Math. Struct. in Comp. Science (2008), *vol.* 18, *pp.* 555–609.
doi:10.1017/S0960129508006762 Printed in the United Kingdom

Computation with classical sequents

STEFFEN VAN BAKEL[†] and PIERRE LESCANNE[‡]

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

Received 10 November 2005; revised 27 June 2007

$\mathscr{X}$ is an untyped continuation-style formal language with a typed subset that provides a Curry–Howard isomorphism for a sequent calculus for implicative classical logic. $\mathscr{X}$ can also be viewed as a language for describing nets by composition of basic components connected by wires. These features make $\mathscr{X}$ 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 $\mathscr{X}$; in order to demonstrate its expressive power, we will show how elaborate calculi can be embedded, such as the λ-calculus, Bloo and Rose's calculus of explicit substitutions $\lambda\mathbf{x}$, Parigot's $\lambda\mu$ and Curien and Herbelin's $\overline{\lambda}\mu\tilde{\mu}$.
$\mathscr{X}$ was first presented in Lengrand (2003), where it was called the $\lambda\xi$-calculus. It can be seen as the pure untyped computational content of the reduction system for the implicative classical sequent calculus of Urban (2000).

1. Introduction

A number of systems have appeared in the literature linking classical logic with a notion of computation. In the past, say before the Ph.D. theses of Herbelin (Herbelin 1995) and Urban (Urban 2000), the study of the relationship between *computation*, *programming languages* and *logic* was concentrated mainly on *natural deduction systems*. In fact, these deservedly carry the description '*natural*'; 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, with much further development, to System F (Girard 1986) and the Calculus of Constructions (Coquand and Huet 1985).

As an example of an approach for representing classical proofs, Parigot's $\lambda\mu$-calculus (see Parigot (1992)) 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 a 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* (Ariola and Herbelin 2003). The link between

[†] Partially supported by École Normale Supérieure de Lyon, France

classical logic and continuations and control was first established for the λ_C-calculus (Griffin 1990) (where C stands for Felleisen's C operator).

The sequent calculus, which was introduced by Gentzen in Gentzen (1935), 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, through an application of the (*cut*)-rule. 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 Barbanera and Berardi (1996), 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 a requirement for the left-introduction rules to manipulate hypotheses, as studied by Herbelin in Herbelin (1995), Curien and Herbelin (2000) and Herbelin (2005). The relation between call-by-name and call-by-value in the fragment of LK with negation and conjunction in Wadler's Dual Calculus was studied in Wadler (2003); as in calculi like $\lambda\mu$ and $\bar{\lambda}\mu\tilde{\mu}$, 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, thereby generating a proof in *normal form* of the same sequent, that is, without a cut. It is defined using rewriting steps, that is, local reductions of the proof-tree, which has, with some differences, the flavour of the evaluation of explicit substitutions (de Bruijn 1978; Abadi *et al.* 1991). Indeed, the typing rule of an explicit substitution, say in $\lambda\mathbf{x}$ (Bloo and Rose 1995), is just a variant of the (*cut*)-rule, and much work has been done to gain a better understanding of the connection between explicit substitutions and local cut-reduction procedures.

This paper, which is a continuation of Lengrand (2003), presents a correspondence *à la* Curry–Howard for LK, bringing together the various features of two different approaches that we compare: that of Urban (Urban 2000; Urban and Bierman 2001; Urban 2001) and that of Curien and Herbelin (Curien and Herbelin 2000). While Curien and Herbelin insist on the duality of the calculus, Urban, in Urban (2000), thoroughly analyses, among other things, Gentzen-like cut-elimination procedures, and defines a very general reduction system for proofs in LK that is strongly normalising, and in which proofs are represented by a syntax of terms. $\mathscr{X}$ is actually the implicative part of his calculus with a new syntax.

In this paper, we will try to take up the gauntlet on behalf of 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 $\mathscr{X}$ that describes nets, and its reduction rules, which join nets. The logic we will consider is restricted to implication, but that is mainly because in this initial phase we have aimed for simplicity; cut-elimination in sequent calculi is notorious for the large number of rules, which will increase many-fold when considering additional logical connectives.

Breaking with the natural deduction paradigm comes at a price, in that *abstraction* and *application* (corresponding to *introduction of implication* and *modus ponens*) are no

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

longer the basic tools of the extracted language. In fact, the language we obtain has more of a *continuation* style, in that it models both the *parameter* as well as the *context* call. However, abstraction and application can be faithfully implemented, and we will show how $\mathscr{X}$ can be used to describe the behaviour of functional programming languages at a very low level of granularity.

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

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

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

The origins of the work presented in this paper lie, first of all, in Lengrand (2002), where Lengrand defined a first approach to the definition of a calculus, which he 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 notation for the sequent calculus first presented by Urban in his Ph.D. thesis (Urban 2000). This was then studied in relation to $\overline{\lambda}\mu\tilde{\mu}$ (Curien and Herbelin 2000) by Lengrand (Lengrand 2003) through the calculus $\lambda\xi$. The decision to study the connection between $\lambda\xi$ and $\overline{\lambda}\mu\tilde{\mu}$, as reported in Lengrand (2003), was not accidental, and was taken as the starting point for our investigations. We changed the name of the calculus from $\lambda\xi$ to $\mathscr{X}$ 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 $\mathscr{X}$ and $\overline{\lambda}\mu\tilde{\mu}$.

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

As we believe strongly in the importance of syntax for gaining a better grasp 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[†]. This is achieved by studying $\mathscr{X}$ in the context of the normal functional programming languages paradigms, but, more importantly, to cut the link between $\mathscr{X}$ and classical logic. We achieve this by studying our language *without types*; in this way we will also consider

[†] The relationship between $\mathscr{X}$ and its predecessors is the same as the relationship between, say, mini-ML (Clement 1986) and the lambda-calculus.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

nets that do *not* correspond to proofs – in particular, we consider non-termination nets. In fact, we aim to study $\mathscr{X}$ outside the context of classical logic in much the same way as the λ-calculus is studied outside the context of intuitionistic logic.

$\mathscr{X}$ 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 describes 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 through the process of *substitution*. This description is rather basic as it says nothing about the actual *cost* of the substitution, which is quite high at run time. Usually, a calculus of *explicit substitutions* (Bloo and Rose 1995; Abadi *et al.* 1991; Lescanne 1994; Lengrand *et al.* 2004) 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 are thinking of $\lambda\mathbf{x}$, the calculus of *explicit substitution with explicit names*, due to Bloo and Rose (Bloo and Rose 1995). $\lambda\mathbf{x}$ 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 $\mathscr{X}$, we reach a 'subatomic' level by decomposing explicit substitutions into smaller components. At this level, the calculus $\mathscr{X}$ explains how substitutions and terms interact.

The calculus is in fact symmetric (Barbanera and Berardi 1996) and, unlike $\lambda\mathbf{x}$ where a substitution is applied to a term, a term in $\mathscr{X}$ can also be applied to a substitution. Their interaction percolates subtly and gently through the term or substitution according to the direction chosen. We will see that the these two kinds of interaction have a direct connection with call-by-value and call-by-name strategies, which both have a natural description in $\mathscr{X}$.

The ingredients of the syntax

It is important to note that $\mathscr{X}$ does *not* have variables[†] – like the λ-calculus or $\bar{\lambda}\mu\tilde{\mu}$ – as possible places where terms might be inserted. Instead, $\mathscr{X}$ has *wires*, which are also called *connectors*, that can occur free or bound in a term. As in the λ-calculus, the binding of a wire indicates that it is *active* in the computation; but unlike in the λ-calculus, 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* (which are reminiscent of values) and *plugs* (which are reminiscent of continuations) – and these correspond to *variables* and *co-variables*, respectively, in Wadler (2003), or, alternatively, to Parigot's lambda-variables

[†] Care should be taken not to become confused by the use of names like x for the class of connectors, which are called plugs; these names are, in fact, inherited from $\bar{\lambda}\mu\tilde{\mu}$.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

and mu-variables (Parigot 1992) (see also Curien and Herbelin (2000)). 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, has a direct analogy in the practice of sailing. Sailors give a very specific name to each rope (*main sail halyard*, *port jib sheet*, and so on), 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 $\mathscr{X}$, these colours naturally become *kinds*: just as a rope has a colour, a wire has a kind.

One specificity of $\mathscr{X}$ is that syntactic constructors bind *two* wires, one of each kind[†]. In $\mathscr{X}$, bound wires receive a hat, so, to show that x is bound, we write $\hat{x}$; note that in using the *'hat'*-notation, we are maintaining the old tradition of *Principia Mathematica* (Whitehead and Russell 1925; Whitehead and Russell 1997).

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, which is exposed and connectable. Frequently, but not invariably, this corresponds to the creation of the wire.

Contents of this paper

In this paper we will present the formal definitions for $\mathscr{X}$ through its 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 $\mathscr{X}$, in that we will define a system of derivable judgements for which the terms of $\mathscr{X}$ are witnesses, and will show a soundness result for this system by showing that a subject-reduction result holds.

We will also compare $\mathscr{X}$ 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 $\mathscr{X}$. For anyone familiar with the problem of expressiveness, in view of the fact that $\mathscr{X}$ is substitution-free, these result are truly novel. With the exception of $\bar{\lambda}\mu\tilde{\mu}$, it is not possible to embed $\mathscr{X}$ easily and naturally in these other calculi in such a way that the major properties are preserved. This can be easily understood from the fact that the vast majority of calculi in our area are confluent (Church–Rosser), whereas $\mathscr{X}$ is not.

A tool (which can be downloaded from `http://www.doc.ic.ac.uk/~jr200/X`) has been developed using the term graph rewriting technology, that allows users to not only input nets from $\mathscr{X}$, but also terms from λ-calculus, using the interpretation of the latter into $\mathscr{X}$ as specified in this paper. Details of the implementation can be found in van Bakel and Raghunandan (2005) and van Bakel and Raghunandan (2006).

[†] This is also the case in Urban (2000), Urban and Bierman (2001) and Urban (2001), but this fact is made very explicit in $\mathscr{X}$ by the use of *Principia*'s notation.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

This paper presents an extended version of results that first appeared in Lengrand (2003) and van Bakel *et al.* (2005), which were themselves deeply inspired by the work of Urban and Bierman (Urban 2000; Urban and Bierman 2001; Urban 2001).

2. The $\mathscr{X}$-calculus

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

2.1. *The operators*

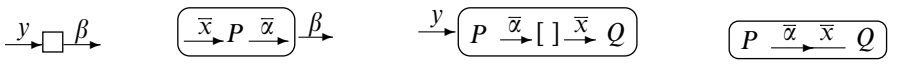
Nets are connected through *wires*, which are named. In our description wires are oriented. This means we know in which direction the 'ether running through our nets' moves, and can say when a wire provides an entrance to a net or when a wire provides an exit. Thus we make the distinction between exit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

When connecting two nets P and Q by an operator, say $\oplus$, we may suppose that P has a plug α and Q has a socket x that 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 names of the plug and socket are forgotten. To be more precise, in $P\hat{\alpha} \oplus \hat{x}Q$, the name α is not reachable outside P and the name x is not reachable outside Q. This is reminiscent of a construction like $\forall x.P$ or $\lambda x.M$ in logic where the name x is not known outside the expressions. These names are said to be *bound*. Likewise, in $\mathscr{X}$, in a construction like $P\hat{\alpha} \oplus \hat{x}Q$ 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 $\mathscr{X}$-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*:

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

As an illustration, we represent the basic nets diagrammatically as



Notice that, using the intuition sketched above, for example, the connector α is not supposed to 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 outputs on the right, as is also done in Wadler (2003).

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

We can motivate the constructions of $\mathscr{X}$ using the example of the 'translations' in a huge international organisation†. The arguments below will be better established and formalised in Section 3 when we will speak about *types*, which are the correct framework. But, ofcourse, $\mathscr{X}$ is basically an untyped language.

Suppose wires carry words in a language such as Estonian or Portuguese, but also translators between languages, such as 'French to Dutch'. A *capsule* $\langle y.\beta\rangle$ 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 in terms of *T-diagrams* like those used in compiling technology for bootstrapping – see Aho *et al.* (1988, 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 from different languages. In order for P and Q to communicate, a translator is needed, and this will be received on the wire y, which is 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*, which 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 is the same as 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 eliminated eventually by building the connection. The calculus, defined by the reduction rules (Section 2.2) gives a detailed explanation of how cuts are distributed through nets to be eventually erased at the level of capsules.

