Computation with Classical Sequents

(Mathematical Structures in Computer Science, 18:555-609, 2008)

Steffen van Bakel* and Pierre Lescanne

Department of Computing, Imperial College London, 180 Queen's Gate London SW7 2BZ, U.K.
École Normale Supérieure de Lyon, 46 Allée d'Italie 69364 Lyon 07, FRANCE
svb@doc.ic.ac.uk,Pierre.Lescanne@ens-lyon.fr

November 10, 2005; Revised June 27, 2007

Abstract

$\mathcal{X}$ is an untyped continuation-style formal language with a typed subset which provides a Curry-Howard isomorphism for a sequent calculus for implicative classical logic. $\mathcal{X}$ can also be viewed as a language for describing nets by composition of basic components connected by wires. These features make $\mathcal{X}$ an expressive platform on which many different (applicative) programming paradigms can be mapped. In this paper we will present the syntax and reduction rules for $\mathcal{X}$; in order to demonstrate its expressive power, we will show how elaborate calculi can be embedded, like the λ-calculus, Bloo and Rose's calculus of explicit substitutions $\lambda\mathbf{x}$, Parigot's $\lambda\mu$ and Curien and Herbelin's $\overline{\lambda}\mu\tilde{\mu}$.
$\mathcal{X}$ was first presented in [35] where it was called the $\lambda\xi$-calculus. It can be seen as the pure untyped computational content of the reduction system for implicative classical sequent calculus of [45].

1 Introduction

There exists a number of systems in the literature that link Classical Logic with a notion of computation. In the past, say before Herbelin's PhD [31] and Urban's PhD [45], the study of the relation between *computation*, *programming languages* and *logic* has concentrated mainly on *natural deduction systems*. In fact, these carry the predicate '*natural*' deservedly; in comparison with, for example, *sequent style systems*, natural deduction systems are easy to understand and reason about. This holds most strongly in the context of *non-classical* logics; for example, the relation between *Intuitionistic Logic* and the *Lambda Calculus* (with types) is well studied and understood, and has resulted in a vast and well investigated area of research, resulting in, amongst others, functional programming languages and much further to system F [26] and the Calculus of Constructions [18].

An example of an approach for representing classical proofs, Parigot's $\lambda\mu$-calculus [40] is a natural deduction system in which there is one main conclusion that is being manipulated and possibly several alternative ones. Adding conflict, $\bot$, as pseudo-type (only negation, or $A \to \bot$, is expressed; $\bot \to A$ is not a type), the $\lambda\mu$-calculus corresponds to *minimal classical logic* [4]. The link between Classical Logic and continuations and control was first established for the λ_C-Calculus [29] (where C stands for Felleisen's C operator).

The sequent calculus, introduced by Gentzen in [24], is a logical system in which the rules only introduce connectives (but on both sides of a sequent), in contrast to natural deduction which uses introduction and elimination rules. The only way to eliminate a connective is to eliminate the whole formula in which it appears, via an application of the (cut)-rule.

* Partially supported by École Normale Supérieure de Lyon, France

Gentzen's calculus for classical logic LK allows sequents of the form $A_1,\ldots,A_n \vdash B_1,\ldots,B_m$, where $A_1,\ldots,A_n$ is to be understood as $A_1\wedge\ldots\wedge A_n$ and $B_1,\ldots,B_m$ is to be understood as $B_1\vee\ldots\vee B_m$. Thus, LK appears as a very symmetrical system.

A symmetrical lambda calculus was defined in [10], essentially allowing an application to be interpreted in two ways, thus encapsulating the non-determinism of cut-elimination in Gentzen's LK. On the other hand, the implicational sequent calculus leads to the necessity of the left-introduction rules to manipulate hypotheses, as studied by Herbelin in [31, 19, 32]. The relation between call-by-name and call-by-value in the fragment of LK with negation and conjunction in Wadler's Dual Calculus is studied in [48]; as in calculi like $\lambda\mu$ and $\overline{\lambda}\mu\tilde{\mu}$, the Dual Calculus considers a logic with *active* formulae.

The (cut)-rule does not increase the expressive power of the system since a *cut-elimination procedure* has been defined that eliminates all applications of the (cut)-rule from the proof of a sequent, generating a proof in *normal form* of the same sequent, that is, without a cut. It is defined via rewriting steps, i.e., local reductions of the proof-tree, which has –with some discrepancies– the flavour of the evaluation of explicit substitutions [15, 1]. Indeed, the typing rule of an explicit substitution, say in $\lambda\mathbf{x}$ [14], is nothing else but a variant of the (cut)-rule, and a lot of work has been done to better understand the connection between explicit substitutions and local cut-reduction procedures.

This paper, which is a continuation of [35], presents a correspondence *à la* Curry-Howard for LK, bringing together the various features of two different approaches that we compare: that of Urban [45, 47, 46] and that of Curien and Herbelin [19]. Whereas Curien and Herbelin insist on the duality of the calculus, in [45], Urban analyses thoroughly, among other things, Gentzen-like cut-elimination procedures, and defines a very general reduction system for proofs in LK which is strongly normalising, and in which proofs are represented by a syntax of terms. $\mathcal{X}$ is actually the implicative part of his calculus with a new syntax.

In this paper, we will try and break a spear for the sequent-style approach, and make some further steps towards the development of a programming language based on cut-elimination for the sequent calculus for classical logic. We will present a language called $\mathcal{X}$ that describes nets, and its reduction rules that join nets. The logic we will consider contains only implication, but that is mainly because we, in this initial phase, aim for simplicity; cut-elimination in sequent calculi is notorious for the great number of rules, which will only increase manifold when considering more logical connectors.

To break with the natural deduction paradigm comes with a price, in that no longer *abstraction* and *application* (corresponding to *introduction of implication* and *modus ponens*) are the basic tools of the extracted language. In fact, the language we obtain is more of a *continuation* style, that models both the *parameter* as well as the *context* call. However, abstraction and application can be faithfully implemented, and we will show how $\mathcal{X}$ can be used to describe the behaviour of functional programming languages at a very low level of granularity.

$\mathcal{X}$ as a language for describing nets

The basic pieces of $\mathcal{X}$ can be understood as components with entrance and exit wires and ways to describe how to connect them to build larger nets. Those components will be quickly surveyed in the introduction and receive a more detailed treatment in Section 2. We call *"nets"* the structures we build, because they are made of components connected by wires.

$\mathcal{X}$ as a syntax for the sequent calculus

The origins of the work presented in this paper lie, first of all, in [34], where Lengrand defined a first approach to the definition of a calculus there called $\lambda\xi$ that would enjoy the Curry-

Howard property for Gentzen's Sequent Calculus. This became the starting point for the results presented here. During discussions in 2002 between Lengrand and the authors of this paper, it became clear that $\lambda\xi$ was similar to the notations for the sequent calculus as presented first by Urban in his PhD thesis [45]. This was then studied in relation with $\overline{\lambda}\mu\tilde{\mu}$ [19] by Lengrand [35] via the calculus $\lambda\xi$. The study of the connection between $\lambda\xi$ and $\overline{\lambda}\mu\tilde{\mu}$ as reported on in [35] was not fortuitous, and was taken as starting point for our investigations. We changed the name of the calculus from $\lambda\xi$ to $\mathcal{X}$ in order to avoid a clash with the (ξ)-rule of the λ-calculus, and because the use of λ unjustifiably suggests that abstraction is part of the syntax. In this paper, we will show in detail the interplay between $\mathcal{X}$ and $\overline{\lambda}\mu\tilde{\mu}$.

The natural context in which to study $\mathcal{X}$ is its role within the context of cut-elimination; for this, $\mathcal{X}$ is naturally typed. From this, it is but a natural, straightforward step to extend the research to an untyped $\mathcal{X}$. This opens the possibility to not only study the relation between $\mathcal{X}$ and other untyped calculi, but also to express recursion via a fixed-point construction, as well as studying normalisation and normalising reduction strategies, and semantics.

As we strongly believe in the importance of the syntax for a better grasping of the concepts, some of the contributions of this paper are to make the notation more intuitive and readable by moving to an infix notation, and to insist on the computational aspect[1]. This is achieved by studying $\mathcal{X}$ in the context of the normal functional programming languages paradigms, but, more importantly, to cut the link between $\mathcal{X}$ and Classical Logic. We achieve this by studying our language *without types*; this way, we also consider nets that do *not* correspond to proofs. In particular, we consider also non-termination nets. In fact, we aim to study $\mathcal{X}$ outside the context of Classical Logic in much the same way as the λ-calculus is studied outside the context of Intuitionistic Logic.

$\mathcal{X}$ as a fine grained operational model of computation

When taking the λ-calculus as a model for programming languages, the operational behaviour is provided by *β-contraction*. As is well known, β-contraction expresses how to calculate the value of a function applied to a parameter. In this, the parameter is used to instantiate occurrences of the bound variable in the body via the process of *substitution*. This description is rather basic as it says nothing on the actual *cost* of the substitution, which is quite high at run-time. Usually, a calculus of *explicit substitutions* [14, 1, 38, 37] is considered better suited for an accurate account of the substitution process and its implementation. When we refer to the calculus of explicit substitution we rather intend $\lambda\mathbf{x}$, the calculus of *explicit substitution with explicit names*, due to Bloo and Rose [14]. $\lambda\mathbf{x}$ gives a better account of substitution as it integrates substitutions as first class citizens, decomposes the process of inserting a term into atomic actions, and explains in detail how substitutions are distributed through terms to be eventually evaluated at the variable level.

In this paper, we will show that the level of description reached by explicit substitutions can in fact be greatly refined. In $\mathcal{X}$, we reach a 'subatomic' level by decomposing explicit substitutions into smaller components. At this level, the calculus $\mathcal{X}$ explains how substitutions and terms interact.

The calculus is actually symmetric [10] and, unlike $\lambda\mathbf{x}$ where a substitution is applied to a term, a term in $\mathcal{X}$ can also be applied to a substitution. Their interaction percolates subtly and gently through the term or substitution according to the direction that has been chosen. We will see that the these two kinds of interaction have a direct connection with call-by-value and call-by-name strategies, that both have a natural description in $\mathcal{X}$.

[1] The relation of $\mathcal{X}$ with its predecessors is the same as this of, say, mini-ML [17] with the lambda-calculus.

The ingredients of the syntax

It is important to note that $\mathcal{X}$ does *not* have variables[2] –like the LC or $\overline{\lambda}\mu\tilde{\mu}$– as possible places where terms might be inserted; instead, $\mathcal{X}$ has *wires*, also called *connectors*, that can occur free or bound in a term. As for the λ-calculus, the binding of a wire indicates that it is *active* in the computation; other than in the λ-calculus, however, the binding is not part of a term that is involved in the interaction, but is part of the interaction itself. There are two kinds of wires: *sockets* (that are reminiscent of values) and *plugs* (that are reminiscent of continuations), that correspond to *variables* and *co-variables*, respectively, in [48], or, alternatively, to Parigot's lambda-variables and mu-variables [40] (see also [19]). Wires are not supposed to denote a location in a term like variables in the λ-calculus. Rather, wires are seen a bit like *ropes* that can be *knotted* or *tightened* (like *chemical bonds*) with ropes of other components.

This, in fact, corresponds in a way to the practice of sailing. Sailors give a name to each rope (*main sail halyard*, *port jib sheet*, etc.), and on a modern competition sailboat every rope has its own colour to be sure that one tightens (or loosens) a rope to (or from) its appropriate place or with (or from) the appropriate rope; loosening the wrong rope can be catastrophic. In $\mathcal{X}$, those colours naturally become *kinds*, and like a rope has a colour, a wire has a kind.

One specificity of $\mathcal{X}$ is that syntactic constructors bind *two* wires, one of each kind[3]. In $\mathcal{X}$, bound wires receive a hat, so to show that x is bound we write $\hat{x}$; note that using the *"hat"*-notation, we are keeping in line with the old tradition of *Principia Mathematica* [49, 50].

That a wire is bound in a net implies, naturally, that this wire is unknown outside that net, but also that it 'interacts' with another 'opposite' wire that is bound in another net. The interaction differs from one constructor to another, and is ruled by basic reductions (see Section 2). In addition to bound wires an introduction rule exhibits a free wire, that is exposed and connectable. Often this exhibition corresponds to the creation of the wire.

Contents of this paper

In this paper we will present the formal definitions for $\mathcal{X}$, via syntax and reduction rules, and will show that the system is well-behaved by stating a number of essential properties. We will define a notion of simple type assignment for terms in $\mathcal{X}$, in that we will define a system of derivable judgements for which the terms of $\mathcal{X}$ are witnesses; we will show a soundness result for this system by showing that a subject-reduction result holds.

We will also compare $\mathcal{X}$ with a number of its predecessors. In fact, we will show that a number of well-known calculi are easily, elegantly and surprisingly effectively implementable in $\mathcal{X}$. For anyone familiar with the problem of expressiveness, in view of the fact that $\mathcal{X}$ is substitution-free, these result are truly novel. Except for $\overline{\lambda}\mu\tilde{\mu}$, it is not possible to easily and naturally embed $\mathcal{X}$ in those calculi in such a way that the major properties are preserved. This can easily be understood from the fact that the vast majority of calculi in our area is confluent (Church-Rosser), whereas $\mathcal{X}$ is not.

A tool (which can be downloaded from `http://www.doc.ic.ac.uk/~jr200/X`) was developed using the term graph rewriting technology, that allows users to not only input nets from $\mathcal{X}$, but also terms from λ-calculus, using the interpretation of the latter into $\mathcal{X}$ as specified in this paper. Details of the implementation can be found in [8, 9].

This paper presents an extended version of results that have appeared first in [35] and [7], themselves deeply inspired by the work of Urban and Bierman [45, 47, 46].

---

[2] We encourage the reader to not become confused by the use of names like x for the class of connectors that are called plugs; these names are, in fact, inherited from $\overline{\lambda}\mu\tilde{\mu}$.

[3] This is also the case in [45, 47, 46], but this fact is made very explicit in $\mathcal{X}$ by the use of *Principia*'s notations.

2 The $\mathcal{X}$-calculus

The nets that are the objects of $\mathcal{X}$ are built with three kinds of building stones, or constructors, called *capsule*, *export* and *import*. In addition there is an operator we call *cut*, which is handy for describing net construction, and which will be eliminated eventually by *rules*.

2.1 The operators

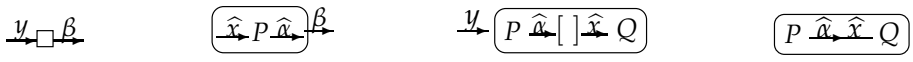
Nets are connected through *wires* that are named. In our description wires are oriented. This means we know in which direction the 'ether running through our nets' moves, and can say when a wire provides an entrance to a net or when a wire provides an exit. Thus we make the distinction between exit wires which we call *plugs* or *outputs* and enter wires which we call *sockets* or *inputs*. Plugs are named with Greek letters $\alpha,\beta,\gamma,\delta$, etc., and sockets are named with Latin letters x,y,z, etc.

When connecting two nets P and Q by an operator, say †, we may suppose that P has a plug α and Q has a socket x which we want to bind together to create a flow from P to Q. After the link has been established, the wires are plugged, and the name of the plug and the name of the socket are forgotten. To be more precise, in $P\widehat{\alpha} \dagger \widehat{x}Q$ the name α is not reachable outside P and the name x is not reachable outside Q. This reminds of construction like $\forall x.P$ or $\lambda x.M$ in logic where the name x is not known outside the expressions. Those names are said to be *bound*. Likewise, in $\mathcal{X}$, in a construction like $P\widehat{\alpha} \dagger \widehat{x}Q$ where α is a plug of P and x is a socket of Q, there are *two* bound names, namely α and x, that are bound in the interaction.

Definition 2.1 (Syntax) The nets of the $\mathcal{X}$-calculus are defined by the following grammar, where $x,y,\ldots$ range over the infinite set of *sockets*, and $\alpha,\beta,\ldots$ over the infinite set of *plugs*.

$$P,Q \;::=\; \langle y\cdot\beta\rangle \mid \widehat{y}P\widehat{\alpha}\cdot\beta \mid P\widehat{\alpha}\,[y]\,\widehat{x}Q \mid P\widehat{\alpha} \dagger \widehat{x}Q$$

As an illustration, we represent the basic nets diagrammatically as:



Notice that, using the intuition sketched above, for example, the connector α is supposed to not occur outside P; this is formalised below by Definition 2.2 and Barendregt's Convention (see also below).

We see sockets as points where nets input, and plugs where they output, and write inputs on the left and output on the right, as is done also in [48].

We could justify the constructions of $\mathcal{X}$ through the example of the "translations" in a huge international organisation[4]. The arguments below will be better established and formalised in Section 3 when we will speak about *types* which are the correct framework. But of course $\mathcal{X}$ is basically an untyped language.

Suppose wires carry words in a language like Estonian, or Portuguese, etc, but also translators from language to language like "French to Dutch." A *capsule* $\langle y\cdot\beta\rangle$ connects inside the socket y with the plug β. Everything entering the capsule on y in one language will leave it in the same language on β. An *export* $(\widehat{x}P\widehat{\alpha}\cdot\beta)$ can be seen as follows: P provides a device that transforms words in a language received on x to words in a(nother) language returned on α, therefore $(\widehat{y}P\widehat{\alpha}\cdot\beta)$ is a translator, which can be seen as a 'higher-order' language, that is returned on a specific wire β which can be connected later on. An export can also be seen as *T-diagrams* like those used in compiling technology for bootstrapping (see [2], Section 11.2). An *import* $(P\widehat{\alpha}\,[y]\,\widehat{x}Q)$ is required when one tries to connect two wires α and x to carry words

[4] see `http://europa.eu.int/comm/translation/index_en.htm`

from different languages. To be able to have P and Q communicate a translator is needed, that will be received on the wire y, a socket.

The operator † is called a *cut* and a term (net) of the form $(P\widehat{\alpha} \dagger \widehat{x}Q)$ is called a *cut net*, and corresponds to an operation on the *switch board*. A cut is specific in that it connects two nets, connecting the socket x of Q to the plug α of P; this assumes that the language expected on x agrees with the language delivered by α. The cut expresses the need for a rewiring of the switch board: a language is on offer on a plug, and demanded on a socket, and "dealing" with the cut, which expresses the need for the connection to be established, will cause the cut to be eventually eliminated by building the connection. The calculus, defined by the reduction rules (Section 2.2) explains in detail how cuts are distributed through nets to be eventually erased at the level of capsules.

The following table gives the correspondence between the notations of $\mathcal{X}$ and those Urban uses. Urban uses the first letters of the Latin alphabet for plugs, and the last for sockets; he expresses input and output behaviour by using a π-calculus-like notation, putting sockets between parentheses and plugs between angles.

$\langle x{\cdot}\alpha\rangle$	$Ax(x,a)$
$\widehat{x}P\widehat{\beta}\cdot\alpha$	$ImpR((x)(b)P,a)$
$P\widehat{\alpha}\,[x]\,\widehat{y}Q$	$ImpL((a)P,(y)Q,x)$
$P\widehat{\alpha}\dagger\widehat{x}Q$	$Cut((a)P,(x)Q)$

It should be noted that, in the π-calculus, input $a(x)$ and output $\overline{a}(c)$ are *actions*, that are consumed in the communication, and written pre-fix because computation runs 'left-to-right': $a(x).P \mid \overline{a}(c).Q \to P[c/x] \mid Q$. In Urban's notation, the brackets have a different meaning: for example, in $ImpR((x)(b)P,a)$, P inputs on x and outputs on b, but (x) and (b) are not *actions*, but *descriptions*. Notice that Urban's notation is pre-fix, which distorts the notion of 'flow' $\mathcal{X}$ expresses; also, in $ImpL((a)P,(y)Q,x)$, it is not clear that x will interface between P and Q.

Above we spoke about bound names; we will introduce now formally those notions with that of free sockets and plugs into $\mathcal{X}$.

Definition 2.2 The *free sockets* and *free plugs* in a net are:

$$\begin{array}{lcllcl}
fs(\langle x{\cdot}\alpha\rangle) & = & \{x\} & fp(\langle x{\cdot}\alpha\rangle) & = & \{\alpha\} \\
fs(\widehat{x}P\widehat{\alpha}\cdot\beta) & = & fs(P)\backslash\{x\} & fp(\widehat{x}P\widehat{\alpha}\cdot\beta) & = & (fp(P)\backslash\{\alpha\})\cup\{\beta\} \\
fs(P\widehat{\alpha}\,[y]\,\widehat{x}Q) & = & fs(P)\cup\{y\}\cup(fs(Q)\backslash\{x\}) & fp(P\widehat{\alpha}\,[y]\,\widehat{x}Q) & = & (fp(P)\backslash\{\alpha\})\cup fp(Q) \\
fs(P\widehat{\alpha}\dagger\widehat{x}Q) & = & fs(P)\cup(fs(Q)\backslash\{x\}) & fp(P\widehat{\alpha}\dagger\widehat{x}Q) & = & (fp(P)\backslash\{\alpha\})\cup fp(Q)
\end{array}$$

A socket x or plug α occurring in P which is not free is called *bound*, written $x \in bs(P)$ and $\alpha \in bp(P)$. We will write $x \notin fs(P,Q)$ for $x \notin fs(P) \wedge x \notin fs(Q)$.

We will normally adopt Barendregt's convention (called convention on variables by Barendregt, but here it will be a convention on names).

Convention on names.: In a net or in a statement, a name is never both bound *and* free in the same context.

We will consider also, for example, x bound in $P[y/x]$ and $P :\cdot \Gamma, x{:}A \vdash_{\mathcal{X}} \Delta$ (this notion will be introduced in Section 3).

As the main concept is that of a name, we define only renaming, i.e., replacement of a name by another name. Indeed it would make no sense to substitute a name by a net. The definition of renaming relies on Barendregt's convention on names; if a binding, say $\hat{x}P$ of

x in P violates Barendregt's convention, one can get it back by renaming, i.e., $\hat{y}P[y/x]$; as is common in calculi with (some form of) explicit substitution, this renaming can be internalised (see Section 2.5).

Definition 2.3 (Renaming of sockets and plugs)

$$\begin{array}{rlrl}
\langle x\cdot\alpha\rangle[y/x] &= \langle y\cdot\alpha\rangle & \langle x\cdot\alpha\rangle[\beta/\alpha] &= \langle x\cdot\beta\rangle \\
\langle z\cdot\alpha\rangle[y/x] &= \langle z\cdot\alpha\rangle \quad (x\neq z) & \langle x\cdot\gamma\rangle[\beta/\alpha] &= \langle x\cdot\gamma\rangle \quad (\alpha\neq\gamma) \\
(\hat{z}P\hat{\alpha}\cdot\beta)[y/x] &= \hat{z}\,(P[y/x])\hat{\alpha}\cdot\beta & (\hat{z}P\hat{\delta}\cdot\alpha)[\beta/\alpha] &= \hat{z}\,(P[\beta/\alpha])\hat{\delta}\cdot\beta \\
(P\hat{\alpha}\,[x]\,\hat{z}Q)[y/x] &= (P[y/x])\,\hat{\alpha}\,[y]\,\hat{z}\,(Q[y/x]) & (\hat{z}P\hat{\delta}\cdot\gamma)[\beta/\alpha] &= \hat{z}\,(P[\beta/\alpha])\hat{\delta}\cdot\beta \quad (\alpha\neq\gamma) \\
(P\hat{\alpha}\,[u]\,\hat{z}Q)[y/x] &= (P[y/x])\,\hat{\alpha}\,[u]\,\hat{z}\,(Q[y/x]) \quad (x\neq u) & (P\hat{\delta}\,[x]\,\hat{z}Q)[\beta/\alpha] &= (P[\beta/\alpha])\,\hat{\delta}\,[x]\,\hat{z}\,(Q[\beta/\alpha]) \\
(P\hat{\alpha}\dagger\hat{z}Q)[y/x] &= (P[y/x])\,\hat{\alpha}\dagger\hat{z}\,(Q[y/x]) & (P\hat{\delta}\dagger\hat{z}Q)[\beta/\alpha] &= (P[\beta/\alpha])\,\hat{\delta}\dagger\hat{z}\,(Q[\beta/\alpha])
\end{array}$$

Renaming will play an important part in dealing with α-conversion, a problem we will discuss in Subsection 2.5.

2.2 The rules

We will now come to the definition of reduction via the elimination of cuts; the intuition in the reduction is that the cut $P\hat{\alpha}\dagger\hat{x}Q$ expresses the intention to connect all αs in P and xs in Q, and that reduction will realise this, by either connecting all αs to all xs, or all xs to all αs (notice that this will not necessarily have the same effect; consider the cases when either α or x does not occur). The base cases occurs when a socket or a plug is *exposed and unique* (is *introduced*, or *connectable*), since then building the connection is straightforward, as is expressed by the first set of rules. Informally, a net P introduces a socket x if P is constructed from sub-nets which do not contain x as free socket, so x only occurs at the "top level." This means that P is either an import with a middle connector $[x]$ or a capsule with left part x. Similarly, a net introduces a plug α if it is an export that "creates" $\cdot\alpha$ or a capsule with right part α. We say now formally what it means for a nets to *introduce* a socket or a plug (Urban uses the terminology "freshly introduce" [45]).

Definition 2.4 (Introduction)(P *introduces* x): $P=\langle x\cdot\beta\rangle$ or $P=R\hat{\alpha}\,[x]\,\hat{y}Q$, with $x\notin fs(R,Q)$.
(P *introduces* β): $P=\langle y\cdot\beta\rangle$ or $P=\hat{x}Q\hat{\alpha}\cdot\beta$ with $\beta\notin fp(Q)$.

We first present a simple family of reduction rules. They specify how to reduce a net that cuts sub-nets which introduce connectors. These rules are naturally divided in four categories; when a capsule is cut with a capsule, an export with a capsule, a capsule with an import or an export with an import. There is no other pattern in which a plug is introduced on the left of a $\dagger$ and a socket is introduced on the right.

Definition 2.5 (Logical Reduction) The logical rules are (assume that the nets of the left-hand sides of the rules *introduce* the socket x and the plug α)

$$\begin{array}{rrcl}
(cap): & \langle y\cdot\alpha\rangle\hat{\alpha}\dagger\hat{x}\langle x\cdot\beta\rangle & \to & \langle y\cdot\beta\rangle \\
(exp): & (\hat{y}P\hat{\beta}\cdot\alpha)\,\hat{\alpha}\dagger\hat{x}\langle x\cdot\gamma\rangle & \to & \hat{y}P\hat{\beta}\cdot\gamma \\
(imp): & \langle y\cdot\alpha\rangle\hat{\alpha}\dagger\hat{x}\,(P\hat{\beta}\,[x]\,\hat{z}Q) & \to & P\hat{\beta}\,[y]\,\hat{z}Q \\
(exp\text{-}imp): & (\hat{y}P\hat{\beta}\cdot\alpha)\,\hat{\alpha}\dagger\hat{x}\,(Q\hat{\gamma}\,[x]\,\hat{z}R) & \to & \begin{cases}(Q\hat{\gamma}\dagger\hat{y}P)\,\hat{\beta}\dagger\hat{z}R\\ Q\hat{\gamma}\dagger\hat{y}\,(P\hat{\beta}\dagger\hat{z}R)\end{cases}
\end{array}$$

As an illustration, we give their diagrammatical representation in Figure 1.

Notice that, in rule $(exp\text{-}imp)$, in addition to the conditions for introduction of the connectors that are active in the cut ($\alpha\notin fp(P)$ and $x\notin fs(Q,R)$) we can also state that $\beta\notin fp(Q)\backslash\{\gamma\}$, as well as that $y\notin fs(R)\backslash\{z\}$, due to Barendregt's convention.

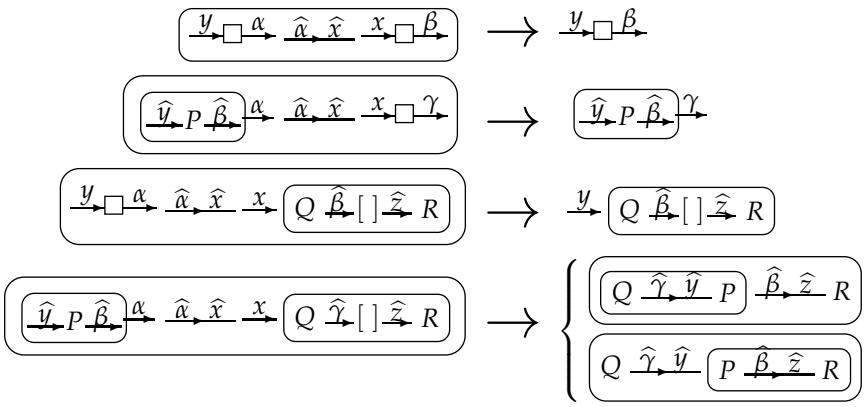


Figure 1: The reduction rules for $\mathcal{X}$ is a diagrammatical representation

Still in rule (*exp-imp*) the reader may have noticed that there are two right-hand sides. These two right-hand sides are a first appearance of the non-determinism in $\mathcal{X}$ which, as noticed by Lafont, is an intrinsic part of cut elimination in classical logic.

