patterns, in: E. Bonelli (Ed.), HOR '10, in: Electronic Proceedings in Theoretical Computer Science, vol. 49, July 2010, pp. 16–30.
[5] T. Balabonski, Optimality for dynamic patterns: extended abstract, in: M. Fernández, T. Kutsia, W. Schreiner (Eds.), PPDP '10, ACM, July 2010, pp. 16–30.
[6] H.P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, Elsevier, Amsterdam, 1984.
[7] G. Berry, Bottom-up computations of recursive programs, RAIRO Theor. Inform. Appl. 10 (3) (1976) 47–82.
[8] E. Bonelli, D. Kesner, C. Lombardi, A. Ríos, Normalisation for dynamic pattern calculi, in: A. Tiwari (Ed.), RTA, in: LIPIcs, vol. 15, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2012, pp. 117–132.
[9] E. Bonelli, D. Kesner, C. Lombardi, A. Ríos, An abstract normalisation result with applications to non-sequential calculi, http://arxiv.org/abs/1412.2118, 2014.
[10] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, Cambridge, 1998.

[11] G. Boudol, Computational semantics of term rewriting systems, in: M. Nivat, J.C. Reynolds (Eds.), Algebraic Methods in Semantics, Cambridge University Press, 1985, pp. 169–236.
[12] H.B. Curry, R. Feys, Combinatory Logic, North-Holland Publishing Company, Amsterdam, 1958.
[13] J. Endrullis, C. Grabmayer, J-W. Klop, V. van Oostrom, On equal μ-terms, Theoret. Comput. Sci. 412 (28) (2011) 3175–3202.
[14] G.P. Huet, J-J. Lévy, Computations in orthogonal rewriting systems, I and II, in: Computational Logic – Essays in Honor of A. Robinson, 1991, pp. 395–443.
[15] B. Jay, Pattern Calculus: Computing with Functions and Structures, Springer Publishing Company, Incorporated, 2009.
[16] B. Jay, D. Kesner, Pure pattern calculus, in: Peter Sestoft (Ed.), ESOP '06, in: LNCS, vol. 3924, Springer-Verlag, 2006, pp. 100–114.
[17] B. Jay, D. Kesner, First-class patterns, J. Funct. Programming 19 (2) (2009) 191–225.
[18] R. Kennaway, Sequential evaluation strategies for parallel-or and related reduction systems, Ann. Pure Appl. Logic 43 (1) (1989) 31–56.
[19] J-W. Klop, Combinatory Reduction Systems, PhD thesis, Utrecht University, 1980.
[20] P-A. Melliès, Description abstraite des Systèmes de Réécriture, PhD thesis, Université Paris VII, 1996.
[21] M.J. O'Donnell, Computing in Systems Described by Equations, LNCS, vol. 58, Springer-Verlag, 1977.
[22] R.C. Sekar, I.V. Ramakrishnan, Programming in equational logic: beyond strong sequentiality, Inform. and Comput. 104 (1) (1993) 78–109.
[23] V. van Oostrom, Normalisation in weakly orthogonal rewriting, in: P. Narendran, M. Rusinowitch (Eds.), RTA, in: LNCS, vol. 1631, Springer-Verlag, 1999, pp. 60–74.
[24] F. van Raamsdonk, Confluence and Normalisation for Higher-Order Rewriting, PhD thesis, Vrije University, 1996.
[25] F. van Raamsdonk, Outermost-fair rewriting, in: P. de Groote (Ed.), TLCA, in: LNCS, vol. 1210, Springer-Verlag, 1997, pp. 284–299.
[26] F. van Raamsdonk, V. van Oostrom, The dynamic pattern calculus as a higher-order pattern rewriting system, in: K. Rose (Ed.), HOR '14, July 2014.