The following table gives the correspondence between the notation of $\mathscr{X}$ and that used by Urban. 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 angled brackets:

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

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

† See `http://europa.eu.int/comm/translation/index_en.htm`

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

We have already referred to the notion of bound names, and we will now give a formal definition together with definitions of free sockets and plugs into $\mathscr{X}$.

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

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

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

We will normally adopt Barendregt's convention (for Barendregt this was a convention on variables, 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 also consider, for example, x bound in $P[y/x]$ and $P \vdots \Gamma, x{:}A \vdash \Delta$ (this notion will be introduced in Section 3).

As the main concept is that of a name, we will only define renaming, that is, 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, that is, $\hat{y}P[y/x]$; 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).

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

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

2.2. *The rules*

We will now come to the definition of reduction through the elimination of cuts; the intuition in reduction is that the cut $P\widehat{\alpha}\dagger\widehat{x}Q$ 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 occur when a socket or a plug is *exposed and unique* (is *introduced*, or *connectable*), since in this case 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 that do not contain x as a 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 will now formally define what it means for a net to *introduce* a socket or a plug (Urban uses the terminology 'freshly introduce' (Urban 2000)).

Definition 2.4 (Introduction).

P *introduces* x means $P = \langle x.\beta\rangle$ or $P = R\widehat{\alpha}\,[x]\,\widehat{y}Q$, with $x \notin fs(R, Q)$.

P *introduces* β means $P = \langle y.\beta\rangle$ or $P = \widehat{x}Q\widehat{\alpha}\cdot\beta$, 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 that introduce connectors. These rules are naturally divided into four categories depending on whether:

— a capsule is cut with a capsule;
— an export is cut with a capsule;
— a capsule is cut with an import; or
— an export is cut 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). Assuming that the nets of the left-hand sides of the rules *introduce* the socket x and the plug α, the logical rules are

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

See Figure 1 for a diagrammatic representation.

Notice that in rule (*exp-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.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

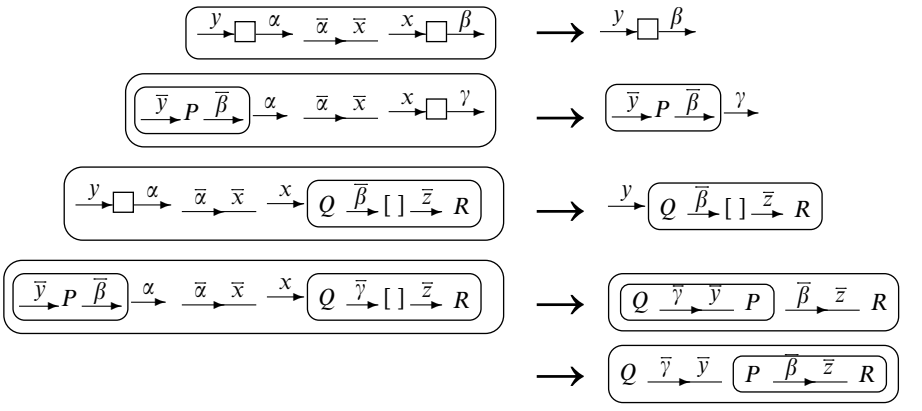


Fig. 1. Diagrammatic representation of logical reduction rules

Again in rule (*exp-imp*), one may observe that there are two right-hand sides: this is a first appearance of the non-determinism in $\mathscr{X}$, which is, as noted by Lafont, an intrinsic part of cut elimination in classical logic.

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

$$P \ ::= \ \ldots \mid P\widehat{\alpha} \not\curlywedge \widehat{x}Q \mid P\widehat{\alpha} \not\curlyvee \widehat{x}Q \, .$$

Intuitively, $P\widehat{\alpha} \not\curlywedge \widehat{x}Q$ 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, that is, is introduced, and $P\widehat{\alpha} \not\curlyvee \widehat{x}Q$ represents the intention to connect the αs to the xs.

Nets where cuts are not activated are called *pure* (the diagrammatical representation of activated cuts is the same as that for straightened out (that is, non-activated) cuts). Activated cuts are propagated through the nets, moving towards occurrences of the connectors mentioned in the cut up to the point where they are connectable, and a logical rule can be applied.

Definition 2.6 (Activating the cuts).

$$(act\text{-L}) : P\widehat{\alpha} \dagger \widehat{x}Q \ \rightarrow \ P\widehat{\alpha} \not\curlywedge \widehat{x}Q, \ \textit{if } P \textit{ does not introduce } \alpha$$
$$(act\text{-R}) : P\widehat{\alpha} \dagger \widehat{x}Q \ \rightarrow \ P\widehat{\alpha} \not\curlyvee \widehat{x}Q, \ \textit{if } Q \textit{ does not introduce } x$$

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

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Definition 2.7 (Propagation reduction). The rules of propagation are:

Left propagation

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

Right propagation

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

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 simply write $P \to Q$ for $P \twoheadrightarrow Q$, to show 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 $\not\swarrow$ and $\not\searrow$ inward. We will also write $P \downarrow Q$ if there exists an R such that $P \twoheadrightarrow R$ and $Q \twoheadrightarrow R$, that is, when P and Q have a common reduct.

The rules (*exp-outs*$\not\swarrow$) and ($\not\searrow$*imp-outs*) deserve some attention. Note that in rule (*exp-outs*$\not\swarrow$) we create a new name γ and that in rule ($\not\searrow$*imp-outs*) we create a new name z. This is done because a cut is duplicated: one copy 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-outs*$\not\swarrow$), 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 γ is then inactivated; this is because, although we now know that γ is introduced, we do not know if x in Q is.

The same thing happens with x in ($\not\searrow$*imp-outs*) and a new name z is created and the external cut is inactivated.

The Renaming Lemma (2.11, which is stated and proved in Section 2.4) shows that in the right-hand side of rules ($\not\swarrow d$) and ($d\not\searrow$) we could have written $P[y/x]$ and $P[\beta/\alpha]$, respectively, instead of the terms we have chosen. We made that choice for three reasons:

(1) We like to make all of the operations explicit and do not want to rely on operations defined in the meta-theory, and internalising name substitutions would have required us to include all the identities of Definition 2.3 as rules in the theory.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

(2) This small-step approach is closer to our philosophy of $\mathscr{X}$, which tends to decompose operations in as fine grained a way as possible.

(3) Note that if α is introduced in $\widehat{y}Q\widehat{\beta}\cdot\alpha$, then (*exp-outs*$\not\curlywedge$) gives, with the Cancellation Lemma (Lemma 2.10),

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

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

2.3. *Strong normalisation*

We should point out that when we use the above rules, 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 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 Urban (2000).

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

$$\begin{array}{lll}
P\widehat{\alpha} \dagger \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{z}Q) & \to & (act\text{-R}) \\
P\widehat{\alpha} \curlywedge \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{z}Q) & \to & (\curlywedge cut) \\
(P\widehat{\alpha} \curlywedge \widehat{x}\langle x.\beta\rangle)\widehat{\beta} \dagger \widehat{z}(P\widehat{\alpha} \curlywedge \widehat{x}Q) & \to & (d\curlywedge), (\curlywedge gc) \\
(P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle)\widehat{\beta} \dagger \widehat{z}Q & \to & (act\text{-L}) \\
(P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle)\widehat{\beta} \not\curlywedge \widehat{z}Q & \to & (cut \not\curlywedge) \\
(P\widehat{\beta} \not\curlywedge \widehat{z}Q)\widehat{\alpha} \dagger \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \not\curlywedge \widehat{z}Q) & \to & (\not\curlywedge d), (\not\curlywedge gc) \\
P\widehat{\alpha} \dagger \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{z}Q)\,. &&
\end{array}$$

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

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

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 that it is sufficient to obtain strong normalisation of typeable nets. He adds the rules

$$\begin{array}{lll}
(P\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle)\widehat{\beta} \not\curlywedge \widehat{y}Q & \to & (P\widehat{\beta} \not\curlywedge \widehat{y}Q)\widehat{\alpha} \dagger \widehat{y}Q \\
P\widehat{\alpha} \curlywedge \widehat{x}(\langle x.\beta\rangle\widehat{\beta} \dagger \widehat{y}Q) & \to & P\widehat{\alpha} \dagger \widehat{y}(P\widehat{\alpha} \curlywedge \widehat{x}Q)\,,
\end{array}$$

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

and gives them priority over the rules (*cut*↛) and (↚*cut*) by changing those to

$$\begin{array}{rcl}(P\widehat{\alpha}\dagger\widehat{x}Q)\widehat{\beta} \not\prec \widehat{y}R & \rightarrow & (P\widehat{\beta} \not\prec \widehat{y}R)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\beta} \not\prec \widehat{u}R),\ Q \neq \langle x.\beta\rangle \\ P\widehat{\alpha} \not\succ \widehat{x}(Q\widehat{\beta}\dagger\widehat{y}R) & \rightarrow & (P\widehat{\alpha} \not\succ \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \not\succ \widehat{x}R),\ Q \neq \langle x.\beta\rangle\,,\end{array}$$

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

Notice that the side condition $Q \neq \langle x.\beta\rangle$ is quite different in character from the rules for $\mathscr{X}$ we presented above, in that we now test for *equality* between circuits, rather than just the occurrence of a connector within a circuit. In fact, it corresponds to replacing, for example, rule (*cut*↛) by the rules

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

and by replacing rule (↚*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. Note that the last two rules might as well be replaced by

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

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

In this section we will define two sub-systems of reduction (that is, restricted versions of the full reduction we defined above), which correspond roughly to call-by-name (CBN) and call-by-value (CBV) reduction. Notice that this is essentially different from the approach of Wadler (2003), which only defines two dual notions of reduction, and no overall notion of reduction.

As in $\bar{\lambda}\mu\tilde{\mu}$, only one notion of reduction is defined in $\mathscr{X}$. When interpreting the λ-calculus or $\lambda\mu$ in $\bar{\lambda}\mu\tilde{\mu}$, 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 (*act*-L) and (*act*-R), can both be fired. The *critical pair* $\langle P\widehat{\alpha} \not\prec \widehat{x}Q, P\widehat{\alpha} \not\succ \widehat{x}Q\rangle$ 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 that favour one kind of activation whenever the above critical pair occurs.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Definition 2.9.

— If a cut can be activated in two ways, the CBV strategy only allows activation through (*act*-L), and we write $P \to_{\mathrm{V}} Q$ in that case. We can reformulate this as the reduction system obtained by replacing rule (*act*-R) by

$$(act\text{-R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \nwarrow \widehat{x}Q, \text{ if } P \text{ introduces } \alpha \text{ and } Q \text{ does not introduce } x.$$

— The CBN strategy can only activate such a cut through (*act*-R), and as above, we write $P \to_{\mathrm{N}} Q$. Again, we can reformulate this as the reduction system obtained by replacing rule (*act*-L) by

$$(act\text{-L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \swarrow \widehat{x}Q, \text{ if } P \text{ does not introduce } \alpha \text{ and } Q \text{ introduces } x.$$

Notice that in CBV, a right cut like $P\widehat{\alpha} \nwarrow \widehat{x}Q$ implies that α is introduced in P (that is, P is a *value*), and that, in CBN, a left cut like $P\widehat{\alpha} \swarrow \widehat{x}Q$ implies that x is introduced in Q (that is, Q is a *name*).

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

Lemma 2.10 (Cancellation).

1 $P\widehat{\alpha} \swarrow \widehat{x}Q \to_{\mathrm{A}} P$ if $\alpha \notin fp(P)$ and P is pure.
2 $P\widehat{\alpha} \dagger \widehat{x}Q \to P$ if $\alpha \notin fp(P)$ and P is pure.
3 $P\widehat{\alpha} \nwarrow \widehat{x}Q \to_{\mathrm{A}} Q$ if $x \notin fs(Q)$ and Q is pure.
4 $P\widehat{\alpha} \dagger \widehat{x}Q \to Q$ if $x \notin fs(Q)$ and Q is pure.

Proof.