We now need to define how to reduce a cut in case when one of its sub-nets does not introduce a socket or a plug. This requires to extend the syntax with two new operators that we call *activated* cuts:

$$P \ ::= \ \ldots \mid P\widehat{\alpha} \not\leftarrow \widehat{x} Q \mid P\widehat{\alpha} \not\rightarrow \widehat{x} Q$$

Intuitively $P\widehat{\alpha} \not\leftarrow \widehat{x} Q$ represents the intention to connect the xs in Q to the αs in P, moving Q inside P to those places where α is connectable, i.e. is introduced, and $P\widehat{\alpha} \not\rightarrow \widehat{x} Q$ represents the intention to connect the αs to the xs. Nets where cuts are not activated are called *pure*. Activated cuts are propagated through the nets, moving towards occurrences of the connectors mentioned in the cut up to the point that that they are connectable, and a logical rule can be applied.

Definition 2.6 (Activating the cuts)

$$\begin{array}{rrcll} (a\not\leftarrow): & P\widehat{\alpha} \dagger \widehat{x} Q & \to & P\widehat{\alpha} \not\leftarrow \widehat{x} Q & (P \textit{ does not introduce } \alpha) \\ (\not\rightarrow a): & P\widehat{\alpha} \dagger \widehat{x} Q & \to & P\widehat{\alpha} \not\rightarrow \widehat{x} Q & (Q \textit{ does not introduce } x) \end{array}$$

Notice that both side-conditions can be valid simultaneously, thereby validating both rewrite rules at the same moment. This gives, in fact, a *critical pair* or *superposition* for our notion of reduction, and is a cause for the loss of confluence.

We will now define how to propagate an activated cut through sub-nets. The direction of the activating shows in which direction the cut should be propagated, hence the two sets of reduction rules.

Definition 2.7 (Propagation Reduction) The rules of propagation are:

Left propagation

$$\begin{array}{rrcll} (d\not\leftarrow): & \langle y{\cdot}\alpha\rangle \widehat{\alpha} \not\leftarrow \widehat{x} P & \to & \langle y{\cdot}\alpha\rangle \widehat{\alpha} \dagger \widehat{x} P & \\ (cap\not\leftarrow): & \langle y{\cdot}\beta\rangle \widehat{\alpha} \not\leftarrow \widehat{x} P & \to & \langle y{\cdot}\beta\rangle & (\beta \neq \alpha) \\ (exp\text{-}out\not\leftarrow): & (\widehat{y} Q \widehat{\beta}{\cdot}\alpha) \widehat{\alpha} \not\leftarrow \widehat{x} P & \to & (\widehat{y} (Q\widehat{\alpha} \not\leftarrow \widehat{x} P) \widehat{\beta}{\cdot}\gamma) \widehat{\gamma} \dagger \widehat{x} P & (\gamma \textit{ fresh}) \\ (exp\text{-}in\not\leftarrow): & (\widehat{y} Q \widehat{\beta}{\cdot}\gamma) \widehat{\alpha} \not\leftarrow \widehat{x} P & \to & \widehat{y} (Q\widehat{\alpha} \not\leftarrow \widehat{x} P) \widehat{\beta}{\cdot}\gamma & (\gamma \neq \alpha) \\ (imp\not\leftarrow): & (Q\widehat{\beta}\,[z]\,\widehat{y} R) \widehat{\alpha} \not\leftarrow \widehat{x} P & \to & (Q\widehat{\alpha} \not\leftarrow \widehat{x} P) \widehat{\beta}\,[z]\,\widehat{y} (R\widehat{\alpha} \not\leftarrow \widehat{x} P) & \\ (cut\not\leftarrow): & (Q\widehat{\beta} \dagger \widehat{y} R) \widehat{\alpha} \not\leftarrow \widehat{x} P & \to & (Q\widehat{\alpha} \not\leftarrow \widehat{x} P) \widehat{\beta} \dagger \widehat{y} (R\widehat{\alpha} \not\leftarrow \widehat{x} P) & \end{array}$$

Right propagation

$$
\begin{array}{rrcll}
(\nwarrow d): & P\widehat{\alpha} \nwarrow \widehat{x}\langle x\cdot\beta\rangle & \to & P\widehat{\alpha} \dagger \widehat{x}\langle x\cdot\beta\rangle & \\
(\nwarrow cap): & P\widehat{\alpha} \nwarrow \widehat{x}\langle y\cdot\beta\rangle & \to & \langle y\cdot\beta\rangle & (y \neq x) \\
(\nwarrow exp): & P\widehat{\alpha} \nwarrow \widehat{x}(\widehat{y}Q\widehat{\beta}\cdot\gamma) & \to & \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\beta}\cdot\gamma & \\
(\nwarrow imp\text{-}out): & P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}\,[x]\,\widehat{y}R) & \to & P\widehat{\alpha} \dagger \widehat{z}((P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\,[z]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R)) & (z \textit{ fresh}) \\
(\nwarrow imp\text{-}in): & P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}\,[z]\,\widehat{y}R) & \to & (P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta}\,[z]\,\widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R) & (z \neq x) \\
(\nwarrow cut): & P\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R) & \to & (P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \nwarrow \widehat{x}R) &
\end{array}
$$

Definition 2.8 We will write $P \to Q$ for the compatible closure of the one-step term rewriting induced by the above rules, and write $P \twoheadrightarrow Q$ for its transitive closure (we will often write simply $P \to Q$ for $P \twoheadrightarrow Q$, indicating that P *reduces to* Q, and reserve $\twoheadrightarrow$ for *a number of steps*). We will subscript the arrow that represents our reduction to indicate certain sub-systems, defined by a reduction strategy: for example, we will write $\to_{\mathrm{A}}$ for the reduction that uses only *Left propagation* or *Right propagation* rules. In fact, $\to_{\mathrm{A}}$ is the reduction that pushes $\not\leftarrow$ and $\nwarrow$ inward. We will also write $P \downarrow_{\mathcal{X}} Q$ if there exists an R such that $P \twoheadrightarrow R$ and $Q \twoheadrightarrow R$, i.e. when P and Q have a common reduct.

The rules $(exp\text{-}out\not\leftarrow)$ and $(\nwarrow imp\text{-}out)$ deserve some attention. Notice that in rule $(exp\text{-}out\not\leftarrow)$ we create a new name γ and that in rule $(\nwarrow imp\text{-}out)$ we create a new name z. This is done because a cut is duplicated, one of which is distributed inside and the other is left outside as an inactive cut. A new name is created to respect Barendregt's hygiene convention; for instance, in the left-hand side of $(exp\text{-}out\not\leftarrow)$, α may occur more than once in $\widehat{y}Q\widehat{\beta}\cdot\alpha$, that is once after the dot and again in Q. The occurrence after the dot is dealt with separately by creating a new name γ. Note that the cut associated with that γ is then inactivated; this is because, although we know that now γ is introduced, we do not know if x in Q is.

The same thing happens with x in $(\nwarrow imp\text{-}out)$ and a new name z is created and the external cut is inactivated.

The Renaming Lemma (2.11, stated and proved in Section 2.4 on page 12) shows that in the right-hand side of rule $(d\not\leftarrow)$ we could have written $P[y/x]$ and in the right-hand side of rule $(\nwarrow d)$ we could have written $P[\beta/\alpha]$ instead of the terms we have chosen. We made that choice for three reasons. Firstly, we like to make all the operations explicit and we do not want to rely on operations defined in the meta-theory. Internalising name substitutions would have required to put as rules in the theory all the identities of Definition 2.3. Secondly, this small-step approach is closer to our philosophy of $\mathcal{X}$ that tends to decompose operations as fine grained as possible. Thirdly, notice that if α is introduced in $\widehat{y}Q\widehat{\beta}\cdot\alpha$ then $(exp\text{-}out\not\leftarrow)$ gives, with the Cancellation Lemma (2.10, see page 11),

$$(\widehat{y}Q\widehat{\beta}\cdot\alpha)\,\widehat{\alpha} \not\leftarrow \widehat{x}P \twoheadrightarrow (\widehat{y}Q\widehat{\beta}\cdot\alpha)\,\widehat{\alpha} \dagger \widehat{x}P$$

and $(d\not\leftarrow)$ and this reduction play similar roles, namely that of "deactivating" cuts.

2.3 Strong normalisation

We should point out that, using the rules above, not all typeable nets are strongly normalisable. This is caused by the fact that arbitrary cut-elimination is too liberal; for example, allowing (inactivated) cuts to propagate over cuts immediately leads to non-termination, since we can always choose the outermost cut as the one to contract. Although the notion of cut-elimination as proposed here has no rule that would allow this behaviour, it can be mimicked, which can lead to non-termination for typeable nets, as already observed in [45].

Assume $x \notin fs(Q), \beta \notin fp(P)$, and P, Q both pure, then:

$$
\begin{array}{lcl}
P\widehat{\alpha} \dagger \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{z}Q) & \to & (\nwarrow a) \\
P\widehat{\alpha} \nwarrow \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{z}Q) & \to & (\nwarrow cut) \\
(P\widehat{\alpha} \nwarrow \widehat{x}\,\langle x{\cdot}\beta\rangle)\,\widehat{\beta} \dagger \widehat{z}\,(P\widehat{\alpha} \nwarrow \widehat{x}Q) & \to & (\nwarrow d), (\nwarrow gc) \\
(P\widehat{\alpha} \dagger \widehat{x}\,\langle x{\cdot}\beta\rangle)\,\widehat{\beta} \dagger \widehat{z}Q & \to & (a\not\leftarrow) \\
(P\widehat{\alpha} \dagger \widehat{x}\,\langle x{\cdot}\beta\rangle)\,\widehat{\beta} \not\leftarrow \widehat{z}Q & \to & (cut\not\leftarrow) \\
(P\widehat{\beta} \not\leftarrow \widehat{z}Q)\,\widehat{\alpha} \dagger \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \not\leftarrow \widehat{z}Q) & \to & (d\not\leftarrow), (gc\not\leftarrow) \\
P\widehat{\alpha} \dagger \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{z}Q) & &
\end{array}
$$

Moreover, assuming $P \vdots \cdot\ \Gamma \vdash_{\mathcal{X}} \alpha{:}A, \Delta$ and $Q \vdots \cdot\ \Gamma, x{:}A \vdash_{\mathcal{X}} \Delta$ (see Section 3), we can construct

$$
\dfrac{\boxed{\phantom{P \vdots \cdot\ \Gamma \vdash \alpha{:}A,\Delta}}\ P \vdots \cdot\ \Gamma \vdash \alpha{:}A,\Delta \qquad \dfrac{\dfrac{}{\langle x{\cdot}\beta\rangle \vdots \cdot\ \Gamma, x{:}A \vdash \beta{:}A,\Delta}\,(Ax) \qquad \dfrac{\boxed{\phantom{Q \vdots \cdot\ \Gamma, x{:}A \vdash \Delta}}\ Q \vdots \cdot\ \Gamma, x{:}A \vdash \Delta}{Q \vdots \cdot\ \Gamma, x{:}A, z{:}A \vdash \Delta}\,(Wk)}{\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{z}Q \vdots \cdot\ \Gamma, x{:}A \vdash \Delta}\,(cut)}{P\widehat{\alpha} \dagger \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{z}Q) \vdots \cdot\ \Gamma \vdash \Delta}\,(cut)
$$

and all the intermediate nets in the reduction above are typeable by the Witness Reduction result (Theorem 3.6; example communicated by Alexander J. Summers).

Urban gives a solution for this unwanted reduction behaviour, and shows it sufficient to obtain strong-normalisation of typeable nets. He adds the rules

$$
\begin{array}{lcl}
(P\widehat{\alpha} \dagger \widehat{x}\,\langle x{\cdot}\beta\rangle)\,\widehat{\beta} \not\leftarrow \widehat{y}Q & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}Q)\,\widehat{\alpha} \dagger \widehat{y}Q \\
P\widehat{\alpha} \nwarrow \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{y}Q) & \to & P\widehat{\alpha} \dagger \widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}Q)
\end{array}
$$

and gives them priority over the rules $(cut\not\leftarrow)$ and $(\nwarrow cut)$ by changing those to

$$
\begin{array}{lcl}
(P\widehat{\alpha} \dagger \widehat{x}Q)\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,(Q\widehat{\beta} \not\leftarrow \widehat{u}R),\ \ Q \neq \langle x{\cdot}\beta\rangle \\
P\widehat{\alpha} \nwarrow \widehat{x}\,(Q\widehat{\beta} \dagger \widehat{y}R) & \to & (P\widehat{\alpha} \nwarrow \widehat{x}Q)\,\widehat{\beta} \dagger \widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}R),\ \ Q \neq \langle x{\cdot}\beta\rangle
\end{array}
$$

thereby blocking the reduction of $P\widehat{\alpha} \not\leftarrow \widehat{x}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta} \dagger \widehat{z}Q)$ to $(P\widehat{\alpha} \dagger \widehat{x}\,\langle x{\cdot}\beta\rangle)\,\widehat{\beta} \dagger \widehat{z}Q$.

Notice that the side-condition $Q \neq \langle x{\cdot}\beta\rangle$ is quite different in character from the rules for $\mathcal{X}$ we presented above, in that now *equality* between circuits is tested, rather than just the occurrence of a connector within a circuit. In fact, it corresponds to replacing, for example, rule $(cut\not\leftarrow)$ by the rules:

$$
\begin{array}{rcl}
(P\widehat{\alpha} \dagger \widehat{x}\,\langle x{\cdot}\beta\rangle)\,\widehat{\beta} \not\leftarrow \widehat{y}Q & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}Q)\,\widehat{\alpha} \dagger \widehat{y}Q \\
(P\widehat{\alpha} \dagger \widehat{x}\,\langle y{\cdot}\gamma\rangle)\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,(\langle y{\cdot}\gamma\rangle\,\widehat{\beta} \not\leftarrow \widehat{y}R) \quad (x \neq y \vee \gamma \neq \beta) \\
(P\widehat{\alpha} \dagger \widehat{x}\,(\widehat{z}Q\widehat{\gamma}{\cdot}\delta))\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,((\widehat{z}Q\widehat{\gamma}{\cdot}\delta)\,\widehat{\beta} \not\leftarrow \widehat{y}R) \\
(P\widehat{\alpha} \dagger \widehat{x}\,(Q_1\,\widehat{\gamma}\,[z]\,\widehat{v}\,Q_2))\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,((Q_1\,\widehat{\gamma}\,[z]\,\widehat{v}\,Q_2)\,\widehat{\beta} \not\leftarrow \widehat{y}R) \\
(P\widehat{\alpha} \dagger \widehat{x}\,(Q_1\,\widehat{\gamma} \dagger \widehat{v}\,Q_2))\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,((Q_1\,\widehat{\gamma} \dagger \widehat{v}\,Q_2)\,\widehat{\beta} \not\leftarrow \widehat{y}R)
\end{array}
$$

as well as replacing rule $(\nwarrow cut)$ by four similar rules, effectively adding eight rules. Although these rules are fine as far as term rewriting is concerned, rewriting is no longer local as they match against sub-nets of the nets involved in the cut. Remark that the last two rules might as well be replaced by:

$$
\begin{array}{rcl}
(P\widehat{\alpha} \dagger \widehat{x}\,(Q_1\,\widehat{\gamma}\,[z]\,\widehat{v}\,Q_2))\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,((Q_1\,\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\gamma}\,[z]\,\widehat{v}\,(Q_2\,\widehat{\beta} \not\leftarrow \widehat{y}R)) \\
(P\widehat{\alpha} \dagger \widehat{x}\,(Q_1\,\widehat{\gamma} \dagger \widehat{v}\,Q_2))\,\widehat{\beta} \not\leftarrow \widehat{y}R & \to & (P\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\alpha} \dagger \widehat{x}\,((Q_1\,\widehat{\beta} \not\leftarrow \widehat{y}R)\,\widehat{\gamma} \dagger \widehat{v}\,(Q_2\,\widehat{\beta} \not\leftarrow \widehat{y}R))
\end{array}
$$

etc.

2.4 Call-by-name and call-by-value

In this section we will define two sub-systems of reduction (i.e. restrictions of the full reduction we defined above), that correspond roughly to call-by-name (CBN) and call-by-value (CBV) reduction. Notice that this is essentially different from the approach of [48], where only two dual notions of reduction are defined, and no overall notion of reduction.

As for $\overline{\lambda}\mu\tilde{\mu}$ also in $\mathcal{X}$ only one notion of reduction is defined. When interpreting the λ-calculus or $\lambda\mu$ in $\overline{\lambda}\mu\tilde{\mu}$, however, different interpretation functions are defined for the CBN and CBV sub-calculi. Here, we will define *one* interpretation function for each calculus, and show that both CBN- and CBV-reduction are respected.

As mentioned above, when P does not introduce α and Q does not introduce x, $P\widehat{\alpha} \dagger \widehat{x}Q$ is a *superposition*, meaning that two rules, namely $(a\not\leftarrow)$ and $(\not\leftarrow a)$, can both be fired. The *critical pair* $\langle P\widehat{\alpha} \not\leftarrow \widehat{x}Q,\ P\widehat{\alpha} \not\leftarrow \widehat{x}Q\rangle$ may lead to different irreducible nets. This is to say that the reduction relation $\rightarrow$ is *not confluent*. Non-determinism is a key feature of both classical logic and rewriting logic.

We introduce two sub-reduction systems which favour one kind of activating whenever the above critical pair occurs.

Definition 2.9 • If a cut can be activated in two ways, the CBV strategy only allows to activate it via $(a\not\leftarrow)$; we write $P \rightarrow_{\mathrm{V}} Q$ in that case. We can reformulate this as the reduction system obtained by replacing rule $(\not\leftarrow a)$ by:

$$(\not\leftarrow a):\ P\widehat{\alpha} \dagger \widehat{x}Q \ \rightarrow\ P\widehat{\alpha} \not\leftarrow \widehat{x}Q \quad (P \textit{ introduces } \alpha \textit{ and } Q \textit{ does not introduce } x)$$

We also choose to only allow one variant of the $(\textit{exp-imp})$ rule:

$$(\textit{exp-imp})_{\mathrm{V}}\ \ (\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha} \dagger \widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R)\ \rightarrow\ Q\widehat{\gamma} \dagger \widehat{y}\,(P\widehat{\beta} \dagger \widehat{z}R)$$

• The CBN strategy can only activate such a cut via $(\not\leftarrow a)$; like above, we write $P \rightarrow_{\mathrm{N}} Q$. Likewise, we can reformulate this as the reduction system obtained by replacing rule $(a\not\leftarrow)$ by:

$$(a\not\leftarrow):\ P\widehat{\alpha} \dagger \widehat{x}Q \ \rightarrow\ P\widehat{\alpha} \not\leftarrow \widehat{x}Q \quad (P \textit{ does not introduce } \alpha \textit{ and } Q \textit{ introduces } x)$$

We also choose to only allow the other variant of the $(\textit{exp-imp})$ rule:

$$(\textit{exp-imp})_{\mathrm{N}}\ \ (\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha} \dagger \widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R)\ \rightarrow\ (Q\widehat{\gamma} \dagger \widehat{y}P)\,\widehat{\beta} \dagger \widehat{z}R$$

Notice that, in CBV, a right cut like $P\widehat{\alpha} \not\leftarrow \widehat{x}Q$ implies that α is introduced in P (i.e. P is a *value*), and that, in CBN, a left cut like $P\widehat{\alpha} \not\leftarrow \widehat{x}Q$ implies that x is introduced in Q (i.e. Q is a *name*).

We will now show some basic properties, which essentially show that the calculus is well behaved; to give the reader an opportunity to see reduction in $\mathcal{X}$ at work, we give the full proof. Recall that a net is pure if it contains no activated cuts.

Lemma 2.10 (CANCELLATION) *i)* $P\widehat{\alpha} \not\leftarrow \widehat{x}Q \rightarrow_{\mathrm{A}} P$ *if* $\alpha \notin fp(P)$ *and* P *is pure.*
ii) $P\widehat{\alpha} \dagger \widehat{x}Q \rightarrow P$ *if* $\alpha \notin fp(P)$ *and* P *is pure.*
iii) $P\widehat{\alpha} \not\leftarrow \widehat{x}Q \rightarrow_{\mathrm{A}} Q$ *if* $x \notin fs(Q)$ *and* Q *is pure.*
iv) $P\widehat{\alpha} \dagger \widehat{x}Q \rightarrow Q$ *if* $x \notin fs(Q)$ *and* Q *is pure.*

Proof: i) By induction on the structure of nets.

$(P = \langle y\cdot\beta\rangle)$: $\langle y\cdot\beta\rangle\,\widehat{\alpha} \not\leftarrow \widehat{x}Q \rightarrow_{\mathrm{A}} (cap\not\leftarrow)\ \langle y\cdot\beta\rangle$

$(P = \widehat{y}R\widehat{\beta}\cdot\gamma)$: $(\widehat{y}R\widehat{\beta}\cdot\gamma)\,\widehat{\alpha} \not\leftarrow \widehat{x}Q\ \rightarrow_{\mathrm{A}}\ (\textit{exp-in}\not\leftarrow)\ \ \widehat{y}\,(R\widehat{\alpha} \not\leftarrow \widehat{x}Q)\widehat{\beta}\cdot\gamma\ \rightarrow_{\mathrm{A}}\ (IH)\ \widehat{y}R\widehat{\beta}\cdot\gamma$

$(P = R\widehat{\beta}\,[z]\,\widehat{y}S)$: $(R\widehat{\beta}\,[z]\,\widehat{y}S)\,\widehat{\alpha} \not\leftarrow \widehat{x}Q \rightarrow_{\mathrm{A}}\ (imp\not\leftarrow)$

$(R\widehat{\alpha} \not\leftarrow \widehat{x}Q)\,\widehat{\beta}\,[z]\,\widehat{y}\,(S\widehat{\alpha} \not\leftarrow \widehat{x}Q) \rightarrow_{\mathrm{A}}\ (IH)\ R\widehat{\beta}\,[z]\,\widehat{y}S$

$(P = R\widehat{\beta} \dagger \widehat{y}S)$: $(R\widehat{\beta} \dagger \widehat{y}S)\,\widehat{\alpha} \not\leftarrow \widehat{x}Q \to_{\mathrm{A}} (cut\not\leftarrow)$
$(R\widehat{\alpha} \not\leftarrow \widehat{x}Q)\,\widehat{\beta} \dagger \widehat{y}\,(S\widehat{\alpha} \not\leftarrow \widehat{x}Q) \to_{\mathrm{A}} (IH)\ R\widehat{\beta} \dagger \widehat{y}S$

ii) $(P$ *introduces* $\alpha)$: Then $\alpha \in fp(P)$, so this is impossible.

$(P$ *does not introduce* $\alpha)$: Then $P\widehat{\alpha} \dagger \widehat{x}Q \to_{\mathrm{A}} (a\not\leftarrow)\ P\widehat{\alpha} \not\leftarrow \widehat{x}Q$, and the result follows from part (*i*).

iii) By induction on the structure of nets.

$(Q = \langle y \cdot \beta \rangle)$: $P\widehat{\alpha} \nwarrow \widehat{x}\langle y \cdot \beta \rangle \to_{\mathrm{A}} (\nwarrow cap)\ \langle y \cdot \beta \rangle$

$(Q = \widehat{y}R\widehat{\beta} \cdot \gamma)$: $P\widehat{\alpha} \nwarrow \widehat{x}\,(\widehat{y}R\widehat{\beta} \cdot \gamma) \to_{\mathrm{A}} (\nwarrow exp)\ \widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}R)\widehat{\beta} \cdot \gamma \to_{\mathrm{A}} (IH)\ \widehat{y}R\widehat{\beta} \cdot \gamma$

$(Q = R\widehat{\beta}\,[z]\,\widehat{y}S)$: $P\widehat{\alpha} \nwarrow \widehat{x}\,(R\widehat{\beta}\,[z]\,\widehat{y}S) \to_{\mathrm{A}} (\nwarrow imp\text{-}in)$
$(P\widehat{\alpha} \nwarrow \widehat{x}R)\,\widehat{\beta}\,[z]\,\widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}S) \to_{\mathrm{A}} (IH)\ R\widehat{\beta}\,[z]\,\widehat{y}S$

$(Q = R\widehat{\beta} \dagger \widehat{y}S)$: $P\widehat{\alpha} \nwarrow \widehat{x}\,(R\widehat{\beta} \dagger \widehat{y}S) \to_{\mathrm{A}} (\nwarrow cut)$
$(P\widehat{\alpha} \nwarrow \widehat{x}R)\,\widehat{\beta} \dagger \widehat{y}\,(P\widehat{\alpha} \nwarrow \widehat{x}S) \to_{\mathrm{A}} (IH)\ R\widehat{\beta} \dagger \widehat{y}S$

iv) $(Q$ *introduces* $x)$: Then $x \in fs(Q)$, so this is impossible.

$(Q$ *does not introduce* $x)$: Then $P\widehat{\alpha} \dagger \widehat{x}Q \to_{\mathrm{A}} (\nwarrow a)\ P\widehat{\alpha} \nwarrow \widehat{x}Q$, and the result follows from part (*iii*). □

We will now show that a cut with a capsule leads to renaming.

Lemma 2.11 (Renaming) *i) $P\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to P[\alpha/\delta]$, if P is pure.*
ii) $P\widehat{\delta} \dagger \widehat{z}\langle z \cdot \alpha \rangle \to P[\alpha/\delta]$, if P is pure.
iii) $\langle z \cdot \alpha \rangle\,\widehat{\alpha} \nwarrow \widehat{x}P \to P[z/x]$, if P is pure.
iv) $\langle z \cdot \alpha \rangle\,\widehat{\alpha} \dagger \widehat{x}P \to P[z/x]$, if P is pure.

Proof: i) By induction on the structure of nets.

$(P = \langle x \cdot \delta \rangle)$: $\langle x \cdot \delta \rangle\,\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} (d\not\leftarrow)\ \langle x \cdot \delta \rangle\,\widehat{\delta} \dagger \widehat{z}\langle z \cdot \alpha \rangle \to (cap)\ \langle x \cdot \alpha \rangle \stackrel{\Delta}{=} \langle x \cdot \delta \rangle[\alpha/\delta]$

$(P = \langle x \cdot \beta \rangle, \beta \neq \delta)$: $\langle x \cdot \beta \rangle\,\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} (cap\not\leftarrow)\ \langle x \cdot \beta \rangle \stackrel{\Delta}{=} \langle x \cdot \beta \rangle[\alpha/\delta]$

$(P = \widehat{y}Q\widehat{\gamma} \cdot \delta)$: $(\widehat{y}Q\widehat{\gamma} \cdot \delta)\,\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} (exp\text{-}out\not\leftarrow),\ \beta\ fresh$
$(\widehat{y}\,(Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle)\widehat{\gamma} \cdot \beta)\,\widehat{\beta} \dagger \widehat{z}\langle z \cdot \alpha \rangle \to (exp)$
$\widehat{y}\,(Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle)\widehat{\gamma} \cdot \alpha \to (IH)$
$\widehat{y}\,Q[\alpha/\delta]\widehat{\gamma} \cdot \alpha \stackrel{\Delta}{=} (\widehat{y}Q\widehat{\gamma} \cdot \delta)[\alpha/\delta]$

$(P = \widehat{y}Q\widehat{\gamma} \cdot \beta, \beta \neq \delta)$: $(\widehat{y}Q\widehat{\gamma} \cdot \beta)\,\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} (exp\text{-}in\not\leftarrow)$
$\widehat{y}\,(Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle)\widehat{\gamma} \cdot \beta \to (IH)$
$\widehat{y}\,Q[\alpha/\delta]\widehat{\gamma} \cdot \beta \stackrel{\Delta}{=} (\widehat{y}Q\widehat{\gamma} \cdot \beta)[\alpha/\delta]$

$(P = Q\widehat{\beta}\,[x]\,\widehat{y}R)$: $(Q\widehat{\beta}\,[x]\,\widehat{y}R)\,\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} (imp\not\leftarrow)$
$(Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle)\,\widehat{\beta}\,[x]\,\widehat{y}\,(R\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle) \to (IH)$
$Q[\alpha/\delta]\,\widehat{\beta}\,[x]\,\widehat{y}R[\alpha/\delta] \stackrel{\Delta}{=} (Q\widehat{\beta}\,[x]\,\widehat{y}R)[\alpha/\delta]$

$(P = Q\widehat{\beta} \dagger \widehat{x}R)$: $(Q\widehat{\beta} \dagger \widehat{x}R)\,\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} (cut\not\leftarrow)$
$(Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle)\,\widehat{\beta} \dagger \widehat{x}\,(R\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle) \to (IH)$
$Q[\alpha/\delta]\,\widehat{\beta} \dagger \widehat{x}R[\alpha/\delta] \stackrel{\Delta}{=} (Q\widehat{\beta} \dagger \widehat{x}R)[\alpha/\delta]$

ii) If $P\widehat{\delta} \dagger \widehat{z}\langle z \cdot \alpha \rangle$, then either:

$(P$ *introduces* $\delta)$: Then either:

$(P = \widehat{x}Q\widehat{\beta} \cdot \delta$, *and* $\delta \notin fp(Q))$: $(\widehat{x}Q\widehat{\beta} \cdot \delta)\,\widehat{\delta} \dagger \widehat{z}\langle z \cdot \alpha \rangle \to (exp)\ \widehat{x}Q\widehat{\beta} \cdot \alpha \stackrel{\Delta}{=} (\widehat{x}Q\widehat{\beta} \cdot \delta)[\alpha/\delta]$

$(P = \langle x \cdot \delta \rangle)$: Then: $\langle x \cdot \delta \rangle\,\widehat{\delta} \dagger \widehat{z}\langle z \cdot \alpha \rangle \to (cap)\ \langle x \cdot \alpha \rangle \stackrel{\Delta}{=} \langle x \cdot \delta \rangle[\alpha/\delta]$.

