

Computational interpretation of classical logic with explicit structural rules

Silvia Ghilezan, Pierre Lescanne, Dragisa Zunic

▶ **To cite this version:**

Silvia Ghilezan, Pierre Lescanne, Dragisa Zunic. Computational interpretation of classical logic with explicit structural rules. 2012. ensl-00681296

HAL Id: ensl-00681296

https://ens-lyon.hal.science/ensl-00681296

Preprint submitted on 21 Mar 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computational interpretation of classical logic with explicit structural rules

S. Ghilezan[a], P. Lescanne[b], D. Žunić[c]

[a] *University of Novi Sad, Faculty of Technical Sciences, Serbia*
[b] *University of Lyon, École Normal Supérieure de Lyon, France*
[c] *Faculty of Economics and Engineering Management, Novi Sad, Serbia*

Abstract

We present a calculus providing a Curry-Howard correspondence to classical logic represented in the sequent calculus with explicit structural rules, namely weakening and contraction. These structural rules introduce explicit erasure and duplication of terms, respectively. We present a type system for which we prove the type-preservation under reduction. A mutual relation with classical calculus featuring implicit structural rules has been studied in detail. From this analysis we derive strong normalisation property.

Keywords: classical logic, Curry-Howard correspondence, lambda calculus, resource control, erasure and duplication

Introduction

The fundamental connection between logic and computation, known as the Curry-Howard correspondence or *formulae-as-types, proofs-as-term* and *proofs-as-programs paradigm*, relates logical and computational systems.

Gentzen's natural deduction is a well established formalism for expressing proofs. Church's simply typed λ-calculus is a core formalism for writing programs. *Simply typed* λ-calculus represents a computational interpretation of *intuitionistic natural deduction*: formulae correspond to types, proofs to terms/programs and simplifying a proof corresponds to executing a program. In its traditional form, terms in the λ-calculus encode proofs in intuitionistic natural deduction; from another perspective the proofs serve as typing

Email addresses: gsilvia@uns.ac.rs (S. Ghilezan), pierre.lescanne@ens-lyon.fr (P. Lescanne), dragisa.zunic@gmail.com (D. Žunić)

derivations for the terms. This correspondence was discovered in the late 1950s and early 1960s independently in logic by Curry, later formulated by Howard; in category theory, Cartesian Closed Categories, by Lambek; and in mechanization of mathematics, the language Automath, by de Brujin.

Griffin extended the Curry-Howard correspondence to classical logic in his seminal 1990 paper [19], by observing that classical tautologies suggest typings for certain control operators. This initiated a vigorous line of research: on the one hand classical calculi can be seen as pure programming languages with explicit representations of control, while at the same time terms can be tools for extracting the constructive content of classical proofs. The $\lambda\mu$-calculus of Parigot [31] expresses the computational content of *classical natural deduction* and has been the basis of a number of investigations into the relationship between classical logic and theories of control in programming languages.

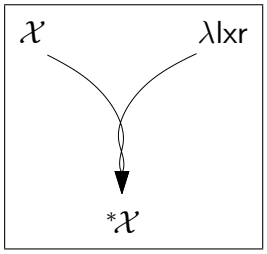
Computational interpretation of *sequent-style logical systems* has come into the picture much later, by the end of 1990s. There were several attempts, over the years, to design a term calculus which would embody the Curry-Howard correspondence for *intuitionistic sequent logic*. The first calculus accomplishing this task is Herbelin's $\bar{\lambda}$-calculus [20]. Recent interest in the Curry-Howard correspondence for intuitinistic sequent logic [20, 5, 13, 14] made it clear that the computational content of sequent derivations and cut-elimination can be expressed through an extension of the λ-calculus. In the classical setting, there are several term calculi based on *classical sequent logic*, in which terms unambiguously encode sequent derivations and reduction corresponds to cut elimination: Barbanera and Berardi's Symmetric Calculus [3], Curien-Herbelin's $\overline{\lambda}\mu\widetilde{\mu}$-calculus [7], Urban-Bierman's calculus [37], Wadler's Dual Calculus [45]. In contrast to natural deduction proof systems, sequent calculi exhibit inherent symmetries in proof structures which create technical difficulties in analyzing the reduction properties of these calculi [12, 11, 16].

The tutorial entitled "Computational interpretations of logics" given by the first author of this paper at ICTAC 2011 in Johannesburg, South Africa, presented a comprehensive overview and a comparison of computational interpretations of intuitionistic and classical logic both in natural deduction and sequent-style setting. In this paper our focus is on the computational interpretations of classical sequent calculus, with explicit structural rules of weakening and contraction.

${}^*\mathcal{X}$ has been designed to provide a correspondence 'a là' Curry-Howard for

the standard formulation of classical sequent calculus, with explicit structural rules (weakening and contraction). The direct correspondence between proofs and terms is achieved by using the technique of labeling formulas by names. These names are used to build terms so that the structure of a term captures the original structure of a corresponding proof. Furthermore, the computation of terms is defined in a way that mirrors the proof-transformation, that is, the cut-elimination.

The inspiration for ${}^*\mathcal{X}$ comes from two sources. On the one hand, the direct predecessor is the classical term language called $\mathcal{X}$. On the other hand, a very strong influence comes from the intuitionistic field and most notably the work on the λlxr-calculus.



In our study we try to respect the underlying principles of these works, and implement them in a way that preserves their good properties.

As a first contribution of this paper, we design ${}^*\mathcal{X}$, which represents the computational interpretation of classical sequent logic with explicit structural rules of contraction and weakening. Further, we propose a simply typed system for which we prove the witness reduction property. We relate the explicit and implicit treatment of structural rules by mutual encoding of ${}^*\mathcal{X}$ and $\mathcal{X}$. Finally, these results leads us to prove strong normalisation of simply typed ${}^*\mathcal{X}$.

Related work. The $\mathcal{X}$ calculus is a term language, introduced in [42] and studied in more detail in [43]. It is a low level language which can easily encode various other calculi and which captures the structure of classical proofs represented in the sequent calculus, espcially cut-elimination. Some of its properties are non-determinism, non-confluence and strong normalizationfor typed terms.

Some closely related computational interpretations have been presented earlier. First of them is the so-called local cut-elimination procedure presented in [38]. It is one of the three cut-elimination procedures studied in detail in [35]. A term assignment is given for proofs in the classical sequent

calculus (formulated with completely implicit structural rules). Then this term language was used as a tool to show the properties of classical sequent calculus. Most importantly, it enabled the authors to use the term-rewriting techniques in order to prove the strong normalization of cut-elimination in classical logic.

A second computational interpretation, very close to $\mathcal{X}$, has been presented by Lengrand in [28], under the name $\lambda\xi$-calculus. There it was studied in relation with $\bar{\lambda}\mu\tilde{\mu}$-calculus of [7], and it was used to infer the strong normalization for $\bar{\lambda}\mu\tilde{\mu}$.

Although there are differences these three formulations are very close. The syntaxes of $\lambda\xi$ and $\mathcal{X}$ are the same (there are minor differences such as the use of $\dagger_{\mathrm{V}}$ in the first, instead of $\not\prec$ in the second). Both the syntax and the reduction rules of $\lambda\xi$ are said to be (in [28]) the subsystems of Urban's local cut-elimination procedure ($\mathcal{T}^{\leftrightarrow}$, $\stackrel{loc}{\longrightarrow}$) (see [35]). However, some differences in the set of reductions exist.

Let us recall here the philosophy behind these calculi. Urban [35] was partly inspired by Danos et al. [9] who consider Gentzen's sequent calculus as a programming language. Their cut-elimination procedure is called LK^{tq}. It is strong normalizing, confluent and strongly connected to linear logic proof nets. Confluence is obtained by assigning color annotations to formulas, which restricts cut-reductions so that the critical pair does not arise. Confluence is essential in LK^{tq} because it enabled the authors to exploit the strong normalization result of proof nets in linear logic. However Urban reveals all the details of the complex classical cut-elimination, by developing a term-notation for proofs, whereas in LK^{tq} concepts are presented informally.

Moreover, it has been shown in [35, 39], using the results of [4], that not all normal forms are reachable using the LK^{tq} interpretation. Secondly, the restrictions introduced by using the colors are not needed to ensure strong normalization.

$\mathcal{X}$ departs from the traditional doctrine of intuitionistic logic, where computation is an equality preserving operation on proofs. Instead, $\mathcal{X}$ accepts that cut-elimination may or may not preserve proof equality, and that nondeterminism is a natural feature of classical logic.

Although mainly concerned with the computational content of classical logic, the ideas presented in this paper come partly from intuitionistic logic, primarily from the $\lambda\mathsf{lxr}$-calculus [22, 23] which had a significant influence. The $\lambda\mathsf{lxr}$-calculus extends $\lambda\mathsf{x}$ [6, 32] by operators for erasure and duplication

in the same way as ${}^*\mathcal{X}$ extends $\mathcal{X}$. The intuitionistic calculi, $\lambda\mathsf{x}$ and $\lambda\mathsf{lxr}$ are related as are $\mathcal{X}$ and ${}^*\mathcal{X}$.

The $\lambda\mathsf{lxr}$-calculus was created as an attempt to relate the two elementary decompositions, namely, the decomposition of intuitionistic connectives in linear logic, and the decomposition of a meta-level λ-calculus substitution. The meta-substitution can be decomposed into more atomic steps, represented within the language [1], thus bringing the theoretical work closer to the actual implementations. It has been shown in [22] that there exists a very strong relation between $\lambda\mathsf{lxr}$-calculus and linear logic proof-nets.

Some works have considered the relation between $\mathcal{X}$ and the π-calculus. The π-calculus, [30, 33], is able to describe concurrent computations, including the communication between processes. The configurations of the interacting processes may change during the computation. The relation of $\mathcal{X}$ and π-calculus has been recently presented in [41], where the $\mathcal{X}$ calculus is encoded into π. This paper seeks for the intuition to what is computational meaning of cut-elimination from the point of view of π.

Some remarks aiming at essential points related to concurrency were given earlier by Urban [35]. He suggested how a form of weak communication can be implemented, using quantifiers, into the classical sequent calculus. Besides that, it was noted that the approach where reduction is not seen as an equality preserving operation, is a standard approach in the calculi of concurrency. Moreover, the substitution mechanism in $\mathcal{X}$-like calculi in which only names may participate, is closer to the π-calculus than the substitution mechanism defined in the λ-calculus which involves terms.

Outline of the paper. Section 1 is a brief overview of the sequent style classical logical systems. Section 2 deals with $\mathcal{X}$ calculus: its syntax, reduction rules, types systems and basic properties. In Section 3 we propose the syntax and operational semantics of ${}^*\mathcal{X}$ as well as the simply typed system. Section 4 provides the relation between the two calculi.

Contents

1 Sequent calculi $G1$ and $G3$ **6**

2 The $\mathcal{X}$ calculus **8**

2.1 The syntax 8

2.2 The computation 9

2.3 The type system 11
2.4 Basic properties 12

3 Erasure and duplication: the $^*\mathcal{X}$ calculus 12
3.1 The syntax 12
3.2 Reduction rules 19
3.3 Operational properties 26
3.4 The type assignment system 27

4 Explicit vs. implicit: relation between $^*\mathcal{X}$ and $\mathcal{X}$ 29
4.1 From $\mathcal{X}$ to $^*\mathcal{X}$ 30
4.2 From $^*\mathcal{X}$ to $\mathcal{X}$ 37
4.3 Strong normalisation of $^*\mathcal{X}$ 41

5 Conclusions 42

1. Sequent calculi *G*1 and *G*3

The basic Genzen systems for classical and intuitionistic logic denoted as $G1, G2$ and $G3$ are formalized in [26] and later revisited in [34]. In brief, the essential difference between $G1$ and $G3$ is the presence or absence of explicit structural rules. The distinguishing point in the case of $G2$ is the use of the so-called mix instead of a cut rule. Although here we focus on the classical systems, we remark that the intuitionistic systems are obtained from classical ones by restricting sequents to having only one formula in the succedent.

The system $G1$. Among the three systems presented by Kleene [26], $G1$ is the closest to Gentzen's original formulation [15]. Despite the fact that Gentzen and Kleene present explicitly exchange rules, which is not the case here, we keep the name $G1$ (Figure 1). Latin symbols $A, B, ...$ are used to denote formulas and Greek symbols $\Gamma, \Delta, \Gamma', \Delta', ...$ to denote contexts, which are in this framework multisets of formulas. Exchange rules are handled by multisets instead of lists, whereas the other structural rules, namely *weakening* and *contraction* are explicitly given. The axiom rules do not involve arbitrary contexts. Inference rules with two premises, namely $(L \rightarrow)$ and *(cut)*, are given in the context-splitting style, which means that when looking bottom-up the contexts of a conclusion is split by premises. It has been shown in [34] that if a context-sharing style was applied one obtains an equivalent system, i. e., a system that proves the same sequents.

$$\frac{}{A \vdash A}\ (ax)$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \to B \vdash \Delta, \Delta'}\ (L\to) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\ (R\to)$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ (cut)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ (weak\text{-}L) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ (weak\text{-}R)$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ (cont\text{-}L) \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ (cont\text{-}R)$$

Figure 1: Sequent system $G1$

The system $G3$. The sequent system $G3$ is obtained from $G1$ by making all structural rules parts of the remaining rules with appropriate forms. In other words, there is no explicit structural rules. Instead structural rules are hidden in the new presentation of the logical rules and of the cut-rule, and thus performed automatically.

This system has been mentioned as $G3a$ in [26] and formalized as classical $G3$ in [34]. It is presented by Figure 2, where $A, B, ...$ range over formulas, while contexts $\Gamma, \Delta, ...$ are finite *sets* of formulas.

$$\frac{}{\Gamma, A \vdash A, \Delta}\ (ax)$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\ (L\to) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\ (R\to)$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\ (cut)$$

Figure 2: Sequent system $G3$

Inference rules with two premises are given in the context-sharing style. The definition of the axiom rule involves contexts, thus allowing arbitrary formulas to be introduced at that level, i.e., weakening rule is hidden in the form of the axiom.

2. The $\mathcal{X}$ calculus

This section presents $\mathcal{X}$ which is, together with $\lambda\mathsf{lxr}$, a predecessor of ${}^*\mathcal{X}$. The design of ${}^*\mathcal{X}$ has been directly inspired by $\mathcal{X}$.

$\mathcal{X}$ was first presented in van Bakel, Lescanne and Lengrand in [42]. The origin of the language is in the notations for classical sequent proofs by Urban [35], introduced as a tool to express the cut-elimination procedure as a term rewriting system, which later allowed him to prove strong normalization of cut-elimination. A close variant of the language has been studied by Lengrand in relation with the $\bar{\lambda}\mu\tilde{\mu}$-calculus, in a calculus he called $\lambda\xi$ [28].

It is argued in [35] that non-determinism, although it leads to non-confluence, should be considered as an intrinsic property of classical logic. This point of view was taken in some earlier works, for example [3, 20, 4] and more recently in [17, 21]. This means that, in classical logic, we depart from the traditional intuitionistic (and linear) logic doctrine, where cut-elimination is an equality preserving operation on proofs.

2.1. The syntax

The $\mathcal{X}$ calculus corresponds to a sequent system with implicit structural rules (Figure 2. Since we consider only the implicative fragment, the only inference rules are axiom, cut, left-arrow introduction and right-arrow introduction. Therefore, in the $\mathcal{X}$ calculus there are four constructors (see Figure 3).

P, Q	$::=$	$\langle x.\alpha \rangle$	*capsule*
	$\mid$	$\widehat{x}\, P\, \widehat{\beta} \cdot \alpha$	*exporter*
	$\mid$	$P\, \widehat{\alpha}\, [x]\, \widehat{y}\, Q$	*importer*
	$\mid$	$P \widehat{\alpha} \dagger \widehat{x} Q$	*cut*

Figure 3: The syntax of $\mathcal{X}$

The term *capsule* corresponds to an axiom rule, *cut* corresponds to a cut-rule, *importer* corresponds to left-arrow introduction rule and *exporter*

corresponds to right-arrow introduction rule.[1] The syntax is then extended by two *active cuts* that reflect the non-deterministic *choice* which exists in the sequent calculus.

$$\begin{array}{rcll} P, Q & ::= & \cdots & \\ & | & P\widehat{\alpha} \nearrow \widehat{x}Q & \textit{left-active cut} \\ & | & P\widehat{\alpha} \nwarrow \widehat{x}Q & \textit{right-active cut} \end{array}$$

2.2. The computation

There are 20 reduction rules in $\mathcal{X}$ which correspond to cut-elimination in the sequent calculus and which are split into logical, activation and propagation clusters and not grouped like [42, 43]. There are named to ease the comparison with ${}^*\mathcal{X}$-rules.

Logical rules

Logical rules say how to eliminate a cut. They apply when the cut refers to two names which are *freshly introduced*.

Definition 1 (Fresh introduction).

- *The term P freshly introduces x if $P = \langle x.\alpha \rangle$ or $P = Q\,\widehat{\alpha}\,[x]\,\widehat{y}\,R$, with $x \notin N(Q), x \notin N(R)$.*
- *The term P freshly introduces α if $P = \langle x.\alpha \rangle$ or $P = \widehat{x}\,Q\,\widehat{\beta} \cdot \alpha$, with $\alpha \notin N(Q)$.*

Informally, names are freshly introducesd if they appear once and only once, at the top level of their corresponding terms.[2] The cut in this position can not be activated. Logical rules are shown by Figure 4.
The first two rules are *renaming*. The last rule, called *insertion*, defines an interaction between an importer and an exporter. It inserts an immediate subterm of an exporter between two immediate subterms of an importer.

[1] In the original papers importer and exporter were called import and mediator.