1 We use induction on the structure of nets:

— $P = \langle y.\beta \rangle$:

$$\langle y.\beta \rangle \widehat{\alpha} \swarrow \widehat{x}Q \to_{\mathrm{A}} (cap\swarrow)\ \langle y.\beta \rangle .$$

— $P = \widehat{y}R\widehat{\beta}\cdot\gamma$:

$$(\widehat{y}R\widehat{\beta}\cdot\gamma)\widehat{\alpha} \swarrow \widehat{x}Q \to_{\mathrm{A}} (exp\text{-}ins\swarrow)\ \widehat{y}(R\widehat{\alpha} \swarrow \widehat{x}Q)\widehat{\beta}\cdot\gamma \to_{\mathrm{A}} (IH)\ \widehat{y}R\widehat{\beta}\cdot\gamma .$$

— $P = R\widehat{\beta}\,[z]\,\widehat{y}S$:

$$\begin{array}{ll} (R\widehat{\beta}\,[z]\,\widehat{y}S)\widehat{\alpha} \swarrow \widehat{x}Q & \to_{\mathrm{A}} (imp\swarrow) \\ (R\widehat{\alpha} \swarrow \widehat{x}Q)\widehat{\beta}\ [z]\ \widehat{y}(S\widehat{\alpha} \swarrow \widehat{x}Q) & \to_{\mathrm{A}} (IH)\ R\widehat{\beta}\,[z]\,\widehat{y}S . \end{array}$$

— $P = R\widehat{\beta} \dagger \widehat{y}S$:

$$\begin{array}{ll} (R\widehat{\beta} \dagger \widehat{y}S)\widehat{\alpha} \swarrow \widehat{x}Q & \to_{\mathrm{A}} (cut\swarrow) \\ (R\widehat{\alpha} \swarrow \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}(S\widehat{\alpha} \swarrow \widehat{x}Q) & \to_{\mathrm{A}} (IH)\ R\widehat{\beta} \dagger \widehat{y}S . \end{array}$$

2 There are two cases to consider:

— P introduces α:

Hence $\alpha \in fp(P)$, so this is impossible.

— P does not introduce α:

Hence $P\widehat{\alpha} \dagger \widehat{x}Q \to_{\mathrm{A}} (act\text{-L})\ P\widehat{\alpha} \swarrow \widehat{x}Q$, and the result follows from part 1.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

3 By induction on the structure of nets:

— $Q = \langle y.\beta\rangle$:

$$P\widehat{\alpha} \leftthreetimes \widehat{x}\langle y.\beta\rangle \to_{\mathrm{A}} (\leftthreetimes cap)\ \langle y.\beta\rangle\,.$$

— $Q = \widehat{y}R\widehat{\beta}\cdot\gamma$:

$$P\widehat{\alpha} \leftthreetimes \widehat{x}(\widehat{y}R\widehat{\beta}\cdot\gamma) \to_{\mathrm{A}} (\leftthreetimes exp)\ \widehat{y}(P\widehat{\alpha} \leftthreetimes \widehat{x}R)\widehat{\beta}\cdot\gamma \to_{\mathrm{A}} (IH)\ \widehat{y}R\widehat{\beta}\cdot\gamma\,.$$

— $Q = R\widehat{\beta}\,[z]\,\widehat{y}S$:

$$\begin{array}{ll} P\widehat{\alpha} \leftthreetimes \widehat{x}(R\widehat{\beta}\,[z]\,\widehat{y}S) & \to_{\mathrm{A}} (\leftthreetimes imp\text{-}ins) \\ (P\widehat{\alpha} \leftthreetimes \widehat{x}R)\widehat{\beta}\ [z]\ \widehat{y}(P\widehat{\alpha} \leftthreetimes \widehat{x}S) & \to_{\mathrm{A}} (IH)\ R\widehat{\beta}\,[z]\,\widehat{y}S\,. \end{array}$$

— $Q = R\widehat{\beta} \dagger \widehat{y}S$:

$$\begin{array}{ll} P\widehat{\alpha} \leftthreetimes \widehat{x}(R\widehat{\beta} \dagger \widehat{y}S) & \to_{\mathrm{A}} (\leftthreetimes cut) \\ (P\widehat{\alpha} \leftthreetimes \widehat{x}R)\widehat{\beta} \dagger \widehat{y}(P\widehat{\alpha} \leftthreetimes \widehat{x}S) & \to_{\mathrm{A}} (IH)\ R\widehat{\beta} \dagger \widehat{y}S\,. \end{array}$$

4 There are two cases to consider:

— Q introduces x:

Hence $x \in fs(Q)$, so this is impossible.

— Q does not introduce x:

Hence $P\widehat{\alpha} \dagger \widehat{x}Q \to_{\mathrm{A}} (act\text{-}\mathrm{R})\ P\widehat{\alpha} \leftthreetimes \widehat{x}Q$, and the result follows from Part 2.10. □

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

Lemma 2.11 (Renaming).

1 $P\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle \to P[\alpha/\delta]$ if P is pure.
2 $P\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle \to P[\alpha/\delta]$ if P is pure.
3 $\langle z.\alpha\rangle\widehat{\alpha} \leftthreetimes \widehat{x}P \to P[z/x]$ if P is pure.
4 $\langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}P \to P[z/x]$ if P is pure.

Proof.

1 By induction on the structure of nets:

— $P = \langle x.\delta\rangle$:

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

— $P = \langle x.\beta\rangle, \beta \neq \delta$:

$$\langle x.\beta\rangle\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle \to_{\mathrm{A}} (cap \nleftarrow)\ \langle x.\beta\rangle \stackrel{\Delta}{=} \langle x.\beta\rangle[\alpha/\delta]\,.$$

— $P = \widehat{y}Q\widehat{\gamma}\cdot\delta$:

$$\begin{array}{lll} (\widehat{y}Q\widehat{\gamma}\cdot\delta)\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle & \to_{\mathrm{A}} & (exp\text{-}outs \nleftarrow),\ \beta\ fresh \\ (\widehat{y}(Q\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle)\widehat{\gamma}\cdot\beta)\widehat{\beta} \dagger \widehat{z}\langle z.\alpha\rangle & \to & (exp) \\ \widehat{y}(Q\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle)\widehat{\gamma}\cdot\alpha & \to & (IH) \\ \widehat{y}Q[\alpha/\delta]\widehat{\gamma}\cdot\alpha & \stackrel{\Delta}{=} & (\widehat{y}Q\widehat{\gamma}\cdot\delta)[\alpha/\delta]\,. \end{array}$$

— $P = \widehat{y}Q\widehat{\gamma}\cdot\beta, \beta \neq \delta$:

$$\begin{array}{lll} (\widehat{y}Q\widehat{\gamma}\cdot\beta)\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle & \to_{\mathrm{A}} & (exp\text{-}ins \nleftarrow) \\ \widehat{y}(Q\widehat{\delta} \nleftarrow \widehat{z}\langle z.\alpha\rangle)\widehat{\gamma}\cdot\beta & \to & (IH) \\ \widehat{y}Q[\alpha/\delta]\widehat{\gamma}\cdot\beta & \stackrel{\Delta}{=} & (\widehat{y}Q\widehat{\gamma}\cdot\beta)[\alpha/\delta]\,. \end{array}$$

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

— $P = Q\widehat{\beta}\,[x]\,\widehat{y}R$:

$$\begin{array}{lll}(Q\widehat{\beta}\,[x]\,\widehat{y}R)\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle & \rightarrow_{\mathrm{A}} & (imp\not\leftarrow)\\ (Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle)\widehat{\beta}\ [x]\ \widehat{y}(R\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle) & \rightarrow & (IH)\\ Q[\alpha/\delta]\widehat{\beta}\ [x]\ \widehat{y}R[\alpha/\delta] & \stackrel{\Delta}{=} & (Q\widehat{\beta}\,[x]\,\widehat{y}R)[\alpha/\delta]\,.\end{array}$$

— $P = Q\widehat{\beta} \dagger \widehat{x}R$:

$$\begin{array}{lll}(Q\widehat{\beta} \dagger \widehat{x}R)\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle & \rightarrow_{\mathrm{A}} & (cut\not\leftarrow)\\ (Q\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle)\widehat{\beta} \dagger \widehat{x}(R\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle) & \rightarrow & (IH)\\ Q[\alpha/\delta]\widehat{\beta} \dagger \widehat{x}R[\alpha/\delta] & \stackrel{\Delta}{=} & (Q\widehat{\beta} \dagger \widehat{x}R)[\alpha/\delta]\,.\end{array}$$

2 If $P\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle$, there are two cases:

— P introduces δ:

Hence either:

– $P = \widehat{x}Q\widehat{\beta}\cdot\delta$ and $\delta \notin fp(Q)$:

So

$$\begin{array}{lll}(\widehat{x}Q\widehat{\beta}\cdot\delta)\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle & \rightarrow & (exp)\\ \widehat{x}Q\widehat{\beta}\cdot\alpha & \stackrel{\Delta}{=} & (\widehat{x}Q\widehat{\beta}\cdot\delta)[\alpha/\delta]\,.\end{array}$$

– $P = \langle x.\delta\rangle$:

So $\langle x.\delta\rangle\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle \rightarrow (cap)\ \langle x.\alpha\rangle \stackrel{\Delta}{=} \langle x.\delta\rangle[\alpha/\delta]$.

— P does not introduce δ:

Hence, $P\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle \rightarrow_{\mathrm{A}} P\widehat{\delta} \not\leftarrow \widehat{z}\langle z.\alpha\rangle$, and the result follows from part 1.

3 By induction on the structure of nets:

— $P = \langle x.\delta\rangle$:

$$\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}\langle x.\delta\rangle \rightarrow_{\mathrm{A}} (d\nwarrow)\ \langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\delta\rangle \rightarrow (cap)\ \langle z.\delta\rangle \stackrel{\Delta}{=} \langle x.\delta\rangle[z/x]\,.$$

— $P = \langle y.\delta\rangle, y \neq x$:

$$\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}\langle y.\delta\rangle \rightarrow_{\mathrm{A}} (\nwarrow cap)\ \langle y.\delta\rangle \stackrel{\Delta}{=} \langle y.\delta\rangle[z/x]\,.$$

— $P = \widehat{y}Q\widehat{\gamma}\cdot\delta$:

$$\begin{array}{lll}\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}(\widehat{y}Q\widehat{\gamma}\cdot\delta) & \rightarrow_{\mathrm{A}} & (\nwarrow exp)\\ \widehat{y}(\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\gamma}\cdot\delta & \rightarrow & (IH)\\ \widehat{y}Q[z/x]\widehat{\gamma}\cdot\delta & \stackrel{\Delta}{=} & (\widehat{y}Q\widehat{\gamma}\cdot\delta)[z/x]\,.\end{array}$$

— $P = Q\widehat{\beta}\,[x]\,\widehat{y}R$:

$$\begin{array}{lll}\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}\,[x]\,\widehat{y}R) & \rightarrow_{\mathrm{A}} & (\nwarrow imp\text{-}outs)\\ \langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{v}((\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\beta}\ [v]\ \widehat{y}(\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}R)) & \rightarrow & (IH)\\ \langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{v}(Q[z/x]\widehat{\beta}\ [v]\ \widehat{y}R[z/x]) & \rightarrow & (imp)\\ Q[z/x]\widehat{\beta}\ [z]\ \widehat{y}R[z/x] & \stackrel{\Delta}{=} & (Q\widehat{\beta}\,[x]\,\widehat{y}R)[z/x]\,.\end{array}$$

— $P = Q\widehat{\beta}\,[v]\,\widehat{y}R, v \neq x$:

$$\begin{array}{lll}\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta}\,[v]\,\widehat{y}R) & \rightarrow_{\mathrm{A}} & (\nwarrow imp\text{-}ins)\\ (\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\beta}\ [v]\ \widehat{y}(\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}R) & \rightarrow & (IH)\\ Q[z/x]\widehat{\beta}\ [v]\ \widehat{y}R[z/x] & \stackrel{\Delta}{=} & (Q\widehat{\beta}\,[v]\,\widehat{y}R)[z/x]\,.\end{array}$$

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

— $P = Q\widehat{\beta} \dagger \widehat{y}R$:

$$\begin{array}{lll} \langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}(Q\widehat{\beta} \dagger \widehat{y}R) & \rightarrow_{\mathrm{A}} & (\nwarrow cut) \\ (\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}Q)\widehat{\beta} \dagger \widehat{y}(\langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}R) & \rightarrow & (IH) \\ Q[z/x]\widehat{\beta} \dagger \widehat{y}R[z/x] & \stackrel{\Delta}{=} & (Q\widehat{\beta} \dagger \widehat{y}R)[z/x]\,. \end{array}$$

4 If $\langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}P$, there are two cases:

— P introduces x:

Hence, either:

– $P = Q\widehat{\beta}\,[x]\,\widehat{y}R$, and x does not occur free in Q, R:
So $\langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\beta}\,[x]\,\widehat{y}R) \rightarrow (imp)\ Q\widehat{\beta}\,[z]\,\widehat{y}R \stackrel{\Delta}{=} (Q\widehat{\beta}\,[x]\,\widehat{y}R)[z/x]$.