$(P$ *does not introduce* $\delta)$: Then $P\widehat{\delta} \dagger \widehat{z}\langle z \cdot \alpha \rangle \to_{\mathrm{A}} P\widehat{\delta} \not\leftarrow \widehat{z}\langle z \cdot \alpha \rangle$, and the result follows from (*i*).

iii) By induction on the structure of nets.

$(P = \langle x\cdot\delta\rangle)$: $\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}\,\langle x\cdot\delta\rangle \to_{\mathrm{A}} (\nwarrow d)\ \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,\langle x\cdot\delta\rangle \to (cap)\ \langle z\cdot\delta\rangle \stackrel{\Delta}{=} \langle x\cdot\delta\rangle[z/x]$

$(P = \langle y\cdot\delta\rangle, y \neq x)$: $\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}\,\langle y\cdot\delta\rangle \to_{\mathrm{A}} (\nwarrow cap)\ \langle y\cdot\delta\rangle \stackrel{\Delta}{=} \langle y\cdot\delta\rangle[z/x]$

$(P = \widehat{y}Q\widehat{\gamma}\cdot\delta)$:
$$\begin{array}{lcll} \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}\,(\widehat{y}Q\widehat{\gamma}\cdot\delta) & \to_{\mathrm{A}} & (\nwarrow exp) \\ \widehat{y}\,(\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}Q)\widehat{\gamma}\cdot\delta & \to & (IH) \\ \widehat{y}\,Q[z/x]\widehat{\gamma}\cdot\delta & \stackrel{\Delta}{=} & (\widehat{y}Q\widehat{\gamma}\cdot\delta)[z/x] \end{array}$$

$(P = Q\widehat{\beta}\,[x]\,\widehat{y}R)$:
$$\begin{array}{lcll} \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}\,(Q\widehat{\beta}\,[x]\,\widehat{y}R) & \to_{\mathrm{A}} & (\nwarrow imp\text{-}out) \\ \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{v}\,((\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}Q)\,\widehat{\beta}\,[v]\,\widehat{y}\,(\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}R)) & \to & (IH) \\ \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{v}\,(Q[z/x]\,\widehat{\beta}\,[v]\,\widehat{y}\,R[z/x]) & \to & (imp) \\ Q[z/x]\,\widehat{\beta}\,[v]\,\widehat{y}\,R[z/x] & \stackrel{\Delta}{=} & (Q\widehat{\beta}\,[x]\,\widehat{y}R)[z/x] \end{array}$$

$(P = Q\widehat{\beta}\,[v]\,\widehat{y}R,\ v \neq x)$:
$$\begin{array}{lcll} \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}\,(Q\widehat{\beta}\,[v]\,\widehat{y}R) & \to_{\mathrm{A}} & (\nwarrow imp\text{-}in) \\ (\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}Q)\,\widehat{\beta}\,[v]\,\widehat{y}\,(\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}R) & \to & (IH) \\ Q[z/x]\,\widehat{\beta}\,[v]\,\widehat{y}\,R[z/x] & \stackrel{\Delta}{=} & (Q\widehat{\beta}\,[v]\,\widehat{y}R)[z/x] \end{array}$$

$(P = Q\widehat{\beta}\,\dagger\,\widehat{y}R)$:
$$\begin{array}{lcll} \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}\,(Q\widehat{\beta}\,\dagger\,\widehat{y}R) & \to_{\mathrm{A}} & (\nwarrow cut) \\ (\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}Q)\,\widehat{\beta}\,\dagger\,\widehat{y}\,(\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}R) & \to & (IH) \\ Q[z/x]\,\widehat{\beta}\,\dagger\,\widehat{y}\,R[z/x] & \stackrel{\Delta}{=} & (Q\widehat{\beta}\,\dagger\,\widehat{y}R)[z/x] \end{array}$$

iv) If $\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}P$ then either:

(P *introduces* x): Then either:

($P = Q\widehat{\beta}\,[x]\,\widehat{y}R$, *and* x *does not occur free in* Q, R): Then:

$$\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,(Q\widehat{\beta}\,[x]\,\widehat{y}R) \to (imp)\ \ Q\widehat{\beta}\,[z]\,\widehat{y}R \stackrel{\Delta}{=} (Q\widehat{\beta}\,[x]\,\widehat{y}R)[z/x]$$

($P = \langle x\cdot\beta\rangle$): Then: $\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,\langle x\cdot\beta\rangle \to (cap)\ \langle z\cdot\beta\rangle \stackrel{\Delta}{=} \langle x\cdot\beta\rangle[z/x]$

(P *does not introduce* x): Then $\langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}P \to_{\mathrm{A}} \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\nwarrow\,\widehat{x}P$ and the result follows from part (*iii*). □

These results motivate the extension (in both sub-systems) of the reduction rules, formulating new rules in the shape of the above results.

$$\begin{array}{rrcll} (gc\not\nwarrow): & P\widehat{\alpha}\,\dagger\,\widehat{x}Q & \to & P & (\text{if } \alpha \notin fp(P),\ P \text{ pure}) \\ (\nwarrow gc): & P\widehat{\alpha}\,\dagger\,\widehat{x}Q & \to & Q & (\text{if } x \notin fs(Q),\ Q \text{ pure}) \\ (\nwarrow ren): & P\widehat{\delta}\,\dagger\,\widehat{z}\,\langle z\cdot\alpha\rangle & \to & P[\alpha/\delta] & (P \text{ pure}) \\ (ren\not\nwarrow): & \langle z\cdot\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}P & \to & P[z/x] & (P \text{ pure}) \end{array}$$

Admissibility of these rules for nets that are *not* pure is shown in [9].

2.5 α-conversion

Normally, renaming is an essential part of *α-conversion*, the process of renaming bound objects in a language to avoid clashes during computation. The most familiar context in which this occurs is of course the λ-calculus, where, when reducing a net like $(\lambda xy.xy)(\lambda xy.xy)$, α-conversion is essential. In this section, we will briefly discuss the solution of [9], that deals accurately with this problem in $\mathcal{X}$.

Example 2.12 Take the following reduction:

$$\begin{array}{ll}
(\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,(\langle x{\cdot}\delta\rangle\,\widehat{\delta}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle) & \to\ (\nwarrow a),(\nwarrow imp\text{-}out)\\
(\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{z}\,(((\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\gamma)\,\widehat{\gamma}\nwarrow\widehat{x}\,\langle x{\cdot}\delta\rangle)\,\widehat{\delta}\,[z]\,\widehat{w}\,((\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\gamma)\,\widehat{\gamma}\nwarrow\widehat{x}\,\langle w{\cdot}\alpha\rangle)) & \\
 & \to\ (\nwarrow d),(exp),(\nwarrow cap)\\
(\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{z}\,((\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\delta)\,\widehat{\delta}\,[z]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle) & \to\ (exp\text{-}imp)\\
((\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\delta)\,\widehat{\delta}\dagger\widehat{y}\,\langle y{\cdot}\rho\rangle)\,\widehat{\rho}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle &
\end{array}$$

In the last term, it is clear that we are in breach with Barendregt's convention: ρ is both free and bound in $(\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\delta)\,\widehat{\delta}\dagger\widehat{y}\,\langle y{\cdot}\rho\rangle$. If we were to continue the reduction, we obtain:

$$\begin{array}{rl}
((\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\delta)\,\widehat{\delta}\dagger\widehat{y}\,\langle y{\cdot}\rho\rangle)\,\widehat{\rho}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle & \to\ (exp)\\
(\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\rho)\,\widehat{\rho}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle &
\end{array}$$

Notice that ρ is not introduced in $\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\rho$, since $\rho\in fp(\langle y{\cdot}\rho\rangle)$. So the cut is propagated, and we obtain:

$$\begin{array}{rl}
 & \to\ (a\not\nearrow),(exp\text{-}out\not\nearrow),(d\not\nearrow)\\
(\widehat{y}\,(\langle y{\cdot}\rho\rangle\,\widehat{\rho}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle)\widehat{\rho}\cdot\theta)\,\widehat{\theta}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle & \to\ (cap),(exp)\\
\widehat{y}\,\langle y{\cdot}\alpha\rangle\widehat{\rho}\cdot\alpha. &
\end{array}$$

This is not correct: $(\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\rho)\,\widehat{\rho}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle$ α-converges to $(\widehat{y}\,\langle y{\cdot}\sigma\rangle\widehat{\sigma}\cdot\rho)\,\widehat{\rho}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle$, where the ρ is introduced, and we should have obtained $\widehat{y}\,\langle y{\cdot}\rho\rangle\widehat{\rho}\cdot\alpha$.

It is clear from this example that α-conversion is needed to some extent in any implementation of $\mathcal{X}$. Three[5] solutions to this problem are proposed in [8, 9], that are compared in terms of efficiency. The first uses a *lazy-copying* strategy to avoid sharing of bound connectors; the second *enforces* Barendregt's convention, by renaming bound connectors when nesting is created; the third avoids *capture* of names, but allows breaches of Barendregt's convention. This is achieved by changing, for example, the rule (*exp-imp*)

$$(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R)\ \to\ \begin{cases} Q\widehat{\gamma}\dagger\widehat{y}\,(P\widehat{\beta}\dagger\widehat{z}R)\\ (Q\widehat{\gamma}\dagger\widehat{y}P)\,\widehat{\beta}\dagger\widehat{z}R\end{cases}$$

A conflict with Barendregt's convention is generated in this rule by the fact that perhaps $\beta=\gamma$ or $y=z$, or β occurs in Q, or y in R. Or, when striving for capture avoidance, it might be that y occurs free in R, or β in Q. In either case, these connectors need to be renamed; one of the great plus points of $\mathcal{X}$ is that this can be done *within* the language itself, unlike for the λ-calculus. In fact, using Lemma 2.11, we can give an α-conflict free version of $\mathcal{X}$ (see below).

In contrast, notice that this is not possible for the λ-calculus. There the only reduction rule is $(\lambda x.M)N\to M[N/x]$, and α-conversion is essential when reducing $(\lambda xy.xy)(\lambda xy.xy)$. Without it, one would get

$$(\lambda xy.xy)(\lambda xy.xy)\ \to\ \lambda y.(\lambda xy.xy)y\ \to\ \lambda yy.yy$$

The conflict is caused by the fact that, in the second step, the right-most y is brought *under* the inner-most binder, which causes variables bound by the outer-most binding to 'swap scope' while reducing.

A particular problem in dealing with α-conversion is that in the β-reduction rule $(\lambda x.M)N\to M[N/x]$, the substitution in the right-hand side is supposed to be *immediate*; since the structure of M and N is anonymous, it is impossible to detect an α-conflict here that typically depends on bindings occurring *inside* M and N. For example, in the first step of the reduction above, the latter term is *identical* to $\lambda y.xy[(\lambda xy.xy)/x]$; the actual performance of the substitution, which *brings* the right-most binder under the left-most is not part of the reduction

[5] NB: De Bruijn indices are not discussed in [9].

system itself, but specified in the auxiliary definition of substitution. This makes α-conversion difficult to tackle in the context of the pure λ-calculus.

To consider substitution as a separate syntactic construct implies moving from the λ-calculus to $\lambda\mathbf{x}$. There the situation is slightly different, in that we can now say that

$$(\lambda y.M)\langle x{=}N\rangle \to \lambda z.(M\langle y{=}z\rangle\langle x{=}N\rangle),$$

thereby preventing a conflict on a possibly free y in N. This is expensive though, as it is performed on *all* substitutions on abstractions, and does not actually detect the conflict, but just prevents it.

In $\mathcal{X}$, not only is the α-conflict solved, but also detected, all within the reduction system of $\mathcal{X}$ itself, by essentially expressing

$$(\lambda y.M)\langle x{=}N\rangle \to \lambda z.(M\langle y{=}z\rangle\langle x{=}N\rangle),\ \textit{if } y \textit{ free in } N$$

As shown in [9], to accurately deal with α-conversion for the case of capture-avoidance, the rule $(\textit{exp-imp})$ needs to be replaced by (assume that α, x are introduced, and that v, δ are fresh):

$$\begin{array}{lcll}
(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) & \to & Q\widehat{\gamma}\dagger\widehat{y}\,(P\widehat{\beta}\dagger\widehat{z}R) & (y\notin fs(R)) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) & \to & Q\widehat{\gamma}\dagger\widehat{v}\,((\langle v{\cdot}\delta\rangle\,\widehat{\delta}\nwarrow\widehat{y}P)\,\widehat{\beta}\dagger\widehat{z}R) & (y\in fs(R)) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) & \to & (Q\widehat{\gamma}\dagger\widehat{y}P)\,\widehat{\beta}\dagger\widehat{z}R & (\beta\notin fp(Q)) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) & \to & (Q\widehat{\gamma}\dagger\widehat{y}\,(P\widehat{\beta}\not\swarrow\widehat{b}\,\langle v{\cdot}\delta\rangle))\,\widehat{\delta}\dagger\widehat{z}R & (\beta\in fp(Q))
\end{array}$$

Almost all propagation rules (exceptions are $(d\not\swarrow)$, $(cap\not\swarrow)$, $(\nwarrow d)$, and $(\nwarrow cap)$) need dealing with as well.

3 Typing for $\mathcal{X}$: from sequent calculus to $\mathcal{X}$

As mentioned in the introduction, $\mathcal{X}$ is inspired by the sequent calculus, so it is worthwhile to recall some of the principles. The sequent calculus we consider has only implication, no structural rules and a changed axiom. It offers an extremely natural presentation of the classical propositional calculus with implication, and is a variant of system LK. It has four rules: *axiom*, *right introduction* of the arrow, *left introduction* and *cut*.

$$(ax):\ \frac{}{\Gamma, A \vdash_{\text{LK}} A, \Delta} \qquad (\Rightarrow L):\ \frac{\Gamma \vdash_{\text{LK}} A, \Delta \quad \Gamma, B \vdash_{\text{LK}} \Delta}{\Gamma, A \Rightarrow B \vdash_{\text{LK}} \Delta}$$

$$(\Rightarrow R):\ \frac{\Gamma, A \vdash_{\text{LK}} B, \Delta}{\Gamma \vdash_{\text{LK}} A \Rightarrow B, \Delta} \qquad (cut):\ \frac{\Gamma \vdash_{\text{LK}} A, \Delta \quad \Gamma, A \vdash_{\text{LK}} \Delta}{\Gamma \vdash_{\text{LK}} \Delta}$$

The elimination of rule (cut) plays a major role in LK, since for proof theoreticians, cut-free proofs enjoy nice properties. Proof reductions by cut-elimination have been proposed by Gentzen; those reductions become the fundamental principle of computation in $\mathcal{X}$.

Another nice property of proof systems is known as the *Curry-Howard correspondence*:

Definition 3.1 (Curry-Howard isomorphism) "*Terms as Proofs, Types as Propositions.*" Let M be a (closed) term, and A a type, then M is of type A if and only if A, read as a logical formula, is provable in the corresponding logic, using a proof which structure corresponds to M.

This isomorphism expresses the fact that one can associate a term with a proof such that propositions become types and proof reductions become term reductions (or computations in $\mathcal{X}$). This phenomenon was first discovered for Combinatory Logic [20], and later the connection between the λ-calculus and intuitionistic logic was put in evidence. The Curry-Howard correspondence for $\mathcal{X}$ is with the classical propositional calculus and is given through the

sequent calculus described above. Propositions receive names; those that appear in the left part of a sequent receive names like x,y,z, etc, and those that appear in the right part of a sequent receive names like α,β,γ, etc.

Definition 3.2 (Types and Contexts) *i*) The set of types is defined by the grammar:

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

where φ is a basic type. The types considered in this paper are normally known as *simple* (or *Curry*) types.

ii) A *context of sockets* Γ is a mapping from sockets to types, denoted as a finite set of *statements* $x{:}A$, such that the *subject* of the statements (x) are distinct. We write $\Gamma, x{:}A$ for the context defined by:

$$\begin{array}{rcll} \Gamma, x{:}A & = & \Gamma \cup \{x{:}A\} & (\Gamma \text{ is not defined on } x) \\ & = & \Gamma & (\text{otherwise}) \end{array}$$

(Notice that the second case implies that $x{:}A \in \Gamma$.) So, when writing a context as $\Gamma, x{:}A$, this implies that $x{:}A \in \Gamma$, or Γ is not defined on x. When we write Γ_1, Γ_2 we mean the union of Γ_1 and Γ_2 when Γ_1 and Γ_2 are coherent (if Γ_1 contains $x{:}A_1$ and Γ_2 contains $x{:}A_2$ then $A_1 = A_2$).

iii) Contexts of *plugs* Δ are defined in a similar way.

The notion of type assignment on $\mathcal{X}$ that we present in this section is the basic implicative system for Classical Logic (Gentzen system LK) as described above. The Curry-Howard property is easily achieved by erasing all term-information.

Definition 3.3 (Typing for $\mathcal{X}$) *i*) *Type judgements* are expressed via a ternary relation $P \vdots \Gamma \vdash_{\mathcal{X}} \Delta$, where Γ is a context of *sockets* and Δ is a context of *plugs*, and P is a net. We say that P is the *witness* of this judgement.

ii) *Type assignment for* $\mathcal{X}$ is defined by the following sequent calculus:

$$(cap): \ \frac{}{\langle y{\cdot}\alpha \rangle \vdots \Gamma, y{:}A \vdash \alpha{:}A, \Delta} \qquad (imp): \ \frac{P \vdots \Gamma \vdash \alpha{:}A, \Delta \quad Q \vdots \Gamma, x{:}B \vdash \Delta}{P\widehat{\alpha}\,[y]\,\widehat{x}Q \vdots \Gamma, y{:}A \to B \vdash \Delta}$$

$$(exp): \ \frac{P \vdots \Gamma, x{:}A \vdash \alpha{:}B, \Delta}{\widehat{x}P\widehat{\alpha}{\cdot}\beta \vdots \Gamma \vdash \beta{:}A \to B, \Delta} \qquad (cut): \ \frac{P \vdots \Gamma \vdash \alpha{:}A, \Delta \quad Q \vdots \Gamma, x{:}A \vdash \Delta}{P\widehat{\alpha} \dagger \widehat{x}Q \vdots \Gamma \vdash \Delta}$$

We write $P \vdots \Gamma \vdash_{\mathcal{X}} \Delta$ if there exists a derivation that has this judgement in the bottom line, and write $\mathcal{D} :: P \vdots \Gamma \vdash_{\mathcal{X}} \Delta$ if we want to name that derivation.

Γ and Δ carry the types of the free connectors in P, as unordered sets. There is no notion of type for P itself, instead the derivable statement shows how P is connectable.

Lemma 3.4 (Weakening) *The following rule is admissible:*

$$\frac{P \vdots \Gamma \vdash_{\mathcal{X}} \Delta}{P \vdots \Gamma' \vdash_{\mathcal{X}} \Delta'}\,(Wk)$$

for any $\Gamma' \supseteq \Gamma$ *and* $\Delta' \supseteq \Delta$.

Proof: The proof is by induction on the proof tree for $P \vdots \Gamma \vdash_{\mathcal{X}} \Delta$. We will only consider two cases, the two other work the same way.

(cap): Then $P \equiv \langle y{\cdot}\alpha \rangle$. Then $\Gamma' \supseteq \Gamma \supseteq \{y{:}A\}$ and $\Delta' \supseteq \Delta \supseteq \{\alpha{:}A\}$, hence $\langle y{\cdot}\alpha \rangle \vdots \Gamma' \vdash_{\mathcal{X}} \Delta'$.

(imp): Then $P \equiv Q\widehat{\alpha}\,[y]\,\widehat{x}R$, $\Gamma \equiv \Gamma_1, y{:}A \to B$ and

$$P :\cdot \Gamma \vdash_{\mathcal{X}} \Delta \;\equiv\; Q\widehat{\alpha}\,[y]\,\widehat{x}R :\cdot \Gamma_1, y{:}A \to B \vdash_{\mathcal{X}} \Delta.$$

We have $Q :\cdot \Gamma_1 \vdash_{\mathcal{X}} \alpha{:}A, \Delta$ and $R :\cdot \Gamma_1, x{:}B \vdash_{\mathcal{X}} \Delta$. If $y{:}A \to B \in \Gamma_1$, i.e., $\Gamma \equiv \Gamma_1$ then we write $\Gamma_1' \equiv \Gamma'$ else we write $\Gamma_1' \equiv \Gamma' \backslash y{:}A \to B$. Notice that $\Gamma_1' \supseteq \Gamma_1$. By induction, we have $Q :\cdot \Gamma_1' \vdash_{\mathcal{X}} \alpha{:}A, \Delta'$ and $R :\cdot \Gamma_1', x{:}B \vdash_{\mathcal{X}} \Delta'$ and $Q\widehat{\alpha}\,[y]\,\widehat{x}R :\cdot \Gamma_1', y{:}A \to B \vdash_{\mathcal{X}} \Delta'$ follows by (imp). □

Example 3.5 (A proof of Peirce's Law) The following is a proof for Peirce's Law in Classical Logic:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A, B}\,(Ax)}{\vdash A \Rightarrow B, A}\,(\Rightarrow R) \qquad \dfrac{}{A \vdash A}\,(Ax)}{(A \Rightarrow B) \Rightarrow A \vdash A}\,(\Rightarrow L)}{\vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A}\,(\Rightarrow R)$$

Inhabiting this proof in $\mathcal{X}$ gives the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle y{\cdot}\delta\rangle :\cdot y{:}A \vdash \delta{:}A, \eta{:}B}\,(cap)}{\widehat{y}\,\langle y{\cdot}\delta\rangle\widehat{\eta}\cdot\phi :\cdot \vdash \phi{:}A{\to}B, \delta{:}A}\,(exp) \qquad \dfrac{}{\langle w{\cdot}\delta\rangle :\cdot w{:}A \vdash \delta{:}A}\,(cap)}{(\widehat{y}\,\langle y{\cdot}\delta\rangle\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[z]\,\widehat{w}\,\langle w{\cdot}\delta\rangle :\cdot z{:}(A{\to}B){\to}A \vdash \delta{:}A}\,(imp)}{\widehat{z}\,((\widehat{y}\,\langle y{\cdot}\delta\rangle\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[z]\,\widehat{w}\,\langle w{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma :\cdot \vdash \gamma{:}((A{\to}B){\to}A){\to}A}\,(exp)$$

The soundness result of simple type assignment with respect to reduction is stated as usual:

Theorem 3.6 (Witness reduction) *If $P :\cdot \Gamma \vdash_{\mathcal{X}} \Delta$, and $P \to Q$, then $Q :\cdot \Gamma \vdash_{\mathcal{X}} \Delta$.*

Proof: By induction on the length of reduction sequences, of which we only show the interesting base cases; notice that, occasionally, weakening will be used.

(*Logical rules*): (cap): $\langle y{\cdot}\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,\langle x{\cdot}\beta\rangle \to \langle y{\cdot}\beta\rangle$

$$\dfrac{\dfrac{}{\langle y{\cdot}\alpha\rangle :\cdot \Gamma, y{:}A \vdash \alpha{:}A, \Delta} \qquad \dfrac{}{\langle x{\cdot}\beta\rangle :\cdot \Gamma, x{:}A \vdash \beta{:}A, \Delta}}{\langle y{\cdot}\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,\langle x{\cdot}\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}A, \Delta} \qquad \dfrac{}{\langle y{\cdot}\beta\rangle :\cdot \Gamma, y{:}A \vdash \beta{:}A, \Delta}$$

(exp): $(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\,\dagger\,\widehat{x}\,\langle x{\cdot}\gamma\rangle \to \widehat{y}P\widehat{\beta}\cdot\gamma$

$$\dfrac{\dfrac{\boxed{\mathcal{D}}}{P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \gamma{:}A{\to}B, \Delta}}{\widehat{y}P\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}A{\to}B, \gamma{:}A{\to}B, \Delta} \qquad \dfrac{}{\langle x{\cdot}\gamma\rangle :\cdot \Gamma, x{:}A{\to}B \vdash \gamma{:}A{\to}B, \Delta}}{(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\,\dagger\,\widehat{x}\,\langle x{\cdot}\gamma\rangle :\cdot \Gamma \vdash \gamma{:}A{\to}B, \Delta}$$

$$\dfrac{\dfrac{\boxed{\mathcal{D}}}{P :\cdot \Gamma, y{:}A \vdash \beta{:}B, \gamma{:}A{\to}B, \Delta}}{\widehat{y}P\widehat{\beta}\cdot\gamma :\cdot \Gamma \vdash \gamma{:}A{\to}B, \Delta}$$

(imp): $\langle y{\cdot}\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,(Q\widehat{\beta}\,[x]\,\widehat{z}R) \to Q\widehat{\beta}\,[y]\,\widehat{z}R$

$$\dfrac{\dfrac{}{\langle y{\cdot}\alpha\rangle :\cdot \Gamma, y{:}A{\to}B \vdash \alpha{:}A{\to}B, \Delta} \qquad \dfrac{\dfrac{\boxed{\mathcal{D}_1}}{Q :\cdot \Gamma, y{:}A{\to}B \vdash \beta{:}A, \Delta} \quad \dfrac{\boxed{\mathcal{D}_2}}{R :\cdot \Gamma, y{:}A{\to}B, z{:}B \vdash \Delta}}{Q\widehat{\beta}\,[x]\,\widehat{z}R :\cdot \Gamma, y{:}A{\to}B, x{:}A{\to}B \vdash \Delta}}{\langle y{\cdot}\alpha\rangle\,\widehat{\alpha}\,\dagger\,\widehat{x}\,(Q\widehat{\beta}\,[x]\,\widehat{z}R) :\cdot \Gamma, y{:}A{\to}B \vdash \Delta}$$

$$\dfrac{\boxed{\mathcal{D}_1}\quad Q \vdots \cdot\, \Gamma, y{:}A{\to}B \vdash \beta{:}A, \Delta \qquad \boxed{\mathcal{D}_2}\quad R \vdots \cdot\, \Gamma, y{:}A{\to}B, z{:}B \vdash \Delta}{Q\,\widehat{\beta}\,[y]\,\widehat{z}R \vdots \cdot\, \Gamma, y{:}A{\to}B \vdash \Delta}$$

$(\mathit{exp\text{-}imp})\colon\ (\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to Q\widehat{\gamma}\dagger\widehat{y}\,(P\widehat{\beta}\dagger\widehat{z}R)$

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad P \vdots \cdot\, \Gamma, y{:}A \vdash \beta{:}B, \Delta}{\widehat{y}P\widehat{\beta}\cdot\alpha \vdots \cdot\, \Gamma \vdash \alpha{:}A{\to}B, \Delta} \qquad \dfrac{\boxed{\mathcal{D}_2}\quad Q \vdots \cdot\, \Gamma \vdash \gamma{:}A, \Delta \qquad \boxed{\mathcal{D}_3}\quad R \vdots \cdot\, \Gamma, z{:}B \vdash \Delta}{Q\widehat{\gamma}\,[x]\,\widehat{z}R \vdots \cdot\, \Gamma, x{:}A{\to}B \vdash \Delta}}{(\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \vdots \cdot\, \Gamma \vdash \Delta}$$