[2] This is more complex than in ${}^*\mathcal{X}$, where the linearity condition guarantees that if a name occurs at the top level, then it does not occur elsewhere.

$$
\begin{array}{lcll}
(ren\text{-}R) & : & P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle & \to \quad P\{\beta/\alpha\} \\
(ren\text{-}L) & : & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}Q & \to \quad Q\{y/x\} \\
(\widehat{y}\,P\,\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\,\widehat{\gamma}\;[x]\;\widehat{z}\,R) & \to & \multicolumn{2}{l}{either \begin{cases} (Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R \\ Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R) \end{cases}} \\
 & & \multicolumn{2}{l}{\alpha \notin N(P),\ x \notin N(Q),\ x \notin N(R)}
\end{array}
$$

Figure 4: Logical rules in $\mathcal{X}$

Activation rules

Activation rules describe the non-determinism of classical cut-elimination. If a cut refers to a name which is not freshly introduced, one has to propagate it according to a chosen direction and activation is then followed by propagation rules (see Figures 6 and 7). This choice has usually been bypassed in the previous interpretations, either by restricting the reduction procedure (a very common one is to not allow cuts to pass over other active cuts), or by giving priority to a specific strategy (like in [8], by assigning colors to formulas). Notice that the cut can be activated in one or the other direction when both conditions are fulfilled at the same time, as shown by Figure 5. This is a source of non-confluence.

$$
\begin{array}{lcl}
(act\text{-}L) & : & P\widehat{\alpha} \dagger \widehat{x}Q \ \to P\widehat{\alpha} \nearrow \widehat{x}Q, \text{ if } \alpha \text{ not freshly introduced by } P \\
(act\text{-}R) & : & P\widehat{\alpha} \dagger \widehat{x}Q \ \to P\widehat{\alpha} \nwarrow \widehat{x}Q, \text{ if } x \text{ not freshly introduced by } Q
\end{array}
$$

Figure 5: Activation rules in $\mathcal{X}$

Propagation rules

Left and *right propagation* rules are given in Figures 6 and 7, respectively. These rules describe how a cut is pushed through a term, but also address situations where deactivation, erasure and duplication occur. This means that in $\mathcal{X}$, several actions can be defined by a single reduction rule. Take for example the rule from Figure 7 which involves propagation, duplication and deactivation:

($\nwarrow$-*prop-dupl-deact*) :

$$P\widehat{\alpha} \nwarrow \widehat{x}(Q\,\widehat{\beta}\;[x]\;\widehat{z}\,R) \to P\widehat{\alpha} \dagger \widehat{x}((P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\;[x]\;\widehat{z}\,(P\widehat{\alpha} \nwarrow \widehat{x}R))$$

The rule labelled ($\nearrow$-(*c*)-*prop-deact*) describes a subtle propagation over a capsule whose both names are bound by cuts (nested cuts with an axiom). This rule is introduced to prevent possible infinite reductions of syntactic nature [9, 35]. Rules ($\nearrow$-*gc*) and ($\nwarrow$-*gc*) collect garbage.

($\nearrow$-*eras*)	:	$\langle x.\alpha\rangle\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $\langle x.\alpha\rangle$,	$\alpha \neq \beta$
($\nearrow$-*deact*)	:	$\langle x.\beta\rangle\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}R$	
($\nearrow$-*prop*)	:	$(\widehat{x}\,P\,\widehat{\gamma}\cdot\alpha)\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $\widehat{x}\,(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\gamma}\cdot\alpha$,	$\alpha \neq \beta$
($\nearrow$-*prop-dupl-deact*)	:	$(\widehat{x}\,P\,\widehat{\gamma}\cdot\beta)\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $(\widehat{x}\,(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\gamma}\cdot\beta)\widehat{\beta} \dagger \widehat{y}R$	
($\nearrow$-*prop-dupl*$_1$)	:	$(P\,\widehat{\alpha}\;[x]\;\widehat{z}Q)\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\alpha}\;[x]\;\widehat{z}\,(Q\widehat{\beta} \nearrow \widehat{y}R)$	
($\nearrow$-(*c*)-*prop-deact*)	:	$(P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle)\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $(P\widehat{\beta} \nearrow \widehat{y}R)\widehat{\alpha} \dagger \widehat{y}R$	
($\nearrow$-*prop-dupl*$_2$)	:	$(P\widehat{\alpha} \dagger \widehat{x}Q)\widehat{\beta} \nearrow \widehat{y}R$	$\to$ $(P\widehat{\beta} \nearrow \widehat{y}R)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\beta} \nearrow \widehat{y}R)$,	$Q \neq \langle x.\beta\rangle$
($\nearrow$-*gc*)	:	$P\widehat{\alpha} \nearrow \widehat{x}Q$	$\to$ P, if $\alpha \notin N(P)$	

Figure 6: Left propagation (erasure/duplication/deactivation) in $\mathcal{X}$

($\nwarrow$-*eras*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}\langle y.\beta\rangle$	$\to$ $\langle y.\beta\rangle$,	$x \neq y$
($\nwarrow$-*deact*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}\langle x.\beta\rangle$	$\to$ $P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle$	
($\nwarrow$-*prop*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(\widehat{y}\,Q\,\widehat{\gamma}\cdot\beta)$	$\to$ $\widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\gamma}\cdot\beta$	
($\nwarrow$-*prop-dupl*$_1$)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(Q\,\widehat{\beta}\;[y]\;\widehat{z}\,R)$	$\to$ $(P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\;[y]\;\widehat{z}\,(P\widehat{\alpha} \nwarrow \widehat{x}R)$,	$x \neq y$
($\nwarrow$-*prop-dupl-deact*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(Q\,\widehat{\beta}\;[x]\;\widehat{z}\,R)$	$\to$ $P\widehat{\alpha} \dagger \widehat{x}((P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\;[x]\;\widehat{z}\,(P\widehat{\alpha} \nwarrow \widehat{x}R))$	
($\nwarrow$-(*c*)-*prop-deact*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}R)$	$\to$ $P\widehat{\alpha} \dagger \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R)$	
($\nwarrow$-*prop-dupl*$_2$)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R)$	$\to$ $(P\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R)$,	$Q \neq \langle x.\beta\rangle$
($\nwarrow$-*gc*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}Q$	$\to$ Q, if $x \notin N(Q)$	

Figure 7: Right propagation (erasure/duplication/deactivation) in $\mathcal{X}$

2.3. The type system

The type assignment system for the $\mathcal{X}$ calculus is given by Figure 8.

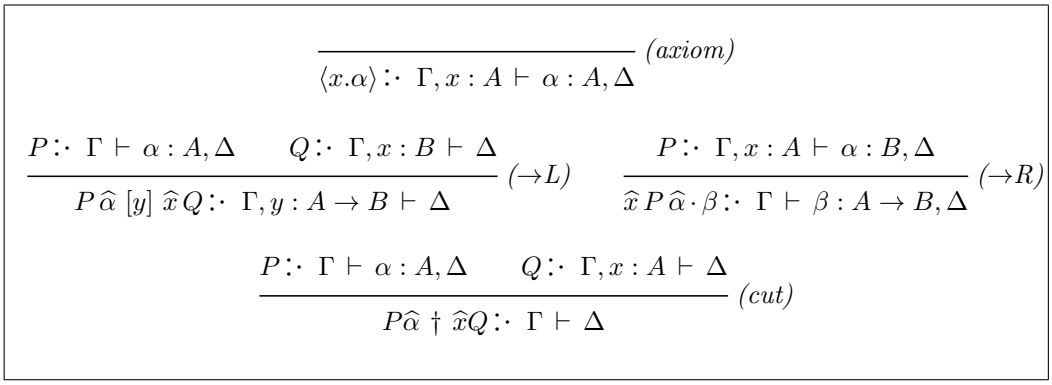


Figure 8: $\mathcal{X}$ type system

2.4. Basic properties

It has been shown in [39] that the computation in $\mathcal{X}$ calculus can be seen as proof-transformation (subject reduction property) and that $\mathcal{X}$, although intrinsically non-deterministic and non confluent, which are indeed properties of classical cut-elimination, is strongly normalising.

3. Erasure and duplication: the ${}^*\mathcal{X}$ calculus

This section presents the rules of untyped ${}^*\mathcal{X}$, followed by the basic operational properties and the definition of typing rules. Although it is presented here as a counterpart of the implicative segment of the sequent calculus for classical logic, the system can naturally be extended to encompass other connectives as well [44].

3.1. The syntax

Names differ essentially from variables in λ-calculus. The difference lies in the fact that a variable can be substituted by an arbitrary term, while a name can only be *renamed* (that is, substituted by another name). In ${}^*\mathcal{X}$ the renaming is explicit, which means that it is expressed within the language itself and is not defined in the meta-theory. The reader will notice the presence of hats on some names. This notation has been borrowed from *Principia Mathematica* [46] and is used to denote name binding.

Free names, bound names and ${}^*\mathcal{X}$-terms

Names can be free or bound. They are defined together with the set of ${}^*\mathcal{X}$-terms also called linear terms.

Linearity In ${}^*\mathcal{X}$, we consider only *linear* terms, which means:
– Every name has at most one free occurrence, and
– Every binder does bind an actual occurrence of a name (and thus only one)

Definition 2 (Free Names and ${}^*\mathcal{X}$- terms). The sets of *free innames* and *free outnames* and the set of (well-formed) ${}^*\mathcal{X}$*-terms* are defined mutually recursively in Figure 9 and Figure 10.

By "mutually recursive" we mean that the definition of an ${}^*\mathcal{X}$-term needs the definition of the free names of its subterms and the definition of the free names supposes that the structure of the subterms is known. We write $N(P)$ the *set* of free names of P, $I(P)$ the *sets* of free innames of P, and $O(P)$ the set of free outnames of P. Thus $N(P) = I(P) \cup O(P)$. A name which occurs in P and which is not free is called a *bound name*. Notice that a construction can bind two names. This can be either two innames, two outnames or an inname and an outname. Moreover, these names sometimes belong to different subterms, as in the case of an importer or a cut. To denote the binding of all names in one list, we use simply $\widehat{\mathcal{I}}$, $\widehat{\mathcal{O}}$. It is sometimes needed to use $\overline{I}(P)$ instead of $\mathcal{I}^P$, and similarly for outnames $\overline{O}(P)$, and names in general $\overline{N}(P)$. The bar is used to denote that we see the given set of names as a list, according to the total order which can be defined for the set of names. To exclude a name from a list, for instance, outname α, we write $\mathcal{O}^P \backslash \alpha$.

Renaming We define the operation $P\{x/y\}$ which denotes the *renaming* of a free name y in P by a fresh name x. It is a meta-operation which replaces a unique occurrence of a free name by another free name. Therefore it is simpler than the meta-substitution of λ-calculus, which denotes the substitution of a free variable (which can occur arbitrary number of times) by an arbitrary term.

Indexing We introduce a special kind of renaming, called *indexing*, in order to simplify the syntax of the reduction rules. For example $P_i = ind(P, N(P), i)$ means that P_i is obtained by indexing free names in P by index i, where $i \in N$. Simple notation P_i for cases such as this one will be used when

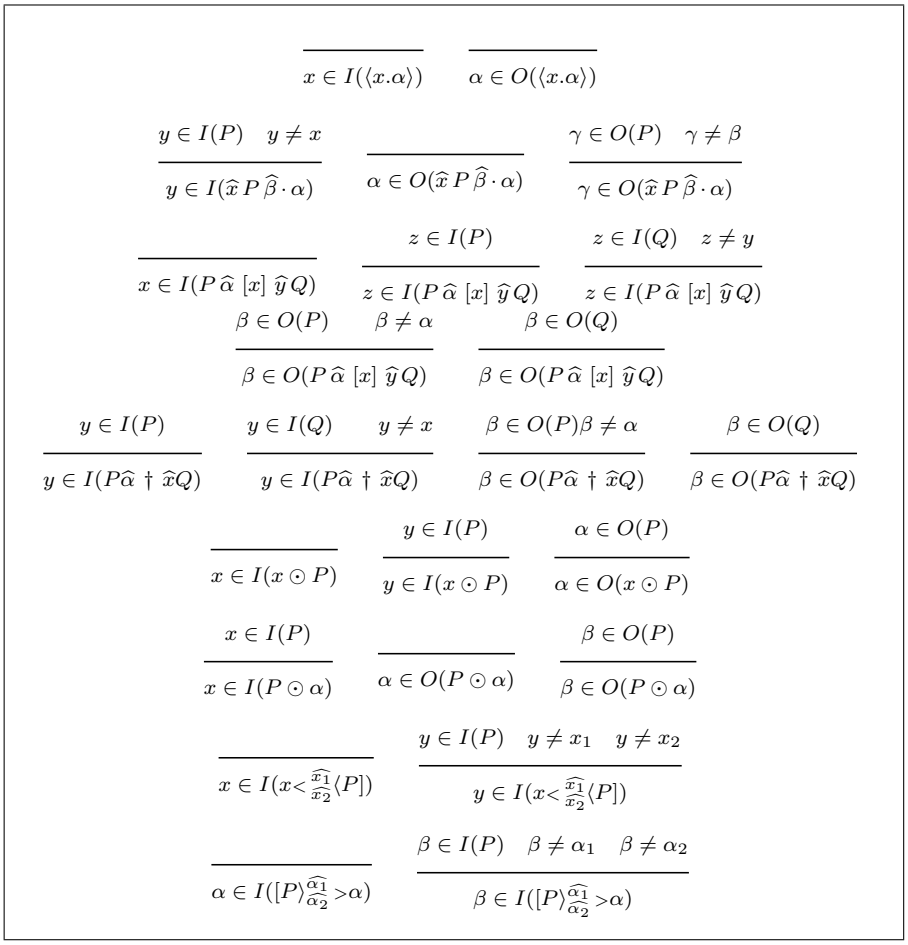


Figure 9: Free names in ${}^*\mathcal{X}$

possible. We assume that indexing always creates fresh names. As we use it indexing preserves linearity.

Modules A module is a part of a term (not a subterm) of the form $\widehat{\alpha} \nearrow \widehat{x}Q$ (*left-module*) and $P\widehat{\beta} \nwarrow \widehat{y}$ (*right-module*) which percolates through the structure of that term (and its subterms) during the computation, as specified by the so-called "propagation rules". It resembles the explicit substitution. We say that α and y are the *handles* of $\widehat{\alpha} \nearrow \widehat{x}Q$ and $P\widehat{\beta} \nwarrow \widehat{y}$, respectively. Two *modules are independent* if the handle of one module does not bind a free

$$\frac{}{\langle x.\alpha\rangle \ \text{wf } {}^*\mathcal{X}\text{-term}}$$

$$\frac{P \ \text{wf } {}^*\mathcal{X}\text{-term},\ x,\beta \in N(P),\ \alpha \notin N(P)}{\widehat{x}\,P\,\widehat{\beta}\cdot\alpha \ \text{wf } {}^*\mathcal{X}\text{-term}}$$

$$\frac{P,Q \ \text{wf } {}^*\mathcal{X}\text{-term},\ \alpha \in N(P),\ x \in N(Q),\ y \notin N(P,Q),\ N(P)\cap N(Q)=\emptyset}{P\,\widehat{\alpha}\,[y]\,\widehat{x}\,Q \ \text{wf } {}^*\mathcal{X}\text{-term}}$$

$$\frac{P,Q \ \text{wf } {}^*\mathcal{X}\text{-term},\ \alpha \in N(P),\ x \in N(Q),\ N(P)\cap N(Q)=\emptyset}{P\widehat{\alpha}\,\dagger\,\widehat{x}Q \ \text{wf } {}^*\mathcal{X}\text{-term}}$$

$$\frac{P \ \text{wf } {}^*\mathcal{X}\text{-term},\ x \notin N(P)}{x \odot P \ \text{wf } {}^*\mathcal{X}\text{-term}} \qquad \frac{P \ \text{wf } {}^*\mathcal{X}\text{-term},\ \alpha \notin N(P)}{P \odot \alpha \ \text{wf } {}^*\mathcal{X}\text{-term}}$$

$$\frac{P \ \text{wf } {}^*\mathcal{X}\text{-term},\ x,y \in N(P),\ z \notin N(P)}{z{<}^{\widehat{x}}_{\widehat{y}}\langle P] \ \text{wf } {}^*\mathcal{X}\text{-term}} \qquad \frac{P \ \text{wf } {}^*\mathcal{X}\text{-term},\ \alpha,\beta \in N(P),\ \gamma \notin N(P)}{[P\rangle^{\widehat{\alpha}}_{\widehat{\beta}}{>}\gamma \ \text{wf } {}^*\mathcal{X}\text{-term}}$$

Figure 10: ${}^*\mathcal{X}$ terms

name inside the other module, and vice-versa, as follows:

independent modules	conditions
$\widehat{\alpha} \nearrow \widehat{x}Q,\ \widehat{\beta} \nearrow \widehat{y}R$	$\alpha \notin N(R),\ \beta \notin N(Q)$
$P\widehat{\alpha} \nwarrow \widehat{x},\ \ Q\widehat{\beta} \nwarrow \widehat{y}$	$x \notin N(Q),\ y \notin N(P)$
$P\widehat{\alpha} \nwarrow \widehat{x},\ \ \widehat{\beta} \nearrow \widehat{y}R$	$x \notin N(R),\ \beta \notin N(P)$

Convention on names We adopt a convention on names: "a name is never both bound and free in the same term". Terms are defined up to α-conversion, that is, the renaming of bound names does not change them.

Every $\mathcal{X}$-term can be translated into an ${}^*\mathcal{X}$-term, using duplicators and erasers. For instance, $\langle x.\alpha\rangle \odot \alpha$, which has two free occurrences of α, can be represented in ${}^*\mathcal{X}$ by the term $[\langle x.\alpha_1\rangle \odot \alpha_2\rangle^{\widehat{a_1}}_{\widehat{\alpha_2}}{>}\alpha$ (notice the role of a duplicator). The term $\widehat{x}\,\langle x.\alpha\rangle\,\widehat{\beta}\cdot\gamma$ binds no free name and corresponds to the ${}^*\mathcal{X}$-term $\widehat{x}\,(\langle x.\alpha\rangle \odot \beta)\,\widehat{\beta}\cdot\gamma$ (notice the role of an eraser).