– $P = \langle x.\beta\rangle$:
So $\langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \rightarrow (cap)\ \langle z.\beta\rangle \stackrel{\Delta}{=} \langle x.\beta\rangle[z/x]$.

— P does not introduce x:
Hence, $\langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}P \rightarrow_{\mathrm{A}} \langle z.\alpha\rangle\widehat{\alpha} \nwarrow \widehat{x}P$ and the result follows from part 1. □

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

$$\begin{array}{rll} (\nearrow gc): & P\widehat{\alpha} \dagger \widehat{x}Q \rightarrow P, & \text{if } \alpha \notin fp(P),\ P \text{ pure} \\ (\nwarrow gc): & P\widehat{\alpha} \dagger \widehat{x}Q \rightarrow Q, & \text{if } x \notin fs(Q),\ Q \text{ pure} \\ (ren\text{-R}): & P\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle \rightarrow P[\alpha/\delta], & P \text{ pure} \\ (ren\text{-L}): & \langle z.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}P \rightarrow P[z/x], & P \text{ pure}. \end{array}$$

The admissibility of these rules for nets that are *not* pure is shown in van Bakel and Raghunandan (2006).

2.5. *α-conversion*

Renaming is normally 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 van Bakel and Raghunandan (2006), which deals precisely with this problem in $\mathscr{X}$.

Example 2.12. Take the following reduction:

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

It is clear that the last term contravenes Barendregt's convention: ρ is both free and bound in $(\widehat{y}\langle y.\rho\rangle\widehat{\rho}\cdot\delta)\widehat{\delta} \dagger \widehat{y}\langle y.\rho\rangle$. If we were to continue the reduction, we would get

$$((\widehat{y}\langle y.\rho\rangle\widehat{\rho}\cdot\delta)\widehat{\delta} \dagger \widehat{y}\langle y.\rho\rangle)\widehat{\rho} \dagger \widehat{w}\langle w.\alpha\rangle \;\to\; (exp)$$
$$(\widehat{y}\langle y.\rho\rangle\widehat{\rho}\cdot\rho)\widehat{\rho} \dagger \widehat{w}\langle w.\alpha\rangle\,.$$

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

$$\to\; (act\text{-}\mathrm{L}), (exp\text{-}outs\not\prec), (\not\prec d)$$
$$(\widehat{y}(\langle y.\rho\rangle\widehat{\rho} \dagger \widehat{w}\langle w.\alpha\rangle)\widehat{\rho}\cdot\theta)\widehat{\theta} \dagger \widehat{w}\langle w.\alpha\rangle \;\to\; (cap), (exp)$$
$$\widehat{y}\langle y.\alpha\rangle\widehat{\rho}\cdot\alpha\,.$$

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

It is clear from this example that α-conversion is needed to some extent in any implementation of $\mathscr{X}$. Three[†] solutions to this problem are proposed in van Bakel and Raghunandan (2005; 2006), and these 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 we may have $\beta = \gamma$ 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 $\mathscr{X}$ 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 $\mathscr{X}$ (see below).

In contrast, this is impossible in the λ-calculus, where 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 innermost binder, which causes variables bound by the outermost 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, which 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 $\lambda y.xy[(\lambda xy.xy)/x]$;

[†] Note that De Bruijn indices are not discussed in van Bakel and Raghunandan (2006).

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

the actual carrying out of the substitution, which *brings* the right-most binder under the left-most is not part of the reduction 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

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

which prevents 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 $\mathscr{X}$ however, not only is the α-conflict solved, but also detected, all within the reduction system of $\mathscr{X}$ itself. This is achieved, essentially, by 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 van Bakel and Raghunandan (2006), in order to deal accurately with α-conversion for the case of capture-avoidance, the rule (*exp-imp*) needs to be replaced by (assuming that α, x are introduced, and that v, δ are fresh):

$$\begin{array}{lll}
(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta}\dagger\widehat{z}R), & y \notin fs(R) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to Q\widehat{\gamma} \dagger \widehat{v}((\langle v.\delta\rangle\widehat{\delta} \nwarrow \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R), & y \in fs(R) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to (Q\widehat{\gamma}\dagger\widehat{y}P)\widehat{\beta} \dagger \widehat{z}R, & \beta \notin fp(Q) \\
(\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to (Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \nearrow \widehat{v}\langle v.\delta\rangle))\widehat{\delta} \dagger \widehat{z}R, & \beta \in fp(Q)\,.
\end{array}$$

We must also deal with almost all of the propagation rules (the exceptions are $(\nearrow d)$, $(cap \nearrow)$, $(d \nwarrow)$, and $(\nwarrow cap)$).

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

As we mentioned in the introduction, $\mathscr{X}$ is inspired by the sequent calculus, so it is worthwhile recalling some of the principles. The sequent calculus we consider just has 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 the four rules *axiom*, *right introduction* of the arrow, *left introduction* and *cut*:

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

$$(\Rightarrow R):\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta} \qquad (cut):\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,.$$

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 were proposed by Gentzen, and these reductions become the fundamental principle of computation in $\mathscr{X}$.

Another nice property of proof systems is known as the *Curry–Howard correspondence*.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Definition 3.1 (Curry–Howard isomorphism). '*Terms as proofs; types as propositions.*' If M is a (closed) term and A is 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 whose 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 $\mathscr{X}$). This phenomenon was first discovered for combinatory logic (Curry *et al.* 1958), and later the connection between the λ-calculus and intuitionistic logic was put in evidence. The Curry–Howard correspondence for $\mathscr{X}$ is with the classical propositional calculus and is given through the sequent calculus described above. Propositions are given names: those that appear in the left part of a sequent are given names like x, y, z, and those that appear in the right part of a sequent are given names like α, β, γ.

Definition 3.2 (Types and contexts).

1 The set of types is defined by the grammar

$$A, B \ ::= \ \varphi \mid A \rightarrow B$$

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

2 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

$$\begin{aligned} \Gamma, x{:}A \ &= \ \Gamma \cup \{x{:}A\}, \ \text{if } \Gamma \text{ is not defined on } x \\ &= \ \Gamma, \qquad\qquad \text{otherwise.} \end{aligned}$$

(Note that the second case implies that $x{:}A \in \Gamma$.) So, writing a context as $\Gamma, x{:}A$ implies $x{:}A \in \Gamma$ 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 $A_1 = A_2$).

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

The notion of type assignment on $\mathscr{X}$ 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 $\mathscr{X}$).

1 *Type judgements* are expressed through a ternary relation $P :\cdot \Gamma \vdash \Delta$, where Γ is a context of *sockets*, Δ is a context of *plugs* and P is a net. We say that P is the *witness* of this judgement.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

2 *Type assignment for* $\mathscr{X}$ is defined by the following sequent calculus:

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

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

We write $P \because \Gamma \vdash \Delta$ if there exists a derivation that has this judgement in the bottom line, and write $\mathrm{D} :: P \because \Gamma \vdash \Delta$ 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:

$$\frac{P \because \Gamma \vdash \Delta}{P \because \Gamma' \vdash \Delta'}\,(W)$$

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

Proof. The proof is by induction on the proof tree for $P \because \Gamma \vdash \Delta$. We will only consider two cases, the other two work in the same way.

(*cap*):

So $P \equiv \langle y.\alpha\rangle$. Hence, $\Gamma' \supseteq \Gamma \supseteq \{y{:}A\}$ and $\Delta' \supseteq \Delta \supseteq \{\alpha{:}A\}$, so $\langle y.\alpha\rangle \because \Gamma' \vdash \Delta'$.

(*imp*):

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

$$P \because \Gamma \vdash \Delta \equiv Q\widehat{\alpha}\,[y]\,\widehat{x}R \because \Gamma_1, y{:}A{\to}B \vdash \Delta.$$

We have $Q \because \Gamma_1 \vdash \alpha{:}A, \Delta$ and $R \because \Gamma_1, x{:}B \vdash \Delta$. If $y{:}A{\to}B \in \Gamma_1$, that is, $\Gamma \equiv \Gamma_1$, we write $\Gamma'_1 \equiv \Gamma'$, otherwise we write $\Gamma'_1 \equiv \Gamma' \setminus y{:}A{\to}B$. Notice that $\Gamma'_1 \supseteq \Gamma_1$. By induction, we have $Q \because \Gamma'_1 \vdash \alpha{:}A, \Delta'$ and $R \because \Gamma'_1, x{:}B \vdash \Delta'$, and $Q\widehat{\alpha}\,[y]\,\widehat{x}R \because \Gamma'_1, y{:}A{\to}B \vdash \Delta'$ follows by (*imp*). □

Example 3.5 (A proof of Peirce's Law). The following is a proof of Peirce's Law in classical logic:

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

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

The soundness result of simple type assignment with respect to reduction is stated as usual in the following way.

Theorem 3.6 (Witness reduction). If $P :\cdot\ \Gamma \vdash \Delta$ and $P \to Q$, then $Q :\cdot\ \Gamma \vdash \Delta$.

Proof. We use induction on the length of reduction sequences. We will only show the interesting base cases; note that weakening will be used occasionally.

Logical rules:

(cap): $\quad \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \to \langle y.\beta\rangle$

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

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

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

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

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

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

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

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

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

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

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

$\not$ propagation

(*exp-outs$\not$*): $(\widehat{y}Q\widehat{\beta}\cdot\alpha)\widehat{\alpha} \not\leftarrow \widehat{x}P \to (\widehat{y}(Q\widehat{\alpha} \not\leftarrow \widehat{x}P)\widehat{\beta}\cdot\gamma)\widehat{\gamma} \dagger \widehat{x}P$

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

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

(*imp$\not$*): $(Q\widehat{\beta}\,[z]\,\widehat{y}R)\widehat{\alpha} \not\leftarrow \widehat{x}P \to (Q\widehat{\alpha} \not\leftarrow \widehat{x}P)\widehat{\beta}\;\;[z]\;\widehat{y}(R\widehat{\alpha} \not\leftarrow \widehat{x}P)$

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

$$\dfrac{\dfrac{\dfrac{\boxed{\mathrm{D}_2}}{R :\cdot \Gamma, y{:}B \vdash \alpha{:}C, \Delta} \quad \dfrac{\boxed{\mathrm{D}_3}}{P :\cdot \Gamma, z{:}A{\to}B, x{:}C \vdash \Delta}}{R :\cdot \Gamma, z{:}A{\to}B, y{:}B \vdash \alpha{:}C, \Delta} \quad \dfrac{\boxed{\mathrm{D}_3}}{P :\cdot \Gamma, z{:}A{\to}B, x{:}C, y{:}B \vdash \Delta}}{R\widehat{\alpha} \not\leftarrow \widehat{x}P :\cdot \Gamma, z{:}A{\to}B, y{:}B \vdash \Delta} \quad \vdots$$

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

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

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

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

$\not\leftarrow$ propagation

$(\not\leftarrow imp\text{-}outs)$: $P\widehat{\alpha} \not\leftarrow \widehat{x}(Q\widehat{\beta}\ [x]\ \widehat{y}R) \to P\widehat{\alpha} \dagger \widehat{z}((P\widehat{\alpha} \not\leftarrow \widehat{x}Q)\widehat{\beta}\ [z]\ \widehat{y}(P\widehat{\alpha} \not\leftarrow \widehat{x}R))$

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

$$\dfrac{\dfrac{\boxed{\mathrm{D}_1}}{P :\cdot \Gamma \vdash \alpha{:}A{\to}B, \Delta} \quad \dfrac{\boxed{\mathrm{D}_3}}{R :\cdot \Gamma, y{:}B \vdash \Delta}}{P :\cdot \Gamma, y{:}B \vdash \alpha{:}A{\to}B, \Delta} \quad \dfrac{\boxed{\mathrm{D}_3}}{R :\cdot \Gamma, x{:}A{\to}B, y{:}B \vdash \Delta}}{P\widehat{\alpha} \not\leftarrow \widehat{x}R :\cdot \Gamma, y{:}B \vdash \Delta}$$

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

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

$(\curlywedge cut)$: $\quad P\widehat{\alpha}\curlywedge\widehat{x}(Q\widehat{\beta}\dagger\widehat{y}R) \to (P\widehat{\alpha}\curlywedge\widehat{x}Q)\widehat{\beta}\dagger\widehat{y}(P\widehat{\alpha}\curlywedge\widehat{x}R)$