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_2}\quad Q \vdots \cdot\, \Gamma \vdash \gamma{:}A, \Delta}{Q \vdots \cdot\, \Gamma \vdash \beta{:}B, \gamma{:}A, \Delta} \qquad \boxed{\mathcal{D}_1}\quad P \vdots \cdot\, \Gamma, y{:}A \vdash \beta{:}B, \Delta}{Q\widehat{\gamma}\dagger\widehat{y}P \vdots \cdot\, \Gamma \vdash \beta{:}B, \Delta} \qquad \boxed{\mathcal{D}_3}\quad R \vdots \cdot\, \Gamma, z{:}B \vdash \Delta}{(Q\widehat{\gamma}\dagger\widehat{y}P)\,\widehat{\beta}\dagger\widehat{z}R \vdots \cdot\, \Gamma \vdash \Delta}$$

$(\mathit{exp\text{-}imp})\colon\ (\widehat{y}P\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to Q\widehat{\gamma}\dagger\widehat{y}\,(P\widehat{\beta}\dagger\widehat{z}R)$

$$\dfrac{\boxed{\mathcal{D}_2}\quad Q \vdots \cdot\, \Gamma \vdash \gamma{:}A, \Delta \qquad \dfrac{\boxed{\mathcal{D}_1}\quad P \vdots \cdot\, \Gamma, y{:}A \vdash \beta{:}B, \Delta \qquad \dfrac{\boxed{\mathcal{D}_3}\quad R \vdots \cdot\, \Gamma, z{:}B \vdash \Delta}{R \vdots \cdot\, \Gamma, y{:}A, z{:}B \vdash \Delta}}{P\widehat{\beta}\dagger\widehat{z}R \vdots \cdot\, \Gamma, y{:}A \vdash \Delta}}{Q\widehat{\gamma}\dagger\widehat{y}\,(P\widehat{\beta}\dagger\widehat{z}R) \vdots \cdot\, \Gamma \vdash \Delta}$$

$(\not\prec\ \mathit{propagation})\colon\ \ (\mathit{exp\text{-}out}\not\prec)\colon\ (\widehat{y}Q\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\not\prec\widehat{x}P \to (\widehat{y}\,(Q\widehat{\alpha}\not\prec\widehat{x}P)\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{x}P$

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad Q \vdots \cdot\, \Gamma, y{:}A \vdash \beta{:}B, \Delta}{\widehat{y}Q\widehat{\beta}\cdot\alpha \vdots \cdot\, \Gamma \vdash \alpha{:}A{\to}B, \Delta} \qquad \boxed{\mathcal{D}_2}\quad P \vdots \cdot\, \Gamma, x{:}A{\to}B \vdash \Delta}{(\widehat{y}Q\widehat{\beta}\cdot\alpha)\,\widehat{\alpha}\not\prec\widehat{x}P \vdots \cdot\, \Gamma \vdash \Delta}$$

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad Q \vdots \cdot\, \Gamma, y{:}A \vdash \beta{:}B, \Delta}{Q \vdots \cdot\, \Gamma, y{:}A \vdash \alpha{:}A{\to}B, \beta{:}B, \Delta} \qquad \dfrac{\boxed{\mathcal{D}_2}\quad P \vdots \cdot\, \Gamma, x{:}A{\to}B \vdash \Delta}{P \vdots \cdot\, \Gamma, x{:}A{\to}B \vdash \beta{:}B, \Delta}}{Q\widehat{\alpha}\not\prec\widehat{x}P \vdots \cdot\, \Gamma, y{:}A \vdash \beta{:}B, \Delta} \qquad \boxed{\mathcal{D}_2}\quad P \vdots \cdot\, \Gamma, x{:}A{\to}B \vdash \Delta}{(\widehat{y}\,(Q\widehat{\alpha}\not\prec\widehat{x}P)\widehat{\beta}\cdot\gamma)\,\widehat{\gamma}\not\prec\widehat{x}P \vdots \cdot\, \Gamma \vdash \Delta}$$

$(\mathit{imp}\not\prec)\colon\ (Q\widehat{\beta}\,[z]\,\widehat{y}R)\,\widehat{\alpha}\not\prec\widehat{x}P \to (Q\widehat{\alpha}\not\prec\widehat{x}P)\,\widehat{\beta}\,[z]\,\widehat{y}\,(R\widehat{\alpha}\not\prec\widehat{x}P)$

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\quad Q \vdots \cdot\, \Gamma \vdash \alpha{:}C, \beta{:}A, \Delta \qquad \boxed{\mathcal{D}_2}\quad R \vdots \cdot\, \Gamma, y{:}B \vdash \alpha{:}C, \Delta}{Q\widehat{\beta}\,[z]\,\widehat{y}R \vdots \cdot\, \Gamma, z{:}A{\to}B \vdash \alpha{:}C, \Delta} \qquad \boxed{\mathcal{D}_3}\quad P \vdots \cdot\, \Gamma, z{:}A{\to}B, x{:}C \vdash \Delta}{(Q\widehat{\beta}\,[z]\,\widehat{y}R)\,\widehat{\alpha}\not\prec\widehat{x}P \vdots \cdot\, \Gamma, z{:}A{\to}B \vdash \Delta}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{Q \vdots \cdot \Gamma \vdash \alpha{:}C, \beta{:}A, \Delta}}{Q \vdots \cdot \Gamma, z{:}A \to B \vdash \alpha{:}C, \beta{:}A, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_3}{P \vdots \cdot \Gamma, z{:}A \to B, x{:}C \vdash \Delta}}{P \vdots \cdot \Gamma, z{:}A \to B, x{:}C \vdash \beta{:}A, \Delta}}{Q\widehat{\alpha} \not\leftarrow \widehat{x} P \vdots \cdot \Gamma, z{:}A \to B \vdash \beta{:}A, \Delta} \quad \dfrac{\dfrac{\dfrac{\mathcal{D}_2}{R \vdots \cdot \Gamma, y{:}B \vdash \alpha{:}C, \Delta}}{R \vdots \cdot \Gamma, z{:}A \to B, y{:}B \vdash \alpha{:}C, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_3}{P \vdots \cdot \Gamma, z{:}A \to B, x{:}C \vdash \Delta}}{P \vdots \cdot \Gamma, z{:}A \to B, x{:}C, y{:}B \vdash \Delta}}{R\widehat{\alpha} \not\leftarrow \widehat{x} P \vdots \cdot \Gamma, z{:}A \to B, y{:}B \vdash \Delta}}{(Q\widehat{\alpha} \not\leftarrow \widehat{x} P)\, \widehat{\beta}\, [z]\, \widehat{y}\, (R\widehat{\alpha} \not\leftarrow \widehat{x} P) \vdots \cdot \Gamma, z{:}A \to B \vdash \Delta}$$

$(cut\not\leftarrow)\colon (Q\widehat{\beta} \dagger \widehat{y} R)\, \widehat{\alpha} \not\leftarrow \widehat{x} P \to (Q\widehat{\alpha} \not\leftarrow \widehat{x} P)\, \widehat{\beta} \dagger \widehat{y}\, (R\widehat{\alpha} \not\leftarrow \widehat{x} P)$

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{Q \vdots \cdot \Gamma \vdash \alpha{:}C, \beta{:}B, \Delta} \quad \dfrac{\mathcal{D}_2}{R \vdots \cdot \Gamma, y{:}B \vdash \alpha{:}C, \Delta}}{Q\widehat{\beta} \dagger \widehat{y} R \vdots \cdot \Gamma \vdash \alpha{:}C, \Delta} \quad \dfrac{\mathcal{D}_3}{P \vdots \cdot \Gamma, x{:}C \vdash \Delta}}{(Q\widehat{\beta} \dagger \widehat{y} R)\, \widehat{\alpha} \not\leftarrow \widehat{x} P \vdots \cdot \Gamma \vdash \Delta}$$

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{Q \vdots \cdot \Gamma \vdash \alpha{:}C, \beta{:}B, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_3}{P \vdots \cdot \Gamma, x{:}C \vdash \Delta}}{P \vdots \cdot \Gamma, x{:}C \vdash \beta{:}B, \Delta}}{Q\widehat{\alpha} \not\leftarrow \widehat{x} P \vdots \cdot \Gamma \vdash \beta{:}B, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_2}{R \vdots \cdot \Gamma, y{:}B \vdash \alpha{:}C, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_3}{P \vdots \cdot \Gamma, x{:}C \vdash \Delta}}{P \vdots \cdot \Gamma, y{:}B, x{:}C \vdash \Delta}}{R\widehat{\alpha} \not\leftarrow \widehat{x} P \vdots \cdot \Gamma, y{:}B \vdash \Delta}}{(Q\widehat{\alpha} \not\leftarrow \widehat{x} P)\, \widehat{\beta} \dagger \widehat{y}\, (R\widehat{\alpha} \not\leftarrow \widehat{x} P) \vdots \cdot \Gamma \vdash \Delta}$$

$(\nwarrow propagation)\colon\ (\nwarrow imp\text{-}out)\colon P\widehat{\alpha} \nwarrow \widehat{x}\, (Q\widehat{\beta}\, [x]\, \widehat{y} R) \to P\widehat{\alpha} \dagger \widehat{z}\, ((P\widehat{\alpha} \nwarrow \widehat{x} Q)\, \widehat{\beta}\, [z]\, \widehat{y}\, (P\widehat{\alpha} \nwarrow \widehat{x} R))$

$$\dfrac{\dfrac{\mathcal{D}_1}{P \vdots \cdot \Gamma \vdash \alpha{:}A \to B, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_2}{Q \vdots \cdot \Gamma \vdash \beta{:}A, \Delta} \quad \dfrac{\mathcal{D}_3}{R \vdots \cdot \Gamma, y{:}B \vdash \Delta}}{Q\widehat{\beta}\, [x]\, \widehat{y} R \vdots \cdot \Gamma, x{:}A \to B \vdash \Delta}}{P\widehat{\alpha} \nwarrow \widehat{x}\, (Q\widehat{\beta}\, [x]\, \widehat{y} R) \vdots \cdot \Gamma \vdash \Delta}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{P \vdots \cdot \Gamma \vdash \alpha{:}A \to B, \Delta} \quad \dfrac{\mathcal{D}_2}{Q \vdots \cdot \Gamma \vdash \beta{:}A, \Delta}}{P \vdots \cdot \Gamma \vdash \alpha{:}A \to B, \beta{:}A, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_1}{P \vdots \cdot \Gamma \vdash \alpha{:}A \to B, \Delta} \quad \dfrac{\mathcal{D}_3}{R \vdots \cdot \Gamma, y{:}B \vdash \Delta}}{P \vdots \cdot \Gamma, y{:}B \vdash \alpha{:}A \to B, \Delta}}{P\widehat{\alpha} \nwarrow \widehat{x} Q \vdots \cdot \Gamma \vdash \beta{:}A, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_1}{P \vdots \cdot \Gamma \vdash \alpha{:}A \to B, \Delta} \quad \dfrac{\mathcal{D}_3}{R \vdots \cdot \Gamma, y{:}B \vdash \Delta}}{R \vdots \cdot \Gamma, x{:}A \to B, y{:}B \vdash \Delta}}{P\widehat{\alpha} \nwarrow \widehat{x} R \vdots \cdot \Gamma, y{:}B \vdash \Delta}}{P\widehat{\alpha} \dagger \widehat{v}\, ((P\widehat{\alpha} \nwarrow \widehat{x} Q)\, \widehat{\beta}\, [v]\, \widehat{y}\, (P\widehat{\alpha} \nwarrow \widehat{x} R)) \vdots \cdot \Gamma \vdash \Delta}$$

where $\mathcal{D}_1$ is $P \vdots \cdot \Gamma \vdash \alpha{:}A \to B, \Delta$ and $(P\widehat{\alpha} \nwarrow \widehat{x} Q)\, \widehat{\beta}\, [v]\, \widehat{y}\, (P\widehat{\alpha} \nwarrow \widehat{x} R) \vdots \cdot \Gamma, v{:}A \to B \vdash \Delta$ is the premise.

$(\nwarrow cut)\colon P\widehat{\alpha} \nwarrow \widehat{x}\, (Q\widehat{\beta} \dagger \widehat{y} R) \to (P\widehat{\alpha} \nwarrow \widehat{x} Q)\, \widehat{\beta} \dagger \widehat{y}\, (P\widehat{\alpha} \nwarrow \widehat{x} R)$

$$\dfrac{\dfrac{\mathcal{D}_1}{P \vdots \cdot \Gamma \vdash \alpha{:}A, \Delta} \quad \dfrac{\dfrac{\mathcal{D}_2}{Q \vdots \cdot \Gamma, x{:}A \vdash \beta{:}B, \Delta} \quad \dfrac{\mathcal{D}_3}{R \vdots \cdot \Gamma, x{:}A, y{:}B \vdash \Delta}}{Q\widehat{\beta} \dagger \widehat{y} R \vdots \cdot \Gamma, x{:}A \vdash \Delta}}{P\widehat{\alpha} \nwarrow \widehat{x}\, (Q\widehat{\beta} \dagger \widehat{y} R) \vdots \cdot \Gamma \vdash \Delta}$$

$$\dfrac{\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P \vdots \cdot \Gamma \vdash \alpha{:}A, \Delta} \quad \boxed{\mathcal{D}_2}}{P \vdots \cdot \Gamma \vdash \alpha{:}A, \beta{:}B, \Delta \quad Q \vdots \cdot \Gamma, x{:}A \vdash \beta{:}B, \Delta} \quad \dfrac{\dfrac{\boxed{\mathcal{D}_1}}{P \vdots \cdot \Gamma \vdash \alpha{:}A, \Delta} \quad \boxed{\mathcal{D}_3}}{R \vdots \cdot \Gamma, y{:}B \vdash \alpha{:}A, \Delta \quad P \vdots \cdot \Gamma, x{:}A, y{:}B \vdash \Delta}}{P\widehat{\alpha}\,\diagdown\,\widehat{x}Q \vdots \cdot \Gamma \vdash \beta{:}B, \Delta \qquad P\widehat{\alpha}\,\diagdown\,\widehat{x}R \vdots \cdot \Gamma, y{:}B \vdash \Delta}}{(P\widehat{\alpha}\,\diagdown\,\widehat{x}Q)\,\widehat{\beta}\,\dagger\,\widehat{y}\,(P\widehat{\alpha}\,\diagdown\,\widehat{x}R) \vdots \cdot \Gamma \vdash \Delta}$$

4 Interpreting the λ-calculus

In this section, we will illustrate the expressive power of $\mathcal{X}$ by showing that we can faithfully interpret the λ-calculus [16, 11] (a similar result was shown in [45]), and in the following sections we will show a comparable result for $\lambda\mathbf{x}$, $\lambda\mu$, and $\overline{\lambda}\mu\tilde{\mu}$. Using the notion of Curry type assignment, we will show that assignable types are preserved by the interpretation.

In part, the interpretation results could be seen as variants of similar results obtained by Curien and Herbelin in [19]. Indeed, we could have defined our mappings using the mappings of λ-calculus and $\lambda\mu$ into $\overline{\lambda}\mu\tilde{\mu}$, and concatenating those to the mapping from $\overline{\lambda}\mu\tilde{\mu}$ to $\mathcal{X}$, but our encoding is more detailed and precise than that and deals with explicit substitution as well.

One should notice that for [19] the preservation of the CBV-evaluation and CBN-evaluation relies on two *distinct* translations of terms. For instance, the CBV- and CBN-λ-calculus can both be encoded into CPS [3], and there it is clear that what accounts for the distinction between CBV and CBN is the encodings themselves, and not the way CPS reduces the encoded terms.

In contrast, in $\mathcal{X}$ we have no need of two separate interpretation functions, but will define only *one*. Combining this with the two sub-reduction systems $\to_{\mathrm{v}}$ and $\to_{\mathrm{N}}$ we can encode the CBV- and CBN-λ-calculus.

We assume the reader to be familiar with the λ-calculus [11]; we just recall the definition of lambda terms and β-contraction.

Definition 4.1 (LAMBDA TERMS AND β-CONTRACTION [11]) *i*) The set λ of *lambda terms* is defined by the syntax':

$$M, N \;::=\; x \mid \lambda x.M \mid MN$$

Terms x and $\lambda x.M$ are called *values*.

ii) The reduction relation $\to_\beta$ is defined as the contextual (i.e. compatible [11]) closure of the rule:

$$(\lambda x.M)N \;\to_\beta\; M[N/x]$$

iii) If the contraction $(\lambda x.M)N \;\to_\beta\; M[N/x]$ is fired only when N is a value, then the reduction is called *call-by-value* (or CBV for short) and written $\to_{\mathrm{v}}$. No confusion is possible with the reduction with the same name in $\mathcal{X}$, since the nets on which both reductions apply are not in the same syntactic category.

This calculus has a notion of type assignment that corresponds nicely to implicational propositional logic, in the framework of natural deduction.

The rules of natural deduction define how to manipulate logical objects called sequents that have the form: $S \vdash A$, where A is a formula of propositional logic, and S is a set of such formulae. The sequent means that A can be proved from the axioms S. The rules of natural deduction either introduce or eliminate connectives in the right-hand side proposition of the sequent.

For instance the implication is introduced by the rule

$$\frac{\Gamma, A \vdash_{\text{LK}} B}{\Gamma \vdash_{\text{LK}} A \Rightarrow B}\,(\Rightarrow I)$$

and eliminated by Modus Ponens

$$\frac{\Gamma \vdash_{\text{LK}} A \Rightarrow B \quad \Gamma \vdash_{\text{LK}} A}{\Gamma \vdash_{\text{LK}} B}\,(\Rightarrow E)$$

We add to that the rule that allows us to use an axiom:

$$\frac{}{\Gamma \vdash_{\text{LK}} A}\,(A \in \Gamma)$$

These three rules form the intuitionistic implicative logic. Notice that this logic is less expressive than classical logic: for instance, from those rules Peirce's law (Example 3.5) cannot be proved.

We can simulate $(\Rightarrow I)$ easily in sequent calculus; the Modus Ponens rule of natural deduction is simulated by a short reasoning:

$$\dfrac{\dfrac{\dfrac{\dfrac{\square}{\Gamma \vdash A \Rightarrow B, \Delta}\,(W)}{\Gamma \vdash A \Rightarrow B, B, \Delta}\,(W) \quad \dfrac{\dfrac{\square}{\Gamma \vdash A, \Delta}\,(W) \quad \dfrac{}{\Gamma, B \vdash B, \Delta}\,(ax)}{\Gamma, A \Rightarrow B \vdash B, \Delta}\,(L\Rightarrow)}{\Gamma \vdash B, \Delta}\,(cut)$$

(cf [24]). Now the situation where the introduction rule of a connective is followed directly by the elimination rule is traditionally called a *cut* in natural deduction [21]. In that case the proof can be easily transformed into a simpler one, by the process of cut-*elimination*. For instance,

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}}{\Gamma, A \vdash_{\text{LK}} B}}{\Gamma \vdash_{\text{LK}} A \Rightarrow B} \qquad \dfrac{\boxed{\mathcal{D}_2}}{\Gamma \vdash_{\text{LK}} A}$$
$$\Gamma \vdash_{\text{LK}} B$$

can be transformed into a simpler proof: the one of $\Gamma, A \vdash B$ in which every time the axiom A is used we replace the use of the axiom by the proof of $\Gamma \vdash A$.

$$\dfrac{\boxed{\mathcal{D}_2}}{\Gamma \vdash_{\text{LK}} A}$$
$$\dfrac{\boxed{\mathcal{D}_1}}{\Gamma \vdash_{\text{LK}} B}$$

Hence the conclusion is $\Gamma \vdash B$ as required.

Now, formulae of propositional logic can be seen as types of functional programming (especially the simply typed λ-calculus) and vice-versa. The implication $A \Rightarrow B$ corresponds to a functional type $A \to B$. And further, the inference rules of intuitionistic propositional logic are isomorphic to the typing rules of simply typed λ-calculus:

Definition 4.2 (Type assignment for the λ-calculus)

$$(Ax):\ \frac{}{\Gamma, x{:}A \vdash x : A} \qquad (\to I):\ \frac{\Gamma, x{:}A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad (\to E):\ \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

We first define the direct encoding of the λ-calculus into $\mathcal{X}$:

Definition 4.3 (Interpretation of the λ-calculus in $\mathcal{X}$)

$$\begin{array}{rcll}
[\![x]\!]^{\lambda}_{\alpha} & \triangleq & \langle x{\cdot}\alpha\rangle & \\
[\![\lambda x.M]\!]^{\lambda}_{\alpha} & \triangleq & \widehat{x}\,[\![M]\!]^{\lambda}_{\beta}\widehat{\beta}\cdot\alpha & \beta \textit{ fresh} \\
[\![MN]\!]^{\lambda}_{\alpha} & \triangleq & [\![M]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\dagger\widehat{x}\,([\![N]\!]^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \gamma,\beta,x,y \textit{ fresh}
\end{array}$$

Observe that every sub-net of $[\![M]\!]^{\lambda}_{\alpha}$ has exactly one free plug, and that this is precisely α. Moreover, notice that, in the λ-calculus, the output (i.e. result) is anonymous; where an operand 'moves' to carries a name via a variable, but where it comes from is not mentioned, since it is implicit. Since in $\mathcal{X}$, a net is allowed to return a result in more than one way, in order to be able to connect outputs to inputs we have to name the outputs; this forces a name on the output of an interpreted λ-term M as well, carried in the sub-script of $[\![M]\!]^{\lambda}_{\alpha}$; this name α is also the name of the current continuation, i.e. the name of the hole in the context in which M occurs.

In [45], a similar interpretation is defined that differs in the last case, where it states:

$$[\![MN]\!]^{\lambda}_{\alpha} \triangleq [\![M]\!]^{\lambda}_{\gamma}\ [\ \gamma := (x)([\![N]\!]^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)\ ]$$

This definition depends on an additional notion of substitution $P\ [\gamma := (x)Q]$, defined as a recursive transformation of P using cuts. Since this substitution is defined in the manner of left propagation, essentially stating

$$[\![MN]\!]^{\lambda}_{\alpha} \triangleq [\![M]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\not\leftarrow\widehat{x}\,([\![N]\!]^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) \quad \gamma,\beta,x,y \textit{ fresh,}$$

Urban's interpretation actually ignores certain reduction paths that are accessible from our interpretation by right-activating the cut, and using Urban's not all our results shown below would be achievable; in fact, it is not possible to show that CBN reduction is modelled using Urban's interpretation.

Also, because of this substitution, reasoning over this interpretation in our proofs below would be much more complicated, and those proofs would lose their elegance. For example, to prove that type assignment is preserved for this interpretation, a substitution lemma needs to be shown, giving a much more involved proof than the one we achieve in Theorem 4.8. Moreover, it is possible to show that

$$[\![M]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\dagger\widehat{x}\,([\![N]\!]^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) \twoheadrightarrow [\![M]\!]^{\lambda}_{\gamma}\ [\ \gamma := (x)([\![N]\!]^{\lambda}_{\beta}\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)\ ].$$

In all, we feel our choice is justified.

It is worthwhile to notice that the interpretation function $[\![\cdot]\!]^{\lambda}_{\alpha}$ does not generate a confluent sub-calculus. We illustrate this by the following:

Example 4.4 First notice that

$$\begin{array}{rcll}
[\![xx]\!]^{\lambda}_{\alpha} & \triangleq & [\![x]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\dagger\widehat{z}\,([\![x]\!]^{\lambda}_{\beta}\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \triangleq \\
 & & \langle x{\cdot}\gamma\rangle\,\widehat{\gamma}\dagger\widehat{z}\,(\langle x{\cdot}\beta\rangle\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \to\ (\textit{imp}) \\
 & & \langle x{\cdot}\beta\rangle\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle &
\end{array}$$

Moreover notice that the above $(\textit{imp})$ reduction is the only possible one, making the reduction deterministic. So we can write $[\![xx]\!]^{\lambda}_{\alpha} \to \langle x{\cdot}\beta\rangle\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle$. Now

$$\begin{array}{lll}
[\![(\lambda x.xx)(yy)]\!]^{\lambda}_{\alpha} & \triangleq & \\
[\![\lambda x.xx]\!]^{\lambda}_{\beta}\,\widehat{\beta}\dagger\widehat{v}\,([\![yy]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\,[v]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle) & \triangleq & \\
(\widehat{x}\,[\![xx]\!]^{\lambda}_{\delta}\widehat{\delta}\cdot\beta)\,\widehat{\beta}\dagger\widehat{v}\,([\![yy]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\,[v]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle) & \to & (\textit{exp-imp}) \\
[\![yy]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\dagger\widehat{x}\,([\![xx]\!]^{\lambda}_{\delta}\,\widehat{\delta}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle) & \to & (\nwarrow\!\textit{ren}) \\
[\![yy]\!]^{\lambda}_{\gamma}\,\widehat{\gamma}\dagger\widehat{x}\,[\![xx]\!]^{\lambda}_{\alpha} & \overset{2}{\to} & (\textit{imp}) \\
(\langle y{\cdot}\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z{\cdot}\gamma\rangle)\,\widehat{\gamma}\dagger\widehat{x}\,(\langle x{\cdot}\tau\rangle\,\widehat{\tau}\,[x]\,\widehat{u}\,\langle u{\cdot}\alpha\rangle) &
\end{array}$$

This net now has one cut only, that can be activated in two ways (notice that neither γ nor x is introduced here). This results in either:

$$\begin{array}{lll}
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha) & \to & (a\not\prec)\\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\not\prec\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha) & \stackrel{\Delta}{=} & \\
(\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\gamma\rangle)\,\widehat{\gamma}\not\prec\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha) & \to & (imp\not\prec)\\
(\langle y\cdot\sigma\rangle\,\widehat{\gamma}\not\prec\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha))\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\not\prec\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha)) & \to & (cap\not\prec),(d\not\prec),(\nwarrow a)\\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\nwarrow\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha)) & \stackrel{\Delta}{=} & \\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\nwarrow\widehat{x}\,(\langle x\cdot\tau\rangle\,\widehat{\tau}\,[x]\,\widehat{u}\,\langle u\cdot\alpha\rangle)) & \to & (\nwarrow imp\text{-}out)\\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\dagger\widehat{v}\,((\langle z\cdot\gamma\rangle\,\widehat{\gamma}\nwarrow\widehat{x}\,\langle x\cdot\tau\rangle)\,\widehat{\tau}\,[v]\,\widehat{u}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\nwarrow\widehat{x}\,\langle u\cdot\alpha\rangle))) & & \\
\qquad & \to & (\nwarrow d),(\nwarrow cap),(cap)\\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\dagger\widehat{v}\,(\langle z\cdot\tau\rangle\,\widehat{\tau}\,[v]\,\widehat{u}\,\langle u\cdot\alpha\rangle)) & \to & (imp)\\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\tau\rangle\,\widehat{\tau}\,[z]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & &
\end{array}$$

or

$$\begin{array}{lll}
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,[\![xx]\!]^{\lambda}(\alpha) & \to & (\nwarrow a),\stackrel{\Delta}{=}\\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\nwarrow\widehat{x}\,(\langle x\cdot\tau\rangle\,\widehat{\tau}\,[x]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & \to & (\nwarrow imp\text{-}out)\\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\dagger\widehat{w}\,((\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\nwarrow\widehat{x}\,\langle x\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,([\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\nwarrow\widehat{x}\,\langle u\cdot\alpha\rangle)) & \to & (\nwarrow d),(a\not\prec),(\nwarrow cap)\\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\dagger\widehat{w}\,((\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\not\prec\widehat{x}\,\langle x\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & \stackrel{\Delta}{=} & \\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\dagger\widehat{w}\,(((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\gamma\rangle)\,\widehat{\gamma}\not\prec\widehat{x}\,\langle x\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & \to & (imp\not\prec)\\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\dagger\widehat{w}\,(((\langle y\cdot\sigma\rangle\,\widehat{\gamma}\not\prec\widehat{x}\,\langle x\cdot\tau\rangle)\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\not\prec\widehat{x}\,\langle x\cdot\tau\rangle))\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & & \\
\qquad & \to & (cap\not\prec),(d\not\prec),(cap),(a\not\prec)\\
[\![yy]\!]^{\lambda}(\gamma)\,\widehat{\gamma}\not\prec\widehat{w}\,((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & \stackrel{\Delta}{=} & \\
(\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\gamma\rangle)\,\widehat{\gamma}\not\prec\widehat{w}\,((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & \to & (imp\not\prec)\\
(\langle y\cdot\sigma\rangle\,\widehat{\gamma}\not\prec\widehat{w}\,((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle))\,\widehat{\sigma}\;[y] & & \\
\qquad\qquad\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\not\prec\widehat{w}\,((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle)) & \to & (cap\not\prec),(d\not\prec),\\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z\cdot\gamma\rangle\,\widehat{\gamma}\dagger\widehat{w}\,((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle)) & \to & (imp)\\
\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,((\langle y\cdot\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z\cdot\tau\rangle)\,\widehat{\tau}\,[w]\,\widehat{u}\,\langle u\cdot\alpha\rangle) & &
\end{array}$$

Notice that both reductions return normal forms, and that these are different.