Definition 3 (Principal names). The following tables define te so-called *principal names.*

a term	L-princip. names
$\langle x.\alpha\rangle$	x, α
$\widehat{x}\, P\, \widehat{\beta}\cdot\alpha$	α
$P\,\widehat{\alpha}\,[x]\,\widehat{y}\,Q$	x
$P\widehat{\alpha}\,\dagger\,\widehat{x}Q$	none

a term	S-princip. names
$x \odot P$	x
$P \odot \alpha$	α
$x{<}^{\widehat{x_1}}_{\widehat{x_2}}\langle P]$	x
$[P\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha$	α

We say that a name is *principal* if it is either L-principal (introduced by a logical term) or S-principal (introduced by a structural term).

Lemma 4. *Every term has at least a free logical outname.*

Proof:. The proof goes by routine induction on the structure of terms.[3] □.

Definition 5 (Contexts). *Contexts* are formally defined as follows:

$C\{\ \}$	::=	$\{\ \}$	|	$\widehat{x}\,\{\ \}\,\widehat{\beta}\cdot\alpha$
	|	$\{\ \}\,\widehat{\alpha}\,[x]\,\widehat{y}\,Q$	|	$P\,\widehat{\alpha}\,[x]\,\widehat{y}\,\{\ \}$
	|	$\{\ \}\widehat{\alpha}\,\dagger\,\widehat{x}Q$	|	$P\widehat{\alpha}\,\dagger\,\widehat{x}\{\ \}$
	|	$x \odot \{\ \}$	|	$\{\ \} \odot \alpha$
	|	$z{<}^{\widehat{x}}_{\widehat{y}}\langle\{\ \}]$	|	$[\{\ \}\rangle^{\widehat{\alpha}}_{\widehat{\beta}}{>}\gamma$
	|	$C\{C\{\ \}\}$		

Remark 6. A context is a term with a hole in which another term can be placed. Therefore $C\{P\}$ denotes placing the term P in the context $C\{\ \}$.

Remark 7. We use $P = Q$ to denote that the terms P and Q are syntactically equal.

Definition 8 (Subterm relation $\preccurlyeq$). A term Q is a *subterm* of a term P, denoted as $Q \preccurlyeq P$ if there is a context $C\{\ \}$ such that $P = C\{Q\}$.

[3]This would no longer be true if we were to extend the system with negation, for details see [44].

Lemma 9. *The* subterm *relation is reflexive, antisymmetric and transitive (i.e., is an order):*

1. Reflexivity $P \preccurlyeq P$
2. Antisymmetry *If* $P \preccurlyeq Q$ *and* $Q \preccurlyeq P$ *then* $P = Q$
3. Transitivity*: If* $P \preccurlyeq Q$ *and* $Q \preccurlyeq R$ *then* $P \preccurlyeq R$

Proof:.

1. The first point is straightforward. If $P \preccurlyeq P$, then by the subterm definition we have $\exists\, C\{\ \}$ such that $P = C\{P\}$. This stands if we choose $C\{\ \}$ to be $\{\ \}$.
2. Let $P \preccurlyeq Q$ and $Q \preccurlyeq R$. By definition $\exists\, C'\{\ \}, C''\{\ \}$ such that $C'\{P\} = Q$ and $C''\{Q\} = P$. From $C'\{C''\{Q\}\} = Q$ we derive $C'\{\ \} = C''\{\ \} = \{\ \}$. Finally we can conclude $P = Q$.
3. On the one hand, from $P \preccurlyeq Q$ by definition we have: $\exists\, C'\{\ \}$ such that $C'\{P\} = Q$. On the other hand, from $Q \preccurlyeq R$ by definition we have: $\exists\, C''\{\ \}$ such that $C''\{Q\} = R$. Thus, $C''\{C'\{P\}\} = R$ and therefore by definition we have $P \preccurlyeq R$. □

The following definition introduces the notion of a simple context, i.e., a context which is not composed of other contexts. Notice that it resembles the definition of context, except that the cases $\{\ \}$ and $C\{C\{\ \}\}$ are omitted.

Definition 10 (Simple context). A context $C\{\ \}$ is said to be *simple* if $C\{\ \}$ is one of the following:

$C\{\ \}$	:=	$\widehat{x}\{\ \}\widehat{\beta}\cdot\alpha$	|	
	|	$\{\ \}\widehat{\alpha}\ [x]\ \widehat{y}\,Q$	|	$P\,\widehat{\alpha}\ [x]\ \widehat{y}\{\ \}$
	|	$\{\ \}\widehat{\alpha}\ \dagger\ \widehat{x}Q$	|	$P\widehat{\alpha}\ \dagger\ \widehat{x}\{\ \}$
	|	$x \odot \{\ \}$	|	$\{\ \} \odot \alpha$
	|	$z{<}\frac{\widehat{x}}{\widehat{y}}\langle\{\ \}]$	|	$[\{\ \}\rangle_{\widehat{\beta}}^{\widehat{\alpha}}{>}\gamma$

Using the definition of a simple context we formulate the notion of immediate subterm.

Definition 11 (Immediate subterm). A term Q is an *immediate subterm* of P if $P = C\{Q\}$ and $C\{\ \}$ is a simple context.

Example 12. A term can have either one, two, or zero immediate subterms. For example, $Q\,\widehat{\alpha}\,[x]\,\widehat{y}\,R$ has two immediate subterms (these are P and Q), $\widehat{x}\,Q\,\widehat{\beta}\cdot\alpha$ has one (a term P), whilst $\langle x.\alpha\rangle$ has zero immediate subterms.

Using the definition of a context $C\{\ \}$, we specify the notion of a *context with two holes*.

Definition 13 (Context with two holes).

$$\begin{array}{|rcl|rcl|}\hline C\{\ ,\ \} & ::= & \{\ \}\,\widehat{\alpha}\,[x]\,\widehat{y}\{\ \} & | & \{\ \}\widehat{\alpha}\,\dagger\,\widehat{x}\{\ \} \\ & | & C\{C\{\ \}, C\{\ \}\} & | & C\{C\{\ ,\ \}\} \\ \hline\end{array}$$

Definition 14 (Simple context with two holes).

$$\begin{array}{|rcl|rcl|}\hline C\{\ ,\ \} & ::= & \{\ \}\,\widehat{\alpha}\,[x]\,\widehat{y}\{\ \} & | & \{\ \}\widehat{\alpha}\,\dagger\,\widehat{x}\{\ \} \\ \hline\end{array}$$

Using this definition, the notion of *immediate subterm* can be naturally extended as to encompass the cases when we speak about *two immediate subterms*.

Remark 15. We allow the use of of $P = P^{\alpha}\{R\}$to denote that the term P has α as principal name and R as an immediate subterm. Similarly for $P = P^{x}\{R_1,\, R_2\}$

Lemma 16. *The following holds:*

1. *If $\alpha \in N(P)$ then there exists a unique term $Q \preccurlyeq P$ such that α is a principal name for Q.*
2. *If $x \in N(P)$ then there exists a unique term $R \preccurlyeq P$ such that x is a principal name for R.*

Remark 17. We will use the notation Q^{α} to specify that Q has α as a principal name. Similarly, we use R^{x} to emphasize that R has x as a principal name.

Proof:. We prove the first point. The proof goes by induction on the structure of a term P and case analysis.

Let $\alpha \in P$.

- Case: α is a principal name for P. Then $Q = P$.
- Case: α is not a principal name for P. Then, either $P = C\{R\}$ or $P = C\{R_1, R_2\}$, where R, R_1 and R_2 denote immediate subterms of P.
  - $P = C\{R\}$. By induction hypothesis, and since by linearity $\alpha \in R$, we have: $\exists Q \preccurlyeq R$ such that α is a principal name for Q. By using transitivity (lemma 9), from $Q \preccurlyeq R$ and $R \preccurlyeq P$ we infer $Q \preccurlyeq P$.
  - $P = C\{R_1, R_2\}$. By the linearity condition we know that α belongs to either $N(R_1)$ or $N(R_2)$ (not to both). Thus we have two subcases, which correspond to the previous case. In the first case $C\{R_1, R_2\}$ is seen as $C'\{R_1\}$, where $C'\{\ \} = C\{\{\ \}, R_2\}$, and in the second case as $C''\{R_2\}$, where $C''\{\ \} = C\{R_1, \{\ \}\}$. Recall that R_1, R_2 are immediate subterms of P by definition.

The second point of the lemma refers to innames instead of outnames, and the proof goes similarly. □

Abbreviations. We introduce some abbreviations in order to represent reduction rules in a convenient form.

instead of	we write
$x_1 \odot (...(x_n \odot P)...)$	$x_1 \odot \ ...x_n \odot P$
$(...(P \odot \alpha_1)\ ...) \odot \alpha_n$	$P \odot \alpha_1\ ...\ \odot \alpha_n$

instead of	we write
$x_1 < \widehat{\frac{y_1}{z_1}} \langle ... x_n < \widehat{\frac{y_n}{z_n}} \langle P \rangle ... \rangle$	$(x_1, ...,\ x_n) < \widehat{\frac{(y_1, ..., y_n)}{(z_1, ..., z_n)}} \langle P \rangle$
$[...[P\rangle \widehat{\frac{\beta_1}{\gamma_1}} > \alpha_1 ...\rangle \widehat{\frac{\beta_n}{\gamma_n}} > \alpha_n$	$[P\rangle \widehat{\frac{(\beta_1, ..., \beta_n)}{(\gamma_1, ..., \gamma_n)}} > (\alpha_1, ..., \alpha_n)$

3.2. Reduction rules

In this section we define the reduction relation, $\rightarrow$. The set of reduction rules is rather large as it captures classical cut-elimination.

Reduction rules are grouped into

1. Activation rules (left and right)
2. Structural actions (left and right)
3. Deactivation rules (left and right)
4. Logical actions
5. Propagation rules (left and right)

Congruence rules. We assume some simple congruence rules which originate from the sequent calculus.

Commuting names in a duplicator

$$\boxed{\begin{array}{rcl} x<\frac{\widehat{x_1}}{\widehat{x_2}}\langle P] & \equiv & x<\frac{\widehat{x_2}}{\widehat{x_1}}\langle P] \\ [P\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\alpha & \equiv & [P\rangle\frac{\widehat{\alpha_2}}{\widehat{\alpha_1}}>\alpha \end{array}}$$

Permuting independent duplicators

$$\boxed{\begin{array}{rcl} x<\frac{\widehat{x_1}}{\widehat{x_2}}\langle y<\frac{\widehat{y_1}}{\widehat{y_2}}\langle P]] & \equiv & y<\frac{\widehat{y_1}}{\widehat{y_2}}\langle x<\frac{\widehat{x_1}}{\widehat{x_2}}\langle P]] \\ [[P\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\alpha\rangle\frac{\widehat{\beta_1}}{\widehat{\beta_2}}>\beta & \equiv & [[P\rangle\frac{\widehat{\beta_1}}{\widehat{\beta_2}}>\beta\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\alpha \\ [x<\frac{\widehat{x_1}}{\widehat{x_2}}\langle P]\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\alpha & \equiv & x<\frac{\widehat{x_1}}{\widehat{x_2}}\langle [P\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\alpha] \end{array}}$$

The conditions in the first rule treating the duplicators are $y \notin \{x_1, x_2\}$ and $x \notin \{y_1, y_2\}$ and in the second $\beta \notin \{\alpha_1, \alpha_2\}$ and $\alpha \notin \{\beta_1, \beta_2\}$. The third rule allows us to drop parenthesis and use a simplified notation

$$x<\frac{\widehat{x_1}}{\widehat{x_2}}\langle P\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\alpha \qquad \text{and more generally} \qquad \mathcal{I}<\frac{\widehat{\mathcal{I}_1}}{\widehat{\mathcal{I}_2}}\langle P\rangle\frac{\widehat{\mathcal{O}_1}}{\widehat{\mathcal{O}_2}}>\mathcal{O}$$

where $\mathcal{I}$ and $\mathcal{O}$ are lists of names. When $\mathcal{I} = ()$, we write $[P\rangle\frac{\widehat{\mathcal{O}_1}}{\widehat{\mathcal{O}_2}}>\mathcal{O}$. The case $\mathcal{O} = ()$ is not possible as stated by Lemma 4.

When the names are triplicated, one can do it in any order:

$$\boxed{\begin{array}{rcl} z<\frac{\widehat{y}}{\widehat{x_3}}\langle y<\frac{\widehat{x_1}}{\widehat{x_2}}\langle P]] & \equiv & z<\frac{\widehat{x_1}}{\widehat{y}}\langle y<\frac{\widehat{x_2}}{\widehat{x_3}}\langle P]] \\ [[P\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}>\beta\rangle\frac{\widehat{\beta}}{\widehat{\alpha_3}}>\gamma & \equiv & [[P\rangle\frac{\widehat{\alpha_2}}{\widehat{\alpha_3}}>\beta\rangle\frac{\widehat{\alpha_1}}{\widehat{\beta}}>\gamma \end{array}}$$

This can be seen as an associativity of names bound by a ternary duplicator.
Permuting the erasers: The following rule suggests that we may drop parenthesis and write $x \odot P \odot \alpha$, and more generally we may write: $\mathcal{I} \odot P \odot \mathcal{O}$.

$$\boxed{\begin{array}{rcl} y \odot x \odot P & \equiv & x \odot y \odot P \\ P \odot \alpha \odot \beta & \equiv & P \odot \beta \odot \alpha \\ (x \odot P) \odot \alpha & \equiv & x \odot (P \odot \alpha) \end{array}}$$

We now present the reduction rules of ${}^*\mathcal{X}$ calculus.

1. Activation rules

Activation rules hold the non-determinism of classical cut-elimination. More precisely, during the process of cut-elimination sometimes we have to choose the left or the right subtree to push the cut through. This choice is captured by the activation rules, which require to extend the syntax with new symbols called *active cuts*.

Definition 18 (Active Cuts). The syntax is extended with two *active cuts*:

$$P, Q ::= \ldots \mid P\widehat{\alpha} \nearrow \widehat{x}Q \mid P\widehat{\alpha} \nwarrow \widehat{x}Q$$

Activation rules are a potential source of non-confluence, an intrinsic property of classical logic, as illustrated by Example 19.

Example 19. Terms $P\widehat{\alpha} \nearrow \widehat{x}Q$ and $P\widehat{\alpha} \nwarrow \widehat{x}Q$ are essentially different. This becomes obvious in the example where both α and x are introduced by erasers. Take

$$P = M \odot \alpha \qquad \text{and} \qquad Q = x \odot N,$$

where M and N are arbitrary terms. Then we have:

$$\begin{array}{lcl} (M \odot \alpha)\widehat{\alpha} \nearrow \widehat{x}(x \odot N) & \to & \mathcal{I}^{N\setminus x} \odot M \odot \mathcal{O}^{N} \\ (M \odot \alpha)\widehat{\alpha} \nwarrow \widehat{x}(x \odot N) & \to & \mathcal{I}^{M} \odot N \odot \mathcal{O}^{M\setminus \alpha} \end{array}$$

This simple example is reminiscent of that of Lafont [18].

Remark 20. By constantly giving priority to either left or right activation, we may remove the non-confluence from the calculus and thus obtain two confluent subcalculi. In the case of $\bar{\lambda}\mu\tilde{\mu}$-calculus, if one gives priority to one of two sides, then one obtains a call-by-name or a call-by-value calculus. In accordance to what was noted for $\mathcal{X}$ in [36] - that this doesn't hold for $\mathcal{X}$, we suspect that it does not hold for ${}^*\mathcal{X}$ either.

2. Structural actions

Structural actions consist of four reduction rules, specifying *erasure* and *duplication* by referring to the situation when an active cut faces an eraser or a duplicator. Structural actions are given in Figure 11. These computational features were studied extensively in the framework of intuitionistic logic [10],[23].

Structural rules specifying duplication employ the so-called simultaneous substitutions. Informally, simultaneous substitution $\langle\!\langle \widehat{\alpha_1\alpha_2} \nearrow \widehat{x}Q \rangle\!\rangle$ in $P\langle\!\langle \widehat{\alpha_1\alpha_2} \nearrow \widehat{x}Q \rangle\!\rangle$ denotes applying independent modules[4]

[4]These modules are independent by definition of ${}^*\mathcal{X}$-terms $\widehat{\alpha_1} \nearrow \widehat{x}Q$ and $\widehat{\alpha_2} \nearrow \widehat{x}Q$ on P depending only on the occurrence of α_1 and α_2 at the top level and the level of immediate subterms of P. Similarly for $\langle\!\langle P\widehat{\alpha} \nwarrow \widehat{x_1x_2} \rangle\!\rangle$ which is symmetrical.

Left :				
($\nearrow$-*eras*)	:	$(P \odot \alpha)\widehat{\alpha} \nearrow \widehat{x}Q$	$\rightarrow$	$\mathcal{I}^Q \odot P \odot \mathcal{O}^Q$
($\nearrow$-*dupl*)	:	$([P\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}>\alpha)\widehat{\alpha} \nearrow \widehat{x}Q$	$\rightarrow$	$P\langle\!\langle\widehat{\alpha_1}\widehat{\alpha_2} \nearrow \widehat{x}Q\rangle\!\rangle$
Right :				
($\nwarrow$-*eras*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(x \odot Q)$	$\rightarrow$	$\mathcal{I}^P \odot Q \odot \mathcal{O}^P$
($\nwarrow$-*dupl*)	:	$P\widehat{\alpha} \nwarrow \widehat{x}(x<^{\widehat{x_1}}_{\widehat{x_2}}\langle Q])$	$\rightarrow$	$\langle\!\langle P\widehat{\alpha} \nwarrow \widehat{x_1}\widehat{x_2}\rangle\!\rangle Q$

Figure 11: Structural actions

Definition 21 (Simultaneous substitutions). We define simultaneous substitutions $\langle\!\langle\widehat{\alpha_1}\widehat{\alpha_2} \nearrow \widehat{x}Q\rangle\!\rangle$ and $\langle\!\langle P\widehat{\alpha} \nwarrow \widehat{x_1}\widehat{x_2}\rangle\!\rangle$ as follows:

- *Left simultaneous substitution*, $P\langle\!\langle\widehat{\alpha_1}\widehat{\alpha_2} \nearrow \widehat{x}Q\rangle\!\rangle$, is defined depending on the structure of term P to which it is applied:

P	$P\langle\!\langle\widehat{\alpha_1}\widehat{\alpha_2} \nearrow \widehat{x}Q\rangle\!\rangle$
$P^{\alpha_1}\{R\}$	$\mathcal{I}^Q <^{\widehat{\mathcal{I}^Q_1}}_{\widehat{\mathcal{I}^Q_2}}\langle(P^{\alpha_1}\{R\widehat{\alpha_2} \nearrow \widehat{x_2}Q_2\})\widehat{\alpha_1} \dagger \widehat{x_1}Q_1\rangle^{\widehat{\mathcal{O}^Q_1}}_{\widehat{\mathcal{O}^Q_2}} > \mathcal{O}^Q$
$P^{\alpha_2}\{R\}$	$\mathcal{I}^Q <^{\widehat{\mathcal{I}^Q_1}}_{\widehat{\mathcal{I}^Q_2}}\langle(P^{\alpha_2}\{R\widehat{\alpha_1} \nearrow \widehat{x_1}Q_1\})\widehat{\alpha_2} \dagger \widehat{x_2}Q_2\rangle^{\widehat{\mathcal{O}^Q_1}}_{\widehat{\mathcal{O}^Q_2}} > \mathcal{O}^Q$
$P^{\beta}\{R\}, \beta \neq \alpha_1, \alpha_2$	$P^{\beta}\{([R\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}>\alpha)\widehat{\alpha} \nearrow \widehat{x}Q\}$
$P\{R_1, R_2\}$	$\mathcal{I}^Q <^{\widehat{\mathcal{I}^Q_1}}_{\widehat{\mathcal{I}^Q_2}}\langle P\{R_1\widehat{\alpha_1} \nearrow \widehat{x_1}Q_1,\ R_2\widehat{\alpha_2} \nearrow \widehat{x_2}Q_2\}\rangle^{\widehat{\mathcal{O}^Q_1}}_{\widehat{\mathcal{O}^Q_2}} > \mathcal{O}^Q$, if $\alpha_1 \in N(R_1),\ \alpha_2 \in N(R_2)$
	$P\{([R_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}>\alpha)\widehat{\alpha} \nearrow \widehat{x}Q,\ R_2\}$ if $\alpha_1, a_2 \in N(R_1)$
	$P\{R_1,\ ([R_2\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}>\alpha)\widehat{\alpha} \nearrow \widehat{x}Q\},$ if $\alpha_1, a_2 \in N(R_2)$
$P^{x}\{R_1, R_2\}$	Analogously to the previous case

  Where R, R_1, R_2 denote immediate subterms of P, and where: $\mathcal{I}^Q = \overline{I}(Q) \setminus x$, $\mathcal{O}^Q = \overline{O}(Q)$ and $Q_i = ind(Q,\ N(Q),\ i)$ for $i = 1, 2$.

- *Right simultaneous substitution*, $\langle\!\langle P\widehat{\alpha} \nwarrow \widehat{x_1}\widehat{x_2}\rangle\!\rangle Q$, is defined depending on the structure of term Q to which it is applied:

3. Deactivation rules

As we will see, active cuts will be blocked by L-principal names. Thus cuts must be deactivated to continue to be distributed through the terms. Deactivation rules are given in Figure 12.

$\underline{Left}$:

$(\nearrow\text{-}deact)\ :\ P\widehat{\alpha} \nearrow \widehat{x}Q \ \rightarrow\ P\widehat{\alpha} \dagger \widehat{x}Q,$ if α is L-principal for P

$\underline{Right}$:

$(\nwarrow\text{-}deact)\ :\ P\widehat{\alpha} \nwarrow \widehat{x}Q \ \rightarrow\ P\widehat{\alpha} \dagger \widehat{x}Q,$ if x is L-principal for Q

Figure 12: Deactivation rules

Activation is dual of deactivation. Activation and deactivation rules are designed is such that they do not allow loops. Indeed the side conditions do not allow an activation of a cut followed by a deactivation of the same cut, or vice versa.

Q	$\langle\!\langle P\widehat{\alpha} \nwarrow \widehat{x_1}\widehat{x_2} \rangle\!\rangle Q$
$Q\{R\}$	$Q\{P\widehat{\alpha} \nwarrow \widehat{x}(x < \frac{\widehat{x_1}}{\widehat{x_2}}\langle R])\}$
$Q^{x_1}\{R_1,\, R_2\}$	$\mathcal{I}^P < \frac{\widehat{\mathcal{I}_1^P}}{\widehat{\mathcal{I}_2^P}}\langle P_1\widehat{\alpha_1} \dagger \widehat{x_1}(Q^{x_1}\{P_2\widehat{\alpha_2} \nwarrow \widehat{x_2}R_1,\, R_2\})\rangle \frac{\widehat{\mathcal{O}_1^P}}{\widehat{\mathcal{O}_2^P}} > \mathcal{O}^P,$ if $x_2 \in N(R_1)$
	$\mathcal{I}^P < \frac{\widehat{\mathcal{I}_1^P}}{\widehat{\mathcal{I}_2^P}}\langle P_1\widehat{\alpha_1} \dagger \widehat{x_1}(Q^{x_1}\{R_1,\, P_2\widehat{\alpha_2} \nwarrow \widehat{x_2}R_2\})\rangle \frac{\widehat{\mathcal{O}_1^P}}{\widehat{\mathcal{O}_2^P}} > \mathcal{O}^P,$ if $x_2 \in N(R_2)$
$Q^{x_2}\{R_1,\, R_2\}$	Analogously to the previous case
$Q\{R_1,\, R_2\}, y \neq x_1, x_2$	$\mathcal{I}^P < \frac{\widehat{\mathcal{I}_1^P}}{\widehat{\mathcal{I}_2^P}}\langle Q\{P_1\widehat{\alpha_1} \nwarrow \widehat{x_1}R_1,\, P_2\widehat{\alpha_2} \nwarrow \widehat{x_2}R_2\}\rangle \frac{\widehat{\mathcal{O}_1^P}}{\widehat{\mathcal{O}_2^P}} > \mathcal{O}^P,$ if $x_1 \in N(R_1),\ x_2 \in N(R_2)$
	$Q\{P\widehat{\alpha} \nwarrow \widehat{x}(x < \frac{\widehat{x_1}}{\widehat{x_2}}\langle R_1]),\, R_2\},$ if $x_1, x_2 \in N(R_1)$
	$Q\{R_1,\, P\widehat{\alpha} \nwarrow \widehat{x}(x < \frac{\widehat{x_1}}{\widehat{x_2}}\langle R_2])\},$ if $x_1, x_2 \in N(R_2)$
$Q^y\{R_1,\, R_2\}$	Analogously to the previous case

Where R, R_1, R_2 denote immediate subterms of Q, and where: $\mathcal{I}^P = \overline{I}(P),\ \mathcal{O}^P = \overline{O}(P) \setminus \alpha$ and $P_i = ind(P,\, N(P),\, i)$ for $i = 1, 2$.

4. Logical actions

The purpose of logical actions is to define reduction when L-principal names are involved in a cut. See Figure 13.

$$
\begin{array}{lcll}
(ren\text{-}L) & : & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}Q & \rightarrow \quad Q\{y/x\} \\
(ren\text{-}R) & : & P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle & \rightarrow \quad P\{\beta/\alpha\} \\
(ei\text{-}insert) & : & (\widehat{y}\,P\,\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\,\widehat{\gamma}\,[x]\,\widehat{z}\,R) & \rightarrow \quad either \left\{ \begin{array}{l} (Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R \\ Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R) \end{array} \right.
\end{array}
$$

Figure 13: Logical actions

The two first logical rules define the *merge* of a capsule with another term using renaming $\{y/x\}$ which is a meta operation, which resembles the λ-calculus meta-substitution. Renaming replaces simply a free name (unique by linearity) by another free name. It does not essentially change the term.

The third logical action describes the direct interaction between an exporter and an importer, which results in *inserting* the (immediate) subterm of an exporter between the two (immediate) subterms of an importer.

5. Propagation rules

Propagation rules describe the propagation of a cut through the structure of a term. This is a step-by-step propagation (the reduction rules "describe" propagation). It is important to note that propagation of a cut over another inactive cut is possible, which allows an elegant representation of β-reduction. The rules are divided into "left" and "right" symmetric groups, see Figures 14 and 15.

Observe for example the first rule in the left group. The rule is denoted as $(exp\nearrow -prop)$ and it shows how an active cut (in fact, a module $\widehat{\beta} \nearrow \widehat{y}R$) enters from the right-hand side through an exporter, up to its immediate subterm. The rules which define propagation over an exporter or a cut require side conditions to decide to which of the two immediate subterms the module will go.

The rules which require additional explanations are $(cut(c)\nearrow\text{-}prop)$ and $(\nwarrow cut(c)\text{-}prop)$. These are the rules which define an exception when performing propagation rules. They handle the case of propagation over a cut with a capsule whose both names are cut-names. If we exclude these rules from the system, we could construct an infinite reduction sequence.

$$
\begin{array}{lcll}
(exp\nearrow\text{-}prop) & : & (\widehat{x}\,P\,\widehat{\gamma}\cdot\alpha)\widehat{\beta}\nearrow\widehat{y}R & \to\ \widehat{x}\,(P\widehat{\beta}\nearrow\widehat{y}R)\,\widehat{\gamma}\cdot\alpha, \quad \alpha\neq\beta \\
\\
(imp\nearrow\text{-}prop_1) & : & (P\,\widehat{\alpha}\,[x]\,\widehat{z}\,Q)\widehat{\beta}\nearrow\widehat{y}R & \to\ (P\widehat{\beta}\nearrow\widehat{y}R)\,\widehat{\alpha}\,[x]\,\widehat{z}\,Q, \quad \beta\in O(P) \\
(imp\nearrow\text{-}prop_2) & : & (P\,\widehat{\alpha}\,[x]\,\widehat{z}\,Q)\widehat{\beta}\nearrow\widehat{y}R & \to\ P\,\widehat{\alpha}\,[x]\,\widehat{z}\,(Q\widehat{\beta}\nearrow\widehat{y}R), \quad \beta\in O(Q) \\
\\
(cut(c)\nearrow\text{-}prop) & : & (P\widehat{\alpha}\dagger\widehat{x}\langle x.\beta\rangle)\widehat{\beta}\nearrow\widehat{y}R & \to\ P\widehat{\alpha}\dagger\widehat{y}R \\
(cut\nearrow\text{-}prop_1) & : & (P\widehat{\alpha}\dagger\widehat{x}Q)\widehat{\beta}\nearrow\widehat{y}R & \to\ (P\widehat{\beta}\nearrow\widehat{y}R)\widehat{\alpha}\dagger\widehat{x}Q, \quad \beta\in O(P),\ Q\neq\langle x.\beta\rangle \\
(cut\nearrow\text{-}prop_2) & : & (P\widehat{\alpha}\dagger\widehat{x}Q)\widehat{\beta}\nearrow\widehat{y}R & \to\ P\widehat{\alpha}\dagger\widehat{x}(Q\widehat{\beta}\nearrow\widehat{y}R), \quad \beta\in O(Q),\ Q\neq\langle x.\beta\rangle \\
\\
(L\text{-}eras\nearrow\text{-}prop) & : & (x\odot P)\widehat{\beta}\nearrow\widehat{y}R & \to\ x\odot(P\widehat{\beta}\nearrow\widehat{y}R) \\
(R\text{-}eras\nearrow\text{-}prop) & : & (P\odot\alpha)\widehat{\beta}\nearrow\widehat{y}R & \to\ (P\widehat{\beta}\nearrow\widehat{y}R)\odot\alpha, \quad \alpha\neq\beta \\
\\
(L\text{-}dupl\nearrow\text{-}prop) & : & (x{<}\frac{\widehat{x_1}}{\widehat{x_2}}\langle P]\rangle)\widehat{\beta}\nearrow\widehat{y}R & \to\ x{<}\frac{\widehat{x_1}}{\widehat{x_2}}\langle P\widehat{\beta}\nearrow\widehat{y}R] \\
(R\text{-}dupl\nearrow\text{-}prop) & : & ([P\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}{>}\alpha)\widehat{\beta}\nearrow\widehat{y}R & \to\ [P\widehat{\beta}\nearrow\widehat{y}R\rangle\frac{\widehat{\alpha_1}}{\widehat{\alpha_2}}{>}\alpha, \quad \alpha\neq\beta
\end{array}
$$

Figure 14: Left propagation

Example 22. An example of an infinite reduction sequence in absence of $(cut(c)\nearrow\text{-}prop)$ and $(\nwarrow cut(c)\text{-}prop)$ rules:

$$
\begin{array}{rl}
 & (P\widehat{\alpha}\dagger\widehat{x}\langle x.\beta\rangle)\widehat{\beta}\dagger\widehat{y}R \\
\to & (P\widehat{\alpha}\dagger\widehat{x}\langle x.\beta\rangle)\widehat{\beta}\nearrow\widehat{y}R \\
\to & P\widehat{\alpha}\dagger\widehat{x}(\langle x.\beta\rangle\widehat{\beta}\nearrow\widehat{y}R) \\
\to & P\widehat{\alpha}\dagger\widehat{x}(\langle x.\beta\rangle\widehat{\beta}\dagger\widehat{y}R) \\
\to & P\widehat{\alpha}\nwarrow\widehat{x}(\langle x.\beta\rangle\widehat{\beta}\dagger\widehat{y}R) \\
\to & (P\widehat{\alpha}\nwarrow\widehat{x}\langle x.\beta\rangle)\widehat{\beta}\dagger\widehat{y}R \\
\to & (P\widehat{\alpha}\dagger\widehat{x}\langle x.\beta\rangle)\widehat{\beta}\dagger\widehat{y}R
\end{array}
$$

Besides that, the solution offered is intuitive as we would expect the terms

$$(P\widehat{\alpha}\dagger\widehat{x}\langle x.\beta\rangle)\widehat{\beta}\nearrow\widehat{y}R \ \text{ and } \ P\widehat{\alpha}\nwarrow\widehat{x}(\langle x.\beta\rangle\widehat{\beta}\dagger\widehat{y}R)$$

to reduce to the same term (which is in this case $P\widehat{\alpha}\dagger\widehat{y}R$).

$$
\begin{array}{lll}
(\nwarrow \mathit{exp\text{-}prop}) & : P\widehat{\alpha} \nwarrow \widehat{x}(\widehat{y}\,Q\,\widehat{\beta}\cdot\gamma) & \to\ \widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\cdot\gamma \\
\\
(\nwarrow \mathit{imp\text{-}prop}_1) & : P\widehat{\alpha} \nwarrow \widehat{x}(Q\,\widehat{\beta}\ [y]\ \widehat{z}\,R) & \to\ (P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\ [y]\ \widehat{z}\,R, \quad x \in I(Q) \\
(\nwarrow \mathit{imp\text{-}prop}_2) & : P\widehat{\alpha} \nwarrow \widehat{x}(Q\,\widehat{\beta}\ [y]\ \widehat{z}\,R) & \to\ Q\,\widehat{\beta}\ [y]\ \widehat{z}\,(P\widehat{\alpha} \nwarrow \widehat{x}R), \quad x \in I(R) \\
\\
(\nwarrow \mathit{cut(c)\text{-}prop}) & : P\widehat{\alpha} \nwarrow \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}R) & \to\ P\widehat{\alpha} \dagger \widehat{y}R \\
(\nwarrow \mathit{cut\text{-}prop}_1) & : P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R) & \to\ (P\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}R, \quad x \in I(Q),\ Q \neq \langle x.\beta\rangle \\
(\nwarrow \mathit{cut\text{-}prop}_2) & : P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R) & \to\ Q\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R), \quad x \in I(R),\ Q \neq \langle x.\beta\rangle \\
\\
(\nwarrow \mathit{L\text{-}eras\text{-}prop}) & : P\widehat{\alpha} \nwarrow \widehat{x}(y \odot Q) & \to\ y \odot (P\widehat{\alpha} \nwarrow \widehat{x}Q), \quad x \neq y \\
(\nwarrow \mathit{R\text{-}eras\text{-}prop}) & : P\widehat{\alpha} \nwarrow \widehat{x}(Q \odot \beta) & \to\ (P\widehat{\alpha} \nwarrow \widehat{x}Q) \odot \beta \\
\\
(\nwarrow \mathit{L\text{-}dupl\text{-}prop}) & : P\widehat{\alpha} \nwarrow \widehat{x}(y\!<\!^{\widehat{y_1}}_{\widehat{y_2}}\langle Q]) & \to\ y\!<\!^{\widehat{y_1}}_{\widehat{y_2}}\langle P\widehat{\alpha} \nwarrow \widehat{x}Q], \quad x \neq y \\
(\nwarrow \mathit{R\text{-}dupl\text{-}prop}) & : P\widehat{\alpha} \nwarrow \widehat{x}([Q\rangle^{\widehat{\beta_1}}_{\widehat{\beta_2}}\!>\!\beta) & \to\ [P\widehat{\alpha} \nwarrow \widehat{x}Q\rangle^{\widehat{\beta_1}}_{\widehat{\beta_2}}\!>\!\beta
\end{array}
$$

Figure 15: Right propagation

3.3. Operational properties

The reduction system enjoys some desirable properties as expressed by the following lemma.