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

$$\dfrac{\dfrac{\dfrac{\boxed{\mathrm{D}_1}}{P \mathrel{:\cdot} \Gamma \vdash \alpha{:}A, \Delta} \quad \boxed{\mathrm{D}_2}}{P \mathrel{:\cdot} \Gamma \vdash \alpha{:}A, \beta{:}B, \Delta \qquad Q \mathrel{:\cdot} \Gamma, x{:}A \vdash \beta{:}B, \Delta}}{P\widehat{\alpha}\curlywedge\widehat{x}Q \mathrel{:\cdot} \Gamma \vdash \beta{:}B, \Delta} \quad \dfrac{\dfrac{\boxed{\mathrm{D}_1}}{P \mathrel{:\cdot} \Gamma \vdash \alpha{:}A, \Delta} \quad \boxed{\mathrm{D}_3}}{R \mathrel{:\cdot} \Gamma, y{:}B \vdash \alpha{:}A, \Delta \qquad P \mathrel{:\cdot} \Gamma, x{:}A, y{:}B \vdash \Delta}}{P\widehat{\alpha}\curlywedge\widehat{x}R \mathrel{:\cdot} \Gamma, y{:}B \vdash \Delta}$$

$$\dfrac{P\widehat{\alpha}\curlywedge\widehat{x}Q \mathrel{:\cdot} \Gamma \vdash \beta{:}B, \Delta \qquad P\widehat{\alpha}\curlywedge\widehat{x}R \mathrel{:\cdot} \Gamma, y{:}B \vdash \Delta}{(P\widehat{\alpha}\curlywedge\widehat{x}Q)\widehat{\beta}\dagger\widehat{y}(P\widehat{\alpha}\curlywedge\widehat{x}R) \mathrel{:\cdot} \Gamma \vdash \Delta}$$

□

4. Interpreting the λ-calculus

In this section, we will illustrate the expressive power of $\mathscr{X}$ by showing that we can faithfully interpret the λ-calculus (Church 1940; Barendregt 1984) – a similar result was shown in Urban (2000). 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 Curien and Herbelin (2000). Indeed, we could have defined our mappings using the mappings of λ-calculus and $\lambda\mu$ into $\overline{\lambda}\mu\tilde{\mu}$, and then concatenating these with the mapping from $\overline{\lambda}\mu\tilde{\mu}$ to $\mathscr{X}$, but our encoding is more detailed and precise than that and also deals with explicit substitution.

One should note that for Curien and Herbelin (2000) the preservation of the CBV and CBN evaluations relies on two *distinct* translations of terms. For instance, the CBV and CBN λ-calculi can both be encoded into CPS (Appel and Jim 1989), 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 $\mathscr{X}$ we have no need for two separate interpretation functions, and we will only define *one*. Combining this with the two sub-reduction systems $\to_{\mathrm{V}}$ and $\to_{\mathrm{N}}$, we can encode the CBV and CBN λ-calculi.

We will assume familiarity with the λ-calculus (Barendregt 1984), and just recall the definition of lambda terms and β-contraction.

Definition 4.1 (Lambda terms and β-contraction (Barendregt 1984)).

1 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*.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

2 The reduction relation $\rightarrow_\beta$ is defined as the contextual (that is, compatible (Barendregt 1984)) closure of the rule

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

3 If the contraction $(\lambda x.M)N \rightarrow_\beta M[N/x]$ is fired only when N is a value, the reduction is called *call-by-value* (or CBV for short) and written $\rightarrow_{\mathrm{v}}$. No confusion is possible with the reduction with the same name in $\mathscr{X}$, 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 proposition on the right-hand side of the sequent.

For instance, the implication is introduced by the rule

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} (\Rightarrow I)$$

and eliminated by *modus ponens*

$$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} (\Rightarrow E).$$

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

$$\frac{}{\Gamma \vdash A} (A \in \Gamma).$$

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

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

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

(*cf.* Gentzen (1935)). 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 (van Dalen 1994). In that case the proof can be easily transformed into a simpler one, by

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

the process of cut-*elimination*. For instance,

$$\dfrac{\dfrac{\boxed{\mathrm{D}_1}}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B} \qquad \boxed{\mathrm{D}_2} \quad \Gamma \vdash A$$
$$\Gamma \vdash B$$

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 $\Gamma \vdash A$.

$$\begin{array}{c}\boxed{\mathrm{D}_2}\\ \Gamma \vdash A\\ \boxed{\mathrm{D}_1}\\ \Gamma \vdash B\end{array}$$

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, furthermore, 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).

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

$$(\to E) : \frac{\Gamma \vdash_\lambda M : A \to B \qquad \Gamma \vdash_\lambda N : A}{\Gamma \vdash_\lambda MN : B}$$

We will begin by defining the direct encoding of the λ-calculus into $\mathscr{X}$.

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

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

Observe that every sub-net of $[\![M]\!]^\lambda_\alpha$ has exactly one free plug, and that this is precisely α. Moreover, note that the output (that is, result) is anonymous in the λ-calculus: the destination an operand 'moves' to carries a name through a variable, but where it comes from is not mentioned, as it is implicit. Since a net is allowed to return a result in more than one way in $\mathscr{X}$, in order for us to connect outputs to inputs, we have to name the outputs, and this forces a name on the output of an interpreted λ-term M too. This name, say α, is carried in the sub-script of $[\![M]\!]^\lambda_\alpha$, and is also the name of the current continuation, that is, the name of the hole in the context in which M occurs.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

A similar interpretation is defined in Urban (2000), but it differs in the final case, where it states

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

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

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

Urban's interpretation actually ignores certain reduction paths that are accessible from our interpretation by right-activating the cut, and if we used Urban's interpretation, some of our results below could not be achieved; in particular, when using Urban's interpretation, it is not possible to show that CBN reduction is modelled.

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, we would need to prove a substitution lemma, which would give a much more involved proof than the one we achieve in Theorem 4.8. Moreover, it is possible to show that

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

In all, we feel our choice is justified.

It is worth noting that the interpretation function $\llbracket \cdot \rrbracket^{\lambda}_{\alpha}$ does not generate a confluent sub-calculus. We illustrate this with the following example.

Example 4.4. First note that

$$\begin{aligned} \llbracket xx \rrbracket^{\lambda}_{\alpha} \triangleq \; & \llbracket x \rrbracket^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{z}(\llbracket x \rrbracket^{\lambda\widehat{\beta}}_{\beta} \, [z] \, \widehat{y}\langle y.\alpha\rangle) \triangleq \\ & \langle x.\gamma\rangle\widehat{\gamma} \dagger \widehat{z}(\langle x.\beta\rangle\widehat{\beta} \, [z] \, \widehat{y}\langle y.\alpha\rangle) \to (\textit{imp}) \\ & \langle x.\beta\rangle\widehat{\beta} \, [x] \, \widehat{y}\langle y.\alpha\rangle \, . \end{aligned}$$

Moreover, notice that the above (*imp*) reduction is the only possible one, making the reduction deterministic. So we can write

$$\llbracket xx \rrbracket^{\lambda}_{\alpha} \to \langle x.\beta\rangle\widehat{\beta} \, [x] \, \widehat{y}\langle y.\alpha\rangle \, .$$

Now

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

$$
\begin{array}{ll}
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{x}[\![xx]\!]^{\lambda}_{\alpha} & \to (act\text{-L}) \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \not\leftarrow \widehat{x}[\![xx]\!]^{\lambda}_{\alpha} & \stackrel{\Delta}{=} \\
(\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\gamma\rangle)\widehat{\gamma} \not\leftarrow \widehat{x}[\![xx]\!]^{\lambda}_{\alpha} & \to (imp\not\leftarrow) \\
(\langle y.\sigma\rangle\widehat{\gamma} \not\leftarrow \widehat{x}[\![xx]\!]^{\lambda}_{\alpha})\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \not\leftarrow \widehat{x}[\![xx]\!]^{\lambda}_{\alpha}) & \to (cap\not\leftarrow), (\not\leftarrow d), (act\text{-R}) \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \nleftarrow \widehat{x}[\![xx]\!]^{\lambda}_{\alpha}) & \stackrel{\Delta}{=} \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \nleftarrow \widehat{x}(\langle x.\tau\rangle\widehat{\tau}\ [x]\ \widehat{u}\langle u.\alpha\rangle)) & \to (\nleftarrow imp\text{-}outs) \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \dagger \widehat{v}((\langle z.\gamma\rangle\widehat{\gamma} \nleftarrow \widehat{x}\langle x.\tau\rangle)\widehat{\tau}\ [v]\ \widehat{u}(\langle z.\gamma\rangle\widehat{\gamma} \nleftarrow \widehat{x}\langle u.\alpha\rangle))) & \\
\qquad\qquad & \to (d\nleftarrow), (\nleftarrow cap), (cap) \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \dagger \widehat{v}(\langle z.\tau\rangle\widehat{\tau}\ [v]\ \widehat{u}\langle u.\alpha\rangle)) & \to (imp) \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\tau\rangle\widehat{\tau}\ [z]\ \widehat{u}\langle u.\alpha\rangle) &
\end{array}
$$

or

$$
\begin{array}{ll}
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{x}[\![xx]\!]^{\lambda}_{\alpha} & \to (act\text{-R}) \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \nleftarrow \widehat{x}[\![xx]\!]^{\lambda}_{\alpha} & \stackrel{\Delta}{=} \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \nleftarrow \widehat{x}(\langle x.\tau\rangle\widehat{\tau}\ [x]\ \widehat{u}\langle u.\alpha\rangle) & \to (\nleftarrow imp\text{-}outs) \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{w}((\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \nleftarrow \widehat{x}\langle x.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}([\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \nleftarrow \widehat{x}\langle u.\alpha\rangle)) & \to (d\nleftarrow), (act\text{-L}), (\nleftarrow cap) \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{w}((\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \not\leftarrow \widehat{x}\langle x.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle) & \stackrel{\Delta}{=} \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{w}(((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\gamma\rangle)\widehat{\gamma} \not\leftarrow \widehat{x}\langle x.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle) & \to (imp\not\leftarrow) \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{w}(((\langle y.\sigma\rangle\widehat{\gamma} \not\leftarrow \widehat{x}\langle x.\tau\rangle)\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \not\leftarrow \widehat{x}\langle x.\tau\rangle))\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle) & \\
\qquad\qquad & \to (cap\not\leftarrow), (\not\leftarrow d), (cap), (act\text{-L}) \\
[\![yy]\!]^{\lambda}_{\gamma}\widehat{\gamma} \not\leftarrow \widehat{w}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle) & \stackrel{\Delta}{=} \\
(\langle y.\sigma\rangle\widehat{\gamma} \not\leftarrow \widehat{w}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\gamma\rangle)\widehat{\gamma} \not\leftarrow \widehat{w}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle) & \to (imp\not\leftarrow) \\
(\langle y.\sigma\rangle\widehat{\gamma} \not\leftarrow \widehat{w}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle))\widehat{\sigma}\ [y] & \\
\qquad \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \not\leftarrow \widehat{w}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle)) & \to (cap\not\leftarrow), (\not\leftarrow d), \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}(\langle z.\gamma\rangle\widehat{\gamma} \dagger \widehat{w}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\tau\rangle)\widehat{\tau}\ [w]\ \widehat{u}\langle u.\alpha\rangle)) & \to (imp) \\
\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}((\langle y.\sigma\rangle\widehat{\sigma}\ [y]\ \widehat{z}\langle z.\tau\rangle)\widehat{\tau}\ [z]\ \widehat{u}\langle u.\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.

1 $[\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nleftarrow \widehat{x}[\![M]\!]^{\lambda}_{\alpha} \to [\![M[N/x]]\!]^{\lambda}_{\alpha}$.
2 $[\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \dagger \widehat{x}[\![M]\!]^{\lambda}_{\alpha} \to [\![M[N/x]]\!]^{\lambda}_{\alpha}$.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Proof.