We will show that the CBN-reduction on the λ-calculus is respected by the interpretation $[\![\;]\!]^{\lambda}(\;)$. First we show a substitution result.

Lemma 4.5 *i)* $[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,[\![M]\!]^{\lambda}(\alpha)\to[\![M[N/x]]\!]^{\lambda}(\alpha)$.
ii) $[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\dagger\widehat{x}\,[\![M]\!]^{\lambda}(\alpha)\to[\![M[N/x]]\!]^{\lambda}(\alpha)$.

Proof: *i*) By induction on the structure of lambda terms.

$(M=x)$: $[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,[\![x]\!]^{\lambda}(\alpha)\;\stackrel{\Delta}{=}\;[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,\langle x\cdot\alpha\rangle\;\to(ren\not\prec)\;[\![N]\!]^{\lambda}(\delta)[\alpha/\delta]\;\stackrel{\Delta}{=}\;[\![N]\!]^{\lambda}(\alpha)\;\stackrel{\Delta}{=}\;[\![x[N/x]]\!]^{\lambda}(\alpha)$

$(M=y\neq x)$: $[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,[\![y]\!]^{\lambda}(\alpha)\;\stackrel{\Delta}{=}\;[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,\langle y\cdot\alpha\rangle\;\to(\nwarrow cap)\;\langle y\cdot\alpha\rangle\;\stackrel{\Delta}{=}\;[\![y]\!]^{\lambda}(\alpha)\;\stackrel{\Delta}{=}\;[\![y[N/x]]\!]^{\lambda}(\alpha)$

$(M=\lambda y.M')$:

$$\begin{array}{lll}
[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,[\![\lambda y.M']\!]^{\lambda}(\alpha) & \stackrel{\Delta}{=} & \\
[\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,(\widehat{y}\,[\![M']\!]^{\lambda}(\beta)\,\widehat{\beta}\cdot\alpha) & \to & (\nwarrow exp)\\
\widehat{y}\,([\![N]\!]^{\lambda}(\delta)\,\widehat{\delta}\nwarrow\widehat{x}\,[\![M']\!]^{\lambda}(\beta))\,\widehat{\beta}\cdot\alpha & \to & (IH)\\
\widehat{y}\,[\![M]\!]^{\lambda}(')[N/x]\beta\widehat{\beta}\cdot\alpha & \stackrel{\Delta}{=} & [\![(\lambda y.M')[N/x]]\!]^{\lambda}(\alpha)
\end{array}$$

$$
\begin{array}{lll}
(M=PQ)\colon\ [\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![PQ]\!]^{\lambda}_{\shriek}(\alpha) & \stackrel{\Delta}{=} & \\
\quad [\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,([\![P]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\dagger\widehat{z}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)) & \to & (\leftthreetimes cut) \\
\quad ([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\gamma))\,\widehat{\gamma}\dagger\widehat{z}\,([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)) & \to & (\leftthreetimes imp\text{-}in) \\
\quad ([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\gamma))\,\widehat{\gamma}\dagger\widehat{z}\,(([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![Q]\!]^{\lambda}_{\shriek}(\beta))\,\widehat{\beta}\,[z]\,\widehat{y}\,([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,\langle y{\cdot}\alpha\rangle)) & & \\
\qquad\qquad & \to & (\leftthreetimes cap) \\
\quad ([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\gamma))\,\widehat{\gamma}\dagger\widehat{z}\,(([\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![Q]\!]^{\lambda}_{\shriek}(\beta))\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \to & (IH) \\
\quad [\![P[N/x]]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\dagger\widehat{z}\,([\![Q[N/x]]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \stackrel{\Delta}{=} & \\
\quad [\![P[N/x]Q[N/x]]\!]^{\lambda}_{\shriek}(\alpha) & \stackrel{\Delta}{=} & [\![(PQ)[N/x]]\!]^{\lambda}_{\shriek}(\alpha)
\end{array}
$$

ii) We have two cases:

($[\![M]\!]^{\lambda}_{\shriek}(\alpha)$ *introduces* x): By Definition 4.3, this is only possible if $M=x$ (and then $[\![M]\!]^{\lambda}_{\shriek}(\alpha)=\langle x{\cdot}\alpha\rangle$). Then $[\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\dagger\widehat{x}\,\langle x{\cdot}\alpha\rangle\to[\![N]\!]^{\lambda}_{\shriek}(\alpha)$ by $(\leftthreetimes ren)$; notice that $[\![N]\!]^{\lambda}_{\shriek}(\alpha)=[\![x[N/x]]\!]^{\lambda}_{\shriek}(\alpha)$.

($[\![M]\!]^{\lambda}_{\shriek}(\alpha)$ *does not introduce* x): Then $[\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\dagger\widehat{x}\,[\![M]\!]^{\lambda}_{\shriek}(\alpha)\to(\leftthreetimes a)\ [\![N]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\leftthreetimes\widehat{x}\,[\![M]\!]^{\lambda}_{\shriek}(\alpha)$ and the result follows from the first part. □

Theorem 4.6 (Simulation of cbn for the λ-calculus) *If* $M\to_{\mathrm{N}}N$ *then* $[\![M]\!]^{\lambda}_{\shriek}(\gamma)\to_{\mathrm{N}}[\![N]\!]^{\lambda}_{\shriek}(\gamma)$.

Proof: By induction on the number of steps, of which we show only the base case. This in turn is proven by induction on the structure of terms, of which we again show the base case, namely $M=(\lambda x.P)Q$. In that case, $(\lambda x.P)Q\to P[Q/x]$, hence we have to prove:

$$
\begin{array}{lll}
[\![(\lambda x.P)Q]\!]^{\lambda}_{\shriek}(\alpha)\ \stackrel{\Delta}{=}\ [\![\lambda x.P]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\dagger\widehat{y}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[y]\,\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \stackrel{\Delta}{=} & \\
\quad (\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\delta)\widehat{\delta}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{y}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[y]\,\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \to & (exp\text{-}imp) \\
\quad [\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,([\![P]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\dagger\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \to & (\leftthreetimes ren) \\
\quad [\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\alpha) & \to & (4.5)\quad [\![P[Q/x]]\!]^{\lambda}_{\shriek}(\alpha)
\end{array}
$$

Notice that if in the above reduction we would have used the other version of *exp-imp* we would have got the same result namely

$$
\begin{array}{lll}
(\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\delta)\widehat{\delta}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{y}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[y]\,\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \to & (exp\text{-}imp) \\
([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\delta))\,\widehat{\delta}\dagger\widehat{z}\,\langle z{\cdot}\alpha\rangle & \to & (\leftthreetimes ren)\quad [\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![P]\!]^{\lambda}_{\shriek}(\alpha)
\end{array}
$$

Notice that, in this reduction, all reduction steps are allowed in $\to_{\mathrm{N}}$.

Notice also that $\langle z{\cdot}\alpha\rangle$ introduces z; if $[\![P]\!]^{\lambda}_{\shriek}(\delta)$ introduces δ, then either rule (cap) or (exp) can be applied. When $[\![P]\!]^{\lambda}_{\shriek}(\delta)$ does not introduce δ, the cut needs to be activated, leading to

$$[\![P]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\dagger\widehat{z}\,\langle z{\cdot}\alpha\rangle\ \to\ [\![P]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\not\leftthreetimes\widehat{z}\,\langle z{\cdot}\alpha\rangle$$

This is allowed by rule $(a\not\leftthreetimes)$ in $\to_{\mathrm{N}}$, since both side-conditions are satisfied. □

When encoding the cbv-λ-calculus, we also use the $[\![\cdot]\!]^{\lambda}_{\shriek}(\alpha)$ interpretation. Notice that a term like $(\lambda x.M)(PQ)$ is not a redex in the cbv-λ-calculus. As above, we get

$$
\begin{array}{lll}
[\![(\lambda x.M)(PQ)]\!]^{\lambda}_{\shriek}(\alpha) & \stackrel{\Delta}{=} & [\![\lambda x.M]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\dagger\widehat{b}\,([\![PQ]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,[v]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle) \\
& \stackrel{\Delta}{=} & (\widehat{x}\,[\![M]\!]^{\lambda}_{\shriek}(\delta)\widehat{\delta}\cdot\beta)\,\widehat{\beta}\dagger\widehat{b}\,([\![PQ]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,[v]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle) \\
& \to & [\![PQ]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,([\![M]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\dagger\widehat{w}\,\langle w{\cdot}\alpha\rangle) \\
& \to & [\![PQ]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,[\![M]\!]^{\lambda}_{\shriek}(\alpha) \\
& \stackrel{\Delta}{=} & ([\![P]\!]^{\lambda}_{\shriek}(\sigma)\,\widehat{\sigma}\dagger\widehat{t}\,([\![Q]\!]^{\lambda}_{\shriek}(\tau)\,\widehat{\tau}\,[t]\,\widehat{u}\,\langle u{\cdot}\gamma\rangle))\,\widehat{\gamma}\dagger\widehat{x}\,[\![M]\!]^{\lambda}_{\shriek}(\alpha)
\end{array}
$$

In particular, γ is not introduced in the outer-most cut, so $(a\not\leftthreetimes)$ can be applied. What the cbv reduction should guarantee, however, is that $(\leftthreetimes a)$ *cannot* be applied; then the propagation of $[\![P]\!]^{\lambda}_{\shriek}(\sigma)\,\widehat{\sigma}\dagger\widehat{t}\,([\![Q]\!]^{\lambda}_{\shriek}(\tau)\,\widehat{\tau}\,[t]\,\widehat{u}\,\langle u{\cdot}\gamma\rangle)$ into $[\![M]\!]^{\lambda}_{\shriek}(\alpha)$ is blocked (which would produce $[\![M[(PQ)/x]]\!]^{\lambda}_{\shriek}(\alpha)$, by Lemma 4.5). Notice that we can only apply rule $(\leftthreetimes a)$ if both $[\![M]\!]^{\lambda}_{\shriek}(\alpha)$ does not introduce x and $[\![P]\!]^{\lambda}_{\shriek}(\sigma)\,\widehat{\sigma}\dagger\widehat{t}\,([\![Q]\!]^{\lambda}_{\shriek}(\tau)\,\widehat{\tau}\,[t]\,\widehat{u}\,\langle u{\cdot}\gamma\rangle)$ introduces γ. This is not the case,

since the second test fails.

On the other hand, if N is a λ-value (i.e. either a variable or an abstraction) then $[\![N]\!]^\lambda_{\shortparallel}(\gamma)$ introduces γ (in fact, N is a value if and only if $[\![N]\!]^\lambda_{\shortparallel}(\gamma)$ introduces γ). Then $[\![N]\!]^\lambda_{\shortparallel}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![M]\!]^\lambda_{\shortparallel}(\alpha)$ cannot be reduced by rule $(a\not\leftarrow)$, but by either rule $(\nwarrow a)$ or a logical rule. As in the proof of Theorem 4.6, this enables the reduction

$$[\![N]\!]^\lambda_{\shortparallel}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![M]\!]^\lambda_{\shortparallel}(\alpha) \;\to_{\mathrm{v}}\; [\![M[N/x]]\!]^\lambda_{\shortparallel}(\alpha).$$

So CBV-reduction for the λ-calculus is respected by the interpretation function, using $\to_{\mathrm{v}}$.

Theorem 4.7 (Simulation of CBV for the λ-calculus) *If $M \to_{\mathrm{v}} N$ then $[\![M]\!]^\lambda_{\shortparallel}(\gamma) \to_{\mathrm{v}} [\![N]\!]^\lambda_{\shortparallel}(\gamma)$.*

Proof: As in the proof of Theorem 4.6, we only show the case $M = (\lambda x.P)Q$. Notice that then also $[\![(\lambda x.P)Q]\!]^\lambda_{\shortparallel}(\delta) \to_{\mathrm{v}} [\![Q]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\delta)$ (so this is true for both strategies). In this reduction, all reduction steps are allowed in $\to_{\mathrm{v}}$; as above, the only activation of a cut that might be required is in the application of $(\nwarrow ren)$, when perhaps

$$[\![P]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,\langle z{\cdot}\alpha\rangle \;\to\; [\![P]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\not\leftarrow\widehat{z}\,\langle z{\cdot}\alpha\rangle$$

when δ is not introduced in P. This is allowed by rule $(a\not\leftarrow)$ in $\to_{\mathrm{v}}$.

Now we show that $[\![Q]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\delta) \to_{\mathrm{v}} [\![P[Q/x]]\!]^\alpha_{\shortparallel}$ if and only if Q is a value:

(*if*): Let Q be a value, so $[\![Q]\!]^\lambda_{\shortparallel}(\beta)$ introduces β.

As in the proof of Theorem 4.6, we now have two cases:

($[\![P]\!]^\lambda_{\shortparallel}(\alpha)$ *introduces* x): By Definition 4.3, this is only possible if $P = x$ (and then $[\![P]\!]^\lambda_{\shortparallel}(\alpha) = \langle x{\cdot}\alpha\rangle$). Then $[\![Q]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,\dagger\,\widehat{x}\,\langle x{\cdot}\alpha\rangle \to_{\mathrm{v}} [\![Q]\!]^\lambda_{\shortparallel}(\alpha)$ by $(\nwarrow ren)$, and $[\![Q]\!]^\lambda_{\shortparallel}(\alpha) = [\![x[Q/x]]\!]^\lambda_{\shortparallel}(\alpha)$.

($[\![P]\!]^\lambda_{\shortparallel}(\alpha)$ *does not introduce* x): Since the side-conditions for rule $(\nwarrow a)$ are satisfied, we get $[\![Q]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\delta) \to_{\mathrm{v}} (\nwarrow a)\,[\![Q]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\nwarrow\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\delta)$, as noticed above.

(*only if*): Let $Q = RS$. Then

$$[\![RS]\!]^\lambda_{\shortparallel}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha) \;\to\; ([\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle))\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha)$$

Now there are two cuts that can be activated, and we get either:

$$\begin{array}{ll}
([\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle))\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha) & \to \\
([\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle))\,\widehat{\gamma}\not\leftarrow\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha) & \to \\
([\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\gamma}\not\leftarrow\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha))\,\widehat{\delta}\,\dagger\,\widehat{z}\,(([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle)\,\widehat{\gamma}\not\leftarrow\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha)) & \to\; \ldots
\end{array}$$

or:

$$\begin{array}{l}
([\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle))\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha) \;\to \\
T\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![P]\!]^\lambda_{\shortparallel}(\alpha)
\end{array}$$

where T is $[\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\not\leftarrow\widehat{z}\,([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle)$ if $[\![R]\!]^\lambda_{\shortparallel}(\delta)$ does not introduce δ, and the result of applying the appropriate logical rule to $[\![R]\!]^\lambda_{\shortparallel}(\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,([\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle)$ if it does; notice that z is introduced in $[\![S]\!]^\lambda_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{v}\,\langle v{\cdot}\gamma\rangle$.

In both cases, the reduction will continue inside the left-hand side of the outermost (non-active) cut. The CBV-reduction will only allow right-activation of the outer-most cut when the reduction of R returns a net that introduces γ, i.e. is a capsule or an export. In any case, the reduction will not lead to $[\![P[(RS)/x]]\!]^\lambda_{\shortparallel}(\alpha)$. □

Notice that we need to show *both* implications in the proof above. Proving just "If Q is a value, then ..." does not guarantee that the contraction will not take place if Q is not a value. Also, we need to show that the contraction has the desired result; the reduction we have in Theorem 4.6 is not necessarily a reduction in CBV, and Lemma 4.5 shows only a result for *right* activated cuts.

Using the last two results, the significance of Example 4.4 becomes clearer. Remember that

$$\begin{array}{lll}[\![(\lambda x.xx)(yy)]\!]^{\lambda}_{\shriek}(\alpha) & \to_{\mathrm{v}} & \langle y{\cdot}\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,(\langle z{\cdot}\tau\rangle\,\widehat{\tau}\,[z]\,\widehat{u}\,\langle u{\cdot}\alpha\rangle)\\ [\![(\lambda x.xx)(yy)]\!]^{\lambda}_{\shriek}(\alpha) & \to_{\mathrm{N}} & \langle y{\cdot}\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,((\langle y{\cdot}\sigma\rangle\,\widehat{\sigma}\,[y]\,\widehat{z}\,\langle z{\cdot}\tau\rangle)\,\widehat{\tau}\,[z]\,\widehat{u}\,\langle u{\cdot}\alpha\rangle)\end{array}$$

In the λ-calculus, $(\lambda x.xx)(yy)$ has different normal forms with respect to CBV and CBN λ-reduction (respectively $(\lambda x.xx)(yy)$ and $yy(yy)$), which are *both* interpreted in $\mathcal{X}$. The net $[\![(\lambda x.xx)(yy)]\!]^{\lambda}_{\shriek}(\alpha)$ is *not* a normal form in $\mathcal{X}$ for $\to_{\mathrm{v}}$; it contains cuts. But, true to its nature, the CBV-reduction will not return $[\![yy(yy)]\!]^{\lambda}_{\shriek}(\alpha)$, but, instead, returns the net $[\![yy]\!]^{\lambda}_{\shriek}$ with the duplication $[\![zz]\!]^{\lambda}_{\shriek}(\alpha)$ 'waiting to be applied' in the continuation.

We can now show that typeability is preserved by $[\![\cdot]\!]^{\lambda}_{\shriek}(\alpha)$:

Theorem 4.8 *If $\Gamma \vdash_{\lambda} M{:}A$, then $[\![M]\!]^{\lambda}_{\shriek}(\alpha) :\cdot \Gamma \vdash_{\mathcal{X}} \alpha{:}A$.*

Proof: By induction on the structure of derivations in $\vdash_{\lambda}$; notice that we use weakening.

(ax): Then $M = x$, and $\Gamma = \Gamma', x{:}A$. Notice that

$$\frac{}{[\![x]\!]^{\lambda}_{\shriek}(\alpha) :\cdot \Gamma', x{:}A \vdash \alpha{:}A}\,(cap)$$

$(\to I)$: Then $M = \lambda x.N$, $A = C{\to}D$, and $\Gamma, x{:}C \vdash_{\lambda} N{:}D$. Then, by induction, a derivation $\mathcal{D} :: [\![N]\!]^{\lambda}_{\shriek}(\beta) :\cdot \Gamma, x{:}C \vdash_{\mathcal{X}} \beta{:}D$ exists, and we can construct:

$$\frac{\boxed{\mathcal{D}}\\ [\![N]\!]^{\lambda}_{\shriek}(\beta) :\cdot \Gamma, x{:}C \vdash \beta{:}D}{\widehat{x}\,[\![N]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\cdot\alpha :\cdot \Gamma \vdash \alpha{:}C{\to}D}\,(exp)$$

Notice that $\widehat{x}\,[\![N]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\cdot\alpha = [\![\lambda x.N]\!]^{\lambda}_{\shriek}(\alpha)$.

$(\to E)$: Then $M = PQ$, and there exists B such that $\Gamma \vdash_{\lambda} P{:}B{\to}A$ and $\Gamma \vdash_{\lambda} Q{:}B$. By induction, there exist derivations $\mathcal{D}_1 :: [\![P]\!]^{\lambda}_{\shriek}(\gamma) :\cdot \Gamma \vdash_{\mathcal{X}} \gamma{:}B{\to}A$ and $\mathcal{D}_2 :: [\![Q]\!]^{\lambda}_{\shriek}(\beta) :\cdot \Gamma \vdash_{\mathcal{X}} \beta{:}B$, and we can construct:

$$\dfrac{\dfrac{\boxed{\mathcal{D}_1}\\ [\![P]\!]^{\lambda}_{\shriek}(\gamma) :\cdot \Gamma \vdash \gamma{:}B{\to}A}{[\![P]\!]^{\lambda}_{\shriek}(\gamma) :\cdot \Gamma \vdash \alpha{:}A, \gamma{:}B{\to}A} \qquad \dfrac{\dfrac{\boxed{\mathcal{D}_2}\\ [\![Q]\!]^{\lambda}_{\shriek}(\beta) :\cdot \Gamma \vdash \beta{:}B}{[\![Q]\!]^{\lambda}_{\shriek}(\beta) :\cdot \Gamma \vdash \beta{:}B, \alpha{:}A} \qquad \dfrac{}{\langle y{\cdot}\alpha\rangle :\cdot \Gamma, y{:}A \vdash \alpha{:}A}}{[\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle :\cdot \Gamma, x{:}B{\to}A \vdash \alpha{:}A}}{[\![P]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) :\cdot \Gamma \vdash \alpha{:}A}$$

Notice that $[\![PQ]\!]^{\lambda}_{\shriek}(\alpha) = [\![P]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,([\![Q]\!]^{\lambda}_{\shriek}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)$, and that this derivation corresponds (of course) to the simulation of the Modus Ponens rule as discussed above. □

As already suggested in Section 4, this theorem is in fact a reformulation of Gentzen's correctness result on the embedding of natural deduction in the sequent calculus.

To strengthen the fact that we consider more than just those nets that represent proofs, we will show an example of a non-terminating reduction sequence.

Example 4.9 (REDUCING $[\![\Delta\Delta]\!]^{\lambda}_{\shriek}(\beta)$) Remember that $[\![xx]\!]^{\lambda}_{\shriek}(\beta) \to \langle x{\cdot}\delta\rangle\,\widehat{\delta}\,[x]\,\widehat{y}\,\langle y{\cdot}\beta\rangle$. Now $[\![\Delta\Delta]\!]^{\lambda}_{\shriek}(\beta)$ reduces as follows:

$$\begin{array}{lll}[\![\Delta\Delta]\!]^{\lambda}_{\shriek}(\beta) & \triangleq & [\![\lambda x.xx]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{z}\,([\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,[z]\,\widehat{y}\,\langle y{\cdot}\beta\rangle)\\ & \triangleq & (\widehat{x}\,[\![xx]\!]^{\lambda}_{\shriek}(\alpha)\,\widehat{\alpha}\cdot\delta)\,\widehat{\delta}\,\dagger\,\widehat{z}\,([\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,[z]\,\widehat{y}\,\langle y{\cdot}\beta\rangle)\\ & \to (\textit{exp-imp}) & [\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,([\![xx]\!]^{\lambda}_{\shriek}(\alpha)\,\widehat{\alpha}\,\dagger\,\widehat{y}\,\langle y{\cdot}\beta\rangle)\\ & \to (\textit{?ren}) & [\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,[\![xx]\!]^{\lambda}_{\shriek}(\beta) \;\triangleq\; [\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{x}\,(\langle x{\cdot}\delta\rangle\,\widehat{\delta}\,[x]\,\widehat{y}\,\langle y{\cdot}\beta\rangle)\\ & \to (\textit{?a}) \wedge (\textit{?imp-out}) & [\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{z}\,((\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,?\,\widehat{x}\,\langle x{\cdot}\delta\rangle)\,\widehat{\delta}\,[z]\,\widehat{y}\,([\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,?\,\widehat{x}\,\langle y{\cdot}\beta\rangle))\\ & \to (\textit{?cap}) & [\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{z}\,((\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,?\,\widehat{x}\,\langle x{\cdot}\delta\rangle)\,\widehat{\delta}\,[z]\,\widehat{y}\,\langle y{\cdot}\beta\rangle)\\ & \to (\textit{?d}) \wedge (\textit{?ren}) & [\![\Delta]\!]^{\lambda}_{\shriek}(\gamma)\,\widehat{\gamma}\,\dagger\,\widehat{z}\,([\![\Delta]\!]^{\lambda}_{\shriek}(\delta)\,\widehat{\delta}\,[z]\,\widehat{y}\,\langle y{\cdot}\beta\rangle) \quad \triangleq\; [\![\Delta\Delta]\!]^{\lambda}_{\shriek}(\beta)\end{array}$$

5 Interpreting $\lambda\mathbf{x}$

In this section we will interpret a calculus of explicit substitutions, $\lambda\mathbf{x}$, introduced by Bloo and Rose [14], where a β-reduction of the λ-calculus is split into several more atomic steps of computation. We will show that $\mathcal{X}$ has a fine level of atomicity as it simulates each reduction step of $\lambda\mathbf{x}$ by describing how the explicit substitutions interact with nets.

Bloo and Rose introduce the concept of substitution within the syntax of the calculus, making it *explicit*, by adding the operator $M\langle x=N\rangle$:

Definition 5.1 (Bloo &Rose 1995) The syntax of $\lambda\mathbf{x}$ is an extension of that of the λ-calculus:

$$M,N \;::=\; x \mid \lambda x.M \mid MN \mid M\langle x=N\rangle$$

Type assignment on $\lambda\mathbf{x}$ is defined as for the λ-calculus; the added syntactic construct is dealt with by the (cut)-rule:

$$(Ax):\ \frac{}{\Gamma,x{:}A\vdash x:A} \qquad (cut):\ \frac{\Gamma,x{:}A\vdash M:B \quad \Gamma\vdash N:A}{\Gamma\vdash M\langle x=N\rangle:B}$$

$$(\to I):\ \frac{\Gamma,x{:}A\vdash M:B}{\Gamma\vdash \lambda x.M:A\to B} \qquad (\to E):\ \frac{\Gamma\vdash M:A\to B \quad \Gamma\vdash N:A}{\Gamma\vdash MN:B}$$

Notice that the (cut)-rule does not enable us to prove sequents that were not provable beforehand in $\vdash_\lambda$. We can derive from $\Gamma,x{:}A\vdash_\lambda M:B$ and $\Gamma\vdash_\lambda N:A$ also $\Gamma\vdash_\lambda(\lambda x.M)N:B$; also, the Substitution Lemma shows that the (cut)-rule is admissible in $\vdash_\lambda$, replacing the explicit substitution by the implicit: if $\Gamma,x{:}A\vdash_\lambda M:B$ and $\Gamma\vdash_\lambda N:A$, then also $\Gamma\vdash_\lambda M[N/x]:B$.

Explicit substitution describes explicitly the process of executing a β-reduction, i.e. expresses syntactically the details of the computation as a succession of atomic, constant-time steps (like in a first-order rewriting system), where the β-reduction is split into several steps.

Definition 5.2 (Bloo & Rose 1995) The reduction relation is defined by the following rules

$$\begin{array}{llll}
(\mathsf{B}): & (\lambda x.M)P \to M\langle x:=P\rangle & (\mathsf{Var}): & x\langle x:=P\rangle \to P\\
(\mathsf{App}): & (MN)\langle x:=P\rangle \to M\langle x:=P\rangle N\langle x:=P\rangle & (\mathsf{VarK}): & y\langle x:=P\rangle \to y\\
(\mathsf{Abs}): & (\lambda y.M)\langle x:=P\rangle \to \lambda y.(M\langle x:=P\rangle) & (\mathsf{gc}): & M\langle x:=P\rangle \to M, \quad (x\notin fv(M))
\end{array}$$

The notion of reduction $\lambda\mathbf{x}$ is obtained by deleting rule (gc), and the notion of reduction $\lambda\mathbf{x}_{\mathsf{gc}}$ is obtained by deleting rule (VarK). The rule (gc) is called 'garbage collection', as it removes useless substitutions.

Our notion of cbv-$\lambda\mathbf{x}$ is naturally inspired by that of the λ-calculus.

Definition 5.3 (Call by value in $\lambda\mathbf{x}$) Just as in the λ-calculus, a term in $\lambda\mathbf{x}$ is a *value* if it is a variable or an abstraction. In a cbv-β-reduction, the argument must be a value, so that means that when it is simulated by cbv-$\lambda\mathbf{x}$, all the substitutions created are of the form $M\langle x:=N\rangle$ where N is a value, that is, either a variable or an abstraction.

Hence, we build the cbv-$\lambda\mathbf{x}$ by a syntactic restriction:

$$M \;::=\; x \mid \lambda x.M \mid M_1M_2 \mid M\langle x:=\lambda x.N\rangle \mid M\langle x:=y\rangle.$$

The cbv-β-reduction is the reduction generated by the rules of Definition 5.2, when rule (B) is applied only when P is value.

The Subject Reduction still holds as we can still see the computation in $\lambda\mathbf{x}$ as cut-elimination: now this process consists in discarding the situations where the elimination of a connective follows its introduction, by using the (cut)-rule and moving it towards the applications of the axiom-rule until it disappears.