Theorem 23 (Basic properties of $\to$).

1. *Preservation of* free names*: If $P \to Q$ then $N(P) = N(Q)$.*
2. *Preservation of* linearity*: If P is linear and $P \to Q$ then Q is linear.*

Proof:. These properties can be confirmed by checking carefully each rule. □

Preservation of free names holds in ${}^*\mathcal{X}$ due to the use of erasers and duplicators in rewrite rules (like in $\lambda\mathsf{lxr}$ [23]). This property is sometimes referred to as *interface preservation* like in interaction nets [27]. The property of closure under reduction is a minimal requirement, it is a kind of linearity preservation for ${}^*\mathcal{X}$-terms.

Simplification rules. We define the *simplification rules*, denoted $\dashrightarrow$, which can be seen as an efficient way to simplify terms. They are not reduction rules as they do not involve cuts. The point is that applying a duplicator to an eraser is of no interest and can be avoided by using simplification rules, as defined by:

$$\boxed{\begin{array}{lcll} (s_L) & : & x{<}\frac{\widehat{y}}{\widehat{z}}\langle z \odot P] & \dashrightarrow \quad P\{x/y\} \\ (s_R) & : & [P \odot \gamma\rangle^{\widehat{\beta}}_{\widehat{\gamma}}{>}\alpha & \dashrightarrow \quad P\{\alpha/\beta\} \end{array}}$$

They are run before reduction rules, that is, we give them higher priority during computation. One can see them as a kind of garbage collection, as they simplify computation by preventing the situation when we duplicate a term to erase one or both copies in the next step. It is easy to see that the simplification rules preserve free names, linearity and types. The rules can be given in a more general way:

$$\boxed{\begin{array}{lcll} (s^g_L) & : & \mathcal{I}{<}\frac{\widehat{\mathcal{I}_1}}{\widehat{\mathcal{I}_2}}\langle \mathcal{I}_2 \odot P] & \dashrightarrow \quad P\{\mathcal{I}/\mathcal{I}_1\} \\ (s^g_R) & : & [P \odot \mathcal{O}_2\rangle^{\widehat{\mathcal{O}_1}}_{\widehat{\mathcal{O}_2}}{>}\mathcal{O} & \dashrightarrow \quad P\{\mathcal{O}/\mathcal{O}_1\} \end{array}}$$

3.4. The type assignment system

We restrict now to terms to which we can attach types of the form:

$$A, B ::= T \mid A \to B.$$

The *type assignment* of an ${}^*\mathcal{X}$-term P is expressed as $P :\cdot\ \Gamma \vdash \Delta$, where Γ is the antecedent whose domain is made of free innames of P and Δ is the succedent whose domain is made of free outnames of P. Contexts are sets of pairs (name, formula). For example, Γ is a set of type declarations for innames like $x : A,\ y : B$, while Δ as a set of declarations for outnames like $\alpha : A$, $\beta : A \to B$, $\gamma : C$. Comma in the expression Γ, Γ' stands for set union.

We will say that a term P is *typable* if there exist contexts Γ and Δ such that $P :\cdot\ \Gamma \vdash \Delta$ holds in the system of inference rules given by Figure 16. If we remove term-decoration and names, we get the classical sequent system $G1$ given in Figure 1.

Example 24. An illustration could be the type assignment of the ${}^*\mathcal{X}$-term which codes the proof of Peirce's law.

$$\frac{}{\langle x.\alpha\rangle :\cdot \;\; x:A \vdash \alpha:A}\;(ax)$$

$$\frac{P :\cdot \;\; \Gamma \vdash \alpha:A,\Delta \qquad Q :\cdot \;\; \Gamma', y:B \vdash \Delta'}{P\,\widehat{\alpha}\;[x]\;\widehat{y}\,Q :\cdot \;\; \Gamma,\Gamma', x:A\to B \vdash \Delta,\Delta'}\;(L\to) \qquad \frac{P :\cdot \;\; \Gamma, x:A \vdash \alpha:B,\Delta}{\widehat{x}\,P\,\widehat{\alpha}\cdot\beta :\cdot \;\; \Gamma \vdash \beta:A\to B,\Delta}\;(R\to)$$

$$\frac{P :\cdot \;\; \Gamma \vdash \alpha:A,\Delta \qquad Q :\cdot \;\; \Gamma', x:A \vdash \Delta'}{P\widehat{\alpha}\;\dagger\;\widehat{x}Q :\cdot \;\; \Gamma,\Gamma' \vdash \Delta,\Delta'}\;(cut)$$

$$\frac{P :\cdot \;\; \Gamma \vdash \Delta}{x \odot P :\cdot \;\; \Gamma, x:A \vdash \Delta}\;(weak\text{-}L) \qquad \frac{P :\cdot \;\; \Gamma \vdash \Delta}{P \odot \alpha :\cdot \;\; \Gamma \vdash \alpha:A,\Delta}\;(weak\text{-}R)$$

$$\frac{P :\cdot \;\; \Gamma, x:A, y:A \vdash \Delta}{z{<}^{\widehat{x}}_{\widehat{y}}\langle P] :\cdot \;\; \Gamma, z:A \vdash \Delta}\;(cont\text{-}L) \qquad \frac{P :\cdot \;\; \Gamma \vdash \alpha:A,\beta:A,\Delta}{[P\rangle^{\widehat{\alpha}}_{\widehat{\beta}}{>}\gamma :\cdot \;\; \Gamma \vdash \gamma:A,\Delta}\;(cont\text{-}R)$$

Figure 16: ${}^*\mathcal{X}$ type system

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\langle x.\alpha_1\rangle :\cdot \;\; x:A \vdash \alpha_1:A}\;(ax)}{\langle x.\alpha_1\rangle \odot \beta :\cdot \;\; x:A \vdash \alpha_1:A,\beta:B}\;(weak\text{-}R)}{\widehat{x}\,(\langle x.\alpha_1\rangle \odot \beta)\,\widehat{\beta}\cdot\gamma :\cdot \;\; \vdash \alpha_1:A,\gamma:A\to B}\;(\to R) \qquad \dfrac{}{\langle y.\alpha_2\rangle :\cdot \;\; y:A \vdash \alpha_2:A}\;(ax)}{(\widehat{x}\,(\langle x.\alpha_1\rangle \odot \beta)\,\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\;[z]\;\widehat{y}\,\langle y.\alpha_2\rangle :\cdot \;\; z:(A\to B)\to A \vdash \alpha_1:A,\alpha_2:A}\;(\to L)}{[(\widehat{x}\,(\langle x.\alpha_1\rangle \odot \beta)\,\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\;[z]\;\widehat{y}\,\langle y.\alpha_2\rangle\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha :\cdot \;\; z:(A\to B)\to A \vdash \alpha:A}\;(cont\text{-}R)}{\widehat{z}\,([(\widehat{x}\,(\langle x.\alpha_1\rangle \odot \beta)\,\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\;[z]\;\widehat{y}\,\langle y.\alpha_2\rangle\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\,\widehat{\alpha}\cdot\delta :\cdot \;\; \vdash \delta:((A\to B)\to A)\to A}\;(\to R)$$

Example 25. The ${}^*\mathcal{X}$-term which corresponds to $\lambda xyz.xz(yz)$, known as the S-combinator of λ-calculus, is the following:[5]

$$\widehat{\omega}\,(\widehat{u}\,(\widehat{x}\,(x{<}^{\widehat{x_1}}_{\widehat{x_2}}\langle\langle x_2.\epsilon\rangle\,\widehat{\epsilon}\;[w]\;\widehat{v}\,\underline{\underline{((\langle x_1.\delta\rangle\,\widehat{\delta}\;[u]\;\widehat{y}\,\langle y.\beta\rangle)}}\,\widehat{\beta}\;[v]\;\widehat{z}\,\langle z.\gamma\rangle)])\,\widehat{\gamma}\cdot\eta)\,\widehat{\eta}\cdot\theta)\,\widehat{\theta}\cdot\alpha$$

[5]Some parts of terms are underlined to ease the reading.

The witness reduction property

An ${}^*\mathcal{X}$term is the interpretation of a proof in the sequent calculus. If we use computations as proof-transformations, the property of witness reduction is essential.

Theorem 26 (Witness reduction). *Let S be an ${}^*\mathcal{X}$-term and Γ, Δ two contexts. Then the following holds:*

$$\textit{If } S :\cdot\ \Gamma \vdash \Delta \textit{ and } S \to S', \textit{ then } S' :\cdot\ \Gamma \vdash \Delta$$

Remark 27. Linearity and free names are preserved (Theorem 23).

Proof:. The proof is straightforward and goes by inspecting the reduction rules, and by induction on the structure of terms [44].

Theorem 28 (⇢ preserves types). *Simplification rules preserve types.*

$$\textit{If } S :\cdot\ \Gamma \vdash \Delta \textit{ and } S \dashrightarrow S', \textit{ then } S' :\cdot\ \Gamma \vdash \Delta$$

Proof:. By analyzing the proof trees corresponding to S and S', for both simplification rules.

4. Explicit vs. implicit: relation between ${}^*\mathcal{X}$ and $\mathcal{X}$

The ${}^*\mathcal{X}$ calculus is a low-level language whose syntax is an extension of that of the $\mathcal{X}$ calculus, and therefore its reduction steps decompose reduction steps of $\mathcal{X}$, which on its own is also a low level language.

The expressive power of $\mathcal{X}$ has been illustrated in [42], by encoding various calculi, such as: λ, $\lambda\mathsf{x}$ and $\lambda\mu$. Also the $\mathcal{X}$ calculus is encoded into $\lambda\mu$ in [2]. The first hint on how to relate $\bar{\lambda}\mu\tilde{\mu}$ and Gentzen's sequent calculus for classical logic LK (which corresponds to $\mathcal{X}$) was already given by Curien and Herbelin in [7]. It was studied in detail through the $\lambda\xi$-calculus [28], where mutual embedings are presented. These results were used to give an elegant proof of strong normalization for the $\bar{\lambda}\mu\tilde{\mu}$-calculus.

Our view is that most of the features of the $\mathcal{X}$ calculus can also be shown for ${}^*\mathcal{X}$. Since the ${}^*\mathcal{X}$ calculus has a lower level of granularity, is expected to be at least as expressive as the $\mathcal{X}$ calculus. In case of potential implementation this model is better suited, since it introduces the possibility of controlling both duplication and erasure of parts of a program. In this chapter we study

the relation between ${}^*\mathcal{X}$ and the following calculi; intuitionistic: λ, $\lambda\mathsf{x}$ and $\lambda\mathsf{lxr}$, and classical: $\mathcal{X}$ and $\bar{\lambda}\mu\tilde{\mu}$. In this section we show the relation between $\mathcal{X}$-terms and ${}^*\mathcal{X}$-terms. We present the encodings in both directions, and study the relation between the computations. It is shown that $\mathcal{X}$-reduction steps are decomposed into more atomic steps of ${}^*\mathcal{X}$, due to the linearity and the presence of explicit terms for erasure and duplication. Finally, we study the relation between typing of $\mathcal{X}$-terms and typing of ${}^*\mathcal{X}$-terms.

4.1. From $\mathcal{X}$ to ${}^*\mathcal{X}$

We now describe how to encode $\mathcal{X}$-terms, which are possibly not linear, into terms of the ${}^*\mathcal{X}$ calculus. Before doing that we will introduce two operations to help us formulate the encoding. They will be used in the formal definition and their only purpose is to make definitions easier to read. The first operation, denoted by $\circledcirc$, adds *erasers* where needed.

Definition 29 (Potential eraser: $\circledcirc$). The operation $\circledcirc$ is defined as follows:

$$x \circledcirc P \circledcirc \alpha \;=\; \begin{cases} P, & x, \alpha \in N(P) \\ x \odot P, & x \notin N(P),\ \alpha \in N(P) \\ P \odot \alpha, & x \in N(P),\ \alpha \notin N(P) \\ x \odot P \odot \alpha, & x, \alpha \notin N(P) \end{cases}$$

The second operation, denoted by $\lhd\,(\ )\,\rhd$, adds *contractions* where needed. Typically this will happen when encoding terms which have two immediate subterms, denoted by $C\{P, Q\}$, and it will be used to prevent the multiple occurrences of names. This operation also improves the readability of the encoding, although we could have used actual contractions.

Definition 30 (Potential contractions: $\lhd(\ )\rhd$). The operation $\lhd(\ )\rhd$ is defined as follows:

$$\mathcal{I} \lhd \Big(C\{P, Q\}\Big) \rhd \mathcal{O} \;=\; \begin{cases} C\{P, Q\}, & N(P) \cap N(Q) = \emptyset \\ \mathcal{I} < \frac{\widehat{\mathcal{I}_1}}{\widehat{\mathcal{I}_2}} \langle C\{P, Q\} \rangle \frac{\widehat{\mathcal{O}_1}}{\widehat{\mathcal{O}_2}} > \mathcal{O} & \text{when } N(P) \cap N(Q) \neq \emptyset, \\ & \text{where } \mathcal{I} = I(P) \cap I(Q) \\ & \qquad \mathcal{O} = O(P) \cap O(Q) \end{cases}$$

Definition 31. The encoding of $\mathcal{X}$-terms in ${}^*\mathcal{X}$ is defined by induction, as presented by Figure 17.

$$
\begin{array}{lcl}
\llbracket \langle x.\alpha \rangle \rrbracket^{*\mathcal{X}} & := & \langle x.\alpha \rangle \\
\llbracket \widehat{x}\, P\, \widehat{\beta} \cdot \alpha \rrbracket^{*\mathcal{X}} & := & \left(\widehat{x}\, (x \circledcirc \llbracket P \rrbracket^{*\mathcal{X}} \circledcirc \beta)\, \widehat{\beta} \cdot \alpha \right) \rhd \alpha, \\
\llbracket P\, \widehat{\alpha}\, [x]\, \widehat{y}\, Q \rrbracket^{*\mathcal{X}} & := & \mathcal{I} \lhd \left((\llbracket P \rrbracket^{*\mathcal{X}} \circledcirc \alpha)\, \widehat{\alpha}\, [x]\, \widehat{y}\, (y \circledcirc \llbracket Q \rrbracket^{*\mathcal{X}}) \right) \rhd \mathcal{O}, \\
 & & \quad \text{for } x \notin N(P), x \notin N(Q) \\
\llbracket P\, \widehat{\alpha}\, [x]\, \widehat{y}\, Q \rrbracket^{*\mathcal{X}} & := & \mathcal{I} \lhd \left(x \!<\! \frac{\widehat{x_1}}{\widehat{x_2}} \langle ((\llbracket P\{x_1/x\} \rrbracket^{*\mathcal{X}} \circledcirc \alpha)\, \widehat{\alpha}\, [x_2]\, \widehat{y}\, (y \circledcirc \llbracket Q \rrbracket^{*\mathcal{X}})] \right) \rhd \mathcal{O}, \\
 & & \quad \text{for } x \in N(P), x \notin N(Q) \\
\llbracket P\, \widehat{\alpha}\, [x]\, \widehat{y}\, Q \rrbracket^{*\mathcal{X}} & := & \mathcal{I} \lhd \left(x \!<\! \frac{\widehat{x_1}}{\widehat{x_2}} \langle ((\llbracket P \rrbracket^{*\mathcal{X}} \circledcirc \alpha)\, \widehat{\alpha}\, [x_1]\, \widehat{y}\, (y \circledcirc \llbracket Q\{x_2/x\} \rrbracket^{*\mathcal{X}})] \right) \rhd \mathcal{O}, \\
 & & \quad \text{for } x \notin N(P), x \in N(Q) \\
\llbracket P\, \widehat{\alpha}\, [x]\, \widehat{y}\, Q \rrbracket^{*\mathcal{X}} & := & \mathcal{I} \lhd \left(x \!<\! \frac{\widehat{t}}{\widehat{x_3}} \langle t \!<\! \frac{\widehat{x_1}}{\widehat{x_2}} \langle ((\llbracket P\{x_1/x\} \rrbracket^{*\mathcal{X}} \circledcirc \alpha)\, \widehat{\alpha}\, [x_2]\, \widehat{y}\, (y \circledcirc \llbracket Q\{x_3/x\} \rrbracket^{*\mathcal{X}})]] \right) \rhd \mathcal{O}, \\
 & & \quad \text{for } x \in N(P), x \in N(Q) \\
\llbracket P \widehat{\alpha} \dagger \widehat{x} Q \rrbracket^{*\mathcal{X}} & := & \mathcal{I} \lhd \left((\llbracket P \rrbracket^{*\mathcal{X}} \circledcirc \alpha) \widehat{\alpha} \dagger \widehat{x} (x \circledcirc \llbracket Q \rrbracket^{*\mathcal{X}}) \right) \rhd \mathcal{O},
\end{array}
$$

Figure 17: Encoding the $\mathcal{X}$-terms into ${}^*\mathcal{X}$

Figure 17 defines the encoding of pure $\mathcal{X}$-terms in ${}^*\mathcal{X}$. Active cuts can be encoded in the following way:

$$\llbracket P \widehat{\alpha} \nearrow \widehat{x} Q \rrbracket^{*\mathcal{X}} := \mathcal{I} \lhd \left((\llbracket P \rrbracket^{*\mathcal{X}} \circledcirc \alpha) \widehat{\alpha} \nearrow \widehat{x} (x \circledcirc \llbracket Q \rrbracket^{*\mathcal{X}}) \right) \rhd \mathcal{O}$$

$$\llbracket P \widehat{\alpha} \nwarrow \widehat{x} Q \rrbracket^{*\mathcal{X}} := \mathcal{I} \lhd \left((\llbracket P \rrbracket^{*\mathcal{X}} \circledcirc \alpha) \widehat{\alpha} \nwarrow \widehat{x} (x \circledcirc \llbracket Q \rrbracket^{*\mathcal{X}}) \right) \rhd \mathcal{O}$$

Remark 32. Notice that if the $\mathcal{X}$-term is linear, i.e., if there is no need to use the operations $\circledcirc$ and $\lhd(\ )\rhd$, we get simply

$$
\begin{array}{rcl}
\llbracket \langle x.\alpha \rangle \rrbracket^{*\mathcal{X}} & = & \langle x.\alpha \rangle \\
\llbracket \widehat{x}\, P\, \widehat{\beta} \cdot \alpha \rrbracket^{*\mathcal{X}} & = & \widehat{x}\, \llbracket P \rrbracket^{*\mathcal{X}}\, \widehat{\beta} \cdot \alpha \\
\llbracket P\, \widehat{\alpha}\, [x]\, \widehat{y}\, Q \rrbracket^{*\mathcal{X}} & = & \llbracket P \rrbracket^{*\mathcal{X}}\, \widehat{\alpha}\, [x]\, \widehat{y}\, \llbracket Q \rrbracket^{*\mathcal{X}} \\
\llbracket P \widehat{\alpha} \dagger \widehat{x} Q \rrbracket^{*\mathcal{X}} & = & \llbracket P \rrbracket^{*\mathcal{X}} \widehat{\alpha} \dagger \widehat{x} \llbracket Q \rrbracket^{*\mathcal{X}}
\end{array}
$$

Remark 33. The encoding is defined in such a way that none of the free names is lost. Notice that this is not the case with *occurrences* of free names. If a free name has multiple occurrences in $\mathcal{X}$-term, it will occur only once after the encoding.