1 By induction on the structure of lambda terms:

$M = x$:

$$\begin{array}{l} [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![x]\!]^{\lambda}_{\alpha} \triangleq [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}\langle x.\alpha\rangle \;\rightarrow\; (\mathit{ren}\text{-}\textsc{l}) \\ [\![N]\!]^{\lambda}_{\delta}[\alpha/\delta] \triangleq [\![N]\!]^{\lambda}_{\alpha} \triangleq [\![x[N/x]]\!]^{\lambda}_{\alpha}. \end{array}$$

$M = y \neq x$

$$\begin{array}{l} [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![y]\!]^{\lambda}_{\alpha} \triangleq [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}\langle y.\alpha\rangle \;\rightarrow\; (\nwarrow \mathit{cap}) \\ \langle y.\alpha\rangle \triangleq [\![y]\!]^{\lambda}_{\alpha} \triangleq [\![y[N/x]]\!]^{\lambda}_{\alpha}. \end{array}$$

$M = \lambda y.M'$

$$\begin{array}{lll} [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![\lambda y.M']\!]^{\lambda}_{\alpha} & \triangleq & \\ [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}(\widehat{y}[\![M']\!]^{\lambda}_{\beta}\widehat{\beta}\cdot\alpha) & \rightarrow & (\nwarrow \mathit{exp}) \\ \widehat{y}([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![M']\!]^{\lambda}_{\beta})\widehat{\beta}\cdot\alpha & \rightarrow & (\mathit{IH}) \\ \widehat{y}[\![M'[N/x]]\!]^{\lambda}_{\beta}\widehat{\beta}\cdot\alpha & \triangleq & [\![(\lambda y.M')[N/x]]\!]^{\lambda}_{\alpha}. \end{array}$$

$M = PQ$

$$\begin{array}{lll} [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![PQ]\!]^{\lambda}_{\alpha} & \triangleq & \\ [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}([\![P]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{z}([\![Q]\!]^{\lambda}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\alpha\rangle)) & \rightarrow & (\nwarrow \mathit{cut}) \\ ([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![P]\!]^{\lambda}_{\gamma})\widehat{\gamma} \dagger \widehat{z}([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}([\![Q]\!]^{\lambda}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\alpha\rangle)) & \rightarrow & (\nwarrow \mathit{imp\text{-}ins}) \\ ([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![P]\!]^{\lambda}_{\gamma})\widehat{\gamma} \dagger \widehat{z}(([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![Q]\!]^{\lambda}_{\beta})\widehat{\beta}\ [z]\ \widehat{y}([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}\langle y.\alpha\rangle)) & & \\ & \rightarrow & (\nwarrow \mathit{cap}) \\ ([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![P]\!]^{\lambda}_{\gamma})\widehat{\gamma} \dagger \widehat{z}(([\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![Q]\!]^{\lambda}_{\beta})\widehat{\beta}\ [z]\ \widehat{y}\langle y.\alpha\rangle) & \rightarrow & (\mathit{IH}) \\ [\![P[N/x]]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{z}([\![Q[N/x]]\!]^{\lambda}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\alpha\rangle) & \triangleq & \\ [\![P[N/x]Q[N/x]]\!]^{\lambda}_{\alpha} & \triangleq & [\![(PQ)[N/x]]\!]^{\lambda}_{\alpha}. \end{array}$$

2 There are two cases to consider:

— $[\![M]\!]^{\lambda}_{\alpha}$ introduces x:

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

— $[\![M]\!]^{\lambda}_{\alpha}$ does not introduce x:

So $[\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \dagger \widehat{x}[\![M]\!]^{\lambda}_{\alpha} \rightarrow (\mathit{act}\text{-}\textsc{r})\ [\![N]\!]^{\lambda}_{\delta}\widehat{\delta} \nwarrow \widehat{x}[\![M]\!]^{\lambda}_{\alpha}$ and the result follows from the first part. □

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

Proof. We use induction on the number of steps. We will only show the base case, which is, in turn, proved by induction on the structure of terms: again, we will only show the base case, namely $M = (\lambda x.P)Q$. In this case, $(\lambda x.P)Q \rightarrow P[Q/x]$, hence we have to

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

prove

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

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

$$\begin{array}{ll}
(\widehat{x}[\![P]\!]^{\lambda}_{\delta}\widehat{\delta}\cdot\gamma)\widehat{\gamma} \dagger \widehat{y}([\![Q]\!]^{\lambda}_{\beta}\widehat{\beta}\ [y]\ \widehat{z}\langle z.\alpha\rangle) & \rightarrow\ (\textit{exp-imp}) \\
([\![Q]\!]^{\lambda}_{\beta}\widehat{\beta} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\delta})\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle & \rightarrow\ (\textit{ren}\text{-R}) \\
[\![Q]\!]^{\lambda}_{\beta}\widehat{\beta} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\alpha}. &
\end{array}$$

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

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

$$[\![P]\!]^{\lambda}_{\delta}\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle \ \rightarrow\ [\![P]\!]^{\lambda}_{\delta}\widehat{\delta} \not\curvearrowright \widehat{z}\langle z.\alpha\rangle\,.$$

This is allowed by rule (*act*-L) in $\rightarrow_{\mathrm{N}}$, as both side conditions are satisfied. □

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

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

In particular, γ is not introduced in the outermost cut, so (*act*-L) can be applied. What the CBV reduction should guarantee, however, is that (*act*-R) *cannot* be applied, since then the propagation of $[\![P]\!]^{\lambda}_{\sigma}\widehat{\sigma} \dagger \widehat{t}([\![Q]\!]^{\lambda}_{\tau}\widehat{\tau}\ [t]\ \widehat{u}\langle u.\gamma\rangle)$ into $[\![M]\!]^{\lambda}_{\alpha}$ is blocked (which would produce $[\![M[(PQ)/x]]\!]^{\lambda}_{\alpha}$, by Lemma 4.5). It should be noted that we can only apply rule (*act*-R) if both $[\![P]\!]^{\lambda}_{\sigma}\widehat{\sigma} \dagger \widehat{t}([\![Q]\!]^{\lambda}_{\tau}\widehat{\tau}\ [t]\ \widehat{u}\langle u.\gamma\rangle)$ introduces γ and $[\![M]\!]^{\lambda}_{\alpha}$ does not introduce x. This is not the case, since the first test fails.

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

$$[\![N]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{x}[\![M]\!]^{\lambda}_{\alpha} \ \rightarrow_{\mathrm{v}}\ [\![M[N/x]]\!]^{\lambda}_{\alpha}.$$

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

Theorem 4.7 (Simulation of cbv for the λ-calculus). If $M \rightarrow_{\mathrm{v}} N$, then $[\![M]\!]^{\lambda}_{\gamma} \rightarrow_{\mathrm{v}} [\![N]\!]^{\lambda}_{\gamma}$.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Proof. As in the proof of Theorem 4.6, we only show the case $M = (\lambda x.P)Q$. Notice that we also have $[\![(\lambda x.P)Q]\!]^{\lambda}_{\delta} \to_{\mathrm{v}} [\![Q]\!]^{\lambda\widehat{\beta}}_{\beta} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\delta}$ (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 (*ren*-R), when, perhaps,

$$[\![P]\!]^{\lambda\widehat{\delta}}_{\delta} \dagger \widehat{z}\langle z.\alpha\rangle \;\to\; [\![P]\!]^{\lambda\widehat{\delta}}_{\delta} \nearrow \widehat{z}\langle z.\alpha\rangle$$

when δ is not introduced in P. This is allowed by rule (*act*-L) in $\to_{\mathrm{v}}$.

Now we show that $[\![Q]\!]^{\lambda\widehat{\beta}}_{\beta} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\delta} \to_{\mathrm{v}} [\![P[Q/x]]\!]_{\alpha}$ if and only if Q is a value:

If:

Let Q be a value, so $[\![Q]\!]^{\lambda}_{\beta}$ introduces β.

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

— $[\![P]\!]^{\lambda}_{\alpha}$ introduces x:

By Definition 4.3, this is only possible if $P = x$ (and then $[\![P]\!]^{\lambda}_{\alpha} = \langle x.\alpha\rangle$). So we have $[\![Q]\!]^{\lambda\widehat{\beta}}_{\beta} \dagger \widehat{x}\langle x.\alpha\rangle \to_{\mathrm{v}} [\![Q]\!]^{\lambda}_{\alpha}$ by (*ren*-R), and $[\![Q]\!]^{\lambda}_{\alpha} = [\![x[Q/x]]\!]^{\lambda}_{\alpha}$.

— $[\![P]\!]^{\lambda}_{\alpha}$ does not introduce x:

Since the side conditions for rule (*act*-R) are satisfied, we get

$$[\![Q]\!]^{\lambda\widehat{\beta}}_{\beta} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\delta} \to_{\mathrm{v}} (act\text{-R})[\![Q]\!]^{\lambda\widehat{\beta}}_{\beta} \nwarrow \widehat{x}[\![P]\!]^{\lambda}_{\delta},$$

as noted above.

Only if:

Let $Q = RS$. Hence,

$$[\![RS]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\alpha} \;\to\; ([\![R]\!]^{\lambda\widehat{\delta}}_{\delta} \dagger \widehat{z}([\![S]\!]^{\lambda\widehat{\beta}}_{\beta}\ [z]\ \widehat{v}\langle v.\gamma\rangle))\widehat{\gamma} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\alpha}.$$

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

$$\begin{array}{ll} ([\![R]\!]^{\lambda\widehat{\delta}}_{\delta} \dagger \widehat{z}([\![S]\!]^{\lambda\widehat{\beta}}_{\beta}\ [z]\ \widehat{v}\langle v.\gamma\rangle))\widehat{\gamma} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\alpha} & \to \\ ([\![R]\!]^{\lambda\widehat{\delta}}_{\delta} \dagger \widehat{z}([\![S]\!]^{\lambda\widehat{\beta}}_{\beta}\ [z]\ \widehat{v}\langle v.\gamma\rangle))\widehat{\gamma} \nearrow \widehat{x}[\![P]\!]^{\lambda}_{\alpha} & \to \\ ([\![R]\!]^{\lambda\widehat{\gamma}}_{\delta} \nearrow \widehat{x}[\![P]\!]^{\lambda}_{\alpha})\widehat{\delta} \dagger \widehat{z}(([\![S]\!]^{\lambda\widehat{\beta}}_{\beta}\ [z]\ \widehat{v}\langle v.\gamma\rangle)\widehat{\gamma} \nearrow \widehat{x}[\![P]\!]^{\lambda}_{\alpha}) & \to \ldots \end{array}$$

or

$$\begin{array}{l} ([\![R]\!]^{\lambda\widehat{\delta}}_{\delta} \dagger \widehat{z}([\![S]\!]^{\lambda\widehat{\beta}}_{\beta}\ [z]\ \widehat{v}\langle v.\gamma\rangle))\widehat{\gamma} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\alpha} \to \\ T\widehat{\gamma} \dagger \widehat{x}[\![P]\!]^{\lambda}_{\alpha} \end{array}$$

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

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 outermost cut when the reduction of R returns a net that introduces γ, that is, is a capsule or an export. In any case, the reduction will not lead to $[\![P[(RS)/x]]\!]^{\lambda}_{\alpha}$. □

Notice that we need to show *both* implications in the proof above. Just proving '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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

we have in Theorem 4.6 is not necessarily a reduction in CBV, and Lemma 4.5 only shows a result for *right*-activated cuts.

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

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

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

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

Theorem 4.8. If $\Gamma \vdash_{\lambda} M : A$, then $[\![M]\!]^{\lambda}_{\alpha} :\cdot\ \Gamma \vdash \alpha{:}A$.

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

(ax):

We have $M = x$ and $\Gamma = \Gamma', x{:}A$. Then notice that

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

$(\to I)$:

We have $M = \lambda x.N$, $A = C{\to}D$ and $\Gamma, x{:}C \vdash_{\lambda} N : D$. Hence, by induction, there exists a derivation $\mathrm{D} :: [\![N]\!]^{\lambda}_{\beta} :\cdot\ \Gamma, x{:}C \vdash \beta{:}D$, and we can construct

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

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

$(\to E)$:

We have $M = PQ$, and there exists B such that $\Gamma \vdash_{\lambda} P : B{\to}A$ and $\Gamma \vdash_{\lambda} Q : B$. By induction, there exist derivations $\mathrm{D}_1 :: [\![P]\!]^{\lambda}_{\gamma} :\cdot\ \Gamma \vdash \gamma{:}B{\to}A$ and $\mathrm{D}_2 :: [\![Q]\!]^{\lambda}_{\beta} :\cdot\ \Gamma \vdash \beta{:}B$, and we can construct

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Notice that $[\![PQ]\!]^{\lambda}_{\alpha} = [\![P]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{x}([\![Q]\!]^{\lambda\widehat{\beta}}_{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle)$, 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 reinforce the fact that we consider more than just those nets that represent proofs, we will now give an example of a non-terminating reduction sequence.