Definition 5.4 (Interpretation of $\lambda\mathbf{x}$ in $\mathcal{X}$) We define $[\![\cdot]\!]^{\lambda x}_{\shortparallel}(\alpha)$ as the interpretation $[\![\cdot]\!]^{\lambda}_{\shortparallel}(\alpha)$, by adding a case for the explicit substitution:

$$\begin{array}{rcl}
[\![x]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq & \langle x{\cdot}\alpha\rangle \\
[\![\lambda x.M]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq & \widehat{x}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\cdot\alpha \\
[\![MN]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq & [\![M]\!]^{\lambda x}_{\shortparallel}(\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,([\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) \\
[\![M\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq & [\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\leftthreetimes\widehat{x}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\alpha)
\end{array}$$

Notice that the interpretation of the λ-calculus is just made out of the first three rules. Moreover, notice that the cut is activated in the last alternative; this might seem in contrast with Lemma 4.5, where we justify that the cut $[\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\alpha)$ reduces to $[\![M[N/x]]\!]^{\lambda x}_{\shortparallel}(\alpha)$, but, using the inactivated cut, we cannot not prove that

$$[\![(PQ)\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)\to[\![(P\langle x{=}N\rangle)(Q\langle x{=}N\rangle)]\!]^{\lambda x}_{\shortparallel}(\alpha)$$

as in Theorem 5.7 below, but would only be able to manage

$$[\![(PQ)\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)\downarrow_{\mathcal{X}}[\![(P\langle x{=}N\rangle)(Q\langle x{=}N\rangle)]\!]^{\lambda x}_{\shortparallel}(\alpha).$$

Now notice that, again, N is a value if and only if $[\![N]\!]^{\lambda x}_{\shortparallel}(\alpha)$ introduces α.

Theorem 5.5 *If $\Gamma\vdash_{\lambda x}M\colon A$, then $[\![M]\!]^{\lambda x}_{\shortparallel}(\alpha)\mathrel{:\cdot}\Gamma\vdash_{\mathcal{X}}\alpha{:}A$.*

The proof is a straightforward extension of that for Theorem 4.8.

We will now show that the reductions can be simulated, preserving the evaluation strategies.

Theorem 5.6 (Simulation of rule (B)) (CBN): $[\![(\lambda x.M)N]\!]^{\lambda x}_{\shortparallel}(\alpha)\to_{\mathrm{N}}[\![M\langle x:=N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)$
(CBV): $[\![(\lambda x.M)N]\!]^{\lambda x}_{\shortparallel}(\alpha)\to_{\mathrm{V}}[\![M\langle x:=N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)$ *iff N is a value.*

Proof:

$$\begin{array}{lll}
[\![(\lambda x.M)N]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq & \\
[\![\lambda x.M]\!]^{\lambda x}_{\shortparallel}(\gamma)\,\widehat{\gamma}\dagger\widehat{y}\,([\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\,[y]\,\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \triangleq & \\
(\widehat{x}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\delta)\,\widehat{\delta}\cdot\gamma)\,\widehat{\gamma}\dagger\widehat{y}\,([\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\,[y]\,\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \to & (\textit{exp-imp}) \\
[\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,([\![M]\!]^{\lambda x}_{\shortparallel}(\delta)\,\widehat{\delta}\dagger\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \to & (\leftthreetimes a) \\
[\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\leftthreetimes\widehat{x}\,([\![M]\!]^{\lambda x}_{\shortparallel}(\delta)\,\widehat{\delta}\dagger\widehat{z}\,\langle z{\cdot}\alpha\rangle) & \to & (\leftthreetimes \textit{ren}) \\
[\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\leftthreetimes\widehat{x}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq & [\![M\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)
\end{array}$$

Notice that this reduction sequence is valid in the CBN-evaluation, which proves the first point. As for the CBV-evaluation, the step $\leftthreetimes a$ is possible if and only if $[\![N]\!]^{\lambda x}_{\shortparallel}(\beta)$ introduces β, that is, if and only if N is a value. The proof concludes by Lemma 4.5. □

Notice that we could also show $[\![(\lambda x.M)N]\!]^{\lambda x}_{\shortparallel}(\alpha)\to_{\mathrm{N}}[\![N]\!]^{\lambda}_{\shortparallel}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![M]\!]^{\alpha}_{\shortparallel}$, but that using the definition $[\![M\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)\triangleq[\![N]\!]^{\lambda}_{\shortparallel}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![M]\!]^{\alpha}_{\shortparallel}$ would give problems for the next proof.

Theorem 5.7 (Simulation of the other rules) *Let $M\to N$ by any of the rules* (App), (Abs), (Var), (VarK), (gc), *then $[\![M]\!]^{\lambda x}_{\shortparallel}(\gamma)\to_{\mathrm{V}}[\![N]\!]^{\lambda x}_{\shortparallel}(\gamma)$ and $[\![M]\!]^{\lambda x}_{\shortparallel}(\gamma)\to_{\mathrm{N}}[\![N]\!]^{\lambda x}_{\shortparallel}(\gamma)$.*

Proof: We only show the interesting cases. In what follows we activate the cut to the right which corresponds to both CBV and CBN if N is a value and corresponds to CBN otherwise.

$$\begin{array}{lll}
([\![(\lambda y.M)\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)\to[\![\lambda y.M\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha)): & [\![(\lambda y.M)\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq \\
\quad [\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\leftthreetimes\widehat{x}\,[\![\lambda y.M]\!]^{\lambda x}_{\shortparallel}(\alpha) & \triangleq\ [\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\leftthreetimes\widehat{x}\,(\widehat{y}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\gamma)\,\widehat{\gamma}\cdot\alpha) \ \to\ (\leftthreetimes \textit{exp}) \\
\quad \widehat{y}\,([\![N]\!]^{\lambda x}_{\shortparallel}(\beta)\,\widehat{\beta}\leftthreetimes\widehat{x}\,[\![M]\!]^{\lambda x}_{\shortparallel}(\gamma))\,\widehat{\gamma}\cdot\alpha & \triangleq\ \widehat{y}\,[\![M\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\gamma)\,\widehat{\gamma}\cdot\alpha & \triangleq \\
\quad [\![\lambda y.M\langle x{=}N\rangle]\!]^{\lambda x}_{\shortparallel}(\alpha). &&
\end{array}$$

$$
\begin{array}{ll}
(\llbracket (PQ)\langle x{=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \to \llbracket (P\langle x{=}N\rangle)(Q\langle x{=}N\rangle)\rrparenthesis^{\lambda x}_{\shortparallel}(\alpha)): & \llbracket (PQ)\langle x{=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \stackrel{\Delta}{=} \\
\quad \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket PQ \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) & \stackrel{\Delta}{=} \\
\quad \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, (\llbracket P \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma)\, \widehat{\gamma} \dagger \widehat{y}\, (\llbracket Q \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta}\, [y]\, \widehat{z}\, \langle z{\cdot}\alpha\rangle)) & \to\ (\nwarrow cut) \\
\quad (\llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket P \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma))\, \widehat{\gamma} \dagger \widehat{y}\, (\llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, (\llbracket Q \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta}\, [y]\, \widehat{z}\, \langle z{\cdot}\alpha\rangle)) & \to\ (\nwarrow imp\text{-}in) \\
\quad (\llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket P \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma))\, \widehat{\gamma} \dagger \widehat{y}\, ((\llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket Q \rrparenthesis^{\lambda x}_{\shortparallel}(\beta))\, \widehat{\beta}\, [y]\, \widehat{z}\, (\llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \langle z{\cdot}\alpha\rangle)) & \stackrel{\Delta}{=} \\
\quad \llbracket P\langle x{=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma)\, \widehat{\gamma} \dagger \widehat{y}\, (\llbracket Q\langle x{=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta}\, [y]\, \widehat{z}\, (\llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \langle z{\cdot}\alpha\rangle)) & \to\ (\nwarrow cap) \\
\quad \llbracket P\langle x{=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma)\, \widehat{\gamma} \dagger \widehat{y}\, (\llbracket Q\langle x{=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta}\, [y]\, \widehat{z}\, \langle z{\cdot}\alpha\rangle) & \stackrel{\Delta}{=} \\
\quad \llbracket (P\langle x{=}N\rangle)(Q\langle x{=}N\rangle) \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) &
\end{array}
$$

$$
\begin{array}{lll}
(\llbracket x\langle x{:=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \to \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha)): & \llbracket x\langle x{:=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \stackrel{\Delta}{=} \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket x \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \stackrel{\Delta}{=} & \\
\quad \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \langle x{\cdot}\alpha\rangle \to (\nwarrow d)\ \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \dagger \widehat{x}\, \langle x{\cdot}\alpha\rangle \to (\nwarrow ren)\ \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) & &
\end{array}
$$

$$
\begin{array}{lll}
(\llbracket y\langle x{:=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \to \llbracket y \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha)): & \llbracket y\langle x{:=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \stackrel{\Delta}{=} \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket y \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \stackrel{\Delta}{=} & \\
\quad \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \langle y{\cdot}\alpha\rangle \to (\nwarrow cap)\ \langle y{\cdot}\alpha\rangle \stackrel{\Delta}{=} \llbracket y \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) & &
\end{array}
$$

$$
\begin{array}{lll}
(\llbracket M\langle x{:=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \to \llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha),\ \textit{if } x \notin fv(M)): & \llbracket M\langle x{:=}N\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \stackrel{\Delta}{=} & \\
\quad \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \nwarrow \widehat{x}\, \llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) \to (\nwarrow gc)\ \llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) & &
\end{array}
$$

We can now state that $\lambda\mathbf{x}$-reduction is preserved by interpretation of terms into $\mathcal{X}$.

Theorem 5.8 (Simulation of $\lambda\mathbf{x}$) *i) If $M \to_{\mathrm{v}} N$ then $\llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma) \to_{\mathrm{v}} \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma)$*
ii) If $M \to_{\mathrm{N}} N$ then $\llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma) \to_{\mathrm{N}} \llbracket N \rrparenthesis^{\lambda x}_{\shortparallel}(\gamma)$

We can add the following *composition rule* to the reduction system of $\lambda\mathbf{x}$:

$$M\langle x{=}P\rangle\langle y{=}Q\rangle \to M\langle y{=}Q\rangle\langle x{=}P\langle y{=}Q\rangle\rangle$$

This rule trivially breaks the Strong Normalisation property for typed terms [13] and cannot be simulated. But it can be useful for reasoning by equivalence. If we abandon Strong Normalisation, then we can merge the three kinds of cuts. It can be done for instance by setting the equivalence:

$$P\widehat{\alpha} \nearrow \widehat{x}Q \sim P\widehat{\alpha} \dagger \widehat{x}Q \sim P\widehat{\alpha} \nwarrow \widehat{x}Q.$$

In that case the composition rule can be simulated (in both strategies CBV and CBN) as follows:

$$
\begin{array}{ll}
\llbracket M\langle x{=}P\rangle\langle y{=}Q\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha) & \stackrel{\Delta}{=} \\
\llbracket Q \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \dagger \widehat{y}\, (\llbracket P \rrparenthesis^{\lambda x}_{\shortparallel}(\delta)\, \widehat{\delta} \dagger \widehat{x}\, \llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha)) & \\
(\llbracket Q \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \dagger \widehat{y}\, \llbracket P \rrparenthesis^{\lambda x}_{\shortparallel}(\delta))\, \widehat{\delta} \dagger \widehat{x}\, (\llbracket Q \rrparenthesis^{\lambda x}_{\shortparallel}(\beta)\, \widehat{\beta} \dagger \widehat{y}\, \llbracket M \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha)) & \stackrel{\Delta}{=} \\
\llbracket M\langle y{=}Q\rangle\langle x{=}P\langle y{=}Q\rangle\rangle \rrparenthesis^{\lambda x}_{\shortparallel}(\alpha). &
\end{array}
$$

There are interesting issues which deserves to be explored.

6 Interpreting $\lambda\mu$

Parigot's $\lambda\mu$-calculus [40] is a proof-term syntax for classical logic, but expressed in the setting of Natural Deduction. Let us extend the syntax of formulae with the constant $\bot$ (*false*). Natural deduction deals with classical logic by allowing the logical rule:

$$\frac{\Gamma, A \Rightarrow \bot \vdash_{\mathrm{LK}} \bot}{\Gamma \vdash_{\mathrm{LK}} A}$$

An assumption can now be discharged by either the introduction of '$\Rightarrow$' or by this rule (if the assumption has the form $A \Rightarrow \bot$). This leads to the splitting of the set of axioms into two parts: the first where the assumptions might be discharged only by the introduction

of '$\Rightarrow$', and the second where the assumptions might be discharged only by reasoning by contradiction. Since the latter part is of the form $(A_1 \Rightarrow \bot, \ldots, A_n \Rightarrow \bot)$ where the comma is to be thought as a *conjunction*, the sequent

$$\Gamma, A_1 \Rightarrow \bot, \ldots, A_n \Rightarrow \bot \vdash_{\text{LK}} B$$

is logically equivalent to the multi-conclusion sequent

$$\Gamma \vdash_{\text{LK}} B \mid A_1, \ldots, A_n,$$

since in classical logic $(A \Rightarrow \bot) \Rightarrow B$ is logically equivalent to $A \vee B$; 'both $\mid$' and ',' are to be thought as a *disjunction*. The reasoning by contradiction becomes:

$$\frac{\Gamma \vdash \bot \mid A, \Delta}{\Gamma \vdash A \mid \Delta}$$

which exhibits the neutrality of $\bot$ for the disjunction; notice that here we implicitly assume the truly classical $((A \Rightarrow \bot) \Rightarrow \bot) \Rightarrow A$.

However, when we assumed $A \Rightarrow \bot$, we may have liked to use it not only in a reasoning by contradiction, but also as an implication as such:

$$\frac{\dfrac{}{\Gamma, A \Rightarrow \bot \vdash_{\text{LK}} A \Rightarrow \bot} \qquad \boxed{\phantom{\Gamma, A \Rightarrow \bot \vdash_{\text{LK}} A}}}{\Gamma, A \Rightarrow \bot \vdash_{\text{LK}} \bot}$$

and hence in the multi-conclusion style sequent, we have to add the rule:

$$\frac{\Gamma \vdash A \mid A, \Delta}{\Gamma \vdash \bot \mid A, \Delta}$$

where again the neutrality of $\bot$ for the disjunction is exhibited.

In [40], Parigot created the $\lambda\mu$-calculus, the typing system of which is isomorphic to this multi-conclusion logical system; this is achieved by extending the syntax with two new constructs that act as witness to the two rules discussed above. It uses two disjoint sets of variables (Roman letters and Greek letters). The sequents typing terms are of the form $\Gamma \vdash_{\text{LK}} A \mid \Delta$, marking the conclusion A as *active*.

Definition 6.1 (Terms of $\lambda\mu$, version 1) The terms of $\lambda\mu$, version 1, are:

$$M, N \ ::= \ x \mid \lambda x.M \mid MN \mid [\alpha]M \mid \mu\alpha.M$$

Definition 6.2 (Typing rules for $\lambda\mu$) Type assignment for $\lambda\mu$ is defined by the following natural deduction system; there is a *main*, or *active*, conclusion, labelled by a term of this calculus, and the alternative conclusions are labelled by the set of Greek variables $\alpha, \beta, \ldots$.

$$\frac{}{\Gamma \vdash x : A \mid \Delta}\ (x{:}A \in \Gamma) \qquad \frac{\Gamma, x{:}A \vdash M : B \mid \Delta}{\Gamma \vdash \lambda x.M : A \to B \mid \Delta} \qquad \frac{\Gamma \vdash M : A \to B \mid \Delta \quad \Gamma \vdash N : A \mid \Delta}{\Gamma \vdash MN : B \mid \Delta}$$

$$\frac{\Gamma \vdash M : A \mid \alpha{:}A, \Delta}{\Gamma \vdash [\alpha]M : \bot \mid \alpha{:}A, \Delta} \qquad \frac{\Gamma \vdash M : \bot \mid \alpha{:}A, \Delta}{\Gamma \vdash \mu\alpha.M : A \mid \Delta}$$

We can think of $[\alpha]M$ as storing the type of M amongst the alternative conclusions by giving it the name α - the set of Greek variables is called the set of *name* variables. Also, $\mu\alpha.M$ binds α in M; the notion of α-conversion extends naturally to bound names.

Note that we have the *Weakening Property*: If $\Gamma \vdash_{\lambda\mu} M : A \mid \Delta$ and $\Gamma \subseteq \Gamma'$ and $\Delta \subseteq \Delta'$, then $\Gamma' \vdash_{\lambda\mu} M : A \mid \Delta'$.

It is interesting to note that even if $\bot$ is not included in the language (and hence $(A \to \bot) \to \bot$ is not even a type), we stay in classical logic by collapsing the last two rules into:

$$\frac{\Gamma \vdash M : B \mid \alpha{:}A, \beta{:}B, \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : A \mid \beta{:}B, \Delta} \qquad \frac{\Gamma \vdash M : B \mid \alpha{:}B, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : B \mid \Delta}$$

That is, we force the *naming* to be followed by a *μ-abstraction*. Thus:

Definition 6.3 (Terms of $\lambda\mu$, version 2) The terms of $\lambda\mu$, version 2, are defined by the following grammar:

$$M, N \ ::= \ x \mid \lambda x.M \mid MN \mid \mu\beta.[\alpha]M$$

The syntax of Definition 6.1 is the original definition of [40]; that in the above definition was introduced later in [41] and [30].

Since $\bot$ is not a type in the system we consider here for $\mathcal{X}$, this last variant of $\lambda\mu$ is the system for which we will study the relation with $\mathcal{X}$.

As an example illustrating the fact that this system still is more powerful than the system for the λ-calculus, here is a proof of Peirce's Law (due to Ong and Stewart [39]):

$$\dfrac{\dfrac{\dfrac{}{x{:}((A\to B)\to A)\vdash x : (A\to B)\to A \mid \alpha{:}A} \quad \dfrac{\dfrac{\dfrac{}{x{:}((A\to B)\to A), y{:}A \vdash y : A \mid \alpha{:}A, \beta{:}B}}{x{:}((A\to B)\to A), y{:}A \vdash \mu\beta.[\alpha]y : B \mid \alpha{:}A}}{x{:}((A\to B)\to A)\vdash \lambda y.\mu\beta.[\alpha]y : A\to B \mid \alpha{:}A}}{x{:}((A\to B)\to A)\vdash x(\lambda y.\mu\beta.[\alpha]y) : A \mid \alpha{:}A}}{x{:}((A\to B)\to A)\vdash \mu\alpha.[\alpha](x(\lambda y.\mu\beta.[\alpha]y)) : A \mid}}{\vdash \lambda x.\mu\alpha.[\alpha](x(\lambda y.\mu\beta.[\alpha]y)) : ((A\to B)\to A)\to A \mid}$$

Now as the system is still in a natural deduction style, we still have cut-situations that we might want to eliminate. Hence, we have the rules of computation in $\lambda\mu$:

Definition 6.4 ($\lambda\mu$ reduction) Reduction on $\lambda\mu$-terms is defined as the compatible closure of the rules:

$$\begin{array}{rrcl}
\textit{logical } (\beta): & (\lambda x.M)N & \to & M[N/x] \\
\textit{structural } (\mu): & (\mu\alpha.[\beta]M)N & \to & \mu\gamma.([\beta]M)[N{\cdot}\gamma/\alpha] \\
\textit{renaming}: & \mu\alpha.[\beta](\mu\gamma.[\delta]M) & \to & \mu\alpha.[\delta]M[\beta/\gamma] \\
\textit{erasing}: & \mu\alpha.[\alpha]M & \to & M \quad (\alpha \textit{ does not occur in } M)
\end{array}$$

where $M[N{\cdot}\gamma/\alpha]$ stands for the term obtained from M in which every (pseudo) sub-term of the form $[\alpha]M'$ is substituted by $[\gamma](M'N)$ (γ is a fresh variable).

Note that in this calculus, the computation of substitutions is *implicit*.

Definition 6.5 (Interpretation of $\lambda\mu$ in $\mathcal{X}$) We define $[\![\cdot]\!]^{\lambda\mu}(\alpha)$ as the interpretation $[\![\cdot]\!]^{\lambda}(\alpha)$, by adding an alternative for μ-terms:

$$\begin{array}{rcl}
[\![x]\!]^{\lambda\mu}(\alpha) & \triangleq & \langle x{\cdot}\alpha\rangle \\
[\![\lambda x.M]\!]^{\lambda\mu}(\alpha) & \triangleq & \widehat{x}\,[\![M]\!]^{\lambda\mu}(\beta)\,\widehat{\beta}\cdot\alpha \\
[\![MN]\!]^{\lambda\mu}(\alpha) & \triangleq & [\![M]\!]^{\lambda\mu}(\gamma)\,\widehat{\gamma}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) \\
[\![\mu\delta.[\gamma]M]\!]^{\lambda\mu}(\alpha) & \triangleq & [\![M]\!]^{\lambda\mu}(\gamma)\,\widehat{\delta}\dagger\widehat{x}\,\langle x{\cdot}\alpha\rangle
\end{array}$$

Similarly to the previous sections, we can prove (see Lemma 4.5):

$$\begin{array}{rcl}
[\![N]\!]^{\lambda\mu}(\beta)\,\widehat{\beta}\dagger\widehat{x}\,[\![M]\!]^{\lambda\mu}(\alpha) & \to & [\![M[N/x]]\!]^{\lambda\mu}(\alpha) \\
[\![M]\!]^{\lambda\mu}(\gamma)\,\widehat{\delta}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \to & [\![\mu\tau.([\gamma]M)[N{\cdot}\tau/\delta]]\!]^{\lambda\mu}(\alpha)
\end{array}$$

This result will prove convenient below.

Notice that the last alternative is justified, since we can show:

Lemma 6.6 The following rule is admissible:

$$[\![(\mu\delta.[\gamma]M)N]\!]^{\lambda\mu}_{\shortparallel}(\alpha) \;\to\; [\![M]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\delta}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)$$

$$\begin{array}{lll}
[\![(\mu\delta.[\gamma]M)N]\!]^{\lambda\mu}_{\shortparallel}(\alpha) & \stackrel{\Delta}{=} & \\
[\![\mu\delta.[\gamma]M]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\nu}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \stackrel{\Delta}{=} & \\
([\![M]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\delta}\dagger\widehat{z}\,\langle z{\cdot}\nu\rangle)\,\widehat{\nu}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \to & (\nwarrow ren)\\
[\![M]\!]^{\lambda\mu}_{\shortparallel}(\gamma)[\nu/\delta]\,\widehat{\nu}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & \to & (=_\alpha)\\
[\![M]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\delta}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle) & &
\end{array}$$

Notice the similarity between the net $[\![MN]\!]^{\lambda\mu}_{\shortparallel}(\alpha)$ and the result of running $[\![(\mu\delta.[\gamma]M)N]\!]^{\lambda\mu}_{\shortparallel}(\alpha)$; the difference lies only in a bound plug. This implies that, for the λ-calculus, we can only connect to the plug that corresponds to the name of the term itself, whereas for the $\lambda\mu$-calculus, we can also connect to plugs inside that occur inside, i.e., to named sub-terms.

We will now show that the above definition of the encoding of μ-substitution is correct; notice that, unlike for the λ-calculus, we can only show that the interpretation is preserved modulo equivalence, not modulo reduction; a similar restriction holds also for the interpretation of $\lambda\mu$ in $\overline{\lambda}\mu\tilde{\mu}$ achieved in [19][6].

Lemma 6.7 i) $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle)\downarrow_{\mathcal{X}}[\![M[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)$.
ii) $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle)\downarrow_{\mathcal{X}}[\![M[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\nu)$, *if* $\delta\neq\nu$.

Proof: By simultaneous induction on the structure of terms; we only show the interesting cases.

$$\begin{array}{lll}
i)(M=\lambda z.M'):\ [\![\lambda z.M']\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
(\widehat{z}\,[\![M']\!]^{\lambda\mu}_{\shortparallel}(\sigma)\widehat{\sigma}\cdot\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \to & \\
(\widehat{z}\,([\![M']\!]^{\lambda\mu}_{\shortparallel}(\sigma)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle))\widehat{\sigma}\cdot\tau)\,\widehat{\tau}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \downarrow_{\mathcal{X}} & (IH.2)\\
(\widehat{z}\,[\![M'[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\sigma)\widehat{\sigma}\cdot\tau)\,\widehat{\tau}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![\lambda z.M'[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\tau)\,\widehat{\tau}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![(\lambda z.M')[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\gamma) & &
\end{array}$$

$$\begin{array}{lll}
(M=M_1M_2):\ [\![M_1M_2]\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
([\![M_1]\!]^{\lambda\mu}_{\shortparallel}(\sigma)\,\widehat{\sigma}\dagger\widehat{z}\,([\![M_2]\!]^{\lambda\mu}_{\shortparallel}(\tau)\,\widehat{\tau}\,[z]\,\widehat{v}\,\langle v{\cdot}\delta\rangle))\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \downarrow_{\mathcal{X}} & (IH.2)\\
[\![M_1[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\sigma)\,\widehat{\sigma}\dagger\widehat{z}\,([\![M_2[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\tau)\,\widehat{\tau}\,[z]\,\widehat{v}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[v]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle)) & \leftarrow & \\
([\![M_1[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\sigma)\,\widehat{\sigma}\dagger\widehat{z}\,([\![M_2[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\tau)\,\widehat{\tau}\,[z]\,\widehat{v}\,\langle v{\cdot}\tau\rangle))\,\widehat{\tau}\nprec\widehat{x}([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \leftarrow & \\
[\![M_1[N{\cdot}\gamma/\delta]M_2[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\tau)\,\widehat{\tau}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![(M_1M_2)[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\gamma) & &
\end{array}$$

$$\begin{array}{lll}
(M=\mu\sigma.[\delta]M'):\ [\![\mu\sigma.[\delta]M']\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
([\![M']\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\sigma}\dagger\widehat{z}\,\langle z{\cdot}\delta\rangle)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \to & \\
([\![M']\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle))\,\widehat{\sigma}\dagger\widehat{z}\,(\langle z{\cdot}\delta\rangle\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle)) & \to & \\
([\![M']\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\nprec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle))\,\widehat{\sigma}\dagger\widehat{z}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \downarrow_{\mathcal{X}} & (IH.1)\\
[\![M'[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\sigma}\dagger\widehat{z}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \leftarrow & (=_\alpha)\\
([\![M'[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\sigma}\dagger\widehat{w}\,\langle w{\cdot}\nu\rangle)\,\widehat{\nu}\dagger\widehat{z}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![\mu\sigma.[\gamma]M'[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\nu}\dagger\widehat{z}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y{\cdot}\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![(\mu\sigma.[\gamma]M'[N{\cdot}\gamma/\delta]N)N]\!]^{\lambda\mu}_{\shortparallel}(\gamma) & = & \\
[\![(\mu\sigma.[\delta]M')[N{\cdot}\gamma/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\gamma) & &
\end{array}$$

[6] A corrected version of this paper is available from Herbelin's home page.

$$
\begin{array}{lll}
ii)(M=\mu\sigma.[\delta]M'):\ [\![\mu\sigma.[\delta]M'{}_{\shortparallel}^{\lambda\mu}(\nu)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle) & \triangleq & \\
\quad([\![M'{}_{\shortparallel}^{\lambda\mu}(\delta)\,\widehat{\sigma}\dagger\widehat{z}\,\langle z\cdot\nu\rangle)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle) & \to & \\
\quad([\![M'{}_{\shortparallel}^{\lambda\mu}(\delta)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle))\,\widehat{\sigma}\dagger\widehat{z}\,(\langle z\cdot\nu\rangle\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle))) & \to & \\
\quad([\![M'{}_{\shortparallel}^{\lambda\mu}(\delta)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle))\,\widehat{\sigma}\dagger\widehat{z}\,\langle z\cdot\nu\rangle & \downarrow_{\mathcal{X}} & (IH.1)\\
\quad[\![M'[N\cdot\gamma/\delta]N_{\shortparallel}^{\lambda\mu}(\gamma)\,\widehat{\sigma}\dagger\widehat{z}\,\langle z\cdot\nu\rangle & \leftarrow & (=_{\alpha})\\
\quad([\![M'[N\cdot\gamma/\delta]N_{\shortparallel}^{\lambda\mu}(\gamma)\,\widehat{\sigma}\dagger\widehat{w}\,\langle w\cdot\tau\rangle)\,\widehat{\tau}\dagger\widehat{z}\,\langle z\cdot\nu\rangle & \triangleq & \\
\quad[\![\mu\sigma.[\gamma]M'[N\cdot\gamma/\delta]N_{\shortparallel}^{\lambda\mu}(\tau)\,\widehat{\tau}\dagger\widehat{z}\,\langle z\cdot\nu\rangle & \to & \\
\quad[\![(\mu\sigma.[\gamma]M'[N\cdot\gamma/\delta]N)_{\shortparallel}^{\lambda\mu}(\nu) & = & \\
\quad[\![(\mu\sigma.[\delta]M')[N\cdot\gamma/\delta]_{\shortparallel}^{\lambda\mu}(\nu). & & \\
(M=\mu\sigma.[\tau]M'):\ [\![\mu\sigma.[\tau]M'{}_{\shortparallel}^{\lambda\mu}(\nu)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle) & \triangleq & \\
\quad([\![M'{}_{\shortparallel}^{\lambda\mu}(\tau)\,\widehat{\sigma}\dagger\widehat{z}\,\langle z\cdot\nu\rangle)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle) & \to & \\
\quad([\![M'{}_{\shortparallel}^{\lambda\mu}(\tau)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle)))\,\widehat{\sigma}\dagger\widehat{z}\,(\langle z\cdot\nu\rangle\,\widehat{\delta}\not\prec\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\gamma\rangle))) & \downarrow_{\mathcal{X}} & (IH.2)\\
\quad[\![M'[N\cdot\gamma/\delta]_{\shortparallel}^{\lambda\mu}(\tau)\,\widehat{\sigma}\dagger\widehat{z}\,\langle z\cdot\nu\rangle & \to & \\
\quad[\![(\mu\sigma.[\tau]M')[N\cdot\gamma/\delta]_{\shortparallel}^{\lambda\mu}(\nu). & &
\end{array}
$$

Remark that a number of the 'reverse' reduction steps in this proof could have been avoided by defining $[\![\mu\alpha.[\beta]M_{\shortparallel}^{\lambda\mu}(\gamma) \triangleq [\![M_{\shortparallel}^{\lambda\mu}(\beta)[\gamma/\alpha]$. But, as is evident from case $M=M_1M_2$ in the proof of part (i), this would not give a proof that the interpretation is preserved by reduction.

Notice that, for the $(renaming)$ and $(erasure)$ rules, we have:

$$
\begin{array}{lll}
[\![\mu\alpha.[\beta](\mu\gamma.[\delta]M)_{\shortparallel}^{\lambda\mu}(\sigma) & \triangleq & \\
[\![\mu\gamma.[\delta]M_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\alpha}\dagger\widehat{x}\,\langle x\cdot\sigma\rangle & = & \\
([\![M_{\shortparallel}^{\lambda\mu}(\delta)\,\widehat{\gamma}\dagger\widehat{x}\,\langle x\cdot\beta\rangle)\,\widehat{\alpha}\dagger\widehat{x}\,\langle x\cdot\sigma\rangle & \to & \\
[\![M[\beta/\gamma]_{\shortparallel}^{\lambda\mu}(\delta)\,\widehat{\alpha}\dagger\widehat{x}\,\langle x\cdot\sigma\rangle & = & \\
[\![\mu\alpha.[\delta]M[\beta/\gamma]_{\shortparallel}^{\lambda\mu}(\sigma) & &
\end{array}
\qquad\text{and}\qquad
\begin{array}{lll}
[\![\mu\alpha.[\alpha]M_{\shortparallel}^{\lambda\mu}(\sigma) & \triangleq & \\
[\![M_{\shortparallel}^{\lambda\mu}(\alpha)\,\widehat{\alpha}\dagger\widehat{x}\,\langle x\cdot\sigma\rangle & \to & \\
[\![M_{\shortparallel}^{\lambda\mu}(\sigma) & &
\end{array}
$$

We can now show that $\lambda\mu$'s reduction is preserved by our interpretation.

Theorem 6.8 (Simulation of cbn for $\lambda\mu$) *If $M\to_{\mathrm{N}}N$ then $[\![M_{\shortparallel}^{\lambda\mu}(\alpha)\downarrow_{\mathcal{X}}^{\mathrm{N}}[\![N_{\shortparallel}^{\lambda\mu}(\alpha)$.*

Proof: As before, the proof is by induction on the length of the reductions sequence, of which we show the base case; by the above, we need only focus on rules (β) and (μ). The proof for case (β) is as for Theorem 4.6, and for case (μ), we separate:

$((\mu\delta.[\delta]M)N\to\mu\alpha.([\delta]M)[N\cdot\alpha/\delta])$: By Lemma 6.6, we already know that

$$[\![(\mu\delta.[\delta]M)N_{\shortparallel}^{\lambda\mu}(\alpha)\ \to\ [\![M_{\shortparallel}^{\lambda\mu}(\delta)\,\widehat{\delta}\dagger\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\alpha\rangle)$$

(notice that x is introduced). Again, we separate two cases:

$(\delta$ *is introduced*$)$: Then either

$([\![M_{\shortparallel}^{\lambda\mu}(\delta)=\langle z\cdot\delta\rangle)$: Notice that

$$\langle z\cdot\delta\rangle\,\widehat{\delta}\dagger\widehat{x}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y\cdot\alpha\rangle)\ \to\ [\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[z]\,\widehat{y}\,\langle y\cdot\alpha\rangle.$$

and that

$$
\begin{array}{lll}
[\![\mu\alpha.[\alpha]zN_{\shortparallel}^{\lambda\mu}(\alpha) & \triangleq & [\![zN_{\shortparallel}^{\lambda\mu}(\alpha)\,\widehat{\alpha}\dagger\widehat{v}\,\langle v\cdot\alpha\rangle\\
& \triangleq & ([\![z_{\shortparallel}^{\lambda\mu}(\gamma)\,\widehat{\gamma}\dagger\widehat{w}\,([\![N_{\shortparallel}^{\lambda\mu}(\beta)\,\widehat{\beta}\,[w]\,\widehat{a}\,\langle a\cdot\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{v}\,\langle v\cdot\alpha\rangle\\
& \to & [\![N_{\shortparallel}^{\lambda\mu}(\beta)\beta[z]y\langle y\cdot\alpha\rangle
\end{array}
$$

$([\![M_{\shortparallel}^{\lambda\mu}(\delta)=\widehat{z}\,[\![M'{}_{\shortparallel}^{\lambda\mu}(\beta)\widehat{\beta}\cdot\delta)$: Then $M=\lambda z.M'$, and δ does not occur in M'. Notice that

$$\begin{array}{ll}
[\![\mu\alpha.([\delta](\lambda x.M'))[N{\cdot}\alpha/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\alpha) & = \\
[\![\mu\alpha.[\alpha](\lambda x.M')N]\!]^{\lambda\mu}_{\shortparallel}(\alpha) & \stackrel{\Delta}{=} \\
((\widehat{z}\,[\![M'{}]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\cdot\delta)\,\widehat{\delta}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{v}\,\langle v{\cdot}\alpha\rangle & \to \\
(\widehat{z}\,[\![M'{}]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\cdot\delta)\,\widehat{\delta}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle). &
\end{array}$$

(δ *is not introduced*): Then the latter reduces to $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)$.
By Lemma 6.7, $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\delta)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)\downarrow_{\mathcal{X}}[\![M[N{\cdot}\alpha/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\alpha)$. Notice that

$$[\![M[N{\cdot}\alpha/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\alpha)\ \leftarrow\ [\![M[N{\cdot}\alpha/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\alpha)\,\widehat{\alpha}\dagger\widehat{v}\,\langle v{\cdot}\alpha\rangle\ \stackrel{\Delta}{=}\ [\![\mu\alpha.[\alpha]M[N{\cdot}\alpha/\delta]N]\!]^{\lambda\mu}_{\shortparallel}(\alpha).$$

($(\mu\delta.[\nu]M)N\to\mu\alpha.[\nu](M[N{\cdot}\alpha/\delta])$): Again by Lemma 6.6, we know that

$$[\![(\mu\delta.[\nu]M)N]\!]^{\lambda\mu}_{\shortparallel}(\alpha)\ \to\ [\![M]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\delta}\dagger\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle).$$

Since δ is not introduced, the latter reduces to $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)$.
By Lemma 6.7, $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\delta}\not\prec\widehat{x}\,([\![N]\!]^{\lambda\mu}_{\shortparallel}(\beta)\,\widehat{\beta}\,[x]\,\widehat{y}\,\langle y{\cdot}\alpha\rangle)\downarrow_{\mathcal{X}}[\![M[N{\cdot}\alpha/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\nu)$. Notice that

$$[\![M[N{\cdot}\alpha/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\nu)\ \leftarrow\ [\![M[N{\cdot}\alpha/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\nu)\,\widehat{\alpha}\dagger\widehat{v}\,\langle v{\cdot}\alpha\rangle\ \stackrel{\Delta}{=}\ [\![\mu\alpha.[\nu]M[N{\cdot}\alpha/\delta]]\!]^{\lambda\mu}_{\shortparallel}(\alpha).$$ □

We can also show that types are preserved by the interpretation:

Theorem 6.9 *If* $\Gamma\vdash_{\lambda\mu}M:A\mid\Delta$, *then* $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\alpha)\mathrel{:\cdot}\Gamma\vdash\alpha{:}A,\Delta$.

Proof: As the proof of Theorem 4.8; we only need to focus here on the two rules dealing with μ-abstractions. First, let $M=\mu\delta.[\gamma]N$. Then $[\![M]\!]^{\lambda\mu}_{\shortparallel}(\alpha)\ \stackrel{\Delta}{=}\ [\![N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\delta}\dagger\widehat{x}\,\langle x{\cdot}\alpha\rangle$. Assume $\Gamma\vdash_{\lambda\mu}M:A\mid\gamma{:}B,\Delta$, then also $\Gamma\vdash_{\lambda\mu}N:B\mid\delta{:}A,\gamma{:}B,\Delta$, and $[\![N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\mathrel{:\cdot}\Gamma\vdash\delta{:}A,\gamma{:}B$ follows by induction. Then:

$$\dfrac{\boxed{\phantom{[\![N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\mathrel{:\cdot}\Gamma\vdash\delta{:}A,\gamma{:}B,\Delta}}\qquad\dfrac{}{\langle x{\cdot}\alpha\rangle\mathrel{:\cdot}\Gamma,x{:}A\vdash\alpha{:}A,\gamma{:}B,\Delta}}{[\![N]\!]^{\lambda\mu}_{\shortparallel}(\gamma)\,\widehat{\delta}\dagger\widehat{x}\,\langle x{\cdot}\alpha\rangle\mathrel{:\cdot}\Gamma\vdash\alpha{:}A,\gamma{:}B,\Delta}$$

The alternative $M=\mu\delta.[\delta]N$ is similar. □

7 Interpreting $\overline{\lambda}\mu\tilde{\mu}$

In its typed version, $\mathcal{X}$ is a proof-term syntax for a classical sequent calculus. Another proof-system has been proposed for (a variant of) classical sequent calculus in Curien and Herbelin's $\overline{\lambda}\mu\tilde{\mu}$-calculus [19]. It is interesting to relate those two formalisms and realise that $\overline{\lambda}\mu\tilde{\mu}$ can be interpreted in $\mathcal{X}$ as well.

For the $\overline{\lambda}\mu\tilde{\mu}$-calculus as presented in [19], there are two sets of variables: x,y,z, etc., label the types of the hypotheses and α,β,γ, etc., label the types of the conclusions. Moreover, the syntax of $\overline{\lambda}\mu\tilde{\mu}$ has three different categories: commands, terms, and contexts or coterms. Correspondingly, they are typed by three kinds of sequents: the usual sequents $\Gamma\vdash_{\text{LK}}\Delta$ type commands, while the sequents typing terms (resp. contexts) are of the form $\Gamma\vdash_{\text{LK}}A\mid\Delta$ (resp. $\Gamma\mid A\vdash_{\text{LK}}\Delta$), marking the conclusion (resp. hypothesis) A as *active*.

Definition 7.1 (Commands, Terms and Contexts)

$$\begin{array}{lll}
c & ::= & \langle v\mid e\rangle \qquad (\textit{commands})\\
e & ::= & \alpha\mid v{\cdot}e\mid\tilde{\mu}x.c \qquad (\textit{contexts})\\
v & ::= & x\mid\lambda x.v\mid\mu\beta.c \qquad (\textit{terms})
\end{array}$$

Here $\mu\alpha.c$, $\tilde{\mu}x.c'$ and $\lambda y.v$ respectively bind α in c, x in c' and y in v; as usual, we will consider terms, contexts and commands up to α-conversion, writing them in a way satisfying

Barendregt's convention.

A context can be a variable but can also be more complex (so as to have a typing rule introducing '$\to$' on the left-hand side of a sequent), and commands fill the hole of a context with a term.

Definition 7.2 (Typing for $\overline{\lambda}\mu\tilde{\mu}$)

$$(cut): \frac{\Gamma \vdash_{\overline{\lambda}} v : A \mid \Delta \quad \Gamma \mid e : A \vdash_{\overline{\lambda}} \Delta}{\langle v|e\rangle : \Gamma \vdash_{\overline{\lambda}} \Delta}$$

$$(Ax_L): \frac{}{\Gamma \mid \alpha : A \vdash_{\overline{\lambda}} \alpha{:}A, \Delta} \qquad (Ax_R): \frac{}{\Gamma, x{:}A \vdash_{\overline{\lambda}} x : A \mid \Delta}$$

$$(\to_L): \frac{\Gamma \vdash_{\overline{\lambda}} v : A \mid \Delta \quad \Gamma \mid e : B \vdash_{\overline{\lambda}} \Delta}{\Gamma \mid v{\cdot}e : A{\to}B \vdash_{\overline{\lambda}} \Delta} \qquad (\to_R): \frac{\Gamma, x{:}A \vdash_{\overline{\lambda}} v : B \mid \Delta}{\Gamma \vdash_{\overline{\lambda}} \lambda x.v : A{\to}B \mid \Delta}$$

$$(\tilde{\mu}): \frac{c : \Gamma, x{:}A \vdash_{\overline{\lambda}} \Delta}{\Gamma \mid \tilde{\mu}x.c : A \vdash_{\overline{\lambda}} \Delta} \qquad (\mu): \frac{c : \Gamma \vdash_{\overline{\lambda}} \alpha{:}A, \Delta}{\Gamma \vdash_{\overline{\lambda}} \mu\alpha.c : A \mid \Delta}$$

Note that the type of a context is the type that a term is expected to have in order to fill the hole, much like the import circuit in $\mathcal{X}$. With conventional notations about contexts, $v{\cdot}e$ is to be thought of as $e[[\,]\,v]$. We see here how a term (context) is built either by introducing '$\to$' on the right-hand side (left-hand side) of a sequent, or just by activating one conclusion (hypothesis) from a sequent typing a command: $\mu\alpha.c$ is inherited from $\lambda\mu$, and $\tilde{\mu}x.c$ is to be thought as $\mathsf{let}\ x = [\,]\ \mathsf{in}\ c$.

Proofs can often be simplified, that is, commands can be computed by eliminating the cuts:

Definition 7.3 (Reduction in $\overline{\lambda}\mu\tilde{\mu}$)

$$\begin{array}{rrcl} (\to): & \langle \lambda x.v_1 \mid v_2{\cdot}e\rangle & \to & \langle v_2 \mid \tilde{\mu}x.\langle v_1 \mid e\rangle\rangle \\ (\mu): & \langle \mu\beta.c \mid e\rangle & \to & c[e/\beta] \\ (\tilde{\mu}): & \langle v \mid \tilde{\mu}x.c\rangle & \to & c[v/x] \end{array}$$

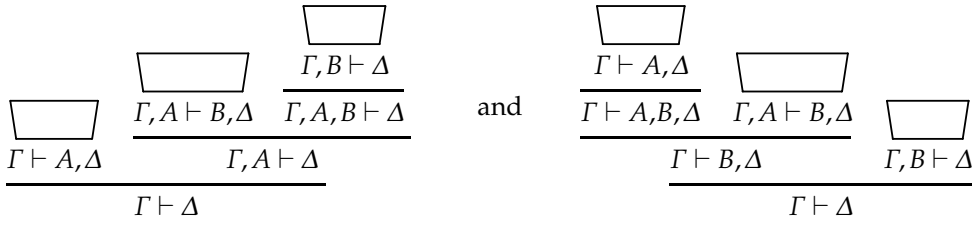
The system has a critical pair $\langle \mu\alpha.c_1 \mid \tilde{\mu}x.c_2\rangle$ and applying in this case rule (μ) gives a call-by-value evaluation, whereas applying rule $(\tilde{\mu})$ gives a call-by-name evaluation. As can be expected, the system with both rules is not confluent, as neither is the *cut*-elimination of the classical sequent calculus.

Herbelin's $\overline{\lambda}\mu\tilde{\mu}$-calculus expresses elegantly the duality of LK's left- and right introduction in a very symmetrical syntax. But the duality goes beyond that: for instance, the symmetry of the reduction rules display syntactically the duality between the CBV and CBN evaluations (see also [48]). Indeed, the call-by-value reduction $\to_{\mathrm{v}}$ is obtained by forbidding a $\tilde{\mu}$-reduction when the redex is also a μ-redex, whereas the call-by-name reduction $\to_{\mathrm{N}}$ forbids a μ-reduction when the redex is also a $\tilde{\mu}$-redex. We show here how $\mathcal{X}$ accounts for this duality.

However, this duality notwithstanding, $\overline{\lambda}\mu\tilde{\mu}$ does not fully represent LK. The LK proof

$$\dfrac{\dfrac{\begin{array}{c}\\ \Gamma, A \vdash B, \Delta\end{array}}{\Gamma \vdash A{\to}B, \Delta}(\to R) \qquad \dfrac{\begin{array}{c}\\ \Gamma \vdash A, \Delta\end{array} \qquad \begin{array}{c}\\ \Gamma, B \vdash \Delta\end{array}}{\Gamma, A{\to}B \vdash \Delta}(\to L)}{\Gamma \vdash \Delta}(cut)$$

(inhabited in $\mathcal{X}$ by the left-hand side of rule $(exp\text{-}imp)$) reduces to both



The first result is represented in the normal reduction system of $\overline{\lambda}\mu\tilde{\mu}$, but the second is not, whereas both are represented in $\mathcal{X}$, by the two right-hand sides of rule *(exp-imp)*. This implies of course that there does not exist a full reduction preserving interpretation of $\mathcal{X}$ into $\overline{\lambda}\mu\tilde{\mu}$.

We can show that there exists an obvious translation from $\mathcal{X}$ into $\overline{\lambda}\mu\tilde{\mu}$:

Definition 7.4 (Translation of $\mathcal{X}$ into $\overline{\lambda}\mu\tilde{\mu}$ [35])

$$\begin{array}{rcl}
[\![\langle x{\cdot}\alpha\rangle]\!]^{\mathcal{X}} & \triangleq & \langle x \mid \alpha\rangle \\
[\![\widehat{x}P\widehat{\alpha}\cdot\beta]\!]^{\mathcal{X}} & \triangleq & \langle \lambda x.\mu\alpha.[\![P]\!]^{\mathcal{X}} \mid \beta\rangle \\
[\![P\widehat{\alpha}\,[x]\,\widehat{y}Q]\!]^{\mathcal{X}} & \triangleq & \langle x \mid (\mu\alpha.[\![P]\!]^{\mathcal{X}}){\cdot}(\tilde{\mu}y.[\![Q]\!]^{\mathcal{X}})\rangle \\
[\![P\widehat{\alpha}\dagger\widehat{x}Q]\!]^{\mathcal{X}} & \triangleq & \langle \mu\alpha.[\![P]\!]^{\mathcal{X}} \mid \tilde{\mu}x.[\![Q]\!]^{\mathcal{X}}\rangle
\end{array}$$

In fact, this is the origin of $\mathcal{X}$: in Remark 4.1 of [19], Curien and Herbelin give a hint on a way to connect $\mathsf{LK}_{\mu\tilde{\mu}}$ and LK. The proofs of LK embed in $\mathsf{LK}_{\mu\tilde{\mu}}$ by considering the following sub-syntax of $\overline{\lambda}\mu\tilde{\mu}$:

$$c \;::=\; \langle x \mid \alpha\rangle \mid \langle \lambda x.\mu\alpha.c \mid \beta\rangle \mid \langle y \mid \mu\alpha.c{\cdot}\tilde{\mu}x.c\rangle \mid \langle \mu\alpha.c \mid \tilde{\mu}x.c\rangle$$

where the typing rules correspond respectively to the axiom rule, the right introduction of '$\Rightarrow$', the left introduction of '$\Rightarrow$', and the rule for a cut. Later it was discovered that this corresponded closely to Urban's approach in [45]; however, as discussed above after Definition 4.3 on page 21 and following, the approaches differ.

Although $\mathcal{X}$ is in fact a sub-syntax of $\overline{\lambda}\mu\tilde{\mu}$, it is not less expressive. On the contrary, on the logical side, two proofs of the same sequent might be considered differently in $\mathsf{LK}_{\mu\tilde{\mu}}$ just because the naming of formulae, or the activation/deactivation of formulae, has not been done the same way.

We can consider proofs up to such differences with equivalence classes, and moreover there is a unique proof of each class in $\mathcal{X}$: the one eagerly naming all formulae it deals with.

In concrete terms, here is a translation of $\overline{\lambda}\mu\tilde{\mu}$ into $\mathcal{X}$ which preserves the typing:

Definition 7.5 (Translation of $\overline{\lambda}\mu\tilde{\mu}$ into $\mathcal{X}$ [35])

$$[\![\langle v \mid e\rangle]\!]^{\overline{\lambda}} \;\triangleq\; [\![v]\!]^{\overline{\lambda}}_{\alpha}\,\widehat{\alpha}\dagger\widehat{x}\,[\![e]\!]^{\overline{\lambda}}_{x}$$

$$\begin{array}{rcl@{\qquad}rcl}
[\![x]\!]^{\overline{\lambda}}_{\alpha} & \triangleq & \langle x{\cdot}\alpha\rangle & [\![\alpha]\!]^{\overline{\lambda}}_{x} & \triangleq & \langle x{\cdot}\alpha\rangle \\
[\![\lambda x.v]\!]^{\overline{\lambda}}_{\alpha} & \triangleq & \widehat{x}\,[\![v]\!]^{\overline{\lambda}}_{\beta}\widehat{\beta}\cdot\alpha & [\![v{\cdot}e]\!]^{\overline{\lambda}}_{x} & \triangleq & [\![v]\!]^{\overline{\lambda}}_{\alpha}\,\widehat{\alpha}\,[x]\,\widehat{y}\,[\![e]\!]^{\overline{\lambda}}_{y} \\
[\![\mu\beta.c]\!]^{\overline{\lambda}}_{\alpha} & \triangleq & [\![c]\!]^{\overline{\lambda}}\,\widehat{\beta}\dagger\widehat{x}\langle x{\cdot}\alpha\rangle & [\![\tilde{\mu}y.c]\!]^{\overline{\lambda}}_{x} & \triangleq & \langle x{\cdot}\beta\rangle\,\widehat{\beta}\dagger\widehat{y}\,[\![c]\!]^{\overline{\lambda}}
\end{array}$$

Example 7.6 In [25] it was shown that Peirce's Law $((A{\to}B){\to}A){\to}A$ can be inhabited in $\overline{\lambda}\mu\tilde{\mu}$ by the term

$$\lambda z.\mu\alpha.\langle z \mid (\lambda y.\mu\beta.\langle y \mid \alpha\rangle){\cdot}\alpha\rangle,$$

a term in normal form which is itself the reduction of the translation of the $\lambda\mu$-term given by [39] (see Section 6) and that is typeable as follows (where $\vdash = \vdash_{\overline{\lambda}}$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z{:}(A{\to}B){\to}A, y{:}A \vdash_{\overline{\lambda}} y : A \mid \alpha{:}A, \beta{:}B}\,(Ax_R) \quad \dfrac{}{z{:}(A{\to}B){\to}A, y{:}A \mid \alpha : A \vdash_{\overline{\lambda}} \alpha{:}A, \beta{:}B}\,(Ax_L)}{\langle y|\alpha\rangle : z{:}(A{\to}B){\to}A, y{:}A \vdash_{\overline{\lambda}} \beta{:}B, \alpha{:}A}\,(cut)}{z{:}(A{\to}B){\to}A, y{:}A \vdash_{\overline{\lambda}} \mu\beta.\langle y|\alpha\rangle : B \mid \alpha{:}A}\,(\mu)}{z{:}(A{\to}B){\to}A \vdash_{\overline{\lambda}} \lambda y.\mu\beta.\langle y|\alpha\rangle : A{\to}B \mid \alpha{:}A}\,(RI) \quad \dfrac{}{z{:}(A{\to}B){\to}A \mid \alpha : A \vdash_{\overline{\lambda}} \alpha{:}A}\,(Ax_L)}{z{:}(A{\to}B){\to}A \mid (\lambda y.\mu\beta.\langle y|\alpha\rangle){\cdot}\alpha : (A{\to}B){\to}A \vdash_{\overline{\lambda}} \alpha{:}A}\,(LI)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z{:}(A{\to}B){\to}A \vdash_{\overline{\lambda}} z : (A{\to}B){\to}A \mid \alpha{:}A}\,(Ax_L) \quad \vdots}{\langle z \mid (\lambda y.\mu\beta.\langle y|\alpha\rangle){\cdot}\alpha\rangle : z{:}(A{\to}B){\to}A \vdash_{\overline{\lambda}} \alpha{:}A}\,(cut)}{z{:}(A{\to}B){\to}A \vdash_{\overline{\lambda}} \mu\alpha.\langle z \mid (\lambda y.\mu\beta.\langle y|\alpha\rangle){\cdot}\alpha\rangle : A \mid}\,(\mu)}{\vdash_{\overline{\lambda}} \lambda z.\mu\alpha.\langle z \mid (\lambda y.\mu\beta.\langle y|\alpha\rangle){\cdot}\alpha\rangle : ((A{\to}B){\to}A){\to}A \mid}$$

When we remove all term information from this derivation, we obtain a semi-proof (with focus) for Peirce's Law in Classical Logic.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(A{\to}B){\to}A, A \vdash A \mid A, B} \quad \dfrac{}{(A{\to}B){\to}A, A \mid A \vdash A, B}}{(A{\to}B){\to}A, A \vdash B, A}}{(A{\to}B){\to}A, A \vdash B \mid A}}{(A{\to}B){\to}A \vdash A{\to}B \mid A} \quad \dfrac{}{(A{\to}B){\to}A \mid A \vdash A}}{(A{\to}B){\to}A \mid (A{\to}B){\to}A \vdash A}}{(A{\to}B){\to}A \vdash (A{\to}B){\to}A \mid A}$$

$$\dfrac{\dfrac{(A{\to}B){\to}A \vdash A}{(A{\to}B){\to}A \vdash A \mid}}{\vdash ((A{\to}B){\to}A){\to}A \mid}$$

Notice that the term $\lambda z.\mu\alpha.\langle z \mid (\lambda y.\mu\beta.\langle y \mid \alpha\rangle){\cdot}\alpha\rangle$ is in normal form, whereas the proof has two cuts that can be eliminated.

Moreover, the two (activation) steps, i.e. the μ-abstractions, in this derivation do not correspond to a logical rule. This means that all provable judgements can be inhabited, but not all derivations correspond to proofs.

Using the translation function $[\![\cdot]\!]^{\overline{\lambda}}$ on the $\overline{\lambda}\mu\tilde{\mu}$-term, we obtain

$$\begin{aligned}
[\![\lambda z.\mu\alpha.\langle z \mid (\lambda y.\mu\beta.\langle y \mid \alpha\rangle){\cdot}\alpha\rangle]\!]^{\overline{\lambda}}_{\gamma} &\stackrel{\Delta}{=}\\
\widehat{z}\,[\![\mu\alpha.\langle z \mid (\lambda y.\mu\beta.\langle y \mid \alpha\rangle){\cdot}\alpha\rangle]\!]^{\overline{\lambda}}_{\delta}\widehat{\delta}\cdot\gamma &\stackrel{\Delta}{=}\\
\widehat{z}\,([\![\langle z \mid (\lambda y.\mu\beta.\langle y \mid \alpha\rangle){\cdot}\alpha\rangle]\!]^{\overline{\lambda}}\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\stackrel{\Delta}{=}\\
\widehat{z}\,([\![z]\!]^{\overline{\lambda}}_{\epsilon}\,\widehat{\epsilon}\dagger\widehat{x}\,[\![(\lambda y.\mu\beta.\langle y \mid \alpha\rangle){\cdot}\alpha]\!]^{\overline{\lambda}}_{x})\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\stackrel{\Delta}{=}\\
\widehat{z}\,((\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,([\![\lambda y.\mu\beta.\langle y \mid \alpha\rangle]\!]^{\overline{\lambda}}_{\phi}\,\widehat{\phi}\,[x]\,\widehat{w}\,[\![\alpha]\!]^{\overline{\lambda}}_{w}))\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\stackrel{\Delta}{=}\\
\widehat{z}\,((\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,((\widehat{y}\,([\![\langle y \mid \alpha\rangle]\!]^{\overline{\lambda}}\,\widehat{\beta}\dagger\widehat{b}\,\langle v{\cdot}\eta\rangle)\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\stackrel{\Delta}{=}\\
\widehat{z}\,((\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,((\widehat{y}\,(([\![y]\!]^{\overline{\lambda}}_{\rho}\,\widehat{\rho}\dagger\widehat{u}\,[\![u]\!]^{\overline{\lambda}}_{\alpha})\,\widehat{\beta}\dagger\widehat{b}\,\langle v{\cdot}\eta\rangle)\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\stackrel{\Delta}{=}\\
\widehat{z}\,((\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,((\widehat{y}\,((\langle y{\cdot}\rho\rangle\,\widehat{\rho}\dagger\widehat{u}\,\langle u{\cdot}\alpha\rangle)\,\widehat{\beta}\dagger\widehat{b}\,\langle v{\cdot}\eta\rangle)\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &
\end{aligned}$$

Notice that the process obtained via translation has four cuts: removing these produces the following (in-side out) reduction sequence.

$$\begin{aligned}
\widehat{z}\,((\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,((\widehat{y}\,(\underline{(\langle y{\cdot}\rho\rangle\,\widehat{\rho}\dagger\widehat{u}\,\langle u{\cdot}\alpha\rangle)}\,\widehat{\beta}\dagger\widehat{b}\,\langle v{\cdot}\eta\rangle)\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\to\\
\widehat{z}\,((\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,((\widehat{y}\,\underline{(\langle y{\cdot}\alpha\rangle\,\widehat{\beta}\dagger\widehat{b}\,\langle v{\cdot}\eta\rangle)}\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle))\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\to\\
\widehat{z}\,(\underline{(\langle z{\cdot}\epsilon\rangle\,\widehat{\epsilon}\dagger\widehat{x}\,((\widehat{y}\,\langle y{\cdot}\alpha\rangle\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[x]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle))}\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\to\\
\widehat{z}\,(((\widehat{y}\,\langle y{\cdot}\alpha\rangle\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[z]\,\widehat{w}\,\langle w{\cdot}\alpha\rangle)\,\widehat{\alpha}\dagger\widehat{x}\,\langle x{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma &\to\\
\widehat{z}\,((\widehat{y}\,\langle y{\cdot}\delta\rangle\widehat{\eta}\cdot\phi)\,\widehat{\phi}\,[z]\,\widehat{w}\,\langle w{\cdot}\delta\rangle)\widehat{\delta}\cdot\gamma. &
\end{aligned}$$

The last term is exactly that of Example 3.5.

The interpretation function preserves typeability:

Lemma 7.7 *i) If* $\Gamma \vdash_{\overline{\lambda}} v : A \mid \Delta$, *then* $[\![v^{\overline{\lambda}}_{\parallel \alpha} :\cdot \Gamma \vdash \alpha{:}A, \Delta$.
ii) If $\Gamma \mid e : A \vdash_{\overline{\lambda}} \Delta$, *then* $[\![e^{\overline{\lambda}}_{\parallel x} :\cdot \Gamma, x{:}A \vdash \Delta$.
iii) If $c : \Gamma \vdash_{\overline{\lambda}} \Delta$, *then* $[\![c^{\overline{\lambda}}_{\parallel} :\cdot \Gamma \vdash \Delta$.

Proof: Straightforward, by simultaneous induction on the structure of derivations.

In [35] it is shown that we can simulate the implicit substitution of $\overline{\lambda}\mu\tilde{\mu}$ in $\mathcal{X}$.

Lemma 7.8 *i)* $[\![c^{\overline{\lambda}}_{\parallel} \widehat{\alpha} \not\frown \widehat{x}\, [\![e^{\overline{\lambda}}_{\parallel x} \to [\![c[e/\alpha^{\overline{\lambda}}_{\parallel}]$.
ii) $[\![v^{\overline{\lambda}}_{\parallel \beta} \widehat{\alpha} \not\frown \widehat{x}\, [\![e^{\overline{\lambda}}_{\parallel x} \to [\![v[e/\alpha^{\overline{\lambda}}_{\parallel}]\beta$.
iii) $[\![e'^{\overline{\lambda}}_{\parallel z} \widehat{\alpha} \not\frown \widehat{x}\, [\![e^{\overline{\lambda}}_{\parallel x} \to [\![e'[e/\alpha^{\overline{\lambda}}_{\parallel}]z$.

Proof: By simultaneous induction on the structure of nets.

Likewise, we can show:

Lemma 7.9 *i)* $[\![v^{\overline{\lambda}}_{\parallel \alpha} \widehat{\alpha} \not\frown \widehat{x}\, [\![c^{\overline{\lambda}}_{\parallel} \to [\![c[v/x^{\overline{\lambda}}_{\parallel}]$.
ii) $[\![v^{\overline{\lambda}}_{\parallel \alpha} \widehat{\alpha} \not\frown \widehat{x}\, [\![v'^{\overline{\lambda}}_{\parallel \beta} \to [\![v'[v/x^{\overline{\lambda}}_{\parallel}]\beta$.
iii) $[\![v^{\overline{\lambda}}_{\parallel \alpha} \widehat{\alpha} \not\frown \widehat{x}\, [\![e^{\overline{\lambda}}_{\parallel y} \to [\![e[v/x^{\overline{\lambda}}_{\parallel}]y$.

We now strengthen these results by stating that this simulation preserves evaluations:

Theorem 7.10 (Simulation of $\overline{\lambda}\mu\tilde{\mu}$) *If* $c \to c'$ *then* $[\![c^{\overline{\lambda}}_{\parallel} \downarrow_{\mathcal{X}} [\![c'^{\overline{\lambda}}_{\parallel}$.

Proof: By induction of the length of the reduction sequence, of which we only show the base cases:
$(\to)$: Then $c = \langle \lambda x.v_1 \mid v_2 \cdot e \rangle$, and $c' = \langle v_2 \mid \tilde{\mu}x.\langle v_1 \mid e \rangle \rangle$. Then:

$$\begin{array}{rcl}
[\![\langle \lambda x.v_1 \mid v_2 \cdot e \rangle^{\overline{\lambda}}_{\parallel} & = & [\![\lambda x.v_1{}^{\overline{\lambda}}_{\parallel \alpha}\, \widehat{\alpha} \dagger \widehat{y}\, [\![v_2 \cdot e^{\overline{\lambda}}_{\parallel y} \\
& = & (\widehat{x}\, [\![v_1{}^{\overline{\lambda}}_{\parallel \beta} \widehat{\beta} \cdot \alpha)\, \widehat{\alpha} \dagger \widehat{y}\, ([\![v_2{}^{\overline{\lambda}}_{\parallel \gamma}\, \widehat{\gamma}\, [y]\, \widehat{z}\, [\![e^{\overline{\lambda}}_{\parallel z}) \\
& \to & [\![v_2{}^{\overline{\lambda}}_{\parallel \gamma}\, \widehat{\gamma} \dagger \widehat{x}\, ([\![v_1{}^{\overline{\lambda}}_{\parallel \beta}\, \widehat{\beta} \dagger \widehat{z}\, [\![e^{\overline{\lambda}}_{\parallel z}) \\
(= \alpha) & \leftarrow & [\![v_2{}^{\overline{\lambda}}_{\parallel \gamma}\, \widehat{\gamma} \dagger \widehat{y}\, (\langle y \cdot \delta \rangle\, \widehat{\delta} \dagger \widehat{x}\, ([\![v_1{}^{\overline{\lambda}}_{\parallel \beta}\, \widehat{\beta} \dagger \widehat{z}\, [\![e^{\overline{\lambda}}_{\parallel z}))) \\
& = & [\![v_2{}^{\overline{\lambda}}_{\parallel \gamma}\, \widehat{\gamma} \dagger \widehat{y}\, (\langle y \cdot \delta \rangle\, \widehat{\delta} \dagger \widehat{x}\, [\![\langle v_1 \mid e \rangle^{\overline{\lambda}}_{\parallel}) \\
& = & [\![v_2{}^{\overline{\lambda}}_{\parallel \gamma}\, \widehat{\gamma} \dagger \widehat{y}\, [\![\tilde{\mu}x.\langle v_1 \mid e \rangle^{\overline{\lambda}}_{\parallel y} \;=\; [\![\langle v_2 \mid \tilde{\mu}x.\langle v_1 \mid e \rangle \rangle^{\overline{\lambda}}_{\parallel}
\end{array}$$

(μ): By Lemma 7.8.
$(\tilde{\mu})$: By Lemma 7.9.

Notice that, had we defined $[\![\mu\beta.c^{\overline{\lambda}}_{\parallel \alpha} = [\![c^{\overline{\lambda}}_{\parallel}[\alpha/\beta]$ and $[\![\tilde{\mu}y.c^{\overline{\lambda}}_{\parallel x} = [\![c^{\overline{\lambda}}_{\parallel}[x/y]$, we could have shown that $c \to c'$ implies $[\![c^{\overline{\lambda}}_{\parallel} \to [\![c'^{\overline{\lambda}}_{\parallel}$.

In fact, we can even show

Theorem 7.11 *i) If* $c \to_{\mathrm{N}} c'$ *then* $[\![c^{\overline{\lambda}}_{\parallel} \downarrow^{\mathrm{N}}_{\mathcal{X}} [\![c'^{\overline{\lambda}}_{\parallel}$.
ii) If $c \to_{\mathrm{V}} c'$ *then* $[\![c^{\overline{\lambda}}_{\parallel} \downarrow^{\mathrm{V}}_{\mathcal{X}} [\![c'^{\overline{\lambda}}_{\parallel}$.

Curien and Herbelin define two encodings:

Definition 7.12 (Curien & Herbelin 2000) The interpretation $\cdot^{\mathrm{V}}$ and interpretation $\cdot^{\mathrm{N}}$ of $\lambda\mu$ into $\overline{\lambda}\mu\tilde{\mu}$ are defined as follows:

$$
\begin{array}{rcl@{\qquad}rcl}
x^{\mathrm{v}} & \triangleq & x & x^{\mathrm{N}} & \triangleq & x \\
(\lambda x.M)^{\mathrm{v}} & \triangleq & \lambda x.M^{\mathrm{v}} & (\lambda x.M)^{\mathrm{N}} & \triangleq & \lambda x.M^{\mathrm{N}} \\
(MN)^{\mathrm{v}} & \triangleq & \mu\alpha.\langle N^{\mathrm{v}} \mid \tilde{\mu}x.\langle M^{\mathrm{v}} \mid x{\cdot}\alpha\rangle\rangle & (MN)^{\mathrm{N}} & \triangleq & \mu\alpha.\langle M^{\mathrm{N}} \mid N^{\mathrm{N}}{\cdot}\alpha\rangle \\
(\mu\beta.c)^{\mathrm{v}} & \triangleq & \mu\beta.c^{\mathrm{v}} & (\mu\beta.c)^{\mathrm{N}} & \triangleq & \mu\beta.c^{\mathrm{N}} \\
([\alpha]M)^{\mathrm{v}} & \triangleq & \langle M^{\mathrm{v}} \mid \alpha\rangle & ([\alpha]M)^{\mathrm{N}} & \triangleq & \langle M^{\mathrm{N}} \mid \alpha\rangle
\end{array}
$$

$\cdot^{\mathrm{v}}$ corresponds to right-to-left call-by-value, and $\cdot^{\mathrm{N}}$ relates to left-to-right call-by-value.

Curien and Herbelin's result is:

Theorem 7.13 (Simulation of $\lambda\mu$ in $\overline{\lambda}\mu\tilde{\mu}$ [19]) *i) If $M \to_{\mathrm{v}} N$ then $M^{\mathrm{v}} \to_{\mathrm{v}} N^{\mathrm{v}}$ up to μ-expansion. ii) If $M \to_{\mathrm{N}} N$ then $M^{\mathrm{N}} \to_{\mathrm{N}} N^{\mathrm{N}}$ up to μ-expansion.*

Theorem 7.13 also holds for the restriction of $\lambda\mu$ to the traditional λ-calculus (but without the restriction of 'up to μ-expansion'). However, one might be disappointed that the preservation of the cbv-evaluation and cbn-evaluation relies on two *distinct* translations of terms.

As above, what accounts for the distinction cbv/cbn is the encodings themselves, and the distinction between cbv and cbn mostly relies on Curien and Herbelin's two distinct encodings rather than the features of $\overline{\lambda}\mu\tilde{\mu}$. The same holds for [48]. Whereas their cbn-translation seems intuitive, they apparently need to twist it in a more complex way than us, in order to give an accurate interpretation of the cbv-λ-calculus, since we can show the following:

Lemma 7.14 $M^{\mathrm{v}} \to M^{\mathrm{N}}$.

Proof: By induction on the structure of terms, the interesting case being:

$$
\begin{array}{rcll}
(NP)^{\mathrm{v}} & \triangleq & \mu\alpha.\langle N^{\mathrm{v}} \mid \tilde{\mu}x.\langle M^{\mathrm{v}} \mid x{\cdot}\alpha\rangle\rangle & \to \quad (IH) \\
 & & \mu\alpha.\langle N^{\mathrm{N}} \mid \tilde{\mu}x.\langle M^{\mathrm{N}} \mid x{\cdot}\alpha\rangle\rangle & \to_{\tilde{\mu}} \quad \mu\alpha.\langle M^{\mathrm{N}} \mid N^{\mathrm{N}}{\cdot}\alpha\rangle \quad \triangleq \quad (NP)^{\mathrm{N}}
\end{array}
$$

This result is a bit disappointing since the cbn-encoding turns out to be more refined than the cbv-encoding.

Of course, by Theorem 7.10, we also have

Lemma 7.15 $[\![M^{\mathrm{v}}]\!]^{\overline{\lambda}}_{\beta} \downarrow_{\mathcal{X}} [\![M^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta}$.

But we now show that we can take the simplest translation (i.e. $[\![\cdot^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta}$) for both cbv and cbn and that the evaluation strategies of $\mathcal{X}$ reflects the distinction between them, in showing that the interpretation $[\![\cdot]\!]^{\lambda}(\alpha)$ is more refined that the composition of $\cdot^{\mathrm{N}}$ and $[\![\cdot]\!]^{\overline{\lambda}}_{\alpha}$.

Lemma 7.16 $[\![M^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta} \to [\![M]\!]^{\lambda}(\beta)$.

Proof: By induction on the structure of terms:

$(M = x)$: $[\![x^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta} \triangleq [\![x]\!]^{\overline{\lambda}}_{\beta} \triangleq \langle x{\cdot}\beta\rangle \triangleq [\![x]\!]^{\lambda}(\beta)$

$(M = \lambda x.M')$: $[\![(\lambda x.M')^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta} \triangleq [\![\lambda x.M'^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta} \triangleq \widehat{x}\,[\![M'^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\alpha}\widehat{\alpha}{\cdot}\beta \to (IH)$
$\widehat{x}\,[\![M']\!]^{\lambda}(\alpha)\widehat{\alpha}{\cdot}\beta \triangleq [\![\lambda x.M']\!]^{\lambda}(\beta)$

$(M = PQ)$:
$$
\begin{array}{rll}
[\![(PQ)^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\beta} \triangleq [\![\mu\alpha\langle P^{\mathrm{N}} \mid Q^{\mathrm{N}}{\cdot}\alpha\rangle]\!]^{\overline{\lambda}}_{\beta} & \triangleq & \\
[\![\langle P^{\mathrm{N}} \mid Q^{\mathrm{N}}{\cdot}\alpha\rangle]\!]^{\overline{\lambda}}\,\widehat{\alpha} \dagger \widehat{x}\,\langle x{\cdot}\beta\rangle & \to & (\curlywedge ren) \\
[\![\langle P^{\mathrm{N}} \mid Q^{\mathrm{N}}{\cdot}\beta\rangle]\!]^{\overline{\lambda}} & \triangleq & \\
[\![P^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\gamma}\,\widehat{\gamma} \dagger \widehat{x}\,[\![Q^{\mathrm{N}}{\cdot}\beta]\!]^{\overline{\lambda}}_{x} & \triangleq & \\
[\![P^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\gamma}\,\widehat{\gamma} \dagger \widehat{x}\,([\![Q^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\alpha}\,\widehat{\alpha}\,[x]\,\widehat{y}\,[\![\beta]\!]^{\overline{\lambda}}_{y}) & \triangleq & \\
[\![P^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\gamma}\,\widehat{\gamma} \dagger \widehat{x}\,([\![Q^{\mathrm{N}}]\!]^{\overline{\lambda}}_{\alpha}\,\widehat{\alpha}\,[x]\,\widehat{y}\,\langle y{\cdot}\beta\rangle) & \to & (IH) \\
[\![P]\!]^{\lambda}(\gamma)\,\widehat{\gamma} \dagger \widehat{x}\,([\![Q]\!]^{\lambda}(\alpha)\,\widehat{\alpha}\,[x]\,\widehat{y}\,\langle y{\cdot}\beta\rangle) & \triangleq & [\![PQ]\!]^{\lambda}(\beta)
\end{array}
$$

8 Conclusions and future work

We have seen that $\mathcal{X}$ is an application- and substitution-free formal language that provides a Curry-Howard isomorphism for a sequent calculus for implicative classical logic. But, of more interest, we have seen $\mathcal{X}$ is very well suited as generic abstract machine for the running of (applicative) programming languages, by building not only an interpretation for λ, $\lambda\mu$, and $\overline{\lambda}\mu\tilde{\mu}$, but also for $\lambda\mathbf{x}$.

A wealth of research lies in the future, of which this paper is but the first step, the seed. We intend to study (strong) normalisation, confluence of the CBN and CBV strategies, to extend $\mathcal{X}$ in order to represent the other logical connectives, study the relation with linear logic, proofnets (both typed and untyped), how to express recursion, functions, etc, etc.

Details of the implementation of the tool for $\mathcal{X}$ can be found in [8, 9]. Polymorphism is introduced in [43], the encoding of other connectors in [42], and the interpretation of $\mathcal{X}$ in $\lambda\mu$ in [6]. The relation between $\mathcal{X}$ and Π is currently under investigation.

Acknowledgements

We would like to thank Alexander Summers, Daniel Hirschkoff, Dragiša Žunić, Harry Mairson, Jamie Gabbay, Jayshan Raghunandan, Luca Cardelli, Luca Roversi, Maria Grazia Vigliotti, Mariangiola Dezani-Ciancaglini, Philippe Audebaud, and Simona Ronchi della Rocca for many fruitful discussions on the topic of this paper.

We are especially grateful to Stéphane Lengrand, who stood at the start of this research, and kindly let us integrate his papers in ours. We would also like to thank Hugo Herbelin for carefully reading our paper, and suggesting many improvements and corrections.

References

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

[2] Aho, A. V., Sethi, R. & Ullman, J. D. 1988. *Compilers: Principles, Techniques and Tools*, Addison-Wesley.

[3] Appel, A. W. & Jim, T. 1989. Continuation-passing, closure-passing style, *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of Programming languages*, ACM Press, pp. 293–302.

[4] Ariola, Z. M. & Herbelin, H. 2003. Minimal classical logic and control operators., *in* J. C. M. Baeten, J. K. Lenstra, J. Parrow & G. J. Woeginger (eds), *ICALP*, Vol. 2719 of *Lecture Notes in Computer Science*, Springer, pp. 871–885.

[5] Ariola, Z. M., Herbelin, H. & Sabry, A. 2004. A type-theoretic foundation of continuations and prompts, *Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming (ICFP '04), Snowbird, Utah, September 19-21, 2004*, ACM, pp. 40–53.

[6] Audebaud, P. & van Bakel, S. 2006. Understanding $\mathcal{X}$ with $\lambda\mu$. Consistent interpretations of the implicative sequent calculus in natural deduction. *Submitted*.

[7] van Bakel, S., Lengrand, S. & Lescanne, P. 2005. The language $\mathcal{X}$: circuits, computations and classical logic, *in* M. Coppo, E. Lodi & G. M. Pinna (eds), *Proceedings of Ninth Italian Conference on Theoretical Computer Science (ICTCS'05), Siena, Italy*, Vol. 3701 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 81–96.

[8] van Bakel, S. & Raghunandan, J. 2005. Implementing $\mathcal{X}$, *Electronic Proceedings of Second International Workshop on* Term Graph Rewriting 2004 *(TermGraph'04), Rome, Italy*, Electronic Notes in Theoretical Computer Science.

[9] van Bakel, S. & Raghunandan, J. 2006. Capture avoidance and garbage collection for $\mathcal{X}$. Presented at the Third International Workshop on *Term Graph Rewriting 2006* (TermGraph'06), Vienna, Austria.

[10] Barbanera, F. & Berardi, S. 1996. A symmetric lambda calculus for classical program extraction, *Information and Computation* **125**(2): 103–117.

[11] Barendregt, H. 1984. *The Lambda Calculus: its Syntax and Semantics*, revised edn, North-Holland, Amsterdam.
[12] Barendregt, H. & Ghilezan, S. 2000. Lambda terms for natural deduction, sequent calculus and cut-elimination, *journal of Functional Porgramming* **10**(1): 121–134.
[13] Bloo, R. & Geuvers, J. H. 1996. Explicit substitution: on the edge of strong normalisation, *Computing Science Reports 96–10*, Eindhoven University of Technology, P.O.box 513, 5600 MB Eindhoven, The Netherlands. To appear in Theoretical Computer Science.
[14] Bloo, R. & Rose, K. 1995. Preservation of strong normalisation in named lambda calculi with explicit substitution and garbage collection, *CSN'95 – Computer Science in the Netherlands*, pp. 62–72. `ftp://ftp.diku.dk/diku/semantics/papers/D-246.ps`
[15] de Bruijn, N. G. 1978. A namefree lambda calculus with facilities for internal definition of expressions and segments, *TH-Report 78-WSK-03*, Technological University Eindhoven, Netherlands, Department of Mathematics.
[16] Church, A. 1940. A formulation of the simple theory of types, *The journal of Symbolic Logic* **5**: 56–68.
[17] Clement, D. 1986. A Simple Applicative Language: MINI-ML, *Technical Report 529*, INRIA centre Sophia Antipolis (France).
[18] Coquand, T. & Huet, G. P. 1985. Constructions: A higher order proof system for mechanizing mathematics, *European Conference on Computer Algebra (1)*, pp. 151–184.
[19] Curien, P.-L. & Herbelin, H. 2000. The Duality of Computation, *Proc. of the 5 th ACM SIGPLAN International Conference on Functional Programming (ICFP'00*, ACM, pp. 233–243.
[20] Curry, H. B., Feys, R. & Craig, W. 1958. *Combinatory Logic*, Vol. 1, Elsevier Science Publishers B. V. (North-Holland), Amsterdam.
[21] van Dalen, D. 1994. *Logic and Structure*, Springer Verlag.
[22] Danos, V., Joinet, J.-B. & Schellinx, H. 1996. Computational isomorphisms in classical logic (extended abstract), *Electronic Notes in Theoretical Computer Science* **3**.
[23] Danos, V., Joinet, J.-B. & Schellinx, H. 1997. A new deconstructive logic: Linear logic, *The journal of Symbolic Logic* **62**.
[24] Gentzen, G. 1935. Untersuchungen über das Logische Schliessen, *Mathematische Zeitschrift* **39**: 176–210 and 405–431. English translation in [44], Pages 68–131.
[25] Ghilezan, S. & Lescanne, P. 2004. Classical proofs, typed processes and intersection types, *in* S. Berardi, M. Coppo & F. Damiani (eds), *TYPES'03*. forthcoming in Lecture Notes in Computer Science.
[26] Girard, J. Y. 1986. The system F of variable types, fifteen years later, *Theoret. Comput. Sci.* **45**: 159–192.
[27] Girard, J.-Y. 1987. Linear logic, *Theoretical Computer Science* **50**: 1–102.
[28] Girard, J.-Y. 1991. A new constrcutive logic: classical logic, *Mathematical Structures in Computer Science* **1**(3): 255–296.
[29] Griffin, T. 1990. A formulae-as-types notion of control, *Proceedings of the 17th Annual ACM Symposium on Principles Of Programming Languages, Orlando (Fla., USA)*, pp. 47–58.
[30] de Groote, P. 1994. On the relation between the $\lambda\mu$-calculus and the syntactic theory of sequential control, *in* Springer-Verlag (ed.), *Proceedings of the 5th International Conference on Logic Programming and Automated Reasoning, (LPAR'94)*, Vol. 822 of *Lecture Notes in Computer Science*, pp. 31–43.
[31] 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 d'université, Université Paris 7.
[32] Herbelin, H. 2005. *C'est maintenant qu'on calcule: au cœur de la dualité*, Mémoire d'habilitation, Université Paris 11.
[33] Krivine, J.-L. 1994. Classical logic, storage operators and second-order lambda-calculus., *Ann. Pure Appl. Logic* **68**(1): 53–78.
[34] Lengrand, S. 2002. *A computational interpretation of the* cut*-rule in classical sequent calculus*, Master's thesis, Mathematical Institute & Computing Laboratory, University of Oxford.
[35] Lengrand, S. 2003. Call-by-value, call-by-name, and strong normalization for the classical sequent calculus, *in* B. Gramlich & S. Lucas (eds), *Post-proceedings of the 3rd Workshop on Reduction Strategies in Rewriting and Programming (WRS 2003)*, Vol. 86 of *Electronic Notes in Theoretical Computer Science*, Elsevier.
[36] Lengrand, S. 2006. *Normalisation & Equivalence in Proof Theory & Type Theory*, PhD thesis, University of Paris VII & University of St Andrews.
[37] Lengrand, S., Lescanne, P., Dougherty, D., Dezani-Ciancaglini, M. & van Bakel, S. 2004. Intersection types for explicit substitutions, *Information and Computation* **189**(1): 17–42.
[38] Lescanne, P. 1994. From $\lambda\sigma$ to $\lambda\upsilon$, a journey through calculi of explicit substitutions, *in* H. Boehm

(ed.), *Proceedings of the 21st Annual ACM Symposium on Principles Of Programming Languages, Portland (Or., USA)*, ACM, pp. 60–69.

[39] Ong, C.-H. L. & Stewart, C. A. 1997. A Curry-Howard foundation for functional computation with control, *Proceedings of the 24th Annual ACM Symposium on Principles Of Programming Languages, Paris (France)*, pp. 215–227.

[40] Parigot, M. 1992. An algorithmic interpretation of classical natural deduction, *Proc. of Int. Conf. on Logic Programming and Automated Reasoning, LPAR'92*, Vol. 624 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 190–201.

[41] Parigot, M. 1993. Classical proofs as programs., *Kurt Gödel Colloquium*, pp. 263–276. Presented at TYPES Workshop, at Bǎstad, June 1992.

[42] Raghunandan, J. & Summers, A. 2006. On the Computational Representation of Classical Logical Connectives. To be presented at 2nd International Workshop on *Developments in Computational Models* (DCM 2006), Venice, Italy.

[43] Summers, A. & van Bakel, S. 2006. Approaches to polymorphism in classical sequent calculus, *in* P. Sestoft (ed.), *Proceedings of 15th European Symposium on Programming (ESOP'06)*, Vienna, Austria, Vol. 3924 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 84 – 99.

[44] Szabo, M. E. (ed.) 1969. *The Collected Papers of Gerhard Gentzen*, Studies in Logic and the Foundations of Mathematics, North-Holland.

[45] Urban, C. 2000. *Classical Logic and Computation*, PhD thesis, University of Cambridge.

[46] Urban, C. 2001. Strong Normalisation for a Gentzen-like Cut-Elimination Procedure', *Proceedings of Typed Lambda Calculus and Applications (TLCA'01)*, Vol. 2044 of *Lecture Notes in Computer Science*, pp. 415–429.

[47] Urban, C. & Bierman, G. M. 2001. Strong normalisation of cut-elimination in classical logic, *Fundamentae Informaticae* **45**(1,2): 123–155.

[48] Wadler, P. 2003. Call-by-Value is Dual to Call-by-Name, *Proceedings of the eighth ACM SIGPLAN international conference on Functional programming*, pp. 189 – 201.

[49] Whitehead, A. N. & Russell, B. 1925. *Principia Mathematica*, 2nd edn, Cambridge University Press.

[50] Whitehead, A. N. & Russell, B. 1997. *Principia Mathematica to *56*, Cambridge Mathematical Library, 2nd edn, Cambridge University Press.