Lemma 34. *The encoding* $[\![\ ]\!]^{*\mathcal{X}}$ *preserves the set of free names.*

$$N(P) \;=\; N([\![P]\!]^{*\mathcal{X}})$$

Proof. By inspection of the encoding rules. □

Example 35. Take for example $P = (\widehat{x}\,\langle x.\alpha\rangle\,\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\;[z]\;\widehat{y}\,\langle y.\alpha\rangle$, where α as a free name occurs twice, and $\widehat{\beta}$ does not bind an occurrence of a free name. The encoding gives:

$$[\![P]\!]^{*\mathcal{X}} = [(\widehat{x}\,(\langle x.\alpha_1\rangle \odot \beta)\,\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\;[z]\;\widehat{y}\,\langle y.\alpha_2\rangle\rangle_{\widehat{\alpha_2}}^{\widehat{\alpha_1}} > \alpha$$

where α has only one occurrence, and $\widehat{\beta}$ does bind an occurrence of a free name.

Notation. We will sometimes annotate the arrow in order to ease the reading: we use $\xrightarrow{*\mathcal{X}}$ to denote ${}^*\mathcal{X}$-reduction and $\xrightarrow{\mathcal{X}}$ to denote $\mathcal{X}$-reduction. Moreover $\rightarrow^*$ and $\rightarrow^+$ are used to denote zero or more, and one or more reduction steps, respectively.

Simulation of $\mathcal{X}$-reduction. In what follows we show that the reduction rules of $\mathcal{X}$ can be simulated in ${}^*\mathcal{X}$. Initially we show that the notion of *introduced name* in $\mathcal{X}$ corresponds to the notion of *L-principal name* in ${}^*\mathcal{X}$.

Lemma 36. *The notion of* introduced name *by a term in $\mathcal{X}$, and that of* L-principal name *of a term in ${}^*\mathcal{X}$, correspond to each other.*

1. *If α is freshly introduced by S, then α is L-principal for $[\![S]\!]^{*\mathcal{X}}$*
2. *If α is L-principal for S, then α is freshly introduced by $[\![S]\!]^{\mathcal{X}}$*

Proof. 1. Case: $S = \langle x.\alpha\rangle$. We have $[\![S]\!]^{*\mathcal{X}} = [\![\langle x.\alpha\rangle]\!]^{*\mathcal{X}} = \langle x.\alpha\rangle$, and thus α is L-principal for $\langle x.\alpha\rangle$.

Case: $S = \widehat{x}\,P\,\widehat{\beta}\cdot\alpha$. Since α is freshly introduced $\alpha \notin N(P)$. We have $[\![S]\!]^{\mathcal{X}} = [\![\widehat{y}\,P\,\widehat{\beta}\cdot\alpha]\!]^{*\mathcal{X}} = \widehat{y}\,(y \circledcirc [\![P]\!]^{*\mathcal{X}} \circledcirc \beta)\,\widehat{\beta}\cdot\alpha$, and thus by definition α is L-principal for $[\![S]\!]^{*\mathcal{X}}$.

2. Case: $S = \langle x.\alpha\rangle$. We have $[\![S]\!]^{\mathcal{X}} = [\![\langle x.\alpha\rangle]\!]^{\mathcal{X}} = \langle x.\alpha\rangle$, where α is freshly introduced by $\langle x.\alpha\rangle$, by definition.

   Case: $S = \widehat{x}\,P\,\widehat{\beta}\cdot\alpha$. By linearity it stands that $\alpha \notin N(P)$. We have $S = [\![\widehat{y}\,P\,\widehat{\beta}\cdot\alpha]\!]^{\mathcal{X}} = \widehat{y}\,[\![P]\!]^{\mathcal{X}}\,\widehat{\beta}\cdot\alpha$, and thus α is freshly introduced by $[\![S]\!]^{\mathcal{X}}$.

It is not difficult to check that the same holds for innames. □

Theorem 37 (Simulation of $\mathcal{X}$-reduction). *Let P and P' be $\mathcal{X}$-terms. Then the following holds:*

$$\textit{If } P \xrightarrow{\mathcal{X}} P' \quad \textit{then} \quad [\![P]\!]^{*\mathcal{X}} \xrightarrow{{}^{*}\mathcal{X}}{}^{+} (\mathcal{I}^{P\setminus P'}) \odot [\![P']\!]^{*\mathcal{X}} \odot (\mathcal{O}^{P\setminus P'})$$

Proof. The proof goes by inspecting the reduction rules and by induction on the structure of terms. We give the proof for some reduction rules.

Logical rules:.

• Take the $(cap - ren)$ rule: $\langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \to \langle y.\beta\rangle$. We have:

$$\begin{array}{rcl} [\![\langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle]\!]^{*\mathcal{X}} & \triangleq & [\![\langle y.\alpha\rangle]\!]^{*\mathcal{X}}\widehat{\alpha} \dagger \widehat{x}[\![\langle x.\beta\rangle]\!]^{*\mathcal{X}} \\ & \triangleq & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \\ & \to & \langle y.b\rangle \\ & \triangleq & [\![\langle y.\beta\rangle]\!]^{*\mathcal{X}} \end{array}$$

• Take the $(exp - ren)$ rule: $(\widehat{y}\,P\,\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle \to \widehat{y}\,P\,\widehat{\beta}\cdot\gamma,\ \alpha \notin N(P)$. We have (assuming for simplicity that $\gamma \notin N(P)$):

$$\begin{array}{rcl} [\![(\widehat{y}\,P\,\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle]\!]^{*\mathcal{X}} & \triangleq & [\![\widehat{y}\,P\,\widehat{\beta}\cdot\alpha]\!]^{*\mathcal{X}}\widehat{\alpha} \dagger \widehat{x}[\![\langle x.\gamma\rangle]\!]^{*\mathcal{X}} \\ & \triangleq & (\widehat{y}\,(y \circledcirc [\![P]\!]^{*\mathcal{X}} \circledcirc \beta)\,\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle \\ & \to & \widehat{y}\,(y \circledcirc [\![P]\!]^{*\mathcal{X}} \circledcirc \beta)\,\widehat{\beta}\cdot\gamma \\ & \triangleq & [\![\widehat{y}\,P\,\widehat{\beta}\cdot\gamma]\!]^{*\mathcal{X}} \end{array}$$

• Take the $(imp - ren)$ rule: $\langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(Q\,\widehat{\beta}\,[x]\,\widehat{z}\,R) \to Q\,\widehat{\beta}\,[y]\,\widehat{z}\,R$, where $x \notin N(Q),\ x \notin N(R)$. We have:

$$\begin{array}{rcl} [\![\langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(Q\,\widehat{\beta}\,[x]\,\widehat{z}\,R)]\!]^{*\mathcal{X}} & \triangleq & [\![\langle y.\alpha\rangle]\!]^{*\mathcal{X}}\widehat{\alpha} \dagger \widehat{x}[\![Q\,\widehat{\beta}\,[x]\,\widehat{z}\,R]\!]^{*\mathcal{X}} \\ & \triangleq & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(\mathcal{I} \lhd \Big(([\![Q]\!]^{*\mathcal{X}} \circledcirc \beta)\,\widehat{\beta}\,[x]\,\widehat{z}\,(z \circledcirc [\![R]\!]^{*\mathcal{X}})\Big) \rhd \mathcal{O}) \\ & \to & \mathcal{I} \lhd \Big(([\![Q]\!]^{*\mathcal{X}} \circledcirc \beta)\,\widehat{\beta}\,[y]\,\widehat{z}\,(z \circledcirc [\![R]\!]^{*\mathcal{X}})\Big) \rhd \mathcal{O} \\ & \triangleq & [\![Q\,\widehat{\beta}\,[y]\,\widehat{z}\,R]\!]^{*\mathcal{X}} \end{array}$$

For simplicity we assumed that $y \notin N(Q)$ and $y \notin N(R)$.

Activation rules.

• Take the $(act{-}L)$ rule: $P\widehat{\alpha}\,\dagger\,\widehat{x}Q \;\rightarrow\; P\widehat{\alpha} \nearrow \widehat{x}Q$, if α not freshly introduced by P. We have:

$$\begin{array}{rcl} [\![P\widehat{\alpha} \dagger \widehat{x}Q]\!]^{*\mathcal{X}} & \triangleq & \mathcal{I}^{P\cap Q} \lhd \Big(([\![P]\!]^{*\mathcal{X}} \circledcirc \alpha)\widehat{\alpha} \dagger \widehat{x}(x \circledcirc [\![Q]\!]^{*\mathcal{X}})\Big) \rhd \mathcal{O}^{P\cap Q} \\ & \xrightarrow{Lem.36} & \mathcal{I}^{P\cap Q} \lhd \Big(([\![P]\!]^{*\mathcal{X}} \circledcirc \alpha)\widehat{\alpha} \nearrow \widehat{x}(x \circledcirc [\![Q]\!]^{*\mathcal{X}})\Big) \rhd \mathcal{O}^{P\cap Q} \\ & \triangleq & [\![P\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{*\mathcal{X}} \end{array}$$

Similarly for the rule $(act - R)$.

Propagation rules.

• Take the $(\nearrow - eras)$ rule: $\langle x.\alpha\rangle\widehat{\beta} \nearrow \widehat{y}R \;\rightarrow\; \langle x.\alpha\rangle$, where $\alpha \neq \beta$.
We will take into consideration the possibility that $x, \alpha \in N(R)$. Thus we have:

$$\begin{array}{rcl} [\![\langle x.\alpha\rangle\widehat{\beta} \nearrow \widehat{y}R]\!]^{*\mathcal{X}} & \triangleq & x \lhd \Big(([\![\langle x.\alpha\rangle]\!]^{*\mathcal{X}} \odot \beta)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc [\![R]\!]^{*\mathcal{X}})\Big) \rhd \alpha \\ & \rightarrow & x \lhd \Big(\mathcal{I}^R \odot [\![\langle x.\alpha\rangle]\!]^{*\mathcal{X}} \odot \mathcal{O}^R\Big) \rhd \alpha \\ & \dashrightarrow & (\mathcal{I}^R \setminus x) \odot [\![\langle x.\alpha\rangle]\!]^{*\mathcal{X}} \odot (\mathcal{O}^R \setminus \alpha) \end{array}$$

• Take the $(\nearrow - deact)$ rule: $\langle x.\beta\rangle\widehat{\beta} \nearrow \widehat{y}R \;\rightarrow\; \langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}R$. We have:

$$\begin{array}{rcl} [\![\langle x.\beta\rangle\widehat{\beta} \nearrow \widehat{y}R]\!]^{*\mathcal{X}} & \triangleq & x \lhd \Big([\![\langle x.\beta\rangle]\!]^{*\mathcal{X}}\widehat{\beta} \nearrow \widehat{y}(y \circledcirc [\![R]\!]^{*\mathcal{X}})\Big) \\ & \triangleq & x \lhd \Big(\langle x.\beta\rangle\widehat{\beta} \nearrow \widehat{y}(y \circledcirc [\![R]\!]^{*\mathcal{X}})\Big) \\ & \rightarrow & x \lhd \Big(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}(y \circledcirc [\![R]\!]^{*\mathcal{X}})\Big) \\ & \triangleq & [\![\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}R]\!]^{*\mathcal{X}} \end{array}$$

• Take the $(\nearrow - prop)$ rule: $(\widehat{x}\,P\,\widehat{\gamma}\cdot\alpha)\widehat{\beta} \nearrow \widehat{y}R \;\rightarrow\; \widehat{x}\,(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\gamma}\cdot\alpha$, $\alpha \neq \beta$.
We assume for simplicity $N(\widehat{x}\,P\,\widehat{\gamma}\cdot\alpha) \cap N(R) = \emptyset$. We have:

$$
\begin{aligned}
\llbracket(\widehat{x}\,P\,\widehat{\gamma}\cdot\alpha)\widehat{\beta} \nearrow \widehat{y}R\rrbracket^{*\mathcal{X}} &\triangleq (\llbracket\widehat{x}\,P\,\widehat{\gamma}\cdot\alpha\rrbracket^{*\mathcal{X}} \circledcirc \beta)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}}) \\
&\triangleq ((\widehat{x}\,(x \circledcirc \llbracket P\rrbracket^{*\mathcal{X}} \circledcirc \gamma)\,\widehat{\gamma}\cdot\alpha) \circledcirc \beta)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}}) \\
&\equiv (\widehat{x}\,((x \circledcirc \llbracket P\rrbracket^{*\mathcal{X}} \circledcirc \gamma) \circledcirc \beta)\,\widehat{\gamma}\cdot\alpha)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}}) \\
&\to \widehat{x}\,(((x \circledcirc \llbracket P\rrbracket^{*\mathcal{X}} \circledcirc \gamma) \circledcirc \beta)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}}))\,\widehat{\gamma}\cdot\alpha \\
&\equiv \widehat{x}\,(x \circledcirc ((\llbracket P\rrbracket^{*\mathcal{X}} \circledcirc \beta)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}})) \circledcirc \gamma)\,\widehat{\gamma}\cdot\alpha \\
&\triangleq \widehat{x}\,(x \circledcirc (P\widehat{\beta} \nearrow \widehat{y}R) \circledcirc \gamma)\,\widehat{\gamma}\cdot\alpha \\
&\triangleq \llbracket\widehat{x}\,(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\gamma}\cdot\alpha\rrbracket^{*\mathcal{X}}
\end{aligned}
$$

• Take the (*↗-prop-dupl-deact*) rule:
$(\widehat{x}\,P\,\widehat{\gamma}\cdot\beta)\widehat{\beta} \nearrow \widehat{y}R \;\to\; (\widehat{x}\,(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\gamma}\cdot\beta)\widehat{\beta} \dagger \widehat{y}R$, and consider $\beta \in N(P)$. We assume for simplicity $N(P) \cap N(R) = \emptyset$, then we have:

$$
\begin{aligned}
&\llbracket(\widehat{x}\,P\,\widehat{\gamma}\cdot\beta)\widehat{\beta} \nearrow \widehat{y}R\rrbracket^{*\mathcal{X}} \triangleq \llbracket\widehat{x}\,P\,\widehat{\gamma}\cdot\beta\rrbracket^{*\mathcal{X}}\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}}) \\
&\triangleq ([\underline{\widehat{x}\,(x \circledcirc (\llbracket P\rrbracket^{*\mathcal{X}}\{\beta_1/\beta\}) \circledcirc \gamma)\,\widehat{\gamma}\cdot\beta_2}\rangle^{\widehat{\beta_1}}_{\widehat{\beta_2}}\!>\beta)\widehat{\beta} \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}}) \\
&\to (\underline{\widehat{x}\,(x \circledcirc (\llbracket P\rrbracket^{*\mathcal{X}}\{\beta_1/\beta\}) \circledcirc \gamma)\,\widehat{\gamma}\cdot\beta_2})\langle\!\langle\widehat{\beta}_1\widehat{\beta}_2 \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}})\rangle\!\rangle \\
&\triangleq \mathcal{I}^R \lhd \Big((\widehat{x}\,(x \circledcirc (\llbracket P\rrbracket^{*\mathcal{X}}\{\beta_1/\beta\}\widehat{\beta}_1 \nearrow \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}})) \circledcirc \gamma)\,\widehat{\gamma}\cdot\beta_2)\widehat{\beta}_2 \dagger \widehat{y}(y \circledcirc \llbracket R\rrbracket^{*\mathcal{X}})\Big) \rhd \mathcal{O}^R \\
&\triangleq \llbracket(\widehat{x}\,(P\widehat{\beta} \nearrow \widehat{y}R)\,\widehat{\gamma}\cdot\beta)\widehat{\beta} \dagger \widehat{y}R\rrbracket^{*\mathcal{X}}
\end{aligned}
$$

• Take the (*↗-gc*) rule: $P\widehat{\alpha} \nearrow \widehat{x}Q \;\to\; P$, if $\alpha \notin N(P)$. Assume $N(P) \cap N(Q) = \emptyset$, we have:

$$
\begin{aligned}
\llbracket P\widehat{\alpha} \nearrow \widehat{x}Q\rrbracket^{*\mathcal{X}} &\triangleq (\llbracket P\rrbracket^{*\mathcal{X}} \odot \alpha)\widehat{\alpha} \nearrow \widehat{x}(x \circledcirc \llbracket Q\rrbracket^{*\mathcal{X}}) \\
&\to \mathcal{I}^{\llbracket Q\rrbracket^{*\mathcal{X}}} \odot \llbracket P\rrbracket^{*\mathcal{X}} \odot \mathcal{O}^{\llbracket Q\rrbracket^{*\mathcal{X}}}, \\
&\quad \text{which is what we expected.}
\end{aligned}
$$

Thus we are done with the proof. □

Preservation of types. We now show that the encoding preserves types. In the typed $\mathcal{X}$ calculus contexts Γ and Δ may contain some auxiliary pairs (name,type). This is due to the fact that weakening is implicit in $\mathcal{X}$, i.e., it is not controlled explicitly. We have to keep that in mind when formulating the lemma.

Lemma 38 (Preservation of types). *If P is an arbitrary $\mathcal{X}$-term such that $P :\cdot\ \Gamma \vdash \Delta$, then*

$$((dom(\Gamma)) \setminus I(P)) \odot [\![P]\!]^{*\mathcal{X}} \odot ((dom(\Delta)) \setminus O(P)) :\cdot\ \Gamma \vdash \Delta$$

Proof. The proof works by case analysis and induction on the structure of terms. We give the detail for encoding of capsule and exporter, whereas the other cases work the same way.

• Rule: $[\![\langle x.\alpha \rangle]\!]^{*\mathcal{X}} := \langle x.\alpha \rangle$.
If $\langle x.\alpha \rangle :\cdot\ \Gamma \vdash \Delta$ where $x : A \in \Gamma$ and $\alpha : A \in \Delta$, then, in ${}^*\mathcal{X}$ we have: $\langle x.\alpha \rangle :\cdot\ x : A \vdash \alpha : A$, which is equivalent to:

$$(dom(\Gamma) \setminus x) \odot \langle x.\alpha \rangle \odot (dom(\Delta) \setminus \alpha) :\cdot\ \Gamma \vdash \Delta$$

• Rule: $[\![\widehat{x}\, P\, \widehat{\beta} \cdot \alpha]\!]^{*\mathcal{X}} := \left(\widehat{x}\, (x \circledcirc [\![P]\!]^{*\mathcal{X}} \circledcirc \beta)\, \widehat{\beta} \cdot \alpha\right) \rhd \alpha$.
If we assume the most generic case, namely for $x, \beta \notin N(P)$ and $\alpha \in N(P)$, then the encoding gives:

$$[\![\widehat{x}\, P\, \widehat{\beta} \cdot \alpha]\!]^{*\mathcal{X}} := [\widehat{x}\, (x \odot ([\![P]\!]^{*\mathcal{X}} \{\alpha_1/\alpha\}) \odot \beta)\, \widehat{\beta} \cdot \alpha_1 \rangle_{\widehat{\alpha_2}}^{\widehat{\alpha_1}} > \alpha$$

On the one hand we have:

$$\frac{P :\cdot\ \Gamma \vdash \alpha : A \to B, \Delta}{\widehat{x}\, P\, \widehat{\beta} \cdot \alpha :\cdot\ \Gamma \vdash \alpha : A \to B, \Delta}\ (\to R)$$

where, as stated previously, $x : A \in \Gamma,\ \beta : B \in \Delta$.

On the other hand,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![P]\!]^{*\mathcal{X}} :\cdot\ \Gamma \vdash \alpha : A \to B, \Delta}{[\![P]\!]^{*\mathcal{X}} \{\alpha_1/\alpha\} :\cdot\ \Gamma \vdash \alpha_1 : A \to B, \Delta}\ (ren)}{x \odot [\![P]\!]^{*\mathcal{X}} \{\alpha_1/\alpha\} :\cdot\ \Gamma, x : A \vdash \alpha_1 : A \to B, \Delta}\ (weak\text{-}L)}{x \odot [\![P]\!]^{*\mathcal{X}} \{\alpha_1/\alpha\} \odot \beta :\cdot\ \Gamma, x : A \vdash \alpha_1 : A \to B, \beta : B, \Delta}\ (weak\text{-}R)}{\widehat{x}\, (x \odot [\![P]\!]^{*\mathcal{X}} \{\alpha_1/\alpha\} \odot \beta)\, \widehat{\beta} \cdot \alpha :\cdot\ \Gamma \vdash \alpha_1 : A \to B, \alpha_2 : A \to B, \Delta}\ (\to R)}{[\widehat{x}\, (x \odot [\![P]\!]^{*\mathcal{X}} \{\alpha_1/\alpha\} \odot \beta)\, \widehat{\beta} \cdot \alpha \rangle_{\widehat{\alpha_2}}^{\widehat{\alpha_1}} > \alpha :\cdot\ \Gamma \vdash \alpha : A \to B, \Delta}\ (cont\text{-}R)$$

□

4.2. From ${}^*\mathcal{X}$ to $\mathcal{X}$

Now we investigate the opposite direction. We show how to represent ${}^*\mathcal{X}$-terms by $\mathcal{X}$-terms and then we show how ${}^*\mathcal{X}$-reductions are simulated by $\mathcal{X}$-reductions.

Definition 39 (Encoding ${}^*\mathcal{X}$ into $\mathcal{X}$). The encoding of ${}^*\mathcal{X}$-terms in $\mathcal{X}$ calculus is defined inductively as shown by Figure 6.2.

$$
\begin{array}{rcl}
\llbracket \langle x.\alpha \rangle \rrbracket^{\mathcal{X}} & := & \langle x.\alpha \rangle \\
\llbracket \widehat{x}\, P\, \widehat{\beta} \cdot \alpha \rrbracket^{\mathcal{X}} & := & \widehat{x}\, \llbracket P \rrbracket^{\mathcal{X}}\, \widehat{\beta} \cdot \alpha \\
\llbracket P\, \widehat{\alpha}\ [x]\ \widehat{y}\, Q \rrbracket^{\mathcal{X}} & := & \llbracket P \rrbracket^{\mathcal{X}}\, \widehat{\alpha}\ [x]\ \widehat{y}\, \llbracket Q \rrbracket^{\mathcal{X}} \\
\llbracket P\widehat{\alpha} \dagger \widehat{x}Q \rrbracket^{\mathcal{X}} & := & \llbracket P \rrbracket^{\mathcal{X}} \widehat{\alpha} \dagger \widehat{x} \llbracket Q \rrbracket^{\mathcal{X}} \\
\llbracket x <^{\widehat{y}}_{\widehat{z}} \langle P \rangle \rrbracket^{\mathcal{X}} & := & \llbracket P \rrbracket^{\mathcal{X}} \{x/y\}\{x/z\} \\
\llbracket [P\rangle^{\widehat{\beta}}_{\widehat{\gamma}} > \alpha \rrbracket^{\mathcal{X}} & := & \llbracket P \rrbracket^{\mathcal{X}} \{\alpha/\beta\}\{\alpha/\gamma\} \\
\llbracket x \odot P \rrbracket^{\mathcal{X}} & := & \llbracket P \rrbracket^{\mathcal{X}} \\
\llbracket P \odot \alpha \rrbracket^{\mathcal{X}} & := & \llbracket P \rrbracket^{\mathcal{X}}
\end{array}
$$

Figure 18: Encoding the ${}^*\mathcal{X}$-terms into $\mathcal{X}$

Encodings are defined without considering the active cuts but it is not difficult to extend it:

$$\llbracket P\widehat{\alpha} \nearrow \widehat{x}Q \rrbracket^{\mathcal{X}} := \llbracket P \rrbracket^{\mathcal{X}} \widehat{\alpha} \nearrow \widehat{x} \llbracket Q \rrbracket^{\mathcal{X}}$$

$$\llbracket P\widehat{\alpha} \nwarrow \widehat{x}Q \rrbracket^{\mathcal{X}} := \llbracket P \rrbracket^{\mathcal{X}} \widehat{\alpha} \nwarrow \widehat{x} \llbracket Q \rrbracket^{\mathcal{X}}$$

The encoding $\llbracket\ \rrbracket^{\mathcal{X}}$ does the opposite to $\llbracket\ \rrbracket^{{}^*\mathcal{X}}$. Namely, it simply removes erasers and duplicators from terms (some renamings are also performed). That is the reason for a possible decrease of free names after the encoding.

Lemma 40 (Properties of $\llbracket\ \rrbracket^{\mathcal{X}}$). *The encoding $\llbracket\ \rrbracket^{\mathcal{X}}$ satisfies the following:*

1. $N(P) \subseteq N(\llbracket P \rrbracket^{\mathcal{X}})$

2. $\llbracket P \rrbracket^{\mathcal{X}}\{x/y\} = \llbracket P\{x/y\} \rrbracket^{\mathcal{X}}$ *if* $x \notin N(P)$

PROOF. The former statement can be checked by carefully inspecting the encoding rules, and the later by case analysis and induction on the structure of terms. □

The computation in ${}^{*}\mathcal{X}$ is simulated by computation in $\mathcal{X}$ in the way expressed by Theorem 42; each reduction step is mapped into one or more reduction steps.

Lemma 41. *Let P be an ${}^{*}\mathcal{X}$-term, and $\llbracket P \rrbracket^{\mathcal{X}}$ its encoding in $\mathcal{X}$. Then the following holds:*

1. $\alpha, x \notin N(P) \rightarrow \alpha, x \notin N(\llbracket P \rrbracket^{\mathcal{X}})$
2. $\alpha, x \in N(\llbracket P \rrbracket^{\mathcal{X}}) \rightarrow \alpha, x \in N(P)$

PROOF. Trivially by inspecting encoding rules. Names are lost during encoding only if they are introduced in ${}^{*}\mathcal{X}$ by weakening.

Theorem 42 (Simulating ${}^{*}\mathcal{X}$-reduction). *Let P and P' be $\mathcal{X}$-terms. Then the following holds:*

$$\textit{If } P \xrightarrow{{}^{*}\mathcal{X}} P' \textit{ then } \llbracket P \rrbracket^{\mathcal{X}} \xrightarrow{\mathcal{X}} + \llbracket P' \rrbracket^{\mathcal{X}}$$

PROOF. The proof goes by inspecting the reduction rules and by induction on the structure of terms. We provide the proof for several reduction rules.

Logical rules.

- Take the $(ren - L)$ rule: $\langle y.\alpha \rangle \widehat{\alpha} \dagger \widehat{x} Q \rightarrow Q\{y/x\}$. We have:

$$\begin{array}{rcl} \llbracket \langle y.\alpha \rangle \widehat{\alpha} \dagger \widehat{x} Q \rrbracket^{\mathcal{X}} & \triangleq & \llbracket \langle y.\alpha \rangle \rrbracket^{\mathcal{X}} \widehat{\alpha} \dagger \widehat{x} \llbracket Q \rrbracket^{\mathcal{X}} \\ & \triangleq & \langle y.\alpha \rangle \widehat{\alpha} \dagger \widehat{x} \llbracket Q \rrbracket^{\mathcal{X}} \\ & \xrightarrow{ren\text{-}L} & \llbracket Q \rrbracket^{\mathcal{X}}\{y/x\} \\ & = & \llbracket Q\{y/x\} \rrbracket^{\mathcal{X}} \end{array}$$

- Take the $(ren - R)$ rule: $P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta \rangle \rightarrow P\{\beta/\alpha\}$. We have :

$$\begin{array}{rcl} \llbracket P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta \rangle \rrbracket^{\mathcal{X}} & \triangleq & \llbracket P \rrbracket^{\mathcal{X}} \widehat{\alpha} \dagger \widehat{x} \llbracket \langle x.\beta \rangle \rrbracket^{\mathcal{X}} \\ & \triangleq & \llbracket P \rrbracket^{\mathcal{X}} \widehat{\alpha} \dagger \widehat{x}\langle x.\beta \rangle \\ & \xrightarrow{ren\text{-}R} & \llbracket P \rrbracket^{\mathcal{X}}\{\beta/\alpha\} \\ & = & \llbracket P\{\beta/\alpha\} \rrbracket^{\mathcal{X}} \end{array}$$

Activation rules.

• Take the $(act-L)$ rule: $P\widehat{\alpha} \dagger \widehat{x}Q \rightarrow P\widehat{\alpha} \nearrow \widehat{x}Q$, if α not L-principal for P. We have:

$$\begin{array}{rcl} [\![P\widehat{\alpha} \dagger \widehat{x}Q]\!]^{\mathcal{X}} & \triangleq & [\![P]\!]^{\mathcal{X}}\widehat{\alpha} \dagger \widehat{x}[\![Q]\!]^{\mathcal{X}} \\ & \xrightarrow{Lem.36} & [\![P]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\ & \triangleq & [\![P\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} \end{array}$$

Similarly for the rule $(act-R)$.

Dectivation rules.

• Take the $(\nearrow\text{-}deact)$ rule: $P\widehat{\alpha} \nearrow \widehat{x}Q \rightarrow P\widehat{\alpha} \dagger \widehat{x}Q$, if α is L-principal for P. We have:

$$\begin{array}{rcl} [\![P\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} & \triangleq & [\![P]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\ & \xrightarrow{Lem.36} & [\![P]\!]^{\mathcal{X}}\widehat{\alpha} \dagger \widehat{x}[\![Q]\!]^{\mathcal{X}} \\ & \triangleq & [\![P\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} \end{array}$$

Similarly for the rule $(\nwarrow\text{-}deact)$.

Structural rules.

• Take the $(\nearrow\text{-}eras)$ rule: $(P \odot \alpha)\widehat{\alpha} \nearrow \widehat{x}Q \rightarrow \mathcal{I}^Q \odot P \odot \mathcal{O}^Q$. We have:

$$\begin{array}{rcl} [\![(P \odot \alpha)\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} & \triangleq & [\![P \odot \alpha]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\ & \triangleq & [\![P]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}},\ \alpha \notin N(P) \\ & \xrightarrow{\nearrow\text{-}gc} & [\![P]\!]^{\mathcal{X}} \\ & \triangleq & [\![\mathcal{I}^Q \odot P \odot \mathcal{O}^Q]\!]^{\mathcal{X}} \end{array}$$

• Take the $(\nearrow\text{-}dupl)$ rule: $([P\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}>\alpha)\widehat{\alpha} \nearrow \widehat{x}Q \rightarrow P\langle\!\langle \widehat{\alpha_1}\widehat{\alpha_2} \nearrow \widehat{x}Q \rangle\!\rangle$. We analyze here several cases of P.
- Take $P = \widehat{y}\, R\, \widehat{\gamma} \cdot \beta,\ \beta \neq \alpha_1, \alpha_2$. By definition of ${}^*\mathcal{X}$ terms $\alpha_1, \alpha_2 \in N(R)$. Notice that P is of the form $P^{\beta}\{R\}$. We have:

$$
\begin{array}{lll}
\llbracket([\widehat{y}\,R\,\widehat{\gamma}\cdot\beta\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q\rrbracket^{\mathcal{X}} & \triangleq & \llbracket[\widehat{y}\,R\,\widehat{\gamma}\cdot\beta\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha\rrbracket^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & (\llbracket\widehat{y}\,R\,\widehat{\gamma}\cdot\beta\rrbracket^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & ((\widehat{y}\,\llbracket R\rrbracket^{\mathcal{X}}\,\widehat{\gamma}\cdot\beta)\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & (\widehat{y}\,(\llbracket R\rrbracket^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\,\widehat{\gamma}\cdot\beta)\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \xrightarrow{\nearrow\text{-}prop} & \widehat{y}\,((\llbracket R\rrbracket^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}Q)\,\widehat{\gamma}\cdot\beta \\
& \triangleq & \widehat{y}\,(\llbracket[R\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha\rrbracket^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}Q)\,\widehat{\gamma}\cdot\beta \\
& \triangleq & \widehat{y}\,\llbracket([R\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q\rrbracket^{\mathcal{X}}\,\widehat{\gamma}\cdot\beta \\
& \triangleq & \llbracket\widehat{y}\,(([R\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q)\,\widehat{\gamma}\cdot\beta\rrbracket^{\mathcal{X}} \\
& \triangleq & \llbracket P^{\beta}\{R\}\langle\langle\widehat{\alpha_1\alpha_2} \nearrow \widehat{x}Q\rangle\rangle\rrbracket^{\mathcal{X}},\ \text{when } \beta \neq \alpha_1, \alpha_2, \\
& & \text{by def. of simultaneous subst. on page 22.}
\end{array}
$$