Example 4.9 (Reducing $[\![\Delta\Delta]\!]^{\lambda}_{\beta}$). Recall that $[\![xx]\!]^{\lambda}_{\beta} \rightarrow \langle x.\delta\rangle\widehat{\delta}\ [x]\ \widehat{y}\langle y.\beta\rangle$. Now $[\![\Delta\Delta]\!]^{\lambda}_{\beta}$ reduces as follows:

$$
\begin{array}{lll}
[\![\Delta\Delta]\!]^{\lambda}_{\beta} & \stackrel{\Delta}{=} & \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{z}([\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma}\ [z]\ \widehat{y}\langle y.\beta\rangle) & \stackrel{\Delta}{=} & \\
(\widehat{x}[\![xx]\!]^{\lambda\widehat{\alpha}}_{\alpha}\cdot\delta)\widehat{\delta} \dagger \widehat{z}([\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma}\ [z]\ \widehat{y}\langle y.\beta\rangle) & \rightarrow & (\textit{exp-imp}) \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{x}([\![xx]\!]^{\lambda\widehat{\alpha}}_{\alpha} \dagger \widehat{y}\langle y.\beta\rangle) & \rightarrow & (\textit{ren}\text{-R}) \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{x}[\![xx]\!]^{\lambda}_{\beta} & \rightarrow & \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{x}(\langle x.\delta\rangle\widehat{\delta}\ [x]\ \widehat{y}\langle y.\beta\rangle) & \rightarrow & (\textit{act}\text{-R})\ \&\ (\curlywedge\textit{imp-outs}) \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{z}(([\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \curlywedge \widehat{x}\langle x.\delta\rangle)\widehat{\delta}\ [z]\ \widehat{y}([\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \curlywedge \widehat{x}\langle y.\beta\rangle)) & \rightarrow & (\curlywedge\textit{cap}) \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{z}(([\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \curlywedge \widehat{x}\langle x.\delta\rangle)\widehat{\delta}\ [z]\ \widehat{y}\langle y.\beta\rangle) & \rightarrow & (d\curlywedge)\ \&\ (\textit{ren}\text{-R}) \\
[\![\lambda x.xx]\!]^{\lambda\widehat{\gamma}}_{\gamma} \dagger \widehat{z}([\![\lambda x.xx]\!]^{\lambda\widehat{\delta}}_{\delta}\ [z]\ \widehat{y}\langle y.\beta\rangle) & \stackrel{\Delta}{=} & [\![\Delta\Delta]\!]^{\lambda}_{\beta}
\end{array}
$$

5. Interpreting $\lambda\mathbf{x}$

In this section we will interpret a calculus of explicit substitutions, $\lambda\mathbf{x}$, which was introduced by Bloo and Rose (Bloo and Rose 1995), and in which a β-reduction of the λ-calculus is split into several atomic steps of computation. We will show that $\mathscr{X}$ 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 and Rose 1995). The syntax of $\lambda\mathbf{x}$ is an extension of the syntax 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, with the added syntactic construct being dealt with by the (*cut*)-rule:

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

$$(\rightarrow I):\ \frac{\Gamma, x{:}A \vdash_{\lambda\mathbf{x}} M : B}{\Gamma \vdash_{\lambda\mathbf{x}} \lambda x.M : A\rightarrow B} \qquad (\rightarrow E):\ \frac{\Gamma \vdash_{\lambda\mathbf{x}} M : A\rightarrow B \qquad \Gamma \vdash_{\lambda\mathbf{x}} N : A}{\Gamma \vdash_{\lambda\mathbf{x}} MN : B}.$$

Notice that the (*cut*)-rule does not enable us to prove sequents that were not provable 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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

substitution by the implicit

$$\text{if } \Gamma, x{:}A \vdash_\lambda M : B \text{ and } \Gamma \vdash_\lambda N : A, \text{ then also } \Gamma \vdash_\lambda M[N/x] : B.$$

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

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

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

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

Our notion of CBV-$\lambda\mathbf{x}$ follows the λ-calculus treatment in a natural way.

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 CBV-$\lambda\mathbf{x}$ by a syntactic restriction:

$$M \ ::= \ x \mid \lambda x.M \mid M_1 M_2 \mid M\langle x=\lambda x.N\rangle \mid M\langle x=y\rangle .$$

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

Subject reduction still holds, as we can still see the computation in $\lambda\mathbf{x}$ as cut-elimination: now this process consists of 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 $\mathscr{X}$). We define $[\![\cdot]\!]^{\lambda\mathrm{x}}_\alpha$ as the interpretation $[\![\cdot]\!]^{\lambda}_\alpha$ by adding a case for the explicit substitution:

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

Note that the interpretation of the λ-calculus comes from just the first three rules. Note also that the cut is activated in the final alternative, which might seem in contrast with Lemma 4.5, where we justified the fact that the cut $[\![N]\!]^{\lambda\mathrm{x}}_{\widehat{\beta}} \dagger \widehat{x}[\![M]\!]^{\lambda\mathrm{x}}_\alpha$ reduces to $[\![M[N/x]]\!]^{\lambda\mathrm{x}}_\alpha$, but, using the inactivated cut, we cannot prove that

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

$$\llbracket (P\,Q)\langle x=N\rangle \rrbracket_{\alpha}^{\lambda x} \downarrow \llbracket (P\,\langle x=N\rangle)(Q\,\langle x=N\rangle)\rrbracket_{\alpha}^{\lambda x}.$$

Now notice that, again, N is a value if and only if $\llbracket N \rrbracket_{\alpha}^{\lambda x}$ introduces α.

Theorem 5.5. If $\Gamma \vdash_{\lambda x} M : A$, then $\llbracket M \rrbracket_{\alpha}^{\lambda x} :\cdot\ \Gamma \vdash \alpha{:}A$.

The proof is a straightforward extension of the proof of Theorem 4.8.

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

Theorem 5.6 (Simulation of rule (B)).

— CBN:

$$\llbracket (\lambda x.M)N \rrbracket_{\alpha}^{\lambda x} \to_{\mathrm{N}} \llbracket M\,\langle x=N\rangle \rrbracket_{\alpha}^{\lambda x}.$$

— CBV:

$$\llbracket (\lambda x.M)N \rrbracket_{\alpha}^{\lambda x} \to_{\mathrm{V}} \llbracket M\,\langle x=N\rangle \rrbracket_{\alpha}^{\lambda x} \text{ if and only if } N \text{ is a value.}$$

Proof.

$$\begin{array}{lll}
\llbracket (\lambda x.M)N \rrbracket_{\alpha}^{\lambda x} & \stackrel{\Delta}{=} & \\
\llbracket \lambda x.M \rrbracket_{\gamma}^{\lambda x}\widehat{\gamma} \dagger \widehat{y}(\llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta}\ [y]\ \widehat{z}\langle z.\alpha\rangle) & \stackrel{\Delta}{=} & \\
(\widehat{x}\llbracket M \rrbracket_{\delta}^{\lambda x}\widehat{\delta}\cdot\gamma)\widehat{\gamma} \dagger \widehat{y}(\llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta}\ [y]\ \widehat{z}\langle z.\alpha\rangle) & \to & (exp\text{-}imp) \\
\llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta} \dagger \widehat{x}(\llbracket M \rrbracket_{\delta}^{\lambda x}\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle) & \to & (act\text{-R}) \\
\llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta} \diagdown \widehat{x}(\llbracket M \rrbracket_{\delta}^{\lambda x}\widehat{\delta} \dagger \widehat{z}\langle z.\alpha\rangle) & \to & (ren\text{-R}) \\
\llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta} \diagdown \widehat{x}\llbracket M \rrbracket_{\alpha}^{\lambda x} & \stackrel{\Delta}{=} & \llbracket M\,\langle x=N\rangle \rrbracket_{\alpha}^{\lambda x}.
\end{array}$$

Note that this reduction sequence is valid in the CBN evaluation, which proves the first point.

For the CBV evaluation, the step *act*-R is possible if and only if $\llbracket N \rrbracket_{\beta}^{\lambda x}$ introduces β, that is, if and only if N is a value. The proof is then completed using Lemma 4.5. □

Notice that we could also show $\llbracket (\lambda x.M)N \rrbracket_{\alpha}^{\lambda x} \to_{\mathrm{N}} \llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta} \dagger \widehat{x}\llbracket M \rrbracket_{\alpha}$, but that using the definition $\llbracket M\,\langle x=N\rangle \rrbracket_{\alpha}^{\lambda x} \stackrel{\Delta}{=} \llbracket N \rrbracket_{\beta}^{\lambda x}\widehat{\beta} \dagger \widehat{x}\llbracket M \rrbracket_{\alpha}$ 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), (VarI), (VarK), (gc). Then $\llbracket M \rrbracket_{\gamma}^{\lambda x} \to_{\mathrm{V}} \llbracket N \rrbracket_{\gamma}^{\lambda x}$ and $\llbracket M \rrbracket_{\gamma}^{\lambda x} \to_{\mathrm{N}} \llbracket N \rrbracket_{\gamma}^{\lambda x}$.

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

— $\llbracket (P\,Q)\langle x=N\rangle \rrbracket_{\alpha}^{\lambda x} \to \llbracket (P\,\langle x=N\rangle)(Q\,\langle x=N\rangle)\rrbracket_{\alpha}^{\lambda x}$:

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

$$\begin{array}{ll}
[\![(PQ)\langle x=N\rangle]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![PQ]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}([\![P]\!]_\gamma^{\lambda x}\widehat{\gamma} \dagger \widehat{y}([\![Q]\!]_\beta^{\lambda x}\widehat{\beta}\ [y]\ \widehat{z}\langle z.\alpha\rangle)) & \to\ (\nwarrow cut) \\
([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![P]\!]_\gamma^{\lambda x})\widehat{\gamma} \dagger \widehat{y}([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}([\![Q]\!]_\beta^{\lambda x}\widehat{\beta}\ [y]\ \widehat{z}\langle z.\alpha\rangle)) & \to\ (\nwarrow imp\text{-}ins) \\
([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![P]\!]_\gamma^{\lambda x})\widehat{\gamma} \dagger \widehat{y}(([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![Q]\!]_\beta^{\lambda x})\widehat{\beta}\ [y]\ \widehat{z}([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}\langle z.\alpha\rangle)) & \stackrel{\Delta}{=} \\
[\![P\langle x=N\rangle]\!]_\gamma^{\lambda x}\widehat{\gamma} \dagger \widehat{y}([\![Q\langle x=N\rangle]\!]_\beta^{\lambda x}\widehat{\beta}\ [y]\ \widehat{z}([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}\langle z.\alpha\rangle)) & \to\ (\nwarrow cap) \\
[\![P\langle x=N\rangle]\!]_\gamma^{\lambda x}\widehat{\gamma} \dagger \widehat{y}([\![Q\langle x=N\rangle]\!]_\beta^{\lambda x}\widehat{\beta}\ [y]\ \widehat{z}\langle z.\alpha\rangle) & \stackrel{\Delta}{=} \\
[\![(P\langle x=N\rangle)(Q\langle x=N\rangle)]\!]_\alpha^{\lambda x}. &
\end{array}$$

— $[\![(\lambda y.M)\langle x=N\rangle]\!]_\alpha^{\lambda x} \to [\![\lambda y.M\langle x=N\rangle]\!]_\alpha^{\lambda x}$:

$$\begin{array}{ll}
[\![(\lambda y.M)\langle x=N\rangle]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![\lambda y.M]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}(\widehat{y}[\![M]\!]_\gamma^{\lambda x}\widehat{\gamma}\cdot\alpha) & \to\ (\nwarrow exp) \\
\widehat{y}([\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![M]\!]_\gamma^{\lambda x})\widehat{\gamma}\cdot\alpha & \stackrel{\Delta}{=} \\
\widehat{y}[\![M\langle x=N\rangle]\!]_\gamma^{\lambda x}\widehat{\gamma}\cdot\alpha & \stackrel{\Delta}{=}\ [\![\lambda y.M\langle x=N\rangle]\!]_\alpha^{\lambda x}.
\end{array}$$

— $[\![x\langle x=N\rangle]\!]_\alpha^{\lambda x} \to [\![N]\!]_\alpha^{\lambda x}$:

$$\begin{array}{ll}
[\![x\langle x=N\rangle]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![x]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}\langle x.\alpha\rangle & \to\ (d\nwarrow) \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \dagger \widehat{x}\langle x.\alpha\rangle & \to\ (ren\text{-}\mathrm{R})\ [\![N]\!]_\alpha^{\lambda x}.
\end{array}$$

— $[\![y\langle x=N\rangle]\!]_\alpha^{\lambda x} \to [\![y]\!]_\alpha^{\lambda x}$:

$$\begin{array}{ll}
[\![y\langle x=N\rangle]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![y]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}\langle y.\alpha\rangle & \to\ (\nwarrow cap) \\
\langle y.\alpha\rangle & \stackrel{\Delta}{=}\ [\![y]\!]_\alpha^{\lambda x}.
\end{array}$$

— $[\![M\langle x=N\rangle]\!]_\alpha^{\lambda x} \to [\![M]\!]_\alpha^{\lambda x}$, if $x \notin fv(M)$:

$$\begin{array}{ll}
[\![M\langle x=N\rangle]\!]_\alpha^{\lambda x} & \stackrel{\Delta}{=} \\
[\![N]\!]_\beta^{\lambda x}\widehat{\beta} \nwarrow \widehat{x}[\![M]\!]_\alpha^{\lambda x} & \to\ (\nwarrow gc)\ [\![M]\!]_\alpha^{\lambda x}.
\end{array}$$

□

We can now say that $\lambda\mathbf{x}$ reduction is preserved by the interpretation of terms into $\mathscr{X}$.

Theorem 5.8 (Simulation of $\lambda\mathbf{x}$).

1 If $M \to_{\mathrm{v}} N$, then $[\![M]\!]_\gamma^{\lambda x} \to_{\mathrm{v}} [\![N]\!]_\gamma^{\lambda x}$

2 If $M \to_{\mathrm{N}} N$, then $[\![M]\!]_\gamma^{\lambda x} \to_{\mathrm{N}} [\![N]\!]_\gamma^{\lambda x}$

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

$$M\langle x=P\rangle\langle y=Q\rangle \to M\langle y=Q\rangle\langle x=P\langle y=Q\rangle\rangle\,.$$

It is trivial to see that this rule breaks the strong normalisation property for typed terms (Bloo and Geuvers 1996) and cannot be simulated. But it can be useful for reasoning by equivalence. If we abandon strong normalisation, we can merge the three kinds of cuts. It can be done, for instance, by setting the equivalence

$$P\widehat{\alpha} \not\leftarrow \widehat{x}Q \sim P\widehat{\alpha} \dagger \widehat{x}Q \sim P\widehat{\alpha} \not\rightarrow \widehat{x}Q\,.$$

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

$$\begin{array}{ll}
[\![M\langle x=P\rangle\langle y=Q\rangle]\!]_{\alpha}^{\lambda x} & \stackrel{\Delta}{=} \\
[\![Q]\!]_{\beta}^{\lambda x}\widehat{\beta} \dagger \widehat{y}([\![P]\!]_{\delta}^{\lambda x}\widehat{\delta} \dagger \widehat{x}[\![M]\!]_{\alpha}^{\lambda x}) & \to \\
([\![Q]\!]_{\beta}^{\lambda x}\widehat{\beta} \dagger \widehat{y}[\![P]\!]_{\delta}^{\lambda x})\widehat{\delta} \dagger \widehat{x}([\![Q]\!]_{\beta}^{\lambda x}\widehat{\beta} \dagger \widehat{y}[\![M]\!]_{\alpha}^{\lambda x}) & \stackrel{\Delta}{=} \\
[\![M\langle y=Q\rangle\langle x=P\langle y=Q\rangle\rangle]\!]_{\alpha}^{\lambda x}. &
\end{array}$$

There are some interesting issues arising from this that deserve to be explored.

6. Interpreting $\lambda\mu$

Parigot's $\lambda\mu$-calculus (Parigot 1992) is a proof-term syntax for classical logic, but expressed in the setting of natural deduction. We extend the syntax of formulae with the constant $\bot$ (*false*). Natural deduction deals with classical logic by allowing the logical rule

$$\frac{\Gamma, A\Rightarrow\bot \vdash \bot}{\Gamma \vdash A}\,.$$

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 a splitting of the set of axioms into two parts: the first where the assumptions might be discharged purely by the introduction of '$\Rightarrow$', and the second where the assumptions might be discharged purely 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

$$\Gamma, A_1\Rightarrow\bot,\ldots,A_n\Rightarrow\bot \vdash B$$

is logically equivalent to the multi-conclusion sequent

$$\Gamma \vdash B \mid A_1,\ldots,A_n\,,$$

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

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

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

$$\frac{\dfrac{}{\Gamma, A \Rightarrow \bot \vdash A \Rightarrow \bot} \qquad \boxed{}}{\Gamma, A \Rightarrow \bot \vdash \bot}$$

Thus, 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 Parigot (1992), 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 witnesses to the two rules discussed above. It uses two disjoint sets of variables (Latin letters and Greek letters). The sequents typing terms are of the form $\Gamma \vdash A \mid \Delta$, 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 $\alpha, \beta, \ldots$.

$$\frac{}{\Gamma \vdash_{\lambda\mu} x{:}A \mid \Delta}\,(x{:}A \in \Gamma) \qquad \frac{\Gamma, x{:}A \vdash_{\lambda\mu} M{:}B \mid \Delta}{\Gamma \vdash_{\lambda\mu} \lambda x.M{:}A{\to}B \mid \Delta}$$

$$\frac{\Gamma \vdash_{\lambda\mu} M{:}A{\to}B \mid \Delta \quad \Gamma \vdash_{\lambda\mu} N{:}A \mid \Delta}{\Gamma \vdash_{\lambda\mu} MN{:}B \mid \Delta}$$

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

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

into

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

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

Definition 6.3 (Terms of $\lambda\mu$ – version 2). The terms of $\lambda\mu$ version 2 are defined by the 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 Parigot (1992); the syntax in Definition 6.3 was introduced later in Parigot (1993) and de Groote (1994).

Since $\bot$ is not a type in the system we consider here for $\mathscr{X}$, it is the relationship between this last variant of $\lambda\mu$ and $\mathscr{X}$ that we will study.

As an example illustrating the fact that this system is still more powerful than the system for the λ-calculus, here is a proof of Peirce's Law (due to Ong and Stewart (1997)):

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

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 following 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

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

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*.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Definition 6.5 (Interpretation of $\lambda\mu$ in $\mathscr{X}$). We define $\llbracket\cdot\rrbracket^{\lambda\mu}_{\alpha}$ as the interpretation $\llbracket\cdot\rrbracket^{\lambda}_{\alpha}$ by adding an alternative for μ-terms:

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

In a similar way to what we did in previous sections (see Lemma 4.5), we can prove

$$\begin{aligned}
\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta} \dagger \widehat{x}\llbracket M\rrbracket^{\lambda\mu}_{\alpha} &\to \llbracket M[N/x]\rrbracket^{\lambda\mu}_{\alpha}\\
\llbracket M\rrbracket^{\lambda\mu}_{\gamma}\widehat{\delta} \dagger \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle) &\to \llbracket \mu\tau.([\gamma]M)[N\cdot\tau/\delta]\rrbracket^{\lambda\mu}_{\alpha}.
\end{aligned}$$

This result will be convenient later.

Notice that the final alternative is justified, since we can show the following result.

Lemma 6.6. The following rule is admissible:

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

Proof.

$$\begin{aligned}
&\llbracket(\mu\delta.[\gamma]M)N\rrbracket^{\lambda\mu}_{\alpha} && \triangleq\\
&\llbracket\mu\delta.[\gamma]M\rrbracket^{\lambda\mu}_{\nu}\widehat{\nu} \dagger \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle) && \triangleq\\
&(\llbracket M\rrbracket^{\lambda\mu}_{\gamma}\widehat{\delta} \dagger \widehat{z}\langle z.\nu\rangle)\widehat{\nu} \dagger \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle) && \to (ren\text{-R})\\
&\llbracket M\rrbracket^{\lambda\mu}_{\gamma}[\nu/\delta]\widehat{\nu} \dagger \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle) && \to (=_{\alpha})\\
&\llbracket M\rrbracket^{\lambda\mu}_{\gamma}\widehat{\delta} \dagger \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle). && \qquad\square
\end{aligned}$$

Notice the similarity between the net $\llbracket MN\rrbracket^{\lambda\mu}_{\alpha}$ and the result of running $\llbracket(\mu\delta.[\gamma]M)N\rrbracket^{\lambda\mu}_{\alpha}$; 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, but for the $\lambda\mu$-calculus we can also connect to plugs that occur inside, that is, to named sub-terms.

We will now show that the above definition of the encoding of μ-substitution is correct: notice that, unlike the case 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 Curien and Herbelin (2000)[†].

Lemma 6.7.

(1) $\llbracket M\rrbracket^{\lambda\mu}_{\delta}\widehat{\delta} \not\sim \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) \downarrow \llbracket M[N\cdot\gamma/\delta]N\rrbracket^{\lambda\mu}_{\gamma}$.

(2) $\llbracket M\rrbracket^{\lambda\mu}_{\nu}\widehat{\delta} \not\sim \widehat{x}(\llbracket N\rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) \downarrow \llbracket M[N\cdot\gamma/\delta]\rrbracket^{\lambda\mu}_{\nu}$, if $\delta \neq \nu$.

Proof. We use simultaneous induction on the structure of terms. We will only show the interesting cases.

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

(1) $M = \lambda z.M'$:

$$\begin{array}{lll}
[\![\lambda z.M']\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
(\widehat{z}[\![M']\!]^{\lambda\mu}_{\sigma}\widehat{\sigma\cdot\delta})\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \to & \\
(\widehat{z}([\![M']\!]^{\lambda\mu}_{\sigma}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle))\widehat{\sigma\cdot\tau})\widehat{\tau} \dagger \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \downarrow & (IH.2)\\
(\widehat{z}[\![M'[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\sigma}\widehat{\sigma\cdot\tau})\widehat{\tau} \dagger \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![\lambda z.M'[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\tau}\widehat{\tau} \dagger \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![(\lambda z.M')[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}. &&
\end{array}$$

$M = M_1M_2$:

$$\begin{array}{lll}
[\![M_1M_2]\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
([\![M_1]\!]^{\lambda\mu}_{\sigma}\widehat{\sigma} \dagger \widehat{z}([\![M_2]\!]^{\lambda\mu}_{\tau}\widehat{\tau}\ [z]\ \widehat{v}\langle v.\delta\rangle))\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \downarrow & (IH.2)\\
[\![M_1[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\sigma}\widehat{\sigma} \dagger \widehat{z}([\![M_2[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\tau}\widehat{\tau}\ [z]\ \widehat{v}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [v]\ \widehat{y}\langle y.\gamma\rangle)) & \leftarrow & \\
([\![M_1[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\sigma}\widehat{\sigma} \dagger \widehat{z}([\![M_2[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\tau}\widehat{\tau}\ [z]\ \widehat{v}\langle v.\tau\rangle))\widehat{\tau} \not\prec \widehat{x} & & \\
\qquad ([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \leftarrow & \\
[\![M_1[N\cdot\gamma/\delta]M_2[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\tau}\widehat{\tau} \dagger \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![(M_1M_2)[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}. &&
\end{array}$$

$M = \mu\sigma.[\delta]M'$:

$$\begin{array}{lll}
[\![\mu\sigma.[\delta]M']\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
([\![M']\!]^{\lambda\mu}_{\delta}\widehat{\sigma} \dagger \widehat{z}\langle z.\delta\rangle)\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \to & \\
([\![M']\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle))\widehat{\sigma} \dagger \widehat{z}(\langle z.\delta\rangle\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle)) \to & & \\
([\![M']\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle))\widehat{\sigma} \dagger \widehat{z}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\gamma\rangle) & \downarrow & (IH.1)\\
[\![M'[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}\widehat{\sigma} \dagger \widehat{z}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\gamma\rangle) & \leftarrow & (=_{\alpha})\\
([\![M'[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}\widehat{\sigma} \dagger \widehat{w}\langle w.\nu\rangle)\widehat{\nu} \dagger \widehat{z}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![\mu\sigma.[\gamma]M'[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\nu}\widehat{\nu} \dagger \widehat{z}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [z]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
[\![(\mu\sigma.[\gamma]M'[N\cdot\gamma/\delta]N)N]\!]^{\lambda\mu}_{\gamma} & = & \\
[\![(\mu\sigma.[\delta]M')[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}. &&
\end{array}$$

(2) $M = \mu\sigma.[\delta]M'$

$$\begin{array}{lll}
[\![\mu\sigma.[\delta]M']\!]^{\lambda\mu}_{\nu}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} & \\
([\![M']\!]^{\lambda\mu}_{\delta}\widehat{\sigma} \dagger \widehat{z}\langle z.\nu\rangle)\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \to & \\
([\![M']\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle))\widehat{\sigma} \dagger \widehat{z}(\langle z.\nu\rangle\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle)) \to & & \\
([\![M']\!]^{\lambda\mu}_{\delta}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle))\widehat{\sigma} \dagger \widehat{z}\langle z.\nu\rangle & \downarrow & (IