Axiomatic Rewriting Theory II

The $\lambda\sigma$-calculus enjoys finite normalisation cones

Paul-André Melliès
Preuves, Programmes, Systèmes
CNRS EPJ-2025, Université Paris 7
<mellies@pps.jussieu.fr>

Abstract

Every needed strategy is normalising in the λ-calculus. Here, we extend the result to the $\lambda\sigma$-calculus, a λ-calculus with explicit substitutions. The extension requires to consider rewriting systems with critical pairs, confluent or non confluent, and to develop for them a satisfactory theory of needed normalisation. Our idea is to count for every term M the number of its normalising paths, up to Lévy permutation equivalence. We deduce from the standardisation theorem in [21] that every needed strategy normalises when this number is finite. The number is zero or one in the λ-calculus, and we show after [20] that it is finite in the $\lambda\sigma$-calculus.

Introduction: the $\lambda\sigma$-calculus

The $\lambda\sigma$-calculus was introduced in [1] as a bridge between the λ-calculus and its machine implementations. In contrast with a usual λ-term, a $\lambda\sigma$-term may contain *explicit substitutions* which are propagated and substituted according to a series of σ-rules. An additional rule, the *Beta*-rule, mimics the β-rule and produces a new explicit substitution, see figure 1.

Many positive results were proved on the calculus. In particular, the σ-calculus is shown strongly normalising and confluent ; and the whole $\lambda\sigma$-calculus confluent on closed terms, see [6, 25]. However, the author shows in [19] that the simply typed $\lambda\sigma$-calculus is *not* strongly normalising. One research direction suggested at the end of the paper is to reformulate the calculus and design a strongly normalising typed version. Another direction investigated here is to localize normalising strategies inside the $\lambda\sigma$-calculus. Of course, every normalising strategy of the λ-calculus defines a normalising $\lambda\sigma$-strategy by translation. But this is not enough: adopting the original philosophy of the $\lambda\sigma$-calculus, we look for normalising $\lambda\sigma$-strategies which do not necessarily mimic the λ-calculus.

Before any exploration of normalisation in the $\lambda\sigma$-calculus, we should remember the incredible amount of work already devoted to understanding normalisation in the λ-calculus, and more generally orthogonal systems. The first description of a normalising strategy in the λ-calculus appears in [5]: the "normal" strategy which reduces at each step the leftmost-outermost redex in the λ-term. Later, the original scheme was generalised by Lévy: a β-redex u in a λ-term M is "needed" when every rewriting path from M to its normal form computes a residual of u, and a strategy is "needed" when it reduces at each step a "needed" redex. Lévy proves in [16] that

Terms	M	$::=$	$\mathbf{1} \mid MN \mid \lambda M \mid M[s]$
Substitutions	s	$::=$	$id \mid \uparrow \mid M \cdot s \mid s \circ t$

Beta	$(\lambda M)N \to M[N \cdot id]$
App	$(MN)[s] \to M[s]N[s]$
Abs	$(\lambda M)[s] \to \lambda(M[\mathbf{1} \cdot (s \circ \uparrow)])$
Clos	$M[s][t] \to M[s \circ t]$
VarCons	$\mathbf{1}[M \cdot s] \to M$
VarId	$\mathbf{1}[id] \to \mathbf{1}$
Map	$(M \cdot s) \circ t \to M[t] \cdot (s \circ t)$
IdL	$id \circ s \to s$
Ass	$(s_1 \circ s_2) \circ s_3 \to s_1 \circ (s_2 \circ s_3)$
ShiftCons	$\uparrow \circ (M \cdot s) \to s$
ShiftId	$\uparrow \circ id \to \uparrow$

Figure 1: The $\lambda\sigma$-calculus

every needed strategy is normalising in the λ-calculus. The proof makes a significant use of the standardisation theorem.

In this paper, we extend Lévy's result to the $\lambda\sigma$-calculus. The task is not easy for three reasons at least:

- The $\lambda\sigma$-calculus is not orthogonal (11 critical pairs, see figure 2). Despite the work of Boudol [4] in the mid-eighties and more recently of Clark and Kennaway [7], our current understanding of non orthogonal dynamics is limited. We take the opportunity here to develop a general theory of *neededness* in rewriting systems with critical pairs, much broader in scope than λ-calculi with explicit substitutions.

- Huet and Lévy prove in [13] that needed strategies normalise in orthogonal systems, but the property breaks in the presence of critical pairs, as we illustrate with an example in section 3.3. Henceforth a frontier must be traced between the "good" and the "bad" systems, with the prerequisite that orthogonal systems *and* the $\lambda\sigma$-calculus should appear on the same side of the frontier,

- The counter-example in [19] illustrates that the $\lambda\sigma$-calculus exhibits unexpected computations, and that the normalisation result we aim at requires a rigorous analysis of its dynamics. Moreover, the model should be as generic as possible to be re-applicable to similar systems, in particular other λ-calculi with explicit substitutions.

The paper is divided in two parts. We develop a theory of normalisation in part I, and apply it to the $\lambda\sigma$-calculus in part II. We choose to expose the theory for axiomatic rewriting systems as they appear in [21], so as not to limit ourself to any particular syntactical formulation. Section 1 summarizes the results of [21]. We explain that every axiomatic rewriting system can be seen as a particular 2-category, and express the standardisation theorem in that perspective. In section 2, we introduce the subcategory of *external* rewriting paths in any axiomatic system, and study its decomposition properties. Finally, in section 3, we define the notion of *needed* strategy and establish a needed normalisation theorem starting from the hypothesis of finite normalisation cones.

In part II, we justify the normalisation theorem proved in part I by showing that the $\lambda\sigma$-calculus enjoys finite normalisation cones. In section 4, we recall the interpretation method developed by Hardin [10], expressed here as an adjunction property between the rewriting orders of the λ-calculus and the $\lambda\sigma$-calculus. Functorial translations between the $\lambda\sigma$-calculus and the λ-calculus are discussed. In section 5, we prove that external rewriting paths in the $\lambda\sigma$-calculus are projected by σ-normalisation to external rewriting paths in the λ-calculus. In section 6 we deduce from section 5 and a simple argument based on König's lemma, on strong normalisation of the substitution σ-calculus, and on normalisation of needed strategies in the λ-calculus, that the $\lambda\sigma$-calculus enjoys finite normalisation cones.

$App + Beta$	$(\lambda a)[s](b[s])$	$\overset{App}{\leftarrow}$	$((\lambda a)b)[s]$	$\overset{Beta}{\rightarrow}$	$a[b \cdot id][s]$
$Clos + App$	$(ab)[s \circ t]$	$\overset{Clos}{\leftarrow}$	$(ab)[s][t]$	$\overset{App}{\rightarrow}$	$(a[s](b[s]))[t]$
$Clos + Abs$	$(\lambda a)[s \circ t]$	$\overset{Clos}{\leftarrow}$	$(\lambda a)[s][t]$	$\overset{Abs}{\rightarrow}$	$(\lambda(a[\mathbf{1} \cdot s \circ \uparrow]))[t]$
$Clos + VarId$	$\mathbf{1}[id \circ s]$	$\overset{Clos}{\leftarrow}$	$\mathbf{1}[id][s]$	$\overset{VarId}{\rightarrow}$	$\mathbf{1}[s]$
$Clos + VarCons$	$\mathbf{1}[(a \cdot s) \circ t]$	$\overset{Clos}{\leftarrow}$	$\mathbf{1}[a \cdot s][t]$	$\overset{VarCons}{\rightarrow}$	$a[t]$
$Clos + Clos$	$a[s][t \circ t']$	$\overset{Clos}{\leftarrow}$	$a[s][t][t']$	$\overset{Clos}{\rightarrow}$	$a[s \circ t][t']$
$Ass + Map$	$(a \cdot s) \circ (t \circ t')$	$\overset{Ass}{\leftarrow}$	$((a \cdot s) \circ t) \circ t'$	$\overset{Map}{\rightarrow}$	$(a[t] \cdot s \circ t) \circ t'$
$Ass + IdL$	$id \circ (s \circ t)$	$\overset{Ass}{\leftarrow}$	$(id \circ s) \circ t$	$\overset{IdL}{\rightarrow}$	$s \circ t$
$Ass + ShiftId$	$\uparrow \circ (id \circ s)$	$\overset{Ass}{\leftarrow}$	$(\uparrow \circ id) \circ s$	$\overset{ShiftId}{\rightarrow}$	$\uparrow \circ s$
$Ass + ShiftCons$	$\uparrow \circ ((a \cdot s) \circ t)$	$\overset{Ass}{\leftarrow}$	$(\uparrow \circ (a \cdot s)) \circ t$	$\overset{ShiftCons}{\rightarrow}$	$s \circ t$
$Ass + Ass$	$(s \circ s') \circ (t \circ t')$	$\overset{Ass}{\leftarrow}$	$((s \circ s') \circ t) \circ t'$	$\overset{Ass}{\rightarrow}$	$(s \circ (s' \circ t)) \circ t'$

Figure 2: The eleven critical pairs of the $\lambda\sigma$-calculus

Part I: Theory

1 Standardisation

The section motivates axiomatic rewriting systems and their 2-categorical presentation.

1.1 The λ-calculus

Consider the λ-calculus, its reduction graph $\mathcal{G}_\lambda$, and the free category $\lambda^{\rightarrow}$ with λ-terms as objects and finite sequences of β-redexes $M \xrightarrow{r_1} \cdots \xrightarrow{r_n} N$ as morphisms. One may enrich $\lambda^{\rightarrow}$ with an equivalence $\equiv$ expressing the fact that two morphisms $f, g : M \to N$ compute the same thing, but in a different order. This is the idea of Lévy in [16], who deduces $\equiv$ from permutation diagrams like in figure 3.

$$
\begin{tikzcd}
 & xP'Q \arrow[dr, "u'"] & \\
xPQ \arrow[ur, "v"] \arrow[dr, "u"'] \arrow[rr, phantom, "(1)"] & & xP'Q' \\
 & xPQ' \arrow[ur, "v'"'] &
\end{tikzcd}
\quad \text{or} \quad
\begin{tikzcd}
 & \Delta P' \arrow[dr, "u'"] & \\
\Delta P \arrow[ur, "v"] \arrow[dr, "u"'] \arrow[rr, phantom, "(2)"] & & P'P' \\
 & PP \arrow[ur, "v_1;v_2"'] &
\end{tikzcd}
\qquad
\begin{array}{c}
P \xrightarrow{\beta} P' \\
Q \xrightarrow{\beta} Q' \\
\Delta = (\lambda x.xx)
\end{array}
$$

Figure 3: Two redex permutations in the λ-calculus

In [20, 21], we start to orient the permutation equivalence $\equiv$ and express that way the familiar *standardisation* procedure. The idea is to replace the equivalence $f \equiv g$ by "standardisation cells" $f \Rightarrow g$ indicating that the rewriting path g is more standard than the rewriting path f. The cells $f \Rightarrow g$ are constructed using a relation $\rhd_\lambda$ on rewriting paths expressing local permutation diagrams, and their "orientation".

The relation $\rhd_\lambda$ is defined as follows. Recall from [16] that a β-redex $M \to N$ is characterized by the tree-occurrence of the β-pattern $(\lambda x.P)Q$ it contracts in M. A β-redex $u : M \to N$ *nests* a β-redex $v : M \to P$ when the occurrence of u is a prefix of the occurrence of v. Finally, two rewriting paths $M \xrightarrow{f,g} N$ verify $f \rhd_\lambda g$ when f and g are two developments of a set $\{u, v\}$ of coinitial and different β-redexes u and v, f contracts v first, g contracts u first, and v does not nest u:

$$f = M \xrightarrow{v}\xrightarrow{u'} N \quad \text{and} \quad g = M \xrightarrow{u}\xrightarrow{h} N$$

Observe that f is of length 2, with u' the (unique) residual of u after v, and that g is of length at least 1, with h a development of the residuals of v after u. See [16] for a definition of residual and developments in the λ-calculus.

Many of the dynamical properties of the λ-calculus are retained by the structure $(\mathcal{G}_\lambda, \rhd_\lambda)$. For instance, we may use $\rhd_\lambda$ to distinguish two kinds of redex permutation in the calculus, keeping in mind that $f \rhd_\lambda g$ indicates that g is more standard than f:

- permutation (1) in figure 3 from $v; u'$ to $u; v'$ is *reversible* because the two redexes u and v are disjoint (u does not nest v and v does not nest u) and therefore the two paths $v; u'$ and $u; v'$ are equivalent from the standardisation perspective. We have $(v; u' \rhd_\lambda u; v')$ and $(u; v' \rhd_\lambda v; u')$.

- permutation (2) in figure 3 from $v; u'$ to $u; v_1; v_2$ is *irreversible* because the redex u nests v, thus the outside-in path $u; v_1; v_2$ is strictly more standard than the inside-out path $v; u'$. We have $(v; u' \rhd_\lambda u; v_1; v_2)$ and $\neg(u; v_1; v_2 \rhd_\lambda v; u')$.

This motivates the following refinement of traditional abstract rewriting systems, as exposed in [12, 15] for example.

1.2 Axiomatic rewriting systems

Definition 1 (ARS) An *axiomatic rewriting system* is a pair $(\mathcal{G}, \rhd)$ where $\mathcal{G}$ is a graph and $\rhd$ is a binary relation between paths of $\mathcal{G}$. By a *graph* $\mathcal{G}$, we mean a set $\mathcal{T}$ of vertices (= terms), a set $\mathcal{R}$ of edges (= redexes), and two functions ∂_0 and ∂_1 from $\mathcal{R}$ to $\mathcal{T}$ (=the source and target functions). Given an edge u, we write $u : M \to N$ when $\partial_0(u) = M$ and $\partial_1(u) = N$. We recall that a *path* in a graph $\mathcal{G}$ is a sequence $f = (M_1, r_1, M_2, ..., M_m, r_m, M_{m+1})$ where $r_i : M_i \to M_{i+1}$ for every $i \in [1...m]$. We write $f : M_1 \to M_{m+1}$. The length of f is m ; f is said to be empty when $m = 0$. Two paths $f : M \to N$ and $g : P \to Q$ are coinitial (resp. cofinal) when $M = P$ (resp. $N = Q$). $f; g : M \to Q$ denotes the concatenation of two paths $f : M \to P$ and $g : P \to Q$ in $\mathcal{G}$.

One benefit of the axiomatic approach is to express various rewriting frameworks under the same elementary form, from Petri nets to the λ-calculus. In [21], we introduce a series of axioms on $(\mathcal{G}, \rhd)$ describing the generic properties of *oriented permutations* in a rewriting system. We show that the system $(\mathcal{G}_\lambda, \rhd_\lambda)$ associated to the λ-calculus in section 1.1 verifies the axioms. We also define for every first order rewriting term (with or without critical pairs) an axiomatic rewriting system $(\mathcal{G}, \rhd)$ with terms as vertices and redexes as edges, and a permutation relation $v; u' \rhd u; h$ precisely when:

- u and v are two different and *compatible* redexes,
- v does not nest u wrt. the tree nesting ordering,
- the redex u' is the unique residual of u after v, and h is a development of the residuals of v after u.

We show in [21] that every such system $(\mathcal{G}, \rhd)$ verifies the axioms of the theory. So, from now on, we identify a first order rewriting system like the $\lambda\sigma$-calculus with the axiomatic rewriting system $(\mathcal{G}_{\lambda\sigma}, \rhd_{\lambda\sigma})$ it defines.

We go a step further. Recall that a 2-category is a category $\mathcal{C}$ where for every two objects P and Q of $\mathcal{C}$, the class $\mathcal{C}(P, Q)$ of morphisms $P \to Q$ is not only a set, also a category. This means that we consider morphisms $\alpha : f \Rightarrow g$ called *cells*, between morphisms $f : P \to Q$ and $g : P \to Q$ of the category $\mathcal{C}$. Cells may be composed "vertically" and "horizontally": we write $\alpha * \beta : f \Rightarrow h$ for the vertical composite of $\alpha : f \Rightarrow g$ and $\beta : g \Rightarrow h$, and $\alpha; \beta : f_1; f_2 \Rightarrow g_1; g_2$ for the horizontal composite of

$\alpha : f_1 \Rightarrow f_2$ and $\alpha : g_1 \Rightarrow g_2$. Diagrammatically:

$$
\begin{tikzcd}
M \arrow[r, bend left=60, "f"] \arrow[r, "g" description] \arrow[r, bend right=60, "h"'] \arrow[r, phantom, shift left=4, "\Downarrow\alpha"] \arrow[r, phantom, shift right=4, "\Downarrow\beta"] & N
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
M \arrow[r, bend left=50, "f"] \arrow[r, bend right=50, "h"'] \arrow[r, phantom, "\Downarrow\alpha*\beta"] & N
\end{tikzcd}
$$

$$
\begin{tikzcd}
M \arrow[r, bend left=50, "f_1"] \arrow[r, bend right=50, "g_1"'] \arrow[r, phantom, "\Downarrow\alpha"] & N \arrow[r, bend left=50, "f_2"] \arrow[r, bend right=50, "g_2"'] \arrow[r, phantom, "\Downarrow\beta"] & P
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
M \arrow[r, bend left=50, "f_1;f_2"] \arrow[r, bend right=50, "f_2;g_2"'] \arrow[r, phantom, "\Downarrow\alpha;\beta"] & P
\end{tikzcd}
$$

Finally, the interchange law asks that composing four cells $\alpha_1 : f_1 \Rightarrow g_1$ and $\beta_1 : g_1 \Rightarrow h_1$, $\alpha_2 : f_2 \Rightarrow g_2$ and $\beta_2 : g_2 \Rightarrow h_2$ vertically then horizontally as $(\alpha_1 * \alpha_2);(\beta_1 * \beta_2) : f_1;f_2 \Rightarrow h_1;h_2$, or horizontally then vertically as $(\alpha_1;\beta_1) * (\alpha_2;\beta_2) : f_1;f_2 \Rightarrow h_1;h_2$ is equivalent.

We interpret any axiomatic rewriting system $(\mathcal{G}, \rhd)$ as a 2-category whose cells $f \Rightarrow g$ are precisely the "standardisation cells" mentioned in section 1.1.

Definition 2 ($\overset{step}{\Rightarrow}$, $\overset{R}{\Rightarrow}$, $\overset{I}{\Rightarrow}$) A *standardisation step* $\alpha : d \overset{step}{\Rightarrow} e$ in $(\mathcal{G}, \rhd)$ is a quadruple $\alpha = (d_1, f, g, d_2)$ of paths verifying:

$$d \; = \overset{d_1}{\to}\overset{f}{\to}\overset{d_2}{\to} \qquad e \; = \overset{d_1}{\to}\overset{g}{\to}\overset{d_2}{\to} \qquad f \rhd g$$

We extend to any $(\mathcal{G}, \rhd)$ the taxonomy observed in $(\mathcal{G}_\lambda, \rhd_\lambda)$ and declare the step α *reversible* when $(g \rhd f)$, *irreversible* when $\neg(g \rhd f)$. We write

1. $d \overset{R}{\Rightarrow} e$ when there exists a (possibly empty) sequence of reversible steps
$$d \overset{step}{\Rightarrow} \cdots \overset{step}{\Rightarrow} e$$

2. $d \overset{I}{\Rightarrow} e$ when there exists a sequence
$$d \overset{step}{\Rightarrow} \cdots \overset{step}{\Rightarrow} e$$
containing one irreversible step at least.

Definition 3 ($\Rightarrow$, $\simeq$) Any axiomatic rewriting system $(\mathcal{G}, \rhd)$ is interpreted as the 2-category:

1. with underlying category the free category on the graph $\mathcal{G}$,

2. with cell $\alpha : d \Rightarrow e$ any triple (d, e, R) such that $d \overset{R}{\Rightarrow} e$, and any triple (d, e, I) such that $d \overset{I}{\Rightarrow} e$. Two cells $\alpha = (d, e, X)$ and $\beta = (e, f, Y)$ compose as $\alpha * \beta = (d, f, X \star Y)$ where $R \star R = R$ and $I \star R = R \star I = I \star I = I$,

3. the identity cell $\mathrm{id}^d : d \Rightarrow d$ is defined as (d, d, R).

Recall that an isomorphism in a category $\mathcal{C}$ is a morphism $f : M \to N$ with an inverse $f^{-1} : N \to M$ such that $f; f^{-1} = \mathrm{id}_M$ and $f^{-1}; f = \mathrm{id}_N$, and that a 2-isomorphism $\alpha : d \Rightarrow e$ between $d, e : P \to Q$ in a 2-category $\mathcal{C}$ is an isomorphism in the hom-category $\mathcal{C}(P, Q)$. Observe that in $(\mathcal{G}, \rhd)$, the 2-isomorphisms $\alpha : d \Rightarrow e$ are precisely the cells $\alpha = (d, e, R)$ constructed with reversible permutations: we write $\alpha : d \simeq e$ in that case. Reinterpreting figure 3 in figure 4, permutation (1) becomes a 2-isomorphism and permutation (2) a proper 2-cell in the axiomatic system $(\mathcal{G}_\lambda, \rhd_\lambda)$.

$$\begin{tikzcd}[column sep=small] & xP'Q \arrow[dr, "u'"] & \\ xPQ \arrow[ur, "v"] \arrow[dr, "u"'] \arrow[rr, phantom, "\simeq"] & & xP'Q' \\ & xPQ' \arrow[ur, "v'"'] & \end{tikzcd} \quad \text{and} \quad \begin{tikzcd}[column sep=small] & \Delta P' \arrow[dr, "u'"] & \\ \Delta P \arrow[ur, "v"] \arrow[dr, "u"'] \arrow[rr, phantom, "\Downarrow"] & & P'P' \\ & PP \arrow[ur, "v_1; v_2"'] & \end{tikzcd}$$

Figure 4: Two standardisation cells in $(\mathcal{G}_\lambda, \rhd_\lambda)$

1.3 Standardisation theorem

Definition 4 ($\equiv$, $\sqsubseteq$) We say that a rewriting path $g : M \to N$ is *more standard* than a rewriting path $f : M \to N$ when there exists a cell $f \Rightarrow g$ in $(\mathcal{G}, \rhd)$. We define the permutation equivalence $\equiv$ as the least equivalence relation containing two paths f and g when $f \Rightarrow g$. So, the equivalence $f \equiv g$ between $f, g : M \to N$ means that f and g are in the same connected component of the hom-category $(\mathcal{G}, \rhd)(M, N)$, alternatively, that there is a "zig-zag"

$$f \Rightarrow h_1 \Leftarrow h_2 \Rightarrow \cdots \Leftarrow h_n \Rightarrow g$$

Observe that the definition is equivalent to Lévy's original definition in the case of $(\mathcal{G}_\lambda, \rhd_\lambda)$. We say that $f : M \to P$ *extends* $g : M \to N$, and write $g \sqsubseteq f$, when there exists a path $h : N \to P$ such that $f \equiv g; h$.

Definition 5 (standard) A rewriting path $f : M \to N$ is *standard* when it cannot be further standardised, meaning that every cell $\alpha : f \Rightarrow g$ starting from f is a 2-isomorphism.

In [21], we prove a *standardisation theorem* for every axiomatic rewriting system $(\mathcal{G}, \rhd)$. The theorem states that:

1. for every rewriting path $d : M \to N$, there exists a standard path $\Downarrow_d : M \to N$ and a unique cell $\alpha : d \Rightarrow \Downarrow_d$,

2. two paths $d, e : M \to N$ equal modulo $\equiv$ have their standard path $\Downarrow_d, \Downarrow_e : M \to N$ equal modulo $\simeq$:
$$\forall M \overset{d,e}{\to} N, \quad d \equiv e \Rightarrow \Downarrow_d \simeq \Downarrow_e$$

$$
\begin{tikzcd}
\Delta(Ia) \arrow[rr, "v"] \arrow[dd, "u"'] & \arrow[d, phantom, "\Downarrow"] & \Delta a \arrow[dd, "u'"] \\
& a(Ia) \arrow[dr, "v_2'"] & \\
(Ia)(Ia) \arrow[ur, "v_1"] \arrow[dr, "v_2"'] & \simeq & aa \\
& (Ia)a \arrow[ur, "v_1'"'] &
\end{tikzcd}
\qquad
\begin{array}{c}
\Delta = \lambda x.xx \\
I = \lambda x.x \\
v; u' \rhd_\lambda u; v_1; v_2' \\
v_1; v_2' \rhd_\lambda v_2; v_1' \\
v_2; v_1' \rhd_\lambda v_1; v_2'
\end{array}
$$

Figure 5: A standardisation procedure in $(\mathcal{G}_\lambda, \rhd_\lambda)$

Property 1. and 2. correspond to what is called *existence* and *uniqueness* of standardisation in [16, 4, 9, 20]. By property 2, there exists a unique standard path (unique modulo $\simeq$) in each Lévy equivalence class $\equiv$. By property 1, this path d is characterised by the equivalence $d \simeq \Downarrow_d$. We illustrate this in figure 5, where the rewriting path $v; u' : \Delta(Ia) \to aa$ enjoys two standard paths $u; v_1; v_2'$ and $u; v_2; v_1'$ which are equal modulo $\simeq$.
In [21], we prove also that every $(\mathcal{G}, \rhd)$ verifies properties A and B:

A. two rewriting paths d and e are standard when their composite $d; e$ is standard. This means that for every pair $M \xrightarrow{d} N \xrightarrow{e} P$ of composable paths:

$$d; e \simeq \Downarrow_{d;e} \quad \Rightarrow \quad d \simeq \Downarrow_d \text{ and } e \simeq \Downarrow_e$$

B. The standardisation algorithm introduced in [9, 20, 21] computes the standard path $\Downarrow_d$ by extracting recursively an "outermost" redex in the rewriting path d. The definition of the algorithm implies that for every triple $M \xrightarrow{d} N \xrightarrow{e} P \xrightarrow{f} Q$ of composable paths:

$$d; \Downarrow_{e;f} \simeq \Downarrow_{d;e;f} \quad \Rightarrow \quad d; \Downarrow_e \simeq \Downarrow_{d;e}$$

We prefer to keep the axiomatics in [21] implicit and unless explicitly mentioned (in section 3) limit ourself to the following definition of an axiomatic system $(\mathcal{G}, \rhd)$: it is a 2-category equipped with an operator $\Downarrow$ on morphisms (the categorician will recognize a *lax* endofunctor $\Downarrow : (\mathcal{G}, \rhd) \to (\mathcal{G}, \rhd)$) verifying properties 1, 2, A and B.

2 The category $\mathcal{E}$ of external rewriting paths

2.1 Motivation and definition

Standard rewriting paths do not compose generally. In the λ-calculus for instance, the two β-redexes

$$(\lambda x.y)(Ia) \xrightarrow{u} (\lambda x.y)a \qquad\qquad (\lambda x.y)a \xrightarrow{v} y$$

are standard but their composite $u; v$ is not standard because of the irreversible permutation:

$$((\lambda x.y)(Ia) \xrightarrow{u} (\lambda x.y)a \xrightarrow{v} y) \quad \rhd_\lambda \quad ((\lambda x.y)(Ia) \xrightarrow{w} y)$$

This suggests to strengthen the notion of standardness and define a *subcategory* of "external computations" expressing in any $(\mathcal{G}, \rhd)$ the idea of leftmost-outermost computation in the λ-calculus. For example, we look for a syntax-free explanation for the "externality" of a β-redex like

$$r : (\lambda x.\lambda y.x)MP \to (\lambda y.M)P$$

In [20], we suggest that r is external *because* any path $r; f$ obtained by extending r with a standard path $f : (\lambda y.M)P \to Q$ is standard. This leads to the following definition in any axiomatic system $(\mathcal{G}, \rhd)$:

Definition 6 (external) A rewriting path $e : M \to N$ is *external* in $(\mathcal{G}, \rhd)$ when for every right-composable rewriting path $f : N \to P$, we have:

$$e; \Downarrow_f \simeq \Downarrow_{e;f}$$

The first observation is that, contrarily to standard paths, external paths do compose.

Lemma 7 The class $\mathcal{E}$ of external paths defines a subcategory of $(\mathcal{G}, \rhd)$.

Proof Identity paths $\mathrm{id}_P : P \to P$ are external. Moreover, given two external paths $d : M \to N$ and $e : N \to P$, and any path $f : P \to Q$:

$$\begin{array}{rcll} d; e; \Downarrow_f & \simeq & d; \Downarrow_{e;f} & \text{by } e \in \mathcal{E} \text{ and left composition of 2-isomorphisms,} \\ & \simeq & \Downarrow_{d;e;f} & \text{by } d \in \mathcal{E} \end{array}$$

We conclude that the composite $d; e$ is external. ■

The following definition appears already in Hilken's analysis of long-$\beta\eta$-normal forms in the λ-calculus, see [11].

Definition 8 (split monic, normal form) A path $f : M \to N$ is *split monic* when there exists a path $g : N \to M$ such that $f; g \simeq \mathrm{id}_M$. A term M is *normal* when every path $f : M \to N$ is a split monic. We call also M a *normal form.* A path $d : M \to N$ is called *normalising* when N is normal.

Lemma 9 Every standard and normalising path is external in $(\mathcal{G}, \rhd)$.

Proof Let $e : M \to N$ be standard and normalising, and $f : N \to P$ be any right-composable path. By definition of a normal form, there exists a path $g : P \to N$ such that $f; g \equiv \mathrm{id}_N$. We claim that id_N is standard. Indeed, it follows from property 2 and $\mathrm{id}_N \equiv \mathrm{id}_N; \Downarrow_{\mathrm{id}_N}$ that $\Downarrow_{\mathrm{id}_N} \simeq \Downarrow_{\mathrm{id}_N; \Downarrow_{\mathrm{id}_N}}$. We have $\mathrm{id}_N; \Downarrow_{\mathrm{id}_N} \simeq \Downarrow_{\mathrm{id}_N; \Downarrow_{\mathrm{id}_N}}$ by definition of id_N, and our claim that $\mathrm{id}_N \simeq \Downarrow_{\mathrm{id}_N}$ follows by property A.

We obtain the series of equivalence:

$$\begin{array}{rcll} e; \Downarrow_{f;g} & \simeq & e; \Downarrow_{\mathrm{id}_N} & \text{by } f; g \equiv \mathrm{id}_N \text{ and property 2,} \\ & \simeq & e & \text{by } \mathrm{id}_N \simeq \Downarrow_{\mathrm{id}_N}, \\ & \simeq & \Downarrow_e & \text{because } e \text{ is standard,} \\ & \simeq & \Downarrow_{e;f;g} & \text{by } e \equiv e; f; g \text{ and property 2.} \end{array}$$

Here, we apply property B. and deduce the equivalence $e; \Downarrow_f \simeq \Downarrow_{e;f}$. We conclude that e is external. ■

2.2 Decomposition properties

In lemma 7, we show that external paths compose well. We study now how they decompose. Let us observe first that

Lemma 10 Every external path is standard in $(\mathcal{G}, \rhd)$.

PROOF Suppose that $e : M \to N$ is external. It follows from $e \equiv e; \Downarrow_{\mathrm{id}_N}$ and property 2 that $\Downarrow_e \simeq \Downarrow_{e;\Downarrow_{\mathrm{id}_N}}$. We deduce that

$$e; \Downarrow_{\mathrm{id}_N} \;\simeq\; \Downarrow_{e;\mathrm{id}_N} \;\simeq\; \Downarrow_e \;\simeq\; \Downarrow_{e;\Downarrow_{\mathrm{id}_N}}$$

by externality of e. We apply property A. and conclude that $e \simeq \Downarrow_e$. ∎

Lemma 11 (right decomposition) The path $N \overset{e}{\to} P$ in a pair $M \overset{d}{\to} N \overset{e}{\to} P$ is always external when the composite $M \overset{d;e}{\to} P$ is external.

PROOF Given a pair $M \overset{d}{\to} N \overset{e}{\to} P$ with external composite $M \overset{d;e}{\to} P$, and any right-composable path $f : P \to Q$, we have:

$$\begin{array}{rcll} d; e; \Downarrow_f & \simeq & \Downarrow_{d;e;f} & \text{because } d;e \text{ is external} \\ & \simeq & \Downarrow_{d;e;\Downarrow_f} & \text{by } d;e;f \equiv d;e;\Downarrow_f \text{ and standardisation} \end{array}$$

It follows from property A. that $e; \Downarrow_f \simeq \Downarrow_{e;\Downarrow_f}$, then from $e; f \equiv e; \Downarrow_f$ and standardisation that $\Downarrow_{e;\Downarrow_f} \simeq \Downarrow_{e;f}$. We conclude that e is external in $(\mathcal{G}, \rhd)$. ∎

Lemma 12 (left decomposition) Suppose that for every two coinitial paths $f : M \to P$ and $g : M \to Q$ in $(\mathcal{G}, \rhd)$, we have two cofinal paths $f' : Q \to N$ and $g' : P \to N$ such that $f; g' \equiv g; f'$. In that case, the path $M \overset{d}{\to} N$ in a pair $M \overset{d}{\to} N \overset{e}{\to} P$ is external when the composite $M \overset{d;e}{\to} P$ is external.

PROOF We prove that $d; \Downarrow_f \simeq \Downarrow_{d;f}$ for any path $f : N \to Q$ right-composable with d, and conclude that d is external. By hypothesis, there are two paths e' and f' such that $f; e' \equiv e; f'$. Property B. and the series of equivalence

$$\begin{array}{rcll} \Downarrow_{d;f;e'} & \simeq & \Downarrow_{d;e;f'} & \text{by } e; f' \equiv f; e', \\ & \simeq & d; e; \Downarrow_{f'} & \text{by } d; e \in \mathcal{E}, \\ & \simeq & d; \Downarrow_{e;f'} & \text{by } e \in \mathcal{E}, \\ & \simeq & d; \Downarrow_{f;e'} & \text{by } e; f' \equiv f; e'. \end{array}$$

implies that $d; \Downarrow_f \simeq \Downarrow_{d;f}$. We conclude. ∎

Let $(\mathcal{G}, \rhd)/\equiv$ be the category obtained by quotienting the 2-category $(\mathcal{G}, \rhd)$ with the relation $\equiv$. Huet and Lévy prove in [13] that the quotient category $(\mathcal{G}, \rhd)/\equiv$ has pushouts when the calculus is orthogonal, see [22] for a discussion. This means that for every pair $P \overset{f}{\leftarrow} M \overset{g}{\to} Q$ of coinitial rewriting paths, there exists a pair $P \overset{g'}{\to} N \overset{f'}{\to} Q$ of cofinal rewriting paths making the diagram (1) commute modulo Lévy equivalence:

$$\begin{tikzcd} M \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "\equiv"] & Q \arrow[d, dotted, "f'"] \\ P \arrow[r, dotted, "g'"'] & N \end{tikzcd} \qquad f; g' \equiv g; f' \tag{1}$$

Furthermore, for every two rewriting paths $h_1 : P \to O$ and $h_2 : Q \to O$ verifying $f; h_1 \equiv g; h_2$, there exists a rewriting path $h : N \to O$ unique modulo $\equiv$, making the diagram (2) commute in $(\mathcal{G}, \rhd)/\equiv$:

$$\begin{tikzcd}
M \arrow[r, "g"] \arrow[d, "f"'] & Q \arrow[d, "f'"'] \arrow[ddr, bend left, "h_2"] & \\
P \arrow[r, "g'"] \arrow[drr, bend right, "h_1"'] & N \arrow[dr, dotted, "h"] & \\
 & & O
\end{tikzcd} \tag{2}$$

This algebraic strengthening of the Church-Rosser property insures that the premises of lemma 12 are verified. Because every external path decomposes on the right *and* on the left by lemma 11 and 12, a rewriting path is external if and only if every redex it contracts is external.

This is not true anymore in a rewriting system with critical pairs: an external path may also contract non external redexes. For instance, Boudol introduces in [4], section 5.1, the following first-order rewriting system:

$$\begin{array}{lcl@{\qquad\qquad}lcl@{\qquad\qquad}lcl}
F(A,B,x) & \to & C & F(A,A,A) & \to & C & \Omega & \to & A \\
F(B,x,A) & \to & C & F(B,B,B) & \to & C & \Omega & \to & B \\
F(x,A,B) & \to & C & & & & & &
\end{array}$$

He observes that the term $F(\Omega, \Omega, \Omega)$ is confluent, normalises to C, but does not contain any external redex (Boudol speaks about "needed" and "strongly needed" redexes. Every strongly needed or "necessary" redex in his sense is external in our sense, but the converse is not true.) This is precisely the fact that $F(-,-,-)$ describes a *non sequential* function, the so-called Gustave function appearing in [3]. For instance, the redex

$$r : F(\Omega,\Omega,\Omega) \to F(A,\Omega,\Omega)$$

is not external because the path

$$F(\Omega,\Omega,\Omega) \stackrel{r}{\to} F(A,\Omega,\Omega) \to F(A,A,\Omega) \to F(A,A,B) \stackrel{F}{\to} C$$

is not standard. However, the redex r may be extended to an external *path*

$$F(\Omega,\Omega,\Omega) \stackrel{r}{\to} F(A,\Omega,\Omega) \to F(A,B,\Omega)$$

which cannot be computed as a sequence of external redexes. Another illustration appears in the $\lambda\sigma$-calculus, equipped with the usual tree nesting ordering. A simple argument shows that for every M, N, s, t, the path

$$e: \quad ((\lambda M)N)[s][t] \quad \stackrel{Beta1}{\to} \quad M[N \cdot id][s][t] \quad \stackrel{Clos}{\to} \quad M[(N \cdot id) \circ s][t]$$

is external. However, the first redex it contracts

$$((\lambda M)N)[s][t] \quad \stackrel{Beta1}{\to} \quad M[N \cdot id][s][t]$$

is not external because of the irreversible permutation

$$Beta1; Clos2 \ \rhd_{\lambda\sigma} \ Clos1; Beta2$$

in the diagram:

$$\begin{tikzcd}
((\lambda M)N)[s][t] \arrow[r, "Beta1"] \arrow[d, "Clos1"'] \arrow[dr, phantom, "\Leftarrow"] & M[N \cdot id][s][t] \arrow[r, "Clos"] \arrow[d, "Clos2"] & M[(N \cdot id) \circ s][t] \\
((\lambda M)N)[s \circ t] \arrow[r, "Beta2"'] & M[N \cdot id][s \circ t] &
\end{tikzcd}$$

It would be easy to show that if we choose M, N, s and t in such a way that $P = ((\lambda M)N)[s][t]$ has a normal form Q, and extend e to a normalising path $e; f : P \to Q$, the normalising path $e; f$ cannot be reorganised modulo permutation as a sequence of external redexes. We conclude that external redexes do not provide a *complete* description of the normalisation space. This explains why we need to introduce the notion of external *path* instead of external *redex* when we decide to analyse the $\lambda\sigma$-calculus, and more generally rewriting systems with critical pairs.

2.3 Another definition of $\mathcal{E}$

We discuss briefly another definition of externality appearing in [20]. In the λ-calculus, an external β-redex like

$$r : (\lambda x.\lambda y.x)MP \to (\lambda y.M)P$$

is neither duplicated or erased by any rewriting path $d : (\lambda x.\lambda y.x)MP \to Q$. Thus, whenever $r \sqsubseteq d$, the redex r appears "preserved" by the path d in the sense that d factors as $d = D; r'; E$ where r' is the (unique) residual of r through D. This may be expressed this other way: for every two rewriting paths $f : (\lambda y.M)P \to Q$ and $d : (\lambda x.\lambda y.x)MP \to Q$,

$$d \equiv r; f \quad \Rightarrow \quad (\exists g, \exists \alpha, \ f \equiv g \ \text{ and } \ \alpha : d \Rightarrow r; g)$$

Here, g is constructed as the path $D'; E$ where D' is a path residual of D after r in the λ-calculus (see [16, 13, 4] for a definition.) We show that the characterization extends to any axiomatic rewriting system $(\mathcal{G}, \rhd)$:

Lemma 13 A rewriting path $e : M \to N$ is external $(\mathcal{G}, \rhd)$ if and only if for every right-composable path $f : N \to P$ and for every path $d : M \to P$:

$$d \equiv e; f \quad \Rightarrow \quad (\exists g : N \to P, \ \exists \alpha, \quad f \equiv g \ \text{ and } \ \alpha : d \Rightarrow e; g) \tag{3}$$

PROOF (only if) take $g = \Downarrow_f$. (if) we choose a path $f : N \to P$ and apply property (3) to $d = \Downarrow_{e;f}$. We deduce that there exists a morphism $g \equiv f$ and a cell α such that $\alpha : \Downarrow_{e;f} \Rightarrow e; g$. Composing α with $\beta : g \Rightarrow \Downarrow_f$, we obtain a cell $\Downarrow_{e;f} \Rightarrow e; \Downarrow_f$. By standardness of $\Downarrow_{e;f}$, we conclude that $\Downarrow_{e;f} \simeq e; \Downarrow_f$. The equivalence is verified for every path $f : N \to P$ right-composable with $e : M \to N$, which is therefore external. ■

3 Finite normalisation cones

3.1 On neededness in the presence of critical pairs

Our next task is to characterise the *needed* paths of an axiomatic system $(\mathcal{G}, \triangleright)$. We start from the definition of Maranget in [18] for orthogonal rewriting systems: A redex $r : M \to N$ is needed when it has a residual (at least) after any rewriting path $f : M \to P$, unless f contracts at some step a residual of r. We want to extend the definition to rewriting systems with critical pairs. Consider a rewriting system like:

$$A \to B \qquad\qquad A \to C$$

The critical pair in A implies that the redex $r : A \to B$ has no residual after the reduction $f : A \to C$. Should we conclude that r is not needed? Not really. Maybe *needed* is not the right word to qualify r and should be replaced by *unavoidable*, since every path from A either contracts a residual of r, or a redex forming a critical pair with a residual of r. But if we choose to keep the word *needed*, then we should definitely consider r as needed, and adapt Maranget's definition accordingly: We say that a redex $r : M \to N$ is needed when every rewriting path $f : M \to P$ extending r ($r \sqsubseteq f$) contracts a residual of r at least. Check that the definition is equivalent with the previous one in an orthogonal rewriting system, and that the redex $r : A \to B$ is needed in our example.

The definition is fine, but not adapted to our current setting: remember that we never use or define the notion of *residual* in our axiomatic systems. This choice is not ornamental: we believe that $\triangleright$ is the fundamental structure, and that the residual relation is one possible way to reconstruct $\triangleright$ in various settings. So, we need to characterize needed paths another way.

Definition 14 ($||-||$) We write $||f||$ the *length* of a rewriting path $f : M \to N$, which is the number of redexes it contracts.

In [21], we extend to any axiomatic rewriting system $(\mathcal{G}, \triangleright)$ a property established by Lévy in the λ-calculus:

C. Given any pair of composable paths $M \xrightarrow{f} N \xrightarrow{g} P$ in $(\mathcal{G}, \triangleright)$, the length $||\Downarrow_g||$ is bounded by the length $||\Downarrow_{f;g}||$:

$$\forall(M, N, P), \forall M \xrightarrow{f} N \xrightarrow{g} P : \qquad ||\Downarrow_{f;g}|| \;\geq\; ||\Downarrow_g||$$

The reason for the inequality is that every redex contracted in $\Downarrow_g$ corresponds to a unique redex contracted in $\Downarrow_{f;g}$. Observe now that the inequality becomes strict when the path $f : M \to N$ is a needed redex $r : M \to N$ because one residual of r at least appears contracted in $\Downarrow_{r;g}$. We use the property to characterise the needed paths of $(\mathcal{G}, \triangleright)$:

Definition 15 (needed path) A rewriting path $f : M \to N$ is *properly needed* in an axiomatic rewriting system $(\mathcal{G}, \triangleright)$ when:

$$\forall P, \forall N \xrightarrow{g} P, \qquad ||\Downarrow_{f;g}|| \;>\; ||\Downarrow_g||$$

We say that $f : M \to N$ is *needed* when f is properly needed, or when f is split monic.

We will prove several results on the class of needed paths in $(\mathcal{G}, \rhd)$. Again, we are careful to limit ourself to five properties of $|| - ||$: it associates an ordinal to every rewriting path f and verifies properties C. D. E. and F:

D. In every axiomatic rewriting system, two paths f and g involved in a reversible permutation $f \rhd g$ have lengths 2. By definition of $\simeq$, this implies:

$$\forall (P, Q), \forall P \stackrel{d,e}{\to} Q : \qquad d \simeq e \;\Rightarrow\; ||d|| = ||e||$$

E. In every axiomatic rewriting system, the length of a path is defined as its number of rewriting steps. We deduce:

$$\forall (M, N, P), \forall M \xrightarrow{f} N \xrightarrow{g} Q : \qquad ||f; g|| \;\geq\; ||f||$$

F. Suppose that two paths $f; \Downarrow_g$ and $\Downarrow_g$ have the same length in $(\mathcal{G}, \rhd)$. Then, the path f is empty and a fortiori split monic.

$$\forall (M, N, P), \forall M \xrightarrow{f} N \xrightarrow{g} Q : \qquad ||f; \Downarrow_g|| = ||\Downarrow_g|| \;\Rightarrow\; f \text{ is split monic.}$$

We prove that:

Lemma 16 The class $\mathcal{N}$ of needed paths defines a subcategory of $(\mathcal{G}, \rhd)$. Moreover, for every pair of rewriting paths $d : L \to M$ and $f : P \to Q$, the path $d; e; f$ is properly needed when $e : M \to P$ is properly needed.

Proof The last assertion follows by definition of properly needed paths, and by property C. We deduce that $\mathcal{N}$ is a category by observing that all identities are needed, and that split monics compose well. ∎

Lemma 17 Every external path is needed in $(\mathcal{G}, \rhd)$.

Proof We claim that any external path $e : M \to N$ is properly needed when it is not split monic. Indeed, we have for any right-composable path $f : N \to P$:

$$\begin{array}{rcll} ||\Downarrow_f|| & \leq & ||\Downarrow_{e;f}|| & \text{by property C,} \\ & = & ||e; \Downarrow_f|| & \text{by property D. and externality of } e, \\ & \neq & ||\Downarrow_f|| & \text{by property F.} \end{array}$$

The strict inequality $||\Downarrow_f|| < ||\Downarrow_{e;f}||$ follows. We conclude that every external path e is either split monic or properly needed — thus in both cases needed. ∎

3.2 Needed strategies

We choose a definition of strategy which generalizes the traditional notion of "one step" strategy, as it appears for instance in [2, 13]. In particular, the definition of $\mathcal{S}$ as a *total* function is not really restrictive because the equality $\mathcal{S}(P) = \mathrm{id}_P$ is always possible.

Definition 18 (strategy) A strategy $\mathcal{S}$ in $(\mathcal{G}, \triangleright)$ is a function which associates to every term P a rewriting path

$$\mathcal{S}(P) : P \to Q$$

The indexed family of paths $\mathcal{S}^n(P) : P \to Q_n$ is constructed by induction on $n \in \mathbb{N}$:

- $\mathcal{S}^0(P) = \mathrm{id}_P$
- $\mathcal{S}^{n+1}(P) = \mathcal{S}^n(P); \mathcal{S}(Q_n)$ when $\mathcal{S}^n(P) : P \to Q_n$.

We say that a rewriting path $d : P \to Q$ *follows* a strategy $\mathcal{S}$ when there exists $n \in \mathbb{N}$ such that $d = \mathcal{S}^n(P)$.

Definition 19 (normalising strategy) A strategy $\mathcal{S}$ is declared *normalising* when for every term P with a normal form, there exists $n \in \mathbb{N}$ such that $\mathcal{S}^n(P) : P \to Q$ computes a normal form Q.

Definition 20 (needed strategy) A strategy $\mathcal{S}$ is declared *needed* when, for every term P:

1. the path $\mathcal{S}(P) : P \to Q$ is needed,
2. the path $\mathcal{S}(P) : P \to Q$ is properly needed when there exists a properly needed path $P \to R$ outgoing P,
3. Q has a normal form when P has a normal form and $\mathcal{S}(P) : P \to Q$.

Condition 3 could be easily replaced by a similar condition on needed paths, asking that a term Q has a normal form whenever P has a normal form and $f : P \to Q$ is needed. This is a matter of taste: by keeping condition 3 here, we ensure lemma 17. Condition 2 forbids $\mathcal{S}$ to stop on a term for no reason.

3.3 Finite normalisation cone

Huet and Lévy prove in [13] that every needed strategy normalises in an orthogonal rewriting system. This is not true any more in the presence of critical pairs. Consider the term A in the first order rewriting system:

$$A \to A \qquad\qquad A \to B$$

Observe that the redex $r : A \to A$ is external and a fortiori needed, and that the term A has normal form B. However, the needed strategy $\mathcal{S}$ defined by $\mathcal{S}(A) = r : A \to A$ and $\mathcal{S}(B) = \mathrm{id}_B$ does not normalise A.

At that point, we introduce a criterion to detect *well-behaved* systems among rewriting systems with critical pairs. The idea is to count the number of normalising paths from a term, modulo Lévy equivalence $\equiv$.

- The number is infinite from A to B in the last example because every rewriting path $A \to A \cdots A \to B$ defines its own permutation class modulo $\equiv$,

- the number is at most one in any orthogonal rewriting system. This follows the existence of pushouts in the category $(\mathcal{G}, \rhd)/\equiv$, see [16, 13]. Indeed, suppose that there exists two different normalising paths $M \xrightarrow{f} P$ and $M \xrightarrow{g} Q$. By existence of pushouts (see section 2.2 for a definition), there exist two rewriting paths $f' : Q \to N$ and $g' : P \to N$ such that $f; g' \equiv g; f'$. But P and Q are normal forms, and f' and g' are therefore split monics. We conclude that $f \equiv g$ in a setting like [16, 13, 4, 21] where the identities are the only split monics. Otherwise, an elementary categorical reasoning shows that there exists two paths $h : P \to Q$ and $h^{-1} : Q \to P$ verifying $h; h^{-1} \equiv \mathrm{id}_P$ and $h^{-1}; h \equiv \mathrm{id}_Q$, such that $f; h \equiv g$ and $g; h^{-1} \equiv f$. This implies that the normalisation cone is at most singleton in definition 21 (it is empty when the term is not normalising).

- We prove in part II that the number is finite from any closed $\lambda\sigma$-term.

This leads to the following definition:

Definition 21 (normalisation cone) We call *normalisation cone* from a term M any cone $(e_i : M \to P_i)_{i \in I}$ of normalising paths such that every normalising path $f : M \to Q$ factors through a unique path $e_i : M \to P_i$ modulo $\equiv$:

$$\exists! i \in I,\ \exists P_i \xrightarrow{h} Q, \qquad M \xrightarrow{f} Q \quad \equiv \quad M \xrightarrow{e_i} P_i \xrightarrow{h} Q$$

An axiomatic rewriting system $(\mathcal{G}, \rhd)$ enjoys finite normalisation cones when from any term M there exists a *finite* normalisation cone $(e_i : M \to P_i)_{i \in I}$.

3.4 Normalisation theorem

The main motivation of definition 21 is theorem 1.

Theorem 1 (normalisation) Suppose that the axiomatic system $(\mathcal{G}, \rhd)$ enjoys finite normalisation cones. Then, every needed strategy $\mathcal{S}$ is normalising in $(\mathcal{G}, \rhd)$.

Proof By hypothesis, we may assign to any term M a finite normalisation cone $(e_i^M : M \to P_i)_{i \in I_M}$, and choose each e_i^M standard. We define $\mathtt{prof}(M)$ as:

1. $\mathtt{prof}(M) = \infty$ when the cone $(e_i^M : M \to P_i)_{i \in I_M}$ is empty, or equivalently, when M has no normal form,

2. otherwise, $\mathtt{prof}(M)$ is the length $||e_i^M||$ of the longest path $e_i^M : M \to P_i$ in the cone $(e_i^M : M \to P_i)_{i \in I_M}$:

$$\mathtt{prof}(M) = \mathtt{max}\{||e_i^M|| \mid i \in I_M\}$$

We use the universality condition in the definition of a normalisation cone, and associate to any rewriting path $f : M \to N$ the function:

$$f^* : I_N \to I_M$$

which maps each index $j \in I_N$ to the unique index $i \in I_M$ such that

$$e_i^M \sqsubseteq f; e_j^N \tag{4}$$

Now, we claim that $||\Downarrow_f|| = ||\Downarrow_g||$ whenever two normalising paths $M \xrightarrow{f} P$ and $M \xrightarrow{g} Q$ verify $f \sqsubseteq g$. Indeed, by definition of $f \sqsubseteq g$, there exists a path $h : P \to Q$ such that $f; h \equiv g$. Because P is normal, the path h is split monic, which means that there exists a path h' such that $h; h' \equiv \mathrm{id}_P$. The equivalence $g; h' \equiv f$ follows. We apply uniqueness of standardisation (property 2) and externality of $\Downarrow_f$ and $\Downarrow_g$ (lemma 9) to obtain the series of equivalence

$$\Downarrow_f ; \Downarrow_h \;\simeq\; \Downarrow_{f;h} \simeq \Downarrow_g \qquad \Downarrow_g ; \Downarrow_{h'} \simeq \Downarrow_{g;h'} \simeq \Downarrow_f$$

We apply property D. and E. and conclude that $||\Downarrow_f|| \leq ||\Downarrow_g|| \leq ||\Downarrow_f||$.

We apply immediately this result to (4) and deduce:

$$\forall j \in I_N, \quad ||e^M_{f^*(j)}|| \quad = \quad ||\Downarrow_{f;e^N_j}|| \tag{5}$$

We claim (α) that $\mathtt{prof}(M) > \mathtt{prof}(N)$ when $f : M \to N$ is properly needed and N has a normal form. By definition of a properly needed path, we have:

$$\forall j \in I_N, \quad ||\Downarrow_{f;e^N_j}|| \quad > \quad ||e^N_j||$$

We apply property (5) and deduce that

$$\forall j \in I_N, \quad ||e^M_{f^*(j)}|| \quad > \quad ||e^N_j||$$

The inequation applies in particular to the index $j \in I_N$ defining $\mathtt{prof}(N)$ as $||e^N_j||$. We conclude that $\mathtt{prof}(M) > \mathtt{prof}(N)$.

We claim (β) that a term M is normal whenever there exists a split monic path $f : M \to P$ to a normal form P. Indeed, by definition of split monic, there exists a path $g : P \to N$ such that $f; g \equiv \mathrm{id}_M$. Now, the path g is split monic too because P is normal. It follows that M and P are isomorphic in the category $(\mathcal{G}, \rhd)/\equiv$. We conclude that M is a normal form.

Let $\mathcal{S}$ be a needed strategy. We prove that $\mathtt{prof}(M) > \mathtt{prof}(N)$ whenever M is non normal and has a normal form. Indeed, let $f : M \to P$ be a normalising path from M. By lemma 9, the path f is external, and by lemma 17, it is needed. By definition of needed, f is either split monic or properly needed, but split monic is forbidden because M is non normal and claim (β). We conclude that f is properly needed. By condition 2 and 3 in the definition of a needed strategy, the path $\mathcal{S}(M) : M \to N$ is properly needed, and N has a normal form. We conclude that $\mathtt{prof}(M) > \mathtt{prof}(N)$ by claim (α).

We have just established that $\mathtt{prof}(M) > \mathtt{prof}(N)$ and that N has a normal form whenever M is non normal and has a normal form. We conclude by well-foundedness that the strategy $\mathcal{S}$ is normalising. ∎

3.5 Fair and hyper normalisation

The reader interested in *fair* normalisation should observe that theorem 1 adapts easily to that setting. The idea appearing with the definition of “eventually outermost” in [8] is to mix fairly efficient and unefficient computations, and prove a normalisation result. Suppose for instance that a strategy $\mathcal{S}$ verifies for every term P that:

1. there exists $n \in \mathbb{N}$ such that $\mathcal{S}^n(P) : P \to Q$ is needed,
2. there exists $n \in \mathbb{N}$ such that $\mathcal{S}^n(P) : P \to Q$ is properly needed when there exists a properly needed path $P \to R$ outgoing P,
3. Q has a normal form when P has a normal form and $\mathcal{S}(P) : P \to Q$.

We may translate any such "eventually needed" or "hyper needed" strategy $\mathcal{S}$ as the strategy $\mathcal{S}'$ which associates to any term P the path

$$\mathcal{S}'(P) = \mathcal{S}^n(P) : P \to Q$$

where $n \in \mathbb{N}$ is chosen such that:

- $\mathcal{S}^n(P) : P \to Q$ is needed,
- $\mathcal{S}^n(P) : P \to Q$ is properly needed when there exists a properly needed path $P \to R$ outgoing P.

The strategy $\mathcal{S}'$ is needed. Moreover, normalisation of $\mathcal{S}'$ implies normalisation of $\mathcal{S}$. By theorem 1 on $\mathcal{S}'$, we deduce that the strategy $\mathcal{S}$ is normalising when the rewriting system $(\mathcal{G}, \triangleright)$ enjoys finite normalisation cones.

Part II: Application to the $\lambda\sigma$-calculus

We devote this part II to a single proof: the fact that the $\lambda\sigma$-calculus enjoys finite normalisation cones. As part I (theorem 1) explains, this implies that every needed strategy of the $\lambda\sigma$-calculus normalises.

4 Remarks on the interpretation method

We recall the so-called interpretation method developed by Hardin in [10]. Every (possibly open) λ-term M may be seen as a (closed) $\lambda\sigma$-term $U(M)$ by de Bruijn translation. This translation is a functor $U : \lambda \to \lambda\sigma$ between the two order categories $(\lambda, \leq_\lambda)$ and $(\lambda\sigma, \leq_{\lambda\sigma})$: the objects of the first category are the (possibly open) λ-terms, with $M \leq_\lambda N$ when M β-rewrites to N, the objects of the second category are the (closed) $\lambda\sigma$-terms, with $M \leq_{\lambda\sigma} N$ when M $\lambda\sigma$-rewrites to N. Conversely, every (closed) $\lambda\sigma$-term may be interpreted as a (possibly open) λ-term by σ-normalisation, this defining a functor $\sigma : \lambda\sigma \to \lambda$. The two functors appear to be adjoint in fact, $U : \lambda \to \lambda\sigma$ right adjoint to $\sigma : \lambda\sigma \to \lambda$:

$$\forall M \in \lambda\sigma, \forall N \in \lambda, \qquad \sigma(M) \leq_\lambda N \iff M \leq_{\lambda\sigma} U(N) \tag{6}$$

Functoriality breaks when we move from preorders $(\lambda, \leq_\lambda)$ and $(\lambda\sigma, \leq_{\lambda\sigma})$ to the free categories $\lambda^{\to}$ and $\lambda\sigma^{\to}$ derived from the reduction graphs of (possibly open) λ-terms and (closed) $\lambda\sigma$-terms respectively. On one hand, there is no canonical "forgetful" functor $U : \lambda^{\to} \to \lambda\sigma^{\to}$ because of the different strategies to σ-normalise a $\lambda\sigma$-term ; on the other hand, no "interpretation" functor $\sigma : \lambda\sigma^{\to} \to \lambda^{\to}$ because generally σ-normalisation duplicates *Beta*-redexes.

A solution is to consider rewriting paths $M \to N$ modulo reversible permutations, in the categories $\lambda^{\simeq} = (\lambda^{\to}/\simeq)$ and $\lambda\sigma^{\simeq} = (\lambda\sigma^{\to}/\simeq)$. First, there is a forgetful functor $U : \lambda^{\to} \to \lambda\sigma^{\simeq}$ obtained by contracting at each β-step $M \to N$ the corresponding *Beta*-redex $U(M) \to P$, followed by a standard σ-normalisation path $P \to U(N)$ avoiding the *Clos* and *Ass*-rules: there is always such a path $P \to \sigma(P)$, and all choices are equivalent modulo $\simeq$ by standardness and absence of critical pair in that fragment of the σ-calculus, see figure 2. Conversely, there is an interpretation functor $\sigma : \lambda\sigma^{\to} \to \lambda^{\simeq}$ obtained by contracting "in parallel" for each *Beta*-redex $M \to N$ the β-redexes obtained after σ-normalisation in $\sigma(M)$. Because every two such β-redexes are disjoint, the different computations of the β-redexes are equal modulo $\simeq$.

We obtain the commutative diagrams:

$$\begin{tikzcd} \lambda^{\simeq} \arrow[dr] & \\ \lambda^{\to} \arrow[u, "\simeq"] \arrow[r, "U"'] & \lambda\sigma^{\simeq} \end{tikzcd} \qquad\qquad \begin{tikzcd} \lambda\sigma^{\simeq} & \\ \lambda\sigma^{\to} \arrow[u, "\simeq"] \arrow[r, "\sigma"'] & \lambda^{\simeq} \end{tikzcd} \tag{7}$$

The existence of a functor $\lambda^{\simeq} \to \lambda\sigma^{\simeq}$ making the first diagram commute is a consequence of the fact that two disjoint β-redexes induce disjoint *Beta* and σ-normalisation procedures in $\lambda\sigma$. As for the second diagram, there exists no functor $\lambda\sigma^{\simeq} \to \lambda^{\simeq}$ making it commute because two disjoint *Beta*-redexes in a $\lambda\sigma$-term M may have nested σ-projections in $\sigma(M)$, for instance when $M = P[P \cdot id]$ and $P = (\lambda\mathbf{2})\mathbf{1} = (\lambda\mathbf{1}[\uparrow])\mathbf{1}$.

5 From standard $\lambda\sigma$-paths to standard λ-paths

Starting from the two functors discussed in section 4

$$U : \lambda^{\simeq} \to \lambda\sigma^{\simeq} \qquad\qquad \sigma : \lambda\sigma^{\to} \to \lambda^{\simeq}$$

we observe that σ does not preserve standardness in general, for instance when it translates the standard rewriting $\lambda\sigma$-path

$$((\lambda\mathbf{11})\mathbf{1})[(\lambda\mathbf{1})\mathbf{1} \cdot id] \to ((\lambda\mathbf{11})\mathbf{1})[\mathbf{1}[\mathbf{1} \cdot id] \cdot id] \to (\mathbf{11})[\mathbf{1} \cdot id][\mathbf{1}[\mathbf{1} \cdot id] \cdot id] \tag{8}$$

as an inside-out computation in the λ-calculus:

$$(\lambda x.xx)((\lambda y.y)\mathbf{1}) \to (\lambda x.xx)\mathbf{1} \to \mathbf{11}$$

We develop in section 7.2 a simple solution to that mismatch by considering that redexes of s are nested by redexes of M in the $\lambda\sigma$-term $M[s]$. This "dynamical" ordering prevents the path (8) to be standard, and we check in section 7.2 that the new and tighter notion of standardness is preserved by interpretation σ.

Here, we adopt a much simpler approach and prove that σ preserves traditional standardness when one restricts to paths $f : M \to U(N)$ ending on the de Bruijn translation $U(N)$ of a λ-term N.

Lemma 22 Every standard rewriting path $f : M \to U(N)$ in $\lambda\sigma^{\to}$ is interpreted as a standard path $\sigma(f) : \sigma(M) \to N$ in $\lambda^{\simeq}$.

We start the proof with a definition and two lemmas.

Definition 23 (enshrine,annihilate,preserve) An occurrence x *enshrines* a redex $u : P \to Q$ when x is a strict prefix of the occurrence of u. A redex u *annihilates* an occurrence x when the occurrence of u is a prefix of x. A path $u_1; \cdots; u_n$ *preserves* an occurrence x when none of the redexes u_i annihilates x.

Lemma 24 Suppose that an occurrence x enshrines a redex $r : P \to Q$. For every rewriting path $\lambda\sigma$-path $f = r; g$,

1. either f preserves the occurrence x,
2. or f is equal modulo $\simeq$ to a rewriting path $h_1; u; v; h_2$ where the path h_1 preserves the occurrence x ; the occurrence x enshrines the redex u ; the redex v annihilates the occurrence x.

PROOF Let $u_1; \cdots; u_j : P \to R$ be a $\lambda\sigma$-path which preserves the occurrence x, and contracts a redex enshrined in x. Let u_k be the last redex enshrined in x. If $k < j$, the redex u_{k+1} which is not enshrined in x may be permuted *reversibly* before u_k:

$$\begin{tikzcd}
 & & P_{k+1} \arrow[dr, "u_{k+1}"] & & \\
P_{k-1} \arrow[r, "u_{k-1}"] & P_k \arrow[ur, "u_k"] \arrow[dr, "u'_k"'] \arrow[rr, phantom, "\simeq"] & & P_{k+2} \arrow[r, "u_{k+2}"] & P_{k+3} \\
 & & P'_{k+1} \arrow[ur, "u'_{k+1}"'] & &
\end{tikzcd}$$

Observe that the resulting path $u_i; \cdots; u'_k; u'_{k+1}; \cdots; u_j$ preserves x, and contracts the redex u'_{k+1} enshrined in x. Repeating the process $j-k$ times, one constructs a path $v_1; \cdots; v_j \simeq u_1; \cdots; u_j$ whose last redex v_j is enshrined in x.

Let $f = r_1; \cdots; r_n$ be a path and x an occurrence which enshrines r_1. We suppose that f does not preserve x. Let r_{j+1} be the first redex in $f = r_1; \cdots; r_n$ which annihilates x. By our previous result, there exists a path $v_1; \cdots; v_j$ equal to $r_1; \cdots; r_j$ modulo $\simeq$ whose last redex $u = v_j$ is enshrined in x. We conclude. ■

Given a pair of composable $\lambda\sigma$-redexes $M \xrightarrow{u} N \xrightarrow{v} P$, we say that the redex u *creates* the redex v when v is not the residual after u of any redex in M. We illustrate the definition with two examples:

$$(\lambda M)N[s] \xrightarrow{u} M[N \cdot id][s] \xrightarrow{v} M[(N \cdot id) \circ s]$$

or

$$(\lambda M)[s]N \xrightarrow{u} (\lambda M[\mathbf{1} \cdot (id \circ \uparrow)])N \xrightarrow{v} M[\mathbf{1} \cdot (id \circ \uparrow)][N \cdot id] \tag{9}$$

Lemma 25 Suppose that an occurrence x enshrines a redex $r : P \to Q$. Every standard path $f = r; g$ preserves the occurrence x when:

- x is a cons-node $(M \cdot s)$,
- x is a λ-node λM,
- x is an application node MN and the path f is a σ-path.

PROOF Suppose that an occurrence x annihilated by v enshrines u in a pair $M \xrightarrow{u} N \xrightarrow{v} P$. Clearly, the path $u; v$ cannot be standard, unless u creates v. Inspecting the eleven rewriting rules of the $\lambda\sigma$-calculus in figure 1, we see that creation is forbidden in three cases at least:

1. When the occurrence x is a cons-node $M \cdot s$ because $M \cdot s$ is "impervious" to any interaction with M or s,
2. When the occurrence x is a λ-node λM because λM is "impervious" to any interaction with M,
3. When the occurrence x is an application node MN and v is a σ-redex: the only possible pattern of creation appears in (9) where u is an *Abs*-redex and v is a *Beta*-redex.

We apply lemma 24 on the standard rewriting path $f = r; g$ and the occurrence x enshrining r. By cases 1—3, the second alternative cannot hold when x is a cons-node or a λ-node, or when x is an application node and f does not contain any *Beta*-redex. We conclude. ■

PROOF OF LEMMA 22. By standard path in $\lambda^{\simeq}$, we mean the equivalence class of a standard rewriting path in $\lambda^{\to}$. The main thing to observe is that no $\lambda\sigma$-redex contracted in the standard path $f : M \to U(N)$ ever appears enshrined in a cons-node $M \cdot s$. We proceed by contradiction. Suppose that there exists a redex r_i contracted in $f = r_1; \cdots; r_n$ enshrined in the occurrence x of a cons-node. We reach a contradiction between two facts: $U(N)$ does not contain any cons-node, and by lemma 25 the path $r_i; \cdots; r_n : P \to U(N)$ preserves x. We conclude.

It follows that no *Beta*-redex contracted in f ever occurs inside a substitution s. Of course, the remark does not extend to the σ-redexes contracted in f, as the following standard path $f : M \to U(N) = P$ illustrates:

$$\mathbf{1}[(\mathbf{1}\cdot id)\circ(P\cdot id)] \overset{Map}{\to} \mathbf{1}[\mathbf{1}[P\cdot id]\cdot id\circ(P\cdot id)] \overset{VarCons}{\to} \mathbf{1}[P\cdot id] \overset{VarCons}{\to} P$$

Anyhow, the property implies that every *Beta*-redex contracted in $f : M \to U(N)$ is translated as a unique β-redex in $\sigma(f)$. This means that whenever $f : M \to U(N)$ is standard, we may consider $\sigma(f)$ as a proper rewriting path in $\lambda^{\to}$, not just an equivalence class of $\simeq$ in $\lambda^{\simeq}$. We show now that any standardisation cell $\alpha : \sigma(f) \Rightarrow h$ in $\lambda^{\to}$ may be mirrored as a standardisation cell $\beta : f \Rightarrow g$ in $\lambda\sigma^{\to}$, in the sense that $\sigma(g) = h$.

Let us write $\sigma(f) = R_1; \cdots; R_p$ and $\phi : \{1,...,p\} \to \{1,...,n\}$ the function which associates to any β-redex R_k in $\sigma(f)$ the corresponding *Beta*-redex $r_{\phi(k)}$ in $f = r_1; \cdots; r_n$. Let R_k and R_{k+1} be two consecutive β-redexes in $\sigma(f)$. The $\lambda\sigma$-path $r_i; \cdots; r_j = r_{\phi(k)+1}; \cdots; r_{\phi(k+1)-1}$ between $r_{\phi(k)}$ and $r_{\phi(k+1)}$ contracts only σ-redexes.

Suppose that the two β-redexes R_k and R_{k+1} are permuted in $\sigma(f)$ using a reversible permutation $R_k; R_{k+1} \simeq R'_k; R'_{k+1}$. We construct $g \simeq f$ such that $\sigma(g) = R_1; \cdots; R'_k; R'_{k+1}; \cdots; R_p$. By lemma 25, the path $r_i; \cdots; r_j$ preserves any application node enshrining $r_{\phi(k)}$, and in particular, the lowest application node appearing above $R_k : P \to Q$ and R_{k+1} in the λ-term $\sigma(P)$. It is straightforward then that $r_{\phi(k)}; r_i; \cdots; r_j; r_{\phi(k+1)}$ may be reorganised modulo $\simeq$ as a rewriting path g with projection $\sigma(g) = R'_k; R'_{k+1}$.

We claim also that the two β-redexes R_k and R_{k+1} cannot be permuted using an irreversible permutation $R_k; R_{k+1} \rhd_\lambda R'_k; F$. We proceed by contradiction. By lemma 25, the path $r_i; \cdots; r_n$ preserves every λ-node enshrining $r_{\phi(k)}$. Among these λ-nodes, the λ-node involved in R_{k+1} after σ-normalisation is annihilated by $r_{\phi(k+1)}$. We reach a contradiction and conclude. ∎

Lemma 26 Every external rewriting path $e : M \to N$ in $\lambda\sigma^{\to}$ is interpreted as an external path $\sigma(e) : \sigma(M) \to \sigma(N)$ in $\lambda^{\to}$.

Proof Because $\sigma : \lambda\sigma^{\to} \to \lambda^{\simeq}$ respects the permutation equivalence $\equiv$, we may reformulate lemma 22 as:

$$\sigma(\Downarrow_f) = \Downarrow_{\sigma(f)}$$

for any $\lambda\sigma$-path $f : M \to U(N)$.

Now, let $e : M \to N$ be an external $\lambda\sigma$-path in the $\lambda\sigma$-calculus, and f any λ-path $f : \sigma(N) \to P$. We call $h : N \to U(\sigma(N))$ any σ-normalisation path from N.

$$\begin{array}{rcll}
\Downarrow_{\sigma(e);f} & = & \Downarrow_{\sigma(e);\mathrm{id}_{\sigma(N)};\sigma(U(f))} & \text{by } \sigma(U(f)) = f, \\
& = & \Downarrow_{\sigma(e;h;U(f))} & \text{by } \sigma(h) = \mathrm{id}_{\sigma(N)} \text{ and functoriality of } \sigma, \\
& = & \sigma(\Downarrow_{e;h;U(f)}) & \text{by lemma 22,} \\
& = & \sigma(e; \Downarrow_{h;U(f)}) & \text{by externality of } e \text{ and diagram (7),} \\
& = & \sigma(e); \sigma(\Downarrow_{h;U(f)}) & \text{by functoriality of } \sigma, \\
& = & \sigma(e); \Downarrow_{\sigma(h;U(f))} & \text{by lemma 22,} \\
& = & \sigma(e); \Downarrow_f & \text{by } \sigma(h;U(f)) = f.
\end{array}$$

We conclude. ∎

6 Main result

We prove that

Theorem 2 Every closed $\lambda\sigma$-term enjoys a finite normalisation cone.

PROOF The proof is by contradiction. Suppose that there exists a closed $\lambda\sigma$-term M with an infinite number of normalising $\lambda\sigma$-path $M \to N$, modulo Lévy permutation equivalence $\equiv$.

First step: König's lemma. We construct the infinite tree T whose nodes are the standard paths $M \to N$ which may be extended to a standard normalising path $M \to P \to N$. The nodes are ordered by prefix ordering on paths. Because every $\lambda\sigma$-term contains at most a finite number of $\lambda\sigma$-redexes, we may apply König's lemma and deduce that there exists an infinite path f_∞ in the tree.

Second step: strong σ-normalisation. By strong normalisation of the substitution σ-calculus established in [6, 25], the infinite path f_∞ contracts an infinite number of $Beta$-redexes.

Third step: lemma 26. By definition of T, every finite prefix $f : M \to P$ of f_∞ may be extended to a standard normalising path $M \xrightarrow{f} P \xrightarrow{g} N$. By lemma 9, the path $f;g$ is external. By lemma 26, its projection $\sigma(f;g)$ is external. Every $Beta$-redex $r : P \xrightarrow{Beta} Q$ contracted in f projects as a β-redex $\sigma(r) : \sigma(P) \to \sigma(Q)$ contracted in the external λ-path $\sigma(f;g)$. The β-redex $\sigma(r)$ is external by the two decomposition lemmas 11 and 12 applied on $\sigma(f;g)$.

Last step: needed normalisation in the λ-calculus. We conclude from step 2. and 3. that projecting f_∞ constructs an infinite external λ-path from $\sigma(M)$. We reach a contradiction between three facts: the λ-term $\sigma(M)$ has a normal form $\sigma(N)$, external paths are needed by lemma 17, and needed strategies are normalising in the λ-calculus. We conclude. ■

We apply theorem 1 and deduce that:

Corollary 27 Every needed strategy is normalising in the closed $\lambda\sigma$-calculus.

We do not prove that the calculus of *open* $\lambda\sigma$-terms enjoys finite normalisation cones. This, we do not know. It should be possible however to extend the proof to the $\lambda\sigma$-calculus of open terms with no substitution variable.

To conclude, let us stress that $\lambda\sigma$-neededness does not depend on neededness in the λ-calculus: it is an internal notion provided by definition 15. We may illustrate it as follows. The *vertebra* of a $\lambda\sigma$-term M is defined inductively as a $\lambda\sigma$-term $\mathsf{vertebra}(M)$ with extra constant $-$:

1. $\mathsf{vertebra}(\mathbf{1}) = \mathbf{1}$,
2. $\mathsf{vertebra}(\lambda P) = \lambda(\mathsf{vertebra}(P))$,
3. $\mathsf{vertebra}(PQ) = (\mathsf{vertebra}(P))-$,
4. $\mathsf{vertebra}(P[s]) = (\mathsf{vertebra}(P))[-]$.

The spine $\mathsf{spine}(M)$ of M is defined as $\mathsf{vertebra}(M)$ when M is not of the form $M = \lambda ... \lambda(\mathbf{i} M_1 ... M_n)$ for a de Bruijn number $\mathbf{i} = \mathbf{1}[\uparrow \circ \cdots \circ \uparrow]$, and as

$$\mathsf{spine}(M) \;=\; \lambda ... \lambda(\mathbf{i}\ \mathsf{spine}(M_1) ... \mathsf{spine}(M_n))$$

when $M = \lambda ... \lambda(\mathbf{i} M_1 ... M_n)$. Call *spine* every $\lambda\sigma$-redex $u : M \to N$ whose occurrence appears not replaced by the constant $-$ in the spine of M. An easy argument shows that every spine redex is properly needed. We apply corollary 27 and deduce that every $\lambda\sigma$-strategy contracting a spine redex at each step (or eventually) is normalising.

7 Appendix

7.1 Head-normalisation

Theorem 2 may be extended to consider *values* instead of just normal forms. We start with the axiomatic definition of a value-set appearing in [23], adapted from Glauert and Khasidashvili's definition in [14]. Given an axiomatic rewriting system $(\mathcal{G}, \triangleright)$, we say that a redex u *nests* another redex v, and write $u \prec v$, when there exists an *irreversible* permutation

$$(M \xrightarrow{v} Q \xrightarrow{h_2} N) \;\triangleright\; (M \xrightarrow{u} P \xrightarrow{h_1} N)$$

We say that two redexes u and v are *disjoint*, and write $u \| v$, when there exists a *reversible* permutation

$$(M \xrightarrow{v} Q \xrightarrow{h_2} N) \;\triangleright\; (M \xrightarrow{u} P \xrightarrow{h_1} N)$$

Two redexes $u : M \to P$ and $v : M \to Q$ are declared *compatible* when $u \prec v$, $u \| v$ or $v \prec u$. The axiomatics of [21] insures that the three cases $u \| v$, $u \prec v$ and $v \prec u$ are exclusive.

A set $\mathcal{V}$ of vertices is declared *open-stable* in $(\mathcal{G}, \triangleright)$ when the three properties below are verified by every permutation diagram $v; u' \triangleright u; f$:

$$\begin{tikzcd} M \arrow[r, "v"] \arrow[d, "u"'] \arrow[dr, phantom, "\Leftarrow"] & Q \arrow[d, "u'"] \\ P \arrow[r, "f"'] & N \end{tikzcd}$$

[head]	if $u \prec v$ and $Q \in \mathcal{V}$, then $M \in \mathcal{V}$,
[stable]	if $u \| v$ and $P, Q \in \mathcal{V}$, then $M \in \mathcal{V}$,
[closed]	if $M \to N$ and $M \in \mathcal{V}$, then $N \in \mathcal{V}$

We prove in [23] that from any term M in $\mathcal{G}$, there exists a cone $(e_i : M \to V_i)_{i \in I}$ of *minimal* rewriting paths to the set $\mathcal{V}$. By *minimal*, we mean that each $V_i \in \mathcal{V}$, and that for every path $f : M \to V$ with $V \in \mathcal{V}$, there exists one and only one $i \in I$ such that f factors as $f \equiv e_i; h$ for some path h.

$$\begin{tikzcd} M \arrow[r, "e_i"] \arrow[dr, "f"'] & V_i \arrow[d, dotted, "h"] \\ & V \end{tikzcd}$$

Moreover, the rewriting path h is unique modulo $\equiv$. Here, we establish that the "head-normalisation" cone $(e_i : M \to V_i)_{i\in I}$ is finite for any *reasonable* notion of value-set $\mathcal{V}$ in the $\lambda\sigma$-calculus. By reasonable, we mean the three following properties of $\mathcal{V}$ in $(\mathcal{G}_{\lambda\sigma}, \rhd_{\lambda\sigma})$:

1. $\mathcal{V}$ is open-stable in $(\mathcal{G}_{\lambda\sigma}, \rhd_{\lambda\sigma})$,
2. its projection $\sigma(\mathcal{V})$ is open-stable in $(\mathcal{G}_{\lambda}, \rhd_{\lambda})$,
3. for every *Beta*-redex $u : M \to N$ between two $\lambda\sigma$-terms M and N with projections $\sigma(M)$ and $\sigma(N)$ in $\sigma(\mathcal{V})$, and every normalisation σ-path $f : N \to U(\sigma(N))$ inducing a standard path $M \xrightarrow{u} N \xrightarrow{f} U(\sigma(N))$, there exists a path $M \xrightarrow{f_1} V \xrightarrow{f_2} U(\sigma(N))$ verifying:
$$M \xrightarrow{u} N \xrightarrow{f} U(\sigma(N)) \;\simeq\; M \xrightarrow{f_1} V \xrightarrow{f_2} U(\sigma(N))$$
such that f_1 is a σ-path and V is an element of $\mathcal{V}$.

Another way to state property 3 is to ask that every path e_i in the cone $(e_i : M \to V_i)_{i\in I}$ from M to $\mathcal{V}$ is a σ-path when M is a $\lambda\sigma$-term with interpretation $\sigma(M)$ in $\sigma(\mathcal{V})$. As an illustration, we call *$\lambda\sigma$-head-normal form* any closed $\lambda\sigma$-term written as $\lambda ... \lambda(\mathbf{i} M_1 ... M_n)$ where $\mathbf{i} = \mathbf{1}[\uparrow \circ \cdots \circ \uparrow]$ is a de Bruijn number, and $M_1, ..., M_n$ are $\lambda\sigma$-terms. An easy syntactical reasoning shows that the set $\mathcal{V}_{\lambda\sigma}$ of $\lambda\sigma$-head-normal forms verifies the three properties 1—3.

Theorem 3 Let $\mathcal{V}$ be a set of closed $\lambda\sigma$-terms verifying properties 1—3, and M a closed $\lambda\sigma$-term. The normalisation cone $(e_i : M \to V_i)_{i\in I}$ from M to $\mathcal{V}$ is finite, and every path $e_i : M \to V_i$ in the cone projects as the unique path $e : \sigma(M) \to V$ in the normalisation cone from $\sigma(M)$ to $\sigma(\mathcal{V})$.

Proof We show that the cone $(e_i : M \to V_i)_{i\in I}$ is finite. By σ-strong normalisation, we only need to prove that the number of *Beta*-redex in e_i is bounded. This will follow from the fact that every path e_i projects as the canonical path $e : \sigma(M) \to V$ defined by the open-stable set $\sigma(\mathcal{V})$ in the λ-calculus. Let $e_i : M \to V_i$ be any path in the cone, and $f : V_i \to U(\sigma(V_i))$ a σ-path normalising V_i. Recall from our stability theorem in [23] that e_i is external. This implies here that $g = e_i; \Downarrow_f$ is standard. By lemma 22, its projection $\underline{g} = \sigma(g)$ is standard. We claim that $\underline{g}$ is precisely the canonical path $e : M \to V$, modulo $\simeq$. We proceed by contradiction, and suppose that $\neg(\underline{g} \simeq e)$. The proof of lemma 22 informs us that to every λ-path $\underline{h}$ equal to $\underline{g}$ modulo $\simeq$, there corresponds a $\lambda\sigma$-path $h \simeq g$ with projection $\sigma(h) = \underline{h}$. We may reorganise $g = M \xrightarrow{e_i} V_i \xrightarrow{\Downarrow_f} U(\sigma(V_i))$ as $h = M \xrightarrow{h_1} P \xrightarrow{r} Q \xrightarrow{h_2} U(\sigma(V_i))$ modulo $\simeq$, in such a way that r is a *Beta*-redex, that h_2 a σ-path, and that $\sigma(P) \in \sigma(\mathcal{V})$. By property 3, the standard path $r; h_2$ may be reorganised modulo $\simeq$ as a σ-path $P \xrightarrow{h_3} W \in \mathcal{V}$ followed by a $\lambda\sigma$-path $W \xrightarrow{h_4} U(\sigma(V_i))$. We obtain the series of equivalence:

$$\begin{array}{rcl} M \xrightarrow{e_i} V_i \xrightarrow{\Downarrow_f} U(\sigma(V_i)) & \simeq & M \xrightarrow{h_1} P \xrightarrow{r} Q \xrightarrow{h_2} U(\sigma(V_i)) \\ & \simeq & M \xrightarrow{h_1} P \xrightarrow{h_3} W \xrightarrow{h_4} U(\sigma(V_i)) \end{array}$$

It follows from $W \in \mathcal{V}$ and the definition of $(e_i : M \to V_i)_{i \in I}$ that $e_i \sqsubseteq h_1; h_3$. By externality of e_i and standardness of $h_1; h_3$, there exists a path $h_4 : V_i \to W$ verifying

$$e_i; h_4 \simeq h_1; h_3$$

This contradicts the fact that e_i contracts strictly more *Beta*-redexes than $h_1; h_3$. We conclude. ■

7.2 A dynamical order

We relax the assumption that $f : M \to U(N)$ in lemma 22. Observe that the tree nesting ordering on $\lambda\sigma$-terms is not completely satisfactory: a substitution s and a term M are disjoint in the $\lambda\sigma$-term $M[s]$ despite the fact that the substitution s appears nested in M after projection σ. So, we introduce another more "dynamical" ordering $\preceq_\bigstar$ defined as the least order on $\lambda\sigma$-redexes extending the usual tree ordering, closed under context, and verifying:

1. $u \preceq_\bigstar v$ whenever the $\lambda\sigma$-redex u occurs in M and the $\lambda\sigma$-redex v occurs in s in a term $M[s]$,
2. $u \preceq_\bigstar v$ whenever the $\lambda\sigma$-redex u occurs in s and the $\lambda\sigma$-redex v occurs in t in a substitution $s \circ t$.

We define the system $(\mathcal{G}_\bigstar, \rhd_\bigstar)$ as $\mathcal{G}_\bigstar = \mathcal{G}_{\lambda\sigma}$ and $v; u' \rhd_\bigstar u; f$ precisely when $v; u' \rhd_{\lambda\sigma} u; f$ and $\neg(v \preceq_\bigstar u)$. By lack of space, we do not prove here that $(\mathcal{G}_\bigstar, \rhd_\bigstar)$ defines an axiomatic rewriting system on $\lambda\sigma$-redexes. The alteration of $\preceq$ into $\preceq_\bigstar$ allows to improve diagram (7) in section 4.

Lemma 28 We have the following commutative diagrams:

$$\begin{tikzcd}
\lambda^{\simeq} \arrow[dr] & \\
\lambda^{\to} \arrow[u, "\simeq"] \arrow[r, "U\bigstar"'] & \lambda\sigma^{\simeq}_{\bigstar}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\lambda\sigma^{\simeq}_{\bigstar} \arrow[dr] & \\
\lambda\sigma^{\to}_{\bigstar} \arrow[u, "\simeq\bigstar"] \arrow[r, "\sigma\bigstar"'] & \lambda^{\simeq}
\end{tikzcd}$$

where $\lambda\sigma^{\simeq}_{\bigstar} = \lambda\sigma^{\to}_{\bigstar} / \simeq_\bigstar$ denotes the free category $\lambda\sigma^{\to}_{\bigstar}$ quotiented by the relation $\simeq_\bigstar$ derived from $\rhd_\bigstar$.

Proof The construction of the two functors $U_\bigstar$ and $\sigma_\bigstar$ and the completion of the first diagram are straightforward. Here, we complete the second diagram and prove that $\sigma_\bigstar$ respects the relation $\simeq$. This reduces to the claim that two *Beta*-redexes disjoint wrt. $\preceq_\bigstar$ are interpreted as a set of disjoint β-redexes after σ-normalisation. Since *Beta*-redexes usually disappear and reappear in the process of σ-normalisation, we switch to *Beta*-patterns, defined as any pattern of the form $((\lambda M)[s_1] \cdots [s_m])N$ for $m \geq 0$ in a $\lambda\sigma$-term P. Any *Beta*-redex is a *Beta*-pattern, and two *Beta*-patterns u and v are declared disjoint when their application nodes are disjoint wrt. the tree nesting ordering. By residual of a *Beta*-pattern in P through a $\lambda\sigma$-redex $r : P \to Q$, we mean the obvious syntactical notion, keeping in mind that a β-redex obtained by σ-normalisation from a *Beta*-pattern should be its residual. Finally, given any two disjoint $Beta$-patterns u and v in P, we write $u \wedge v$ the longest occurrence above u and v in P wrt. the tree nesting ordering. This node $u \wedge v$ is either a node MN, $M[s]$, $M \cdot s$ or $s \circ t$.

Let us observe that:

1. two different residuals u' and u'' of a $Beta$-pattern u after a $\lambda\sigma$-redex $r : P \to Q$ are necessarily disjoint, with $u' \wedge u''$ an application or a cons node. The σ-redex r is an *App*-redex in the first case, a Map-redex in the second case, and u always lies in the duplicated substitution,

2. let u and v be two disjoint $Beta$-patterns such that $u \wedge v$ is an application or cons node. Let u' and v' be residuals of u and v respectively after a σ-redex $r : P \to Q$. An easy inspection of the ten possible σ-rules for r shows that u' and v' are disjoint and that $u' \wedge v'$ is an application or cons node.

We conclude that two disjoint $Beta$-redexes u and v in a $\lambda\sigma$-term P project by σ-normalisation as a set of disjoint β-redexes in $\sigma(P)$. ■

Lemma 28 does not provide any remarkable adjunction property like (6), even extended to a polyadjunction property à la Taylor [24], see also [23]. However, it verifies that:

Lemma 29 Every standard rewriting path $f : M \to N$ in $(\mathcal{G}_\bigstar, \triangleright_\bigstar)$ projects through $\sigma_\bigstar$ as a standard λ-path (more exactly: $\simeq$-class) in $\lambda^{\simeq}$.

Proof The whole argument is based on an elementary property of standardisation in the λ-calculus. Given any path $d : M \to N$, and a vector $f_i : P_i \to Q_i$ of paths, we may define the path

$$d[f_i/x_i] : M[P_i/x_i] \to N[Q_i/x_i]$$

obtained by applying first the rewriting path $d : M \to N$ on $M[P_i/x_i]$, then each rewriting path $f_i : P_i \to Q_i$ on $N[P_i/x_i]$. The construction is valid modulo $\simeq$, and $d[f_i/x_i]$ is standard whenever d and each of the f_i's is standard.

We define similarly $\mathbf{i}[f] : \mathbf{i}[s] \to \mathbf{i}[t]$ for every $\lambda\sigma$-path $f : s \to t$ between substitutions s and t, by applying $f : s \to t$ in front of the de Bruijn number $\mathbf{i} = \mathbf{1}[\uparrow \circ \cdots \circ \uparrow]$.

We prove that:

- the projection $\sigma_\bigstar(f)$ is standard for every standard path $f : M \to N$,

- the projection $\sigma_\bigstar(\mathbf{i}[f])$ is standard for every standard path $f : s \to t$ and every number $i \geq 1$,

by induction on the length of f, then on the size of M. When the path $f = u; g$ contracts a σ-redex u at first step, the two paths f and g define by $\sigma_\bigstar$ the same $\simeq$-class in $\lambda^{\simeq}$, which is standard by induction hypothesis applied to g. So, we suppose from now on that the first redex u contracted by f is a $Beta$-redex. We proceed by induction on the size of M and s.

1. Suppose that $M = PQ$ or $M = P[s]$, and that f does not preserve the root node. Because f does not preserve any occurrence in M, by lemma 25, the $Beta$-redex u is not enshrined by any cons-node $L \cdot t$ or λ-node λL in M. Because application nodes KL are "impervious" to interactions with L, we could very well extend

lemma 25 and establish that u cannot appear in any subterm L inside a subterm KL of M. All in all, we obtain that the *Beta*-redex u is translated in $\lambda^{\simeq}$ as an external β-redex, and conclude by induction hypothesis.

2. Suppose that $M = PQ$ and that $f : M \to N$ preserves the root node PQ. The path f decomposes as two paths $f_P : P \to P'$ and $f_Q : Q \to Q'$ which are standard in $(\mathcal{G}_\bigstar, \rhd_\bigstar)$. By induction hypothesis, their projection is standard, and we conclude that the projection of f is standard.

3. Suppose that $M = P[s]$ and that $f : M \to N$ preserves the root node $P[s]$. The way we choose to define $\preceq_\bigstar$ implies that f is the sequential composite $f : f_P; f_s$ of a path $f_P : P \to Q$ and a path $f_s : s \to t$. By induction hypothesis, $\sigma_\bigstar(f_P)$ and each $\sigma_\bigstar(\mathbf{i}[f_s])$ is standard in $\lambda^{\simeq}$. We deduce standardness of $\sigma_\bigstar(f)$ from the equality in $\lambda^{\simeq}$:
$$\sigma_\bigstar(f) \ = \ \sigma_\bigstar(f_P)[\sigma_\bigstar(\mathbf{i}[f_s])]$$

4. The case $s = M \cdot s'$ is trivial. We treat the other case $s = s_1 \circ s_2$. The first redex contracted by $f : s \to t$ is a *Beta*-redex and therefore enshrined in a cons-node. The path f which preserves the cons-node preserves a fortiori the root node $s_1 \circ s_2$. Moreover, the choice of $\preceq_\bigstar$ implies that f is the sequential composite $f = f_1; f_2$ of a path $f_1 : s_1 \to t_1$ and a path $f_2 : s_2 \to t_2$. We reason as in case 2. and deduce that each $\mathbf{i}[f] : \mathbf{i}[s] \to \mathbf{i}[t]$ projects as a standard morphism in $\lambda^{\simeq}$.

■

Acknowledgements

The author would like to thank Pierre-Louis Curien and Thérèse Hardin for their enthusiasm and support.

References

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

[2] H. Barendregt, "*The Lambda Calculus: Its Syntax and Semantics*". North Holland, 1985.

[3] G. Berry, "*Modèles complètement adéquats et stables des lambda-calculs typés*". Thèse de Doctorat d'Etat, Université Paris VII, 1979.

[4] G. Boudol, "Computational semantics of term rewriting systems". *Algebraic methods in Semantics*, Maurice Nivat and John C. Reynolds (eds), Cambridge University Press, 1985.

[5] H.-B. Curry, R. Feys, "*Combinatory Logic*". North Holland Volume 1, 1958.

[6] P.-L. Curien, T. Hardin, A. Ríos. "Strong normalization of substitutions". Lecture Notes in Computer Science, 629:209–217, 1992.

[7] D. Clark and R. Kennaway, "Event structures and non-orthogonal term graph rewriting". In *Mathematical Structures in Computer Science*, vol. 6, pp. 545-578, 1996.

[8] M. J. O'Donnell. "*Computing in systems described by equations*". Lecture Notes in Computer Science 58, Springer Verlag, 1977.

[9] G. Gonthier, J.-J. Lévy, P.-A. Melliès, "An abstract standardisation theorem". In *Proceedings Logic In Computer Science 92*, Santa Cruz, 1992.

[10] T. Hardin, "Confluence Results for the Pure Strong Categorical Logic CCL. λ-calculi as subsystems of CCL". *Theoretical Computer Science* volume 65, pages 291–342, 1989.

[11] B. P. Hilken, "Towards a proof theory of rewriting: the simply typed 2λ-calculus". *Theoretical Computer Science* 170, pages 407-444, 1996.

[12] G. Huet, "Confluent reductions: abstract properties and applications to term rewriting systems". *J. Computer and Systems Sciences* 23 (1), pp 11-21, 1981.

[13] G. Huet, J.-J. Lévy, "Call by Need Computations in Non-Ambiguous Linear Term Rewriting Systems". Rapport de recherche INRIA 359, 1979. Republished as "Computations in orthogonal rewriting systems, I and II", in Jean-Louis Lassez and Gordon Plotkin, editors, *Computational logic, essays in honor of Alan Robinson*, pages 395—443. MIT Press, Cambridge, Massachussets, 1991.

[14] J. Glauert, Z. Khasidashvili, "Relative normalisation in deterministic residual structures". In *Proceedings Colloquium on trees in Algebra And Programming 96*, Linköping, Lecture Notes in Computer Science 1059, Springer Verlag, 1996.

[15] J. W. Klop, "Term Rewriting Systems". *Handbook of Logic in Computer Science*, Volume 2, in S. Abramsky, Dov M. Gabbay, T.S.E. Maibaum, editors, Oxford Science Publications, 1992.

[16] J.-J. Lévy, "Réductions correctes et optimales dans le λ-calcul". Thèse de Doctorat d'Etat, Université Paris VII, 1978.

[17] S. Mac Lane, "*Categories for the working mathematician*", Springer Verlag, 1971.

[18] L. Maranget, "*La stratégie paresseuse*". Thèse de l'Université Paris VII, 1992.

[19] P.-A. Melliès, "Typed lambda calculi with explicit substitutions may not terminate". In *Proceedings Typed Lambda Calculus and Applications 95*, Edinburgh, Lecture Notes in Computer Science 902, Springer Verlag, 1995.

[20] P.-A. Melliès, "*Description axiomatique des systèmes de réécriture*". Thèse de l'Université Paris VII, december 1996.

[21] P.-A. Melliès, "Axiomatic Rewriting Theory I: An axiomatic standardisation theorem", submitted, 2000.

[22] P.-A. Melliès, “Axiomatic Rewriting Theory III: A factorisation theorem in Rewriting Theory”. In *Proceedings Category Theory and Computer Science 97*, Santa Margherita Ligure, Lecture Notes in Computer Science 1290, Springer Verlag, pp. 49-68, 1997.

[23] P.-A. Melliès, “Axiomatic Rewriting Theory IV: A stability theorem in Rewriting Theory”. In *Proceedings Logic In Computer Science 98*, Indianapolis, 1998.

[24] P. Taylor, “The trace factorisation of stable functors”, paper available at `http://www.dcs.qmw.ac.uk/~pt/`, 1988.

[25] H. Zantema, “Termination of term rewriting by interpretation”. LNCS 656, Springer Verlag, 1993.