- Take $P = \widehat{y}\,R\,\widehat{\gamma}\cdot\alpha_1$. By definition of ${}^{*}\mathcal{X}$ terms $a_2 \in N(R), \alpha_1 \notin N(R)$. Notice that P is of the form $P^{\alpha_1}\{R\},\ \alpha_2 \in R$. We have:

$$
\begin{array}{lll}
& & \llbracket([\widehat{y}\,R\,\widehat{\gamma}\cdot\alpha_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q\rrbracket^{\mathcal{X}} \\
& \triangleq & \llbracket[\widehat{y}\,R\,\widehat{\gamma}\cdot\alpha_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha\rrbracket^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & (\llbracket\widehat{y}\,R\,\widehat{\gamma}\cdot\alpha_1\rrbracket^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & ((\widehat{y}\,\llbracket R\rrbracket^{\mathcal{X}}\,\widehat{\gamma}\cdot\alpha_1)\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & (\widehat{y}\,(\llbracket R\rrbracket^{\mathcal{X}}\{\alpha/\alpha_2\})\,\widehat{\gamma}\cdot\alpha)\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \xrightarrow{\nearrow\text{-}prop\text{-}dupl\text{-}deact} & (\widehat{y}\,((\llbracket R\rrbracket^{\mathcal{X}}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}})\,\widehat{\gamma}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& = & (\widehat{y}\,(\llbracket R\{\alpha/\alpha_2\}\rrbracket^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}})\,\widehat{\gamma}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & (\widehat{y}\,\llbracket R\{\alpha/\alpha_2\}\widehat{\alpha} \nearrow \widehat{x}Q\rrbracket^{\mathcal{X}}\,\widehat{\gamma}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & \llbracket\widehat{y}\,(R\{\alpha/\alpha_2\}\widehat{\alpha} \nearrow \widehat{x}Q)\,\widehat{\gamma}\cdot\alpha\rrbracket^{\mathcal{X}}\widehat{\alpha} \dagger \widehat{x}\llbracket Q\rrbracket^{\mathcal{X}} \\
& \triangleq & \llbracket\mathcal{I}^{Q} {}^{\widehat{\mathcal{I}_1^Q}}\!\!<_{\widehat{\mathcal{I}_2^Q}}\langle(\widehat{y}\,(R\widehat{\alpha_2} \nearrow \widehat{x}_2Q_2)\,\widehat{\gamma}\cdot\alpha_1)\widehat{\alpha_1} \dagger \widehat{x}_1Q_1\rangle^{\widehat{\mathcal{O}_1^Q}}_{\widehat{\mathcal{O}_2^Q}}{>}\mathcal{O}^{Q}\rrbracket^{\mathcal{X}} \\
& \triangleq & \llbracket P^{\alpha_1}\{R\}\langle\langle\widehat{\alpha_1\alpha_2} \nearrow \widehat{x}Q\rangle\rangle\rrbracket^{\mathcal{X}}, \\
& & \text{by def. of simultaneous subst. on page 22.}
\end{array}
$$

- Take $P = R_1\,\widehat{\gamma}\ [y]\ \widehat{z}\,R_2$, and assume $\alpha_1, \alpha_2 \in N(R_1)$. Notice that P is of the form $P^{y}\{R_1,\, R_2\}$. We have:

$$
\begin{array}{rl}
 & [\![([R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} \\
\triangleq & [\![[R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & ([\![R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2]\!]^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & (([\![R_1]\!]^{\mathcal{X}}\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2]\!]^{\mathcal{X}})\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & (([\![R_1]\!]^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_1\})\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2]\!]^{\mathcal{X}})\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & ([\![R_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha]\!]^{\mathcal{X}}\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2]\!]^{\mathcal{X}})\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\xrightarrow{\nearrow\text{-}prop\text{-}dupl_1} & ([\![R_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}})\,\widehat{\gamma}\,[y]\,\widehat{z}\,([\![R_2]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}}) \\
\xrightarrow{\nearrow\text{-}gc} & ([\![R_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}})\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2]\!]^{\mathcal{X}} \\
\triangleq & [\![([R_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}}\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2]\!]^{\mathcal{X}} \\
\triangleq & [\![(([R_1\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q)\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2]\!]^{\mathcal{X}} \\
\triangleq & [\![P^y\{R_1,\,R_2\}\langle\!\langle\widehat{\alpha_1\alpha_2} \nearrow \widehat{x}Q\rangle\!\rangle]\!]^{\mathcal{X}},\ \text{when } \alpha_1,\alpha_2 \in N(R_1)
\end{array}
$$

- Take $P = R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2$, and assume $\alpha_1 \in N(R_1),\ \alpha_2 \in N(R_2)$. We have:

$$
\begin{array}{rl}
 & [\![([R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha)\widehat{\alpha} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} \\
\triangleq & [\![[R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2\rangle^{\widehat{\alpha_1}}_{\widehat{\alpha_2}}{>}\alpha]\!]^{\mathcal{X}}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & ([\![R_1\,\widehat{\gamma}\,[y]\,\widehat{z}\,R_2]\!]^{\mathcal{X}}\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & (([\![R_1]\!]^{\mathcal{X}}\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2]\!]^{\mathcal{X}})\{\alpha/\alpha_1\}\{\alpha/\alpha_2\})\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\triangleq & (([\![R_1]\!]^{\mathcal{X}}\{\alpha/\alpha_1\})\,\widehat{\gamma}\,[y]\,\widehat{z}\,([\![R_2]\!]^{\mathcal{X}}\{\alpha/\alpha_2\}))\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}} \\
\xrightarrow{\nearrow\text{-}prop\text{-}dupl_1} & ([\![R_1]\!]^{\mathcal{X}}\{\alpha/\alpha_1\}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}})\,\widehat{\gamma}\,[y]\,\widehat{z}\,([\![R_2]\!]^{\mathcal{X}}\{\alpha/\alpha_2\}\widehat{\alpha} \nearrow \widehat{x}[\![Q]\!]^{\mathcal{X}}) \\
\triangleq & [\![R_1\widehat{\alpha_1} \nearrow \widehat{x}Q]\!]^{\mathcal{X}}\,\widehat{\gamma}\,[y]\,\widehat{z}\,[\![R_2\widehat{\alpha_2} \nearrow \widehat{x}Q]\!]^{\mathcal{X}} \\
\triangleq & [\![\mathcal{I}^Q <^{\widehat{\mathcal{I}^Q_1}}_{\widehat{\mathcal{I}^Q_2}} \langle(R_1\widehat{\alpha_1} \nearrow \widehat{x}_1Q_1)\,\widehat{\gamma}\,[y]\,\widehat{z}\,(R_2\widehat{\alpha_2} \nearrow \widehat{x}_2Q_2)\rangle^{\widehat{\mathcal{O}^Q_1}}_{\widehat{\mathcal{O}^Q_2}}{>}\mathcal{O}^Q]\!]^{\mathcal{X}} \\
\triangleq & [\![P^y\{R_1,\,R_2\}\langle\!\langle\widehat{\alpha_1\alpha_2} \nearrow \widehat{x}Q\rangle\!\rangle]\!]^{\mathcal{X}},\ \text{when } \alpha_1 \in N(R_1), \alpha_2 \in N(R_2)
\end{array}
$$

The proof for propagation group of rules is straightforward. □

4.3. Strong normalisation of ${}^*\mathcal{X}$

Exploiting the strong normalisation property of simply typed $\mathcal{X}$ [36], we prove that ${}^*\mathcal{X}$ is strongly normalising. We first prove that the previously defined encoding of ${}^*\mathcal{X}$ into $\mathcal{X}$ preserves typeability.

Lemma 43 (Preservation of types). *For an arbitrary* ${}^*\mathcal{X}$*-term* P *such that* $P :\cdot\ \Gamma \vdash \Delta$*, it stands*

$$[\![P]\!]^{\mathcal{X}} :\cdot\ \Gamma \vdash \Delta$$

PROOF. By induction on typing derivations along the lines of Lemma 38. □

This section presents the proof of strong normalisation for ${}^*\mathcal{X}$ calculus.

Theorem 44 (Strong Normalisation). *The reduction system of ${}^*\mathcal{X}$ is strongly normalising on simply-typed terms.*

PROOF. Let $P :\cdot\ \Gamma \vdash \Delta$. Assume that P is not strongly normalising, which means that there is an infinite reduction starting with P

$$P \xrightarrow{{}^*\mathcal{X}} P_1 \xrightarrow{{}^*\mathcal{X}} \ldots \xrightarrow{{}^*\mathcal{X}} P_n \xrightarrow{{}^*\mathcal{X}} \ldots$$

then by Theorem 42,

$$[\![P]\!]^{\mathcal{X}} \xrightarrow{\mathcal{X}}+ [\![P_1]\!]^{\mathcal{X}} \xrightarrow{\mathcal{X}}+ \ldots \xrightarrow{\mathcal{X}}+ [\![P_n]\!]^{\mathcal{X}} \xrightarrow{\mathcal{X}}+ \ldots$$

On the other hand according to Lemma 43,

$$[\![P]\!]^{\mathcal{X}} :\cdot\ \Gamma \vdash \Delta$$

and the fact that $\mathcal{X}$ calculus is strongly normalising on typed terms ([36]), we conclude that $[\![P]\!]^{\mathcal{X}}$ is strongly normalising, which contradicts the assumption. Hence, P is strongly normalising. □

5. Conclusions

We have presented two calculi implementing the Curry-Howard correspondence for classical logic sequent calculi. The first one, called $\mathcal{X}$ provides terms for sequent proofs in the calculus $G3$ and a description of cut elimination by reductions. A type system for this calculus assigns types to terms. The type of a term is the proposition that the proof associated with the term proves. We designed the calculus ${}^*\mathcal{X}$ in some sense as an extension of $\mathcal{X}$ with rules for explicit structural rules known in the sequent calculus $G1$ as *weakening* and *contraction*. In ${}^*\mathcal{X}$, the operator associated with weakening is an erasure and the operator associated with contraction is a duplication. Like $\mathcal{X}$, ${}^*\mathcal{X}$ is associated with a type system to represent proofs in a sequent calculus with weakening and contraction. We have explored the connection between the logic calculus $G3$ (resp. $G1$) and its implementation $\mathcal{X}$ (resp. ${}^*\mathcal{X}$). We have also shown how $\mathcal{X}$ can be embedded in ${}^*\mathcal{X}$ and vice-versa. As a low level language, it reveals details in both, structure of terms and

computation, but in the same time this explicitness yields the essence of classical proofs and classical computations. We know that the λ-calculus is the framework of functional sequential programming and ${}^*\mathcal{X}$ can be seen as an extension of λ-calculus. An interesting direction for future work could be to explore the connections between ${}^*\mathcal{X}$ and non deterministic distributed calculi like what has been done by van Bakel, Cardelli and Vigliotti [41].

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

[2] Audebaud, P., van Bakel, S., 2007. A completeness result for λ_μ, preprint.

[3] Barbanera, F., Berardi, S., 1994. A symmetric lambda calculus for "classical" program extraction. In: TACS. pp. 495–515.

[4] Barbanera, F., Berardi, S., Schivalocchi, M., 1997. "Classical" programming-with-proofs in λ^{sym}: an analysis of non-confluence. In: TACS. pp. 365–390.

[5] Barendregt, H., Ghilezan, S., 2000. Lambda terms for natural deduction, sequent calculus and cut-elimination. J. Funct. Programming 10 (1), 121–134.

[6] Bloo, R., Rose, K., 1995. Preservation of strong normalisation in named lambda calculi with explicit substitution and garbage collection. In: CSN'95 Computer Science in the Netherlands. pp. 62–72.
URL `ftp://ftp.diku.dk/diku/semantics/papers/D-246.ps`

[7] Curien, P.-L., Herbelin, H., 2000. The duality of computation. In: Proc. 5 th ACM SIGPLAN Int. Conf. on Functional Programming (ICFP'00). ACM, pp. 233–243.

[8] Danos, V., Joinet, J.-B., Schellinx, H., 1996. Computational isomorphisms in classical logic (extended abstract). Electronic Notes in Theoretical Computer Science 3.

[9] Danos, V., Joinet, J.-B., Schellinx, H., 1997. A new deconstructive logic: Linear logic. Journal of Symbolic Logic 62.

[10] David, R., Guillaume, B., 2001. A lambda-calculus with explicit weakening and explicit substitution. Mathematical Structures in Computer Science 11 (1), 169–206.

[11] Dougherty, D., Ghilezan, S., Lescanne, P., 2008. Characterizing strong normalization in the Curien-Herbelin symmetric lambda calculus: extending the Coppo-Dezani heritage. Theor. Comput. Sci. 398 (1-3), 114–128.

[12] Dougherty, D., Ghilezan, S., Lescanne, P., Likavec, S., 2005. Strong normalization of the dual classical sequent calculus. In: 12th Int. Conf. LPAR. Vol. 3835 of Lecture Notes in Computer Science. pp. 169–183.

[13] Espírito Santo, J., 2007. Completing Herbelin's programme. In: Proceedings of Types Lambda Calculus and Application, TLCA'07. Vol. 4583 of LNCS. pp. 118–132.

[14] Espírito Santo, J., Ghilezan, S., Ivetić, J., 2008. Characterising strongly normalising intuitionistic sequent terms. In: International Workshop TYPES'07 (Selected Papers). Vol. 4941 of Lecture Notes in Computer Science. pp. 85–99.

[15] Gentzen, G., 1935. Untersuchungen über das logische Schließen. Math. Z. 39, 176–210, 405–431.

[16] Ghilezan, S., 2007. Terms for natural deduction, sequent calculus and cut elimination in classical logic. In: Reflections on Type Theory, Lambda Calculus, and the Mind - Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday.
URL `http://www.cs.ru.nl/barendregt60/essays/ghilezan/`

[17] Girard, J.-Y., 2001. Locus solum: From the rules of logic to the logic of rules. Mathematical Structures in Computer Science 11 (3), 301–506.

[18] Girard, J.-Y., Lafont, Y., Taylor, P., 1989. Proofs and Types. Vol. 7 of Cambridge Tracts in Theoret Computer Science. Cambridge University Press.

[19] Griffin, T., 1990. A formulae-as-types notion of control. In: Proceedings of the 17th ACM symposium on Principles of programming languages, POPL. pp. 47–58.

[20] Herbelin, H., 1995. Séquents qu'on calcule: de l'interprétation du calcul des séquents comme calcul de λ-termes et comme calcul de stratégies gagnantes. Thèse de doctorat, Université Paris VII.

[21] Hyland, J. M. E., 2002. Proof theory in the abstract. Annals of Pure and Applied Logic 114 (1-3), 43–78.

[22] Kesner, D., Lengrand, S., 2005. Extending the explicit substitution paradigm. In: RTA. pp. 407–422.

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

[24] Kesner, D., Renaud, F., 2009. The prismoid of resources. In: Královic, R., Niwinski, D. (Eds.), MFCS. Vol. 5734 of Lecture Notes in Computer Science. Springer, pp. 464–476.

[25] Kesner, D., Renaud, F., 2011. A prismoid framework for languages with resources. Theor. Comput. Sci. 412 (37), 4867–4892.

[26] Kleene, S., 1952. Introduction to Metamathematics. No. 1 in Bibliotheca mathematica. North-Holland, revised edition, Wolters-Noordhoff, 1971.

[27] Lafont, Y., 1995. From proof-nets to interaction nets. In: Advances in linear logic. Cambridge University Press, pp. 225–247.

[28] Lengrand, S., 2003. Call-by-value, call-by-name, and strong normalization for the classical sequent calculus. In: Electronic Notes in Theoretical Computer Science. Vol. 86.

[29] Lescanne, P., Zunic, D., 2008. Computing with diagrams in classical logic. In: Inf. Proc. of WRS, Reduction Strategies in Rewriting and Programming. Vol. 08-09. Research Institute for Symbolic Computation, Linz, Austria, pp. 91–109.

[30] Milner, R., 1995. Communication and concurrency. Prentice Hall International (UK) Ltd., Hertfordshire, UK.

[31] Parigot, M., 1992. An algorithmic interpretation of classical natural deduction. In: Int. Conf. LPAR. Vol. 624 of Lecture Notes in Computer Science. pp. 190–201.

[32] Rose, K., Bloo, R., Lang, F., 2011. On explicit substitution with names. Journal of Automated Reasoning, 1–26.

[33] Sangiorgi, D., Walker, D., 2001. π-Calculus: A Theory of Mobile Processes. Cambridge University Press, New York, USA.

[34] Troelstra, A. S., Schwichtenberg, H., 1996. Basic Proof Theory. Cambridge University Press, New York, NY, USA.

[35] Urban, C., 2000. Classical logic and computation. Ph.D. thesis, Univ. of Cambridge.

[36] Urban, C., 2001. Strong normalisation for a Gentzen-like cut-elimination procedure. In: Typed Lambda Calculus and Applications. Vol. 2044 of Lecture Notes in Computer Science. pp. 415–429.

[37] Urban, C., Bierman, G. M., 1999. Strong normalisation of cut-elimination in classical logic. In: Typed Lambda Calculus and Applications, TLCA'99. Vol. 1581 of Lecture Notes in Computer Science. pp. 365–380.

[38] Urban, C., Bierman, G. M., 2001. Strong normalisation of cut-elimination in classical logic. Fundamenta Informaticae 45 (1-2), 123–155, (appeared also at *TLCA* in 1999).

[39] Urban, C., Bierman, G. M., 2001. Strong normalisation of cut-elimination in classical logic. Fundam. Inf. 45 (1,2), 123–155.

[40] van Bakel, S., 2012. Completeness and soundness results for $\mathcal{X}$ with intersection and union types. Fundamenta Informaticae To appear.

[41] van Bakel, S., Cardelli, L., Vigliotti, M. G., 2011. From $\mathcal{X}$ to π; representing the classical sequent calculus in the pi-calculus. CoRR abs/1109.4817.

[42] van Bakel, S., Lengrand, S., Lescanne, P., 2005. The language $\mathcal{X}$: circuits, computations and classical logic. In: Proc.9th Italian Conf. on Theoretical Computer Science (ICTCS'05). Vol. 3701 of Lecture Notes in Computer Science. pp. 81–96.

[43] van Bakel, S., Lescanne, P., 2008. Computation with classical sequents. Mathematical Structures in Computer Science 18 (3), 555–609.

[44] Žunić, D., 2007. Computing with sequent and diagrams in classical logic - calculi ${}^{*}\mathcal{X}$, ${}^{©}\mathcal{X}$ and ${}^{d}\mathcal{X}$. Ph.D. thesis, Ecole Normale Supériéure de Lyon, France.
URL `http://tel.archives-ouvertes.fr/tel-00265549`

[45] Wadler, P., 2003. Call-by-value is dual to call-by-name. In: Proc.8th Int. Conf. on Functional Programming.

[46] Whitehead, A. N., Russell, B., 1925. Principia Mathematica, 2nd Edition. Cambridge University Press.