The prismoid of resources

Delia Kesner and Fabien Renaud

PPS, CNRS and Université Paris Diderot

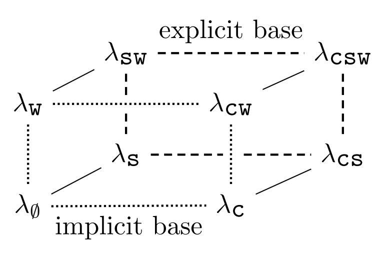
Abstract. We define a framework called the *prismoid of resources* where each vertex refines the λ-calculus by using a different choice to make explicit or implicit (meta-level) the definition of the contraction, weakening, and substitution operations. For all the calculi in the prismoid we show simulation of β-reduction, confluence, preservation of β-strong normalisation and strong normalisation for typed terms. Full composition also holds for all the calculi of the prismoid handling explicit substitutions. The whole development of the prismoid is done by making the set of resources a parameter, so that the properties for each vertex are obtained as a particular case of the general abstract proofs.

1 Introduction

Linear Logic [5] gives a logical framework to formalise the notion of control of resources by means of weakening, contraction and linear substitution. A succinct representation of Linear Logic proofs is given by Proof-Nets [5] which are often used as a semantical support to define λ-calculi with explicit control operators [19, 18, 9, 7].

In this paper we develop an homogeneous framework of λ-calculi called the *prismoid of resources*. Each vertex is a specialised λ-calculus parametrised by a set of *sorts* wich are of two kinds : resources $\mathtt{w}$ (weakening) and $\mathtt{c}$ (contraction), and cut-elimination operation $\mathtt{s}$ (substitution). If a sort in $\{\mathtt{c}, \mathtt{s}, \mathtt{w}\}$ belongs to a given calculus, then management of the corresponding operations is explicit in this calculus. Explicit resources will allow more refined cut-elimination procedures. Each edge is an operation to simulate and/or project one vertex into the other one. The eight calculi of the prismoid correspond to 2^3 different ways to combine sorts by means of explicit or implicit (meta-level) operations.

The asymmetry between different sorts will be reflected in the prismoid by means of its two bases. The base $\mathfrak{B}_I$ contains all the calculi without explicit substitutions and the base $\mathfrak{B}_E$ only contains those with explicit substitutions. The bases are of different nature as they will not enjoy exactly the same properties.



Thus for example, the $\lambda_{\texttt{cs}}$-calculus has only explicit control of contraction and substitution, the λ-calculus has no explicit control at all, and the $\lambda_{\texttt{csw}}$-calculus – a slight variation of $\lambda\texttt{lxr}$ [9] – has explicit control of everything.

For all calculi of the prismoid we show simulation of β-reduction, confluence, preservation of β-strong normalisation (PSN) and strong normalisation (SN) for simply typed terms. Thus in particular, none of the calculi suffers from Mellies' counter-example [14]. Full composition, stating that explicit substitution is able to implement the underlying notion of higher-order substitution, is also shown for all calculi with sort $\texttt{s}$, ie. those included in the explicit substitution base. Each property is stated and proved by making the set of sorts a parameter, so that the properties for each vertex of the prismoid turn out to be a particular case of some general abstract proof, which may hold for the whole prismoid or just for only one base.

While both implicit and explicit substitutions are usually [1, 6, 13] defined by means of the propagation of an operator through the structure of terms, the behaviour of calculi of the prismoid can be understood as a mechanism to decrease the multiplicity of variables that are affected by substitutions. This notion is close in spirit to MELL Proof-Nets, and shares common ideas with calculi by Milner [15] and Accattoli and Guerrini [2]. However their formalisms only handle the substitution operation as explicit, leaving weakening and contraction as implicit functions.

Road Map: Section 2 introduces syntax and operational semantics of the prismoid. Section 3 explores how to enrich the λ-calculus by adding more explicit control of resources, while Section 4 deals with the dual operation which forgets rich information given by explicit weakening and contraction. Section 5 is devoted to PSN and confluence on untyped terms. Finally, typed terms are introduced in Section 6 together with a SN proof for them. We conclude and give future directions of work in Section 7.

2 Terms and Rules of the Prismoid

We assume a denumerable set of variable symbols $x, y, z, \ldots$. Lists and sets of variables are denoted by capital Greek letters $\Gamma, \Delta, \Pi, \ldots$. We write $\Gamma; y$ for $\Gamma \cup \{y\}$ when $y \notin \Gamma$. We use $\Gamma \setminus \Delta$ for **set difference** and $\Gamma \setminus\!\!\setminus \Delta$ for **obligation set difference** which is only defined if $\Delta \subseteq \Gamma$.

Terms are given by the grammar $t, u ::= x \mid \lambda x.t \mid tu \mid t[x/u] \mid \mathcal{W}_x(t) \mid \mathcal{C}_x^{y|z}(t)$. The terms x, $\lambda x.t$, tu, $t[x/u]$, $\mathcal{W}_x(t)$ and $\mathcal{C}_x^{y|z}(t)$ are respectively called **term variable**, **abstraction**, **application**, **closure**, **weakening** and **contraction**. **Free** and **bound** variables of t, respectively written $\texttt{fv}(t)$ and $\texttt{bv}(t)$, are defined as usual: $\lambda x.u$ and $u[x/v]$ bind x in u, $\mathcal{C}_x^{y|z}(u)$ binds y and z in u, x is free in $\mathcal{C}_x^{y|z}(u)$ and in $\mathcal{W}_x(t)$.

Given three lists of variables $\Gamma = x_1, \ldots, x_n$, $\Delta = y_1, \ldots, y_n$ and $\Pi = z_1, \ldots, z_n$ of the same length, the notations $\mathcal{W}_\Gamma(t)$ and $\mathcal{C}_\Gamma^{\Delta|\Pi}(t)$ mean, respectively, $\mathcal{W}_{x_1}(\ldots \mathcal{W}_{x_n}(t))$ and $\mathcal{C}_{x_1}^{y_1|z_1}(\ldots \mathcal{C}_{x_n}^{y_n|z_n}(t))$. These notations will extend naturally to sets of variables of same size thanks to the equivalence relation in Figure 1. The particular cases $\mathcal{C}_{\emptyset}^{\emptyset|\emptyset}(t)$ and $\mathcal{W}_{\emptyset}(t)$ mean simply t.

Given lists $\Gamma = x_1, \ldots, x_n$ and $\Delta = y_1, \ldots, y_n$, the **renaming** of Γ by Δ in t, written $R^{\Gamma}_{\Delta}(t)$, is the capture-avoiding simultaneous substitution of y_i for every free occurrence of x_i in t. For example $R^{x_1x_2}_{y_1y_2}(\mathcal{C}^{y|z}_{x_1}(x_2yz)) = \mathcal{C}^{y|z}_{y_1}(y_2yz)$.

Alpha-conversion is the (standard) congruence generated by *renaming* of bound variables. For example, $\lambda x_1.x_1\mathcal{C}^{y_1|z_1}_{x}(y_1z_1) \equiv_\alpha \lambda x_2.x_2\mathcal{C}^{y_2|z_2}_{x}(y_2z_2)$. All the operations defined along the paper are considered modulo alpha-conversion so that in particular capture of variables is not possible.

The set of **positive free variables** of a term t, written $\mathtt{fv}^+(t)$, denotes the free variables of t which represent a term variable at the end of some (possibly empty) contraction chain. Formally,

$$
\begin{array}{ll}
\mathtt{fv}^+(y) & := \{y\} \\
\mathtt{fv}^+(\lambda y.u) & := \mathtt{fv}^+(u) \setminus \{y\} \\
\mathtt{fv}^+(uv) & := \mathtt{fv}^+(u) \cup \mathtt{fv}^+(v) \\
\mathtt{fv}^+(\mathcal{W}_y(u)) & := \mathtt{fv}^+(u) \\
\mathtt{fv}^+(u[y/v]) & := (\mathtt{fv}^+(u) \setminus \{y\}) \cup \mathtt{fv}^+(v) \\
\mathtt{fv}^+(\mathcal{C}^{z|w}_y(u)) & := (\mathtt{fv}^+(u) \setminus \{z,w\}) \cup \{y\} \text{ if } z \in \mathtt{fv}^+(u) \text{ or } w \in \mathtt{fv}^+(u) \\
\mathtt{fv}^+(\mathcal{C}^{z|w}_y(u)) & := \mathtt{fv}^+(u) \setminus \{z,w\} \text{ otherwise}
\end{array}
$$

The **number of occurrences** of the positive free variable x in the term t is written $|\mathtt{fv}^+(t)|_x$. We extend this definition to sets by $|\mathtt{fv}^+(t)|_\Gamma = \Sigma_{x\in\Gamma}|\mathtt{fv}^+(t)|_x$. Thus for example, given $t = \mathcal{W}_{x_1}(xx)\ \mathcal{W}_x(y)\ \mathcal{C}^{z_1|z_2}_{z}(z_2)$, we have $x, y, z \in \mathtt{fv}^+(t)$ with $|\mathtt{fv}^+(t)|_x = 2$, $|\mathtt{fv}^+(t)|_y = |\mathtt{fv}^+(t)|_z = 1$ but $x_1 \notin \mathtt{fv}^+(t)$.

We write $t_{[y]_x}$ for the **non-deterministic replacement** of *one positive* occurrence of x in t by a *fresh* variable y. Thus for example, $(\mathcal{W}_x(t)\ x\ x)_{[y]_x}$ may denote either $\mathcal{W}_x(t)\ y\ x$ or $\mathcal{W}_x(t)\ x\ y$, but neither $\mathcal{W}_y(t)\ x\ x$ nor $\mathcal{W}_x(t)\ y\ y$.

The **deletion** function removes non positive free variables in Γ from t:

$$
\begin{array}{lll}
\mathtt{del}_\Gamma(y) & := y & \\
\mathtt{del}_\Gamma(u\ v) & := \mathtt{del}_\Gamma(u)\ \mathtt{del}_\Gamma(v) & \\
\mathtt{del}_\Gamma(\lambda y.u) & := \lambda y.\mathtt{del}_\Gamma(u) & \text{if } y \notin \Gamma \\
\mathtt{del}_\Gamma(u[y/v]) & := \mathtt{del}_\Gamma(u)[y/\mathtt{del}_\Gamma(v)] & \text{if } y \notin \Gamma \\
\mathtt{del}_\Gamma(\mathcal{W}_x(u)) & := \begin{cases} u \\ \mathcal{W}_x(\mathtt{del}_\Gamma(u)) \end{cases} & \begin{array}{l} \text{if } x \in \Gamma \\ \text{if } x \notin \Gamma \end{array} \\
\mathtt{del}_\Gamma(\mathcal{C}^{y|z}_x(u)) & := \begin{cases} \mathtt{del}_{\Gamma\setminus x\cup\{y,z\}}(u) \\ \mathcal{C}^{y|z}_x(\mathtt{del}_\Gamma(u)) \end{cases} & \begin{array}{l} \text{if } x \in \Gamma\ \&\ y,z \notin \Gamma\ \&\ x \notin \mathtt{fv}^+(\mathcal{C}^{y|z}_x(u)) \\ \text{otherwise} \end{array}
\end{array}
$$

This operation does not increase the size of the term. Moreover, if $x \in \mathtt{fv}(t) \setminus \mathtt{fv}^+(t)$, then $\mathtt{size}(\mathtt{del}_x(t)) < \mathtt{size}(t)$.

Now, let us consider a set of **resources** $\mathcal{R} = \{\mathtt{c}, \mathtt{w}\}$ and a set of **sorts** $\mathcal{S} = \mathcal{R} \cup \{\mathtt{s}\}$. For every subset $\mathcal{B} \subseteq \mathcal{S}$, we define a calculus $\lambda_\mathcal{B}$ in the **prismoid of resources** which is equipped with a set $\mathcal{T}_\mathcal{B}$ of **well-formed** terms, called $\mathcal{B}$-terms, together with a reduction relation $\to_\mathcal{B}$ given by a *subset* of the reduction system described in Figure 1. Each calculus belongs to a **base** : the explicit substitution base $\mathfrak{B}_E$ which contains all the calculi having at least sort $\mathtt{s}$ and the implicit substitution base $\mathfrak{B}_I$ containing all the other calculi. A term t is in $\mathcal{T}_\mathcal{B}$ iff $\exists\ \Gamma$ s.t. $\Gamma \Vdash_\mathcal{B} t$ is derivable in the following system :

$$\frac{}{x \Vdash_\mathcal{B} x} \quad \frac{\Gamma \Vdash_\mathcal{B} u \quad \Delta \Vdash_\mathcal{B} v}{\Gamma \uplus_\mathcal{B} \Delta \Vdash_\mathcal{B} uv} \quad \frac{\Gamma \Vdash_\mathcal{B} u}{\Gamma \setminus\!\!\setminus_\mathcal{B} x \Vdash_\mathcal{B} \lambda x.u} \quad \frac{\Gamma \Vdash_\mathcal{B} u}{\Gamma; x \Vdash_\mathcal{B} \mathcal{W}_x(u)}\ (\mathtt{w} \in \mathcal{B})$$

$$\frac{\Gamma \Vdash_\mathcal{B} v \quad \Delta \Vdash_\mathcal{B} u}{\Gamma \uplus_\mathcal{B} (\Delta \setminus\!\!\setminus_\mathcal{B} x) \Vdash_\mathcal{B} u[x/v]}\ (\mathtt{s} \in \mathcal{B}) \quad \frac{\Gamma \Vdash_\mathcal{B} u}{x; (\Gamma \setminus\!\!\setminus_\mathcal{B} \{y,z\}) \Vdash_\mathcal{B} \mathcal{C}_x^{y|z}(u)}\ (\mathtt{c} \in \mathcal{B})$$

In the previous rules, $\uplus_\mathcal{B}$ means standard union if $\mathtt{c} \notin \mathcal{B}$ and disjoint union if $\mathtt{c} \in \mathcal{B}$. Similarly, $\Gamma \setminus\!\!\setminus_\mathcal{B} \Delta$ is used for $\Gamma \setminus \Delta$ if $\mathtt{w} \notin \mathcal{B}$ and for $\Gamma \setminus\!\!\setminus \Delta$ if $\mathtt{w} \in \mathcal{B}$.

Notice that variables, applications and abstractions belong to all calculi of the prismoid while weakening, contraction and substitutions only appear in calculi having the corresponding sort. If t is a $\mathcal{B}$-term, then $\mathtt{w} \in \mathcal{B}$ implies that bound variables of t cannot be useless, and $\mathtt{c} \in \mathcal{B}$ implies that no free variable of t has more than one free occurrence. Thus for example the term $\lambda z.xy$ belongs to the calculus $\lambda_\mathcal{B}$ only if $\mathtt{w} \notin \mathcal{B}$ (thus it belongs to $\lambda_\emptyset$, $\lambda_\mathtt{c}$, $\lambda_\mathtt{s}$, $\lambda_\mathtt{cs}$), and $(xz)[z/yx]$ belongs to $\lambda_\mathcal{B}$ only if $\mathtt{s} \in \mathcal{B}$ and $\mathtt{c} \notin \mathcal{B}$ (thus it belongs to $\lambda_\mathtt{s}$ and $\lambda_\mathtt{sw}$). A useful property is that $\Gamma \Vdash_\mathcal{B} t$ implies $\Gamma = \mathtt{fv}(t)$.

In order to introduce the reduction rules of the prismoid we need a meta-level notion of substitution defined on alpha-equivalence classes; it is the one implemented by the explicit control of resources. A $\mathcal{B}$-**substitution** is a pair of the form $\{x/v\}$ with $v \in \mathcal{T}_\mathcal{B}$. The **application of a $\mathcal{B}$-substitution** $\{x/u\}$ to a $\mathcal{B}$-term t is defined as follows: if $|\mathtt{fv}^+(t)|_x = 0$ we have to check if x occurs negatively. If $|\mathtt{fv}(t)|_x = 0$ or $\mathtt{w} \notin \mathcal{B}$ then $t\{x/u\} := \mathtt{del}_x(t)$. Otherwise, $t\{x/u\} := \mathcal{W}_{\mathtt{fv}(u)\setminus\mathtt{fv}(t)}(\mathtt{del}_x(t))$. If $|\mathtt{fv}^+(t)|_x = n+1 \geq 2$, then $t\{x/u\} := t_{[y]_x}\{y/u\}\{x/u\}$. If $|\mathtt{fv}^+(t)|_x = 1$, $t\{x/u\} := \mathtt{del}_x(t)\{\!\{x/u\}\!\}$ where $\{\!\{x/u\}\!\}$ is defined as follows :

$$\begin{array}{lll}
x\{\!\{x/u\}\!\} & := u & \\
y\{\!\{x/u\}\!\} & := y & x \neq y \\
(s\ v)\{\!\{x/u\}\!\} & := s\{\!\{x/u\}\!\}\ v\{\!\{x/u\}\!\} & \\
(\lambda y.v)\{\!\{x/u\}\!\} & := \lambda y.v\{\!\{x/u\}\!\} & x \neq y\ \&\ y \notin \mathtt{fv}(u) \\
s[y/v]\{\!\{x/u\}\!\} & := s\{\!\{x/u\}\!\}[y/v\{\!\{x/u\}\!\}] & x \neq y\ \&\ y \notin \mathtt{fv}(u) \\
\mathcal{W}_y(v)\{\!\{x/u\}\!\} & := \mathcal{W}_{y\setminus\mathtt{fv}(u)}(v\{\!\{x/u\}\!\}) & x \neq y \\
\mathcal{C}_y^{z|w}(v)\{\!\{x/u\}\!\} & := \begin{cases} \mathcal{C}_\Gamma^{\Delta|\Pi}(v\{z/R_\Delta^\Gamma(u)\}\{w/R_\Pi^\Gamma(u)\}) \\ \mathcal{C}_y^{z|w}(v\{\!\{x/u\}\!\}) \end{cases} & \begin{cases} x = y\ \&\ \Gamma = \mathtt{fv}(u) \\ \Delta, \Pi \text{ are fresh} \end{cases} \\
 & & x \neq y\ \&\ z, w \notin \mathtt{fv}(u)
\end{array}$$

This definition looks complex, this is because it is covering all the calculi of the prismoid by a unique homogeneous specification. The restriction of this operation to particular subsets of resources results in simplified notions of substitutions. As a typical example, the previous definition can be shown to be equivalent to the well-known notion of higher-order substitution on $\mathtt{s}$-terms [8].

We now introduce the reduction system of the prismoid. In the last column of Figure 1 we use the notation $\mathcal{A}^+$ (resp. $\mathcal{A}^-$) to specify that the equation/rule belongs to the calculus $\lambda_{\mathcal{B}}$ iff $\mathcal{A} \subseteq \mathcal{B}$ (resp. $\mathcal{A} \cap \mathcal{B} = \emptyset$). Thus, each calculus $\lambda_{\mathcal{B}}$ contains only a strict subset of the reduction rules and equations in Figure 1.

Equations :

$(\mathtt{CC}_{\mathcal{A}})$	$\mathcal{C}^{x\|z}_w(\mathcal{C}^{y\|p}_x(t))$	$\equiv \mathcal{C}^{x\|y}_w(\mathcal{C}^{z\|p}_x(t))$		$\mathtt{c}^+$
$(\mathtt{C}_{\mathcal{C}})$	$\mathcal{C}^{y\|z}_x(t)$	$\equiv \mathcal{C}^{z\|y}_x(t)$		$\mathtt{c}^+$
$(\mathtt{CC}_{\mathcal{C}})$	$\mathcal{C}^{b\|c}_a(\mathcal{C}^{y\|z}_x(t))$	$\equiv \mathcal{C}^{y\|z}_x(\mathcal{C}^{b\|c}_a(t))$	$x \neq b, c$ & $a \neq y, z$	$\mathtt{c}^+$
$(\mathtt{WW}_{\mathcal{C}})$	$\mathcal{W}_x(\mathcal{W}_y(t))$	$\equiv \mathcal{W}_y(\mathcal{W}_x(t))$		$\mathtt{w}^+$
$(\mathtt{SS}_{\mathcal{C}})$	$t[x/u][y/v]$	$\equiv t[y/v][x/u]$	$y \notin \mathtt{fv}(u)$ & $x \notin \mathtt{fv}(v)$	$\mathtt{s}^+$

Rules :

(β)	$(\lambda x.t)\ u$	$\to t\{x/u\}$		$\mathtt{s}^-$
$(\mathtt{B})$	$(\lambda x.t)\ u$	$\to t[x/u]$		$\mathtt{s}^+$
$(\mathtt{V})$	$x[x/u]$	$\to u$		$\mathtt{s}^+$
$(\mathtt{SGc})$	$t[x/u]$	$\to t$	$x \notin \mathtt{fv}(t)$	$\mathtt{s}^+$ & $\mathtt{w}^-$
$(\mathtt{SDup})$	$t[x/u]$	$\to t_{[y]_x}[x/u][y/u]$	$\|\mathtt{fv}^+(t)\|_x > 1$ & y fresh	$\mathtt{s}^+$ & $\mathtt{c}^-$
$(\mathtt{SL})$	$(\lambda y.t)[x/u]$	$\to \lambda y.t[x/u]$		$\mathtt{s}^+$
$(\mathtt{SA}_{\mathtt{L}})$	$(t\ v)[x/u]$	$\to t[x/u]\ v$	$x \notin \mathtt{fv}(v)$	$\mathtt{s}^+$
$(\mathtt{SA}_{\mathtt{R}})$	$(t\ v)[x/u]$	$\to t\ v[x/u]$	$x \notin \mathtt{fv}(t)$	$\mathtt{s}^+$
$(\mathtt{SS})$	$t[x/u][y/v]$	$\to t[x/u[y/v]]$	$y \in \mathtt{fv}^+(u) \setminus \mathtt{fv}(t)$	$\mathtt{s}^+$
$(\mathtt{SW}_1)$	$\mathcal{W}_x(t)[x/u]$	$\to \mathcal{W}_{\mathtt{fv}(u)\setminus\mathtt{fv}(t)}(t)$		$(\mathtt{sw})^+$
$(\mathtt{SW}_2)$	$\mathcal{W}_y(t)[x/u]$	$\to \mathcal{W}_{y\setminus\mathtt{fv}(u)}(t[x/u])$	$x \neq y$	$(\mathtt{sw})^+$
$(\mathtt{LW})$	$\lambda x.\mathcal{W}_y(t)$	$\to \mathcal{W}_y(\lambda x.t)$	$x \neq y$	$\mathtt{w}^+$
$(\mathtt{AW}_{\mathtt{l}})$	$\mathcal{W}_y(u)\ v$	$\to \mathcal{W}_{y\setminus\mathtt{fv}(v)}(u\ v)$		$\mathtt{w}^+$
$(\mathtt{AW}_{\mathtt{r}})$	$u\ \mathcal{W}_y(v)$	$\to \mathcal{W}_{y\setminus\mathtt{fv}(u)}(u\ v)$		$\mathtt{w}^+$
$(\mathtt{SW})$	$t[x/\mathcal{W}_y(u)]$	$\to \mathcal{W}_{y\setminus\mathtt{fv}(t)}(t[x/u])$		$(\mathtt{sw})^+$
$(\mathtt{SCa})$	$\mathcal{C}^{y\|z}_x(t)[x/u]$	$\to \mathcal{C}^{\Delta\|\Pi}_{\Gamma}(t[y/R^{\Gamma}_{\Delta}(u)][z/R^{\Gamma}_{\Pi}(u)])$	$y, z \in \mathtt{fv}^+(t)$; $\Gamma = \mathtt{fv}(u)$; Δ and Π are fresh	$(\mathtt{cs})^+$
$(\mathtt{CL})$	$\mathcal{C}^{y\|z}_w(\lambda x.t)$	$\to \lambda x.\mathcal{C}^{y\|z}_w(t)$		$\mathtt{c}^+$
$(\mathtt{CA}_{\mathtt{L}})$	$\mathcal{C}^{y\|z}_w(t\ u)$	$\to \mathcal{C}^{y\|z}_w(t)\ u$	$y, z \notin \mathtt{fv}(u)$	$\mathtt{c}^+$
$(\mathtt{CA}_{\mathtt{R}})$	$\mathcal{C}^{y\|z}_w(t\ u)$	$\to t\ \mathcal{C}^{y\|z}_w(u)$	$y, z \notin \mathtt{fv}(t)$	$\mathtt{c}^+$
$(\mathtt{CS})$	$\mathcal{C}^{y\|z}_w(t[x/u])$	$\to t[x/\mathcal{C}^{y\|z}_w(u)]$	$y, z \in \mathtt{fv}^+(u)$	$(\mathtt{cs})^+$
$(\mathtt{SCb})$	$\mathcal{C}^{y\|z}_w(t)[x/u]$	$\to \mathcal{C}^{y\|z}_w(t[x/u])$	$x \neq w$ & $y, z \notin \mathtt{fv}(u)$	$(\mathtt{cs})^+$
$(\mathtt{CW}_1)$	$\mathcal{C}^{y\|z}_w(\mathcal{W}_y(t))$	$\to R^z_w(t)$		$(\mathtt{cw})^+$
$(\mathtt{CW}_2)$	$\mathcal{C}^{y\|z}_w(\mathcal{W}_x(t))$	$\to \mathcal{W}_x(\mathcal{C}^{y\|z}_w(t))$	$x \neq y, z$	$(\mathtt{cw})^+$
$(\mathtt{CGc})$	$\mathcal{C}^{y\|z}_w(t)$	$\to R^z_w(t)$	$y \notin \mathtt{fv}(t)$	$\mathtt{c}^+$ & $\mathtt{w}^-$

Fig. 1. The reduction rules and equations of the prismoid

All the equations and rules can be understood by means of MELL Proof-Nets reduction (see for example [9]). The reduction rules can be split into four groups: the first one fires implicit/explicit substitution, the second one implements substitution by decrementing multiplicity of variables and/or performing propagation, the third one pulls weakening operators as close to the top as possible and the fourth one pushes contractions as deep as possible. Alpha-conversion guarantees that no capture of variables occurs during reduction. The use of positive conditions (conditions on positive free variables) in some of the rules will become clear when discussing projection at the end of Section 4.

The notations $\Rightarrow_{\mathcal{R}}$, $\equiv_{\mathcal{E}}$ and $\rightarrow_{\mathcal{R}\cup\mathcal{E}}$, mean, respectively, the rewriting (resp. equivalence and rewriting modulo) relation generated by the rules $\mathcal{R}$ (resp. equations $\mathcal{E}$ and rules $\mathcal{R}$ modulo equations $\mathcal{E}$). Similarly, $\Rightarrow_{\mathcal{B}}$, $\equiv_{\mathcal{B}}$ and $\rightarrow_{\mathcal{B}}$ mean, respectively, the rewriting (resp. equivalence and rewriting modulo) relation generated by the rules (resp. the equations and rules modulo equations) of the calculus $\lambda_{\mathcal{B}}$. Thus for example the reduction relation $\rightarrow_{\emptyset}$ is only generated by the β-rule exactly as in λ-calculus. Another example is $\rightarrow_{\mathtt{c}}$ which can be written $\rightarrow_{\{\beta,\mathtt{CL},\mathtt{CA_L},\mathtt{CA_R},\mathtt{CGc}\}\cup\{\mathtt{CC}_{\mathcal{A}},\mathtt{C}_{\mathcal{C}},\mathtt{CC}_{\mathcal{C}}\}}$. Sometimes we mix both notations to denote particular subrelations, thus for example $\rightarrow_{\mathtt{c}\setminus\beta}$ means $\rightarrow_{\{\mathtt{CL},\mathtt{CA_L},\mathtt{CA_R},\mathtt{CGc}\}\cup\{\mathtt{CC}_{\mathcal{A}},\mathtt{C}_{\mathcal{C}},\mathtt{CC}_{\mathcal{C}}\}}$.

Among the eight calculi of the prismoid we can distinguish the $\lambda_{\emptyset}$-calculus, known as λ-calculus, which is defined by means of the $\rightarrow_{\emptyset}$-reduction relation on $\emptyset$-terms. Another language of the prismoid is the $\lambda_{\mathtt{csw}}$-calculus, a variation of $\lambda\mathtt{lxr}$ [9], defined by means of the $\rightarrow_{\{\mathtt{c},\mathtt{s},\mathtt{w}\}}$-reduction relation on $\{\mathtt{c},\mathtt{s},\mathtt{w}\}$-terms. A last example is the $\lambda_{\mathtt{w}}$-calculus given by means of $\rightarrow_{\mathtt{w}}$-reduction, that is, $\rightarrow_{\{\beta,\mathtt{LW},\mathtt{AW_l},\mathtt{AW_r}\}\cup\{\mathtt{WW}_{\mathcal{C}}\}}$.

A $\mathcal{B}$-term t is in **$\mathcal{B}$-normal form** is there is no u s.t. $t \rightarrow_{\mathcal{B}} u$. A $\mathcal{B}$-term t is said to be **$\mathcal{B}$-strongly normalising**, written $t \in \mathcal{SN}_{\mathcal{B}}$, iff there is no infinite $\mathcal{B}$-reduction sequence starting at t.

The system enjoys the following properties.

Lemma 1 (Preservation of Well-Formed Terms by Reduction). *If $\Gamma \Vdash_{\mathcal{B}} t$ and $t \rightarrow_{\mathcal{B}} u$, then $\exists\ \Delta \subseteq \Gamma$ s.t. $\Delta \Vdash_{\mathcal{B}} u$. Moreover $\mathtt{w} \in \mathcal{B}$ implies $\Delta = \Gamma$.*

Lemma 2 (Full Composition). *Let $t[y/v] \in \mathcal{T}_{\mathcal{B}}$. Then $t[y/v] \rightarrow^*_{\mathcal{B}} t\{y/v\}$.*

3 Adding Resources

This section is devoted to the simulation of the $\lambda_{\emptyset}$-calculus into richer calculi having more resources. The operation is only defined in the calculi of the base $\mathfrak{B}_I$. We consider the function $\mathtt{AR}_{\mathcal{A}}(\_) : \mathcal{T}_{\emptyset} \mapsto \mathcal{T}_{\mathcal{A}}$ for $\mathcal{A} \subseteq \mathcal{R}$ which enriches a $\lambda_{\emptyset}$-term in order to fulfill the constraints needed to be an $\mathcal{A}$-term. Adding is done not only on a static level (the terms) but also on a dynamic level (reduction).

$$
\begin{array}{lll}
\mathtt{AR}_{\mathcal{A}}(x) & := x & \\
\mathtt{AR}_{\mathcal{A}}(\lambda x.t) & := \begin{cases} \lambda x.\mathcal{W}_x(\mathtt{AR}_{\mathcal{A}}(t)) \\ \lambda x.\mathtt{AR}_{\mathcal{A}}(t) \end{cases} & \begin{array}{l} \mathtt{w} \in \mathcal{A}\ \&\ x \notin \mathtt{fv}(t) \\ \text{otherwise} \end{array} \\
\mathtt{AR}_{\mathcal{A}}(t\ u) & := \begin{cases} \mathcal{C}_{\Gamma}^{\Delta|\Pi}(R^{\Gamma}_{\Delta}(\mathtt{AR}_{\mathcal{A}}(t))R^{\Gamma}_{\Pi}(\mathtt{AR}_{\mathcal{A}}(u))) \\ \mathtt{AR}_{\mathcal{A}}(t)\ \mathtt{AR}_{\mathcal{A}}(u) \end{cases} & \begin{array}{l} \begin{cases} \mathtt{c} \in \mathcal{A}\ \&\ \Gamma = \mathtt{fv}(t) \cap \mathtt{fv}(u) \\ \Delta \text{ and } \Pi \text{ are fresh} \end{cases} \\ \text{otherwise} \end{array}
\end{array}
$$

For example, adding resource $\mathtt{c}$ (resp. $\mathtt{w}$) to $t = \lambda x.yy$ gives $\lambda x.\mathcal{C}_y^{y_1|y_2}(y_1 y_2)$ (resp. $\lambda x.\mathcal{W}_x(yy)$), while adding both of them gives $\lambda x.\mathcal{W}_x(\mathcal{C}_y^{y_1|y_2}(y_1 y_2))$.

We now establish the relation between $\mathtt{AR}_{\mathcal{A}}()$ and implicit substitution, which is the technical key lemma of the paper.

Lemma 3. *Let $t, u \in \mathcal{T}_\emptyset$. Then*

- *If $\mathtt{c} \notin \mathcal{A}$, then $\mathtt{AR}_{\mathcal{A}}(t)\{x/\mathtt{AR}_{\mathcal{A}}(u)\} = \mathtt{AR}_{\mathcal{A}}(t\{x/u\})$.*
- *If $\mathtt{c} \in \mathcal{A}$, then $\mathcal{C}_\Gamma^{\Delta|\Pi}(R_\Delta^\Gamma(\mathtt{AR}_{\mathcal{A}}(t))\{x/R_\Pi^\Gamma(\mathtt{AR}_{\mathcal{A}}(u))\}) \to_{\mathcal{A}}^* \mathtt{AR}_{\mathcal{A}}(t\{x/u\})$, where $\Gamma = (\mathtt{fv}(t) \setminus x) \cap \mathtt{fv}(u)$ and Δ, Π are fresh sets of variables.*

Proof. By induction on t, using the usual lambda-calculus substitution definition [3] for x, as the source calculus is the lambda-calculus.

Theorem 1 (Simulation). *Let $t \in \mathcal{T}_\emptyset$ such that $t \to_\emptyset t'$.*

- *If $\mathtt{w} \in \mathcal{A}$, then $\mathtt{AR}_{\mathcal{A}}(t) \to_{\mathcal{A}}^+ \mathcal{W}_{\mathtt{fv}(t)\setminus\mathtt{fv}(t')}(\mathtt{AR}_{\mathcal{A}}(t'))$.*
- *If $\mathtt{w} \notin \mathcal{A}$, then $\mathtt{AR}_{\mathcal{A}}(t) \to_{\mathcal{A}}^+ \mathtt{AR}_{\mathcal{A}}(t')$.*

Proof. By induction on the reduction relation $\to_\beta$ using Lemma 3.

While Theorem 1 states that adding resources to the $\lambda_\emptyset$-calculus is well behaved, this does not necessarily hold for *any* arbitrary calculus of the prismoid. Thus for example, what happens when the $\lambda_{\mathtt{s}}$-calculus is enriched with resource $\mathtt{w}$? Is it possible to simulate each $\mathtt{s}$-reduction step by a sequence of $\mathtt{sw}$-reduction steps? Unfortunately the answer is no: we have $t_1 = (x\ y)[z/v] \to_{\mathtt{s}} x\ y[z/v] = t_2$ but $\mathtt{AR}_{\mathtt{w}}(t_1) = \mathcal{W}_z(x\ y)[z/v] \not\to_{\mathtt{sw}} x\ \mathcal{W}_z(y)[z/v] = \mathtt{AR}_{\mathtt{w}}(t_2)$.

4 Removing Resources

In this section we give a mechanism to remove resources, that is, to change the status of weakening and/or contraction from explicit to implicit. This is dual to the operation allowing to add resources to terms presented in Section 3. Whereas adding is only defined within the implicit base, removing is defined in both bases. As adding, removing is not only done on a static level, but also on a dynamic one. Thus for example, removing translates any $\mathtt{csw}$-reduction sequence into a $\mathcal{B}$-reduction sequence, for any $\mathcal{B} \in \{\mathtt{s}, \mathtt{cs}, \mathtt{sw}\}$.

Given two lists of variables $\Gamma = y_1 \dots y_n$ (with all y_i distinct) and $\Delta = z_1 \dots z_n$, then $(\Gamma \mapsto \Delta)(y)$ is y if $y \notin \Gamma$, or z_i if $y = y_i$ for some i. The **collapsing** function of a term without contractions is then defined modulo α-conversion as follows:

$$
\begin{array}{lll}
\mathtt{S}_\Delta^\Gamma(y) & := (\Gamma \mapsto \Delta)(y) & \\
\mathtt{S}_\Delta^\Gamma(uv) & := \mathtt{S}_\Delta^\Gamma(u)\mathtt{S}_\Delta^\Gamma(v) & \\
\mathtt{S}_\Delta^\Gamma(\lambda y.u) & := \lambda y.\mathtt{S}_\Delta^\Gamma(u) & y \notin \Gamma \\
\mathtt{S}_\Delta^\Gamma(u[y/v]) & := \mathtt{S}_\Delta^\Gamma(u)[y/\mathtt{S}_\Delta^\Gamma(v)] & y \notin \Gamma \\
\mathtt{S}_\Delta^\Gamma(\mathcal{W}_y(v)) & := \begin{cases} \mathtt{S}_\Delta^\Gamma(v) \\ \mathcal{W}_y(\mathtt{S}_\Delta^\Gamma(v)) \end{cases} & \begin{array}{l} (\Gamma \mapsto \Delta)(y) \in \mathtt{fv}(\mathtt{S}_\Delta^\Gamma(v)) \\ (\Gamma \mapsto \Delta)(y) \notin \mathtt{fv}(\mathtt{S}_\Delta^\Gamma(v)) \end{array}
\end{array}
$$

This function renames the variables of a term in such a way that every occurrence of $\mathcal{W}_x(t)$ in the term implies $x \notin \mathtt{fv}(t)$. For example $\mathtt{S}^{y,z}_{x,x}(\mathcal{W}_y(\mathcal{W}_z(x))) = x$.

The function $\mathtt{RR}_{\mathcal{A}}(\_) : \mathcal{T}_{\mathcal{B}} \mapsto \mathcal{T}_{\mathcal{B}\setminus\mathcal{A}}$ removes $\mathcal{A} \subseteq \mathcal{R}$ from a $\mathcal{B}$-term .

$$\begin{array}{llll}
\mathtt{RR}_{\mathcal{A}}(x) & := x & \mathtt{RR}_{\mathcal{A}}(t[x/u]) & := \mathtt{RR}_{\mathcal{A}}(t)[x/\mathtt{RR}_{\mathcal{A}}(u)] \\
\mathtt{RR}_{\mathcal{A}}(\lambda x.t) & := \lambda x.\mathtt{RR}_{\mathcal{A}}(t) & \mathtt{RR}_{\mathcal{A}}(\mathcal{W}_x(t)) & := \begin{cases} \mathtt{RR}_{\mathcal{A}}(t) & \text{if } \mathtt{w} \in \mathcal{A} \\ \mathcal{W}_x(\mathtt{RR}_{\mathcal{A}}(t)) & \text{if } \mathtt{w} \notin \mathcal{A} \end{cases} \\
\mathtt{RR}_{\mathcal{A}}(t\ u) & := \mathtt{RR}_{\mathcal{A}}(t)\ \mathtt{RR}_{\mathcal{A}}(u) & \mathtt{RR}_{\mathcal{A}}(\mathcal{C}^{y|z}_x(t)) & := \begin{cases} \mathtt{S}^{y,z}_{x,x}(\mathtt{RR}_{\mathcal{A}}(t)) & \text{if } \mathtt{c} \in \mathcal{A} \\ \mathcal{C}^{y|z}_x(\mathtt{RR}_{\mathcal{A}}(t)) & \text{if } \mathtt{c} \notin \mathcal{A} \end{cases}
\end{array}$$

Lemma 4. *Let* $t, u \in \mathcal{T}_{\mathcal{A}}$ *and* $\mathtt{b} \in \mathcal{R}$. *Then* $\mathtt{RR}_{\mathtt{b}}(t\{x/u\}) = \mathtt{RR}_{\mathtt{b}}(t)\{x/\mathtt{RR}_{\mathtt{b}}(u)\}$.

Calculi of the prismoid include rules/equations to handle substitution but also other rules/equations to handle resources $\{\mathtt{c}, \mathtt{w}\}$. Moreover, implicit (resp. explicit) substitution is managed by the β-rule (resp. the whole system $\mathtt{s}$). We can then split the reduction relation $\to_{\mathcal{B}}$ in two different parts: one for (implicit or explicit) substitution, which can be *s*trictly projected into itself, and another one for weakening and contraction, which can be projected into a more subtle way given by the following statement.

Theorem 2. *Let* $\mathcal{A} \subseteq \mathcal{R}$ *such that* $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{S}$ *and let* $t \in \mathcal{T}_{\mathcal{B}}$. *If* $t \equiv_{\mathcal{B}} u$, *then* $\mathtt{RR}_{\mathcal{A}}(t) \equiv_{\mathcal{B}\setminus\mathcal{A}} \mathtt{RR}_{\mathcal{A}}(u)$. *Otherwise, we sum up in the following array:*

$\mathtt{s} \notin \mathcal{B}$	$t \Rightarrow_\beta u$	$\mathtt{RR}_{\mathcal{A}}(t) \to^+_\beta \mathtt{RR}_{\mathcal{A}}(u)$	$\mathtt{s} \in \mathcal{B}$	$t \Rightarrow_{\mathtt{s}} u$	$\mathtt{RR}_{\mathcal{A}}(t) \to^+_{\mathtt{s}} \mathtt{RR}_{\mathcal{A}}(u)$
	$t \Rightarrow_{\mathcal{B}\setminus\beta} u$	$\mathtt{RR}_{\mathcal{A}}(t) \to^*_{\mathcal{B}\setminus\beta\setminus\mathcal{A}} \mathtt{RR}_{\mathcal{A}}(u)$		$t \Rightarrow_{\mathcal{B}\setminus\mathtt{s}} u$	$\mathtt{RR}_{\mathcal{A}}(t) \to^*_{\mathcal{B}\setminus\mathtt{s}\setminus\mathcal{A}} \mathtt{RR}_{\mathcal{A}}(u)$
		$\mathtt{RR}_{\mathcal{B}}(t) = \mathtt{RR}_{\mathcal{B}}(u)$			$\mathtt{RR}_{\mathcal{B}}(t) = \mathtt{RR}_{\mathcal{B}}(u)$

Proof. By induction on the reduction relation using Lemma 4. For the points involving $\mathtt{RR}_{\mathcal{A}}(\_)$, one can first consider the case where $\mathcal{A}$ is a singleton. Then the general result follows from two successive applications of the simpler property.

It is now time to discuss the need of positive conditions (conditions involving positive free variables) in the specification of the reduction rules of the prismoid. For that, let us consider a relaxed form of $\mathtt{SS}_1$ rule $t[x/u][y/v] \to t[x/u[y/v]]$ if $y \in \mathtt{fv}(u) \setminus \mathtt{fv}(t)$ (instead of $y \in \mathtt{fv}^+(u) \setminus \mathtt{fv}(t)$).

The need of the condition $y \in \mathtt{fv}(u)$ is well-known [4], otherwise PSN does not hold. The need of the condition $y \notin \mathtt{fv}(t)$ is also natural if one wants to preserve well-formed terms. Now, the reduction step $t_1 = x[x/\mathcal{W}_y(z)][y/y'] \to_{\mathtt{SS}_1} x[x/\mathcal{W}_y(z)[y/y']] = t_2$ in the calculus with sorts $\{\mathtt{s}, \mathtt{w}\}$ cannot be projected into $\mathtt{RR}_{\mathtt{W}}(t_1) = x[x/z][y/y'] \to_{\mathtt{SS}_1} x[x/z[y/y']] = \mathtt{RR}_{\mathtt{W}}(t_2)$ since $y \notin \mathtt{fv}(z)$. Similar examples can be given to justify positive conditions in rules $\mathtt{SDup}, \mathtt{SCa}$ and $\mathtt{CS}$.

Lemma 5. *Let* $t \in \mathcal{T}_\emptyset$ *and let* $\mathcal{A} \subseteq \mathcal{R}$. *Then* $\mathtt{RR}_{\mathcal{A}}(\mathtt{AR}_{\mathcal{A}}(t)) = t$.

The following property states that administration of weakening and/or contraction is terminating in any calculus. The proof can be done by interpreting reduction steps by a strictly decreasing arithmetical measure.

Lemma 6. *If* $\mathtt{s} \notin \mathcal{B}$*, then the reduction relation* $\to_{\mathcal{B}\setminus\beta}$ *is terminating. If* $\mathtt{s} \in \mathcal{B}$*, then the reduction relation* $\to_{\mathcal{B}\setminus\mathtt{s}}$ *is terminating.*

We conclude this section by relating adding and removing resources :

Corollary 1. *Let* $\emptyset \neq \mathcal{A} \subseteq \mathcal{R}$*. Then, the unique* $\mathcal{A}$*-normal form of* $t \in \mathcal{T}_{\mathcal{A}}$ *is* $\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(t))$ *if* $\mathtt{w} \notin \mathcal{A}$*, and* $\mathcal{W}_{\mathtt{fv}(t)\setminus\mathtt{fv}(\mathtt{RR}_{\mathcal{A}}(t))}(\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(t)))$ *if* $\mathtt{w} \in \mathcal{A}$*.*

Proof. Suppose $\mathtt{w} \in \mathcal{A}$. Termination of $\to_{\mathcal{A}}$ (Lemma 6) implies that there is t' in $\mathcal{A}$-normal form such that $t \to^*_{\mathcal{A}} t'$. By Lemma 1, $\mathtt{fv}(t) = \mathtt{fv}(t')$ and by Theorem 2, $\mathtt{RR}_{\mathcal{A}}(t) = \mathtt{RR}_{\mathcal{A}}(t')$. Since t' is in $\mathcal{A}$-normal form, then we get $t' \equiv_{\mathcal{A}} \mathcal{W}_{\mathtt{fv}(t')\setminus\mathtt{fv}(\mathtt{RR}_{\mathcal{A}}(t'))}(\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(t')))$ by a simple induction. Hence, $t' \equiv_{\mathcal{A}} \mathcal{W}_{\mathtt{fv}(t)\setminus\mathtt{fv}(\mathtt{RR}_{\mathcal{A}}(t))}(\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(t)))$. To show uniqueness, let us consider two $\mathcal{A}$-normal forms t'_1 and t'_2 of t. By the previous remark, both t'_1 and t'_2 are congruent to the term $\mathcal{W}_{\mathtt{fv}(t)\setminus\mathtt{fv}(\mathtt{RR}_{\mathcal{A}}(t))}(\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(t)))$ which concludes the case. The case $\mathtt{w} \notin \mathcal{A}$ is similar.

5 Untyped Properties

We first show PSN for all the calculi of the prismoid. The proof will be split in two different subcases, one for each base. This dissociation comes from the fact that redexes are erased by β-reduction in base $\mathfrak{B}_I$ while they are erased by $\mathtt{SGc}$ and/or $\mathtt{SW_1}$-reduction in base $\mathfrak{B}_E$.

Theorem 3 (PSN for the prismoid). *Let* $\mathcal{B} \subseteq \mathcal{S}$ *and* $\mathcal{A} = \mathcal{B} \setminus \{\mathtt{s}\}$*. If* $t \in \mathcal{T}_{\emptyset}$ & $t \in \mathcal{SN}_{\emptyset}$*, then* $\mathtt{AR}_{\mathcal{A}}(t) \in \mathcal{SN}_{\mathcal{B}}$*.*

Proof. There are three cases, one for $\mathfrak{B}_I$ and two subcases for $\mathfrak{B}_E$.

- Suppose $\mathtt{s} \notin \mathcal{B}$. We first show that $u \in \mathcal{T}_{\mathcal{B}}$ & $\mathtt{RR}_{\mathcal{B}}(u) \in \mathcal{SN}_{\emptyset}$ imply $u \in \mathcal{SN}_{\mathcal{B}}$. For that we apply Theorem 6 in the appendix with $\mathtt{A}_1 =\to_{\beta}$, $\mathtt{A}_2 =\to_{\mathcal{B}\setminus\beta}$, $\mathtt{A} =\to_{\beta}$ and $\mathcal{R} = \mathtt{RR}_{\mathcal{B}}(\_)$, using Theorem 2 and Lemma 6. Take $u = \mathtt{AR}_{\mathcal{B}}(t)$. Then $\mathtt{RR}_{\mathcal{B}}(\mathtt{AR}_{\mathcal{B}}(t)) =_{L.5} t \in \mathcal{SN}_{\emptyset}$ by hypothesis. Thus, $\mathtt{AR}_{\mathcal{B}}(t) \in \mathcal{SN}_{\mathcal{B}}$.
- Suppose $\mathcal{B} = \{\mathtt{s}\}$. The proof of $\mathtt{AR}_{\mathtt{s}}(t) = t \in \mathcal{SN}_{\mathtt{s}}$ follows a modular proof technique to show PSN of calculi with full composition which is completely developed in [8]. Details concerning the $\mathtt{s}$-calculus can be found in [17].
- Suppose $\mathtt{s} \in \mathcal{B}$. Then $\mathcal{B} = \{\mathtt{s}\} \cup \mathcal{A}$. We show that $u \in \mathcal{T}_{\mathcal{B}}$ & $\mathtt{RR}_{\mathcal{A}}(u) \in \mathcal{SN}_{\mathtt{s}}$ imply $u \in \mathcal{SN}_{\mathcal{B}}$. For that we apply Theorem 6 in the appendix with $\mathtt{A}_1 =\to_{\mathtt{s}}$, $\mathtt{A}_2 =\to_{\mathcal{B}\setminus\mathtt{s}}$, $\mathtt{A} =\to_{\mathtt{s}}$ and $\mathcal{R} = \mathtt{RR}_{\mathcal{A}}(\_)$, using Theorem 2 and Lemma 6. Now, take $u = \mathtt{AR}_{\mathcal{A}}(t)$. We have $\mathtt{RR}_{\mathcal{A}}(\mathtt{AR}_{\mathcal{A}}(t)) = \mathtt{RR}_{\mathcal{A}}(\mathtt{AR}_{\mathcal{A}}(t)) =_{L.5} t \in \mathcal{SN}_{\emptyset}$ by hypothesis and $t \in \mathcal{SN}_{\mathtt{s}}$ by the previous point. Thus, $\mathtt{AR}_{\mathcal{A}}(t) \in \mathcal{SN}_{\mathcal{B}}$.

Confluence of each calculus of the prismoid is based on that of the $\lambda_{\emptyset}$-calculus [3]. For any $\mathcal{A} \subseteq \mathcal{R}$, consider $\mathtt{xc} : \mathcal{T}_{\{\mathtt{s}\}\cup\mathcal{A}} \mapsto \mathcal{T}_{\mathcal{A}}$ which replaces explicit by implicit substitution.

$$
\begin{array}{llll}
\mathtt{xc}(y) & := y & \mathtt{xc}(\mathcal{W}_y(t)) & := \mathcal{W}_y(\mathtt{xc}(t)) \\
\mathtt{xc}(t\ u) & := \mathtt{xc}(t)\ \mathtt{xc}(u) & \mathtt{xc}(\mathcal{C}^{y_1|y_2}_y(t)) & := \mathcal{C}^{y_1|y_2}_y(\mathtt{xc}(t)) \\
\mathtt{xc}(\lambda y.t) & := \lambda y.\mathtt{xc}(t) & \mathtt{xc}(t[y/u]) & := \mathtt{xc}(t)\{y/\mathtt{xc}(u)\}
\end{array}
$$

Lemma 7. *Let $t \in \mathcal{T}_{\mathcal{B}}$. Then 1) $t \to^*_{\mathcal{B}} \mathtt{xc}(t)$, 2) $\mathtt{RR}_{\mathcal{B}\setminus\mathtt{S}}(\mathtt{xc}(t)) = \mathtt{xc}(\mathtt{RR}_{\mathcal{B}\setminus\mathtt{S}}(t))$. 3) if $t \to_{\mathtt{S}} u$, then $\mathtt{xc}(t) \to^*_{\beta} \mathtt{xc}(u)$.*

Proof. The first and the second property are shown by induction on t using respectively Lemmas 2 and 4. The third property is shown by induction on $t \to_{\mathtt{S}} u$.

Theorem 4. *All the languages of the prismoid are confluent.*

Proof. Let $t \to_{\mathcal{B}} t_1$ and $t \to_{\mathcal{B}} t_2$. We remark that $\mathcal{B} = \mathcal{A}$ or $\mathcal{B} = \{\mathtt{s}\} \cup \mathcal{A}$, with $\mathcal{A} \subseteq \mathcal{R}$. We have $\mathtt{RR}_{\mathcal{A}}(t) \to^*_{\mathcal{B}\setminus\mathcal{A}} \mathtt{RR}_{\mathcal{A}}(t_i)$ (i=1,2) by Theorem 2; $\mathtt{xc}(\mathtt{RR}_{\mathcal{A}}(t)) \to^*_{\beta} \mathtt{xc}(\mathtt{RR}_{\mathcal{A}}(t_i))$ (i=1,2) by Lemma 7; and $\mathtt{xc}(\mathtt{RR}_{\mathcal{A}}(t_i)) \to^*_{\beta} t_3$ (i=1,2) for some $t_3 \in \mathcal{T}_{\emptyset}$ by confluence of the λ-calculus [3]. Also, $\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(\mathtt{xc}(t_i))) =_{L.\ 7} \mathtt{AR}_{\mathcal{A}}(\mathtt{xc}(\mathtt{RR}_{\mathcal{A}}(t_i))) \to^*_{\mathcal{A}} \mathcal{W}_{\Delta_i}(\mathtt{AR}_{\mathcal{A}}(t_3))$ for some Δ_i (i=1,2) by Theorem 1.

But $t_i \to^*_{\mathcal{B}}$ $(L.\ 7)$ $\mathtt{xc}(t_i) \to^*_{\mathcal{A}}$ $(C.\ 1)$ $\mathcal{W}_{\Gamma_i}(\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(\mathtt{xc}(t_i))))$ for some Γ_i (i=1,2). Then $\mathcal{W}_{\Gamma_i}(\mathtt{AR}_{\mathcal{A}}(\mathtt{RR}_{\mathcal{A}}(\mathtt{xc}(t_i)))) \to^*_{\mathcal{A}} \mathcal{W}_{\Gamma_i \cup \Delta_i}(\mathtt{AR}_{\mathcal{A}}(t_3))$ (i=1,2). Now, $\to^*_{\mathcal{A}} \subseteq \to^*_{\mathcal{B}}$ so in order to close the diagram we reason as follows.

If $\mathtt{w} \notin \mathcal{B}$, then $\Gamma_1 \cup \Delta_1 = \Gamma_2 \cup \Delta_2 = \emptyset$ and we are done. If $\mathtt{w} \in \mathcal{B}$, then $\to_{\mathcal{B}}$ preserves free variables by Lemma 1 so that $\mathtt{fv}(t) = \mathtt{fv}(t_i) = \mathtt{fv}(\mathcal{W}_{\Gamma_i \cup \Delta_i}(\mathtt{AR}_{\mathcal{A}}(t_3)))$ (i=1,2) which gives $\Gamma_1 \cup \Delta_1 = \Gamma_2 \cup \Delta_2$

6 Typing

We now introduce simply typed terms for all the calculi of the prismoid, and show that they all enjoy strong normalisation. **Types** are built over a countable set of atomic symbols and the type constructor $\to$.

An **environment** is a finite set of pairs of the form $x : T$. If $\Gamma = \{x_1 : T_1, ..., x_n : T_n\}$ is an environment then the domain of Γ is $\mathtt{dom}(\Gamma) = \{x_1, ..., x_n\}$. Two environments Γ and Δ are said to be **compatible** if $x : T \in \Gamma$ and $x : U \in \Delta$ imply $T = U$. Two environments Γ and Δ are said to be **disjoint** if there is no common variable between them. Compatible union (resp. disjoint union) is defined to be the union of compatible (resp. disjoint) environments.

Typing judgements have the form $\Gamma \vdash t : T$ for t a term, T a type and Γ an environment. **Typing rules** extend the inductive rules for well-formed terms (Section 2) with type annotations. Thus, typed terms are necessarily well-formed and each set of sorts $\mathcal{B}$ has its own set of typing rules.

$$\frac{}{x : T \vdash_{\mathcal{B}} x : T} \qquad \frac{\Gamma \vdash_{\mathcal{B}} t : T}{x : U; (\Gamma \setminus\!\!\setminus_{\mathcal{B}} \{y : U, z : U\}) \vdash_{\mathcal{B}} \mathcal{C}^{y|z}_x(t) : T}\ (\mathtt{c} \in \mathcal{B})$$

$$\frac{\Gamma \vdash_{\mathcal{B}} t : T}{\Gamma; x : U \vdash_{\mathcal{B}} \mathcal{W}_x(t) : T}\ (\mathtt{w} \in \mathcal{B}) \qquad \frac{\Gamma \vdash_{\mathcal{B}} u : U \quad \Delta \vdash_{\mathcal{B}} t : T}{\Gamma \uplus_{\mathcal{B}} (\Delta \setminus\!\!\setminus_{\mathcal{B}} x : U) \vdash_{\mathcal{B}} t[x/u] : T}\ (\mathtt{s} \in \mathcal{B})$$

$$\frac{\Gamma \vdash_{\mathcal{B}} t : U}{\Gamma \setminus\!\!\setminus_{\mathcal{B}} x : T \vdash_{\mathcal{B}} \lambda x.t : T \to U} \qquad \frac{\Gamma \vdash_{\mathcal{B}} t : T \to U \quad \Delta \vdash_{\mathcal{B}} u : T}{\Gamma \uplus_{\mathcal{B}} \Delta \vdash_{\mathcal{B}} tu : U}$$

A term $t \in \mathcal{T}_{\mathcal{B}}$ is said to **have type** T (written $t \in \mathcal{T}_{\mathcal{B}}^{T}$) iff there is Γ s.t. $\Gamma \vdash_{\mathcal{B}} t : T$. A term $t \in \mathcal{T}_{\mathcal{B}}$ is said to be **well-typed** iff there is T s.t. $t \in \mathcal{T}_{\mathcal{B}}^{T}$. Remark that every well-typed $\mathcal{B}$-term has a unique type.

Lemma 8. *If $\Gamma \vdash_{\mathcal{B}} t : T$, then 1) $\mathtt{fv}(t) = \mathtt{dom}(\Gamma)$, 2) $\Gamma \setminus \Pi; \Delta \vdash_{\mathcal{B}} R_{\Delta}^{\Pi}(t) : T$, for every $\Pi \subseteq \Gamma$ and fresh Δ, 3) $\mathtt{RR}_{\mathcal{A}}(t) \in \mathcal{T}_{\mathcal{B}\setminus\mathcal{A}}^{T}$, for every $\mathcal{A} \subseteq \mathcal{R}$.*

Proof. By induction on $\Gamma \vdash_{\mathcal{B}} t : T$.

Theorem 5 (Subject Reduction). *If $t \in \mathcal{T}_{\mathcal{B}}^{T}$ & $t \to_{\mathcal{B}} u$, then $u \in \mathcal{T}_{\mathcal{B}}^{T}$.*

Proof. By induction on the reduction relation using Lemma 8.

Corollary 2 (Strong normalization). *Let $t \in \mathcal{T}_{\mathcal{B}}^{T}$, then $t \in \mathcal{SN}_{\mathcal{B}}$.*

Proof. Let $\mathcal{A} \subseteq \mathcal{R}$ so that $\mathcal{B} = \mathcal{A}$ or $\mathcal{B} = \mathcal{A} \cup \{\mathtt{s}\}$. It is well-known that (simply) typed $\lambda_{\emptyset}$-calculus is strongly normalising (see for example [3]). It is also straightforward to show that PSN for the $\lambda_{\mathtt{s}}$-calculus implies strong normalisation for well-typed $\mathtt{s}$-terms (see for example [7]). By Theorem 2 any infinite $\mathcal{B}$-reduction sequence starting at t can be projected into an infinite $(\mathcal{B}\setminus\mathcal{A})$-reduction sequence starting at $\mathtt{RR}_{\mathcal{A}}(t)$. By Lemma 8 $\mathtt{RR}_{\mathcal{A}}(t)$ is a well-typed $(\mathcal{B} \setminus \mathcal{A})$-term, that is, a well-typed term in $\lambda_{\emptyset}$ or $\lambda_{\mathtt{s}}$. This leads to a contradiction.

7 Conclusion and Future Work

The prismoid of resources is proposed as an homogeneous framework to define λ-calculi being able to control weakening, contraction and linear substitution. The formalism is based on MELL Proof-Nets so that the computational behaviour of substitution is not only based on the propagation of substitution through terms but also on the decreasingness of the multiplicity of variables that are affected by substitutions. All calculi of the prismoid enjoy sanity properties such as simulation of β-reduction, confluence, preservation of β-strong normalisation and strong normalisation for typed terms.

The technology used in the prismoid could also be applied to implement higher-order rewriting systems. Indeed, it seems possible to extend these ideas to different frameworks such as CRSs [11], ERSs [10] or HRSs [16].

Another open problem concerns meta-confluence, that is, confluence for terms with meta-variables. This could be useful in the framework of Proof Assistants.

Finally, a more technical question is related to the operational semantics of the calculi of the prismoid. It seems possible to extend the ideas in [2] to our framework in order to identify those reduction rules of the prismoid that could be transformed into equations. Equivalence classes will be bigger, but reduction rules will coincide exactly with those of Nets [2]. While the operational semantics proposed in this paper is more adapted to implementation issues, the opposite direction would give a more abstract and flexible framework to study denotational properties.

References

1. M. Abadi, L. Cardelli, P. L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 4(1):375–416, 1991.
2. B. Accattoli and S. Guerrini. Jumping Boxes. Representing lambda-calculus boxes by jumps, 2009. To appear in *Proceedings of CSL*, 2009.
3. H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984.
4. R. Bloo. *Preservation of Termination for Explicit Substitution*. PhD thesis, Eindhoven University of Technology, 1997.
5. J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.
6. F. Kamareddine and A. Ríos. A λ-calculus à la de Bruijn with explicit substitutions. In *PLILP*, LNCS 982, 1995.
7. D. Kesner. The theory of explicit substitutions revisited. In *CSL*, LNCS 4646, 2007.
8. D. Kesner. Perpetuality for full and safe composition (in a constructive setting). In *ICALP*, Part II, LNCS 5126, 2008.
9. D. Kesner and S. Lengrand. Resource operators for lambda-calculus. *Information and Computation*, 205(4):419–473, 2007.
10. Z. Khasidashvili. Expression reduction systems. In *Proceedings of IN Vekua Institute of Applied Mathematics*, volume 36, Tbilisi, 1990.
11. J.-W. Klop. *Combinatory Reduction Systems*, volume 127 of *Mathematical Centre Tracts*. Mathematisch Centrum, Amsterdam, 1980. PhD Thesis.
12. S. Lengrand. *Normalisation and Equivalence in Proof Theory and Type Theory*. PhD thesis, University Paris 7 and University of St Andrews, Nov. 2006.
13. P. Lescanne and J. Rouyer-Degli. The calculus of explicit substitutions $\lambda\nu$. Technical report, INRIA, Lorraine, 1994.
14. P.-A. Melliès. Typed λ-calculi with explicit substitutions may not terminate. In *TLCA*, LNCS 902, 1995.
15. R. Milner. Local bigraphs and confluence: two conjectures. In *EXPRESS*, ENTCS 175, 2006.
16. T. Nipkow. Higher-order critical pairs. In *6th Annual IEEE Symposium on Logic in Computer Science (LICS)*, 1991.
17. F. Renaud. Preservation of strong normalisation for lambda-s, 2008. Available on `http://www.pps.jussieu.fr/~renaud`.
18. F.-R. Sinot, M. Fernández, and I. Mackie. Efficient reductions with director strings. In *CSL*, LNCS 2706, 2003.
19. V. van Oostrom. Net-calculus. Course Notes available on `http://www.phil.uu.nl/~oostrom/cmitt/00-01/net.ps`, 2001.

A Appendix

Theorem 6 ([12]). *Let* $\mathtt{A}_1$ *and* $\mathtt{A}_2$ *be two reduction relations on the set* $\mathtt{k}$ *and let* $\mathtt{A}$ *be a reduction relation on the set* $\mathtt{K}$*. Let* $\mathcal{R} \subseteq \mathtt{k} \times \mathtt{K}$*. Suppose*

- *For every* u, v, U *(*$u \mathrel{\mathcal{R}} U$ *&* $u \mathrel{\mathtt{A}_1} v$ *imply* $\exists V$ *s.t.* $v \mathrel{R} V$ *and* $U \mathrel{\mathtt{A}^+} V$*).*
- *For every* u, v, U *(*$u \mathrel{\mathcal{R}} U$ *&* $u \mathrel{\mathtt{A}_2} v$ *imply* $\exists V$ *s.t.* $v \mathrel{\mathcal{R}} V$ *and* $U \mathrel{\mathtt{A}^*} V$*).*
- *The relation* $\mathtt{A}_2$ *is well-founded.*

Then, $t \mathrel{\mathcal{R}} T$ *&* $T \in \mathcal{SN}_{\mathtt{A}}$ *imply* $t \in \mathcal{SN}_{\mathtt{A}_1 \cup \mathtt{A}_2}$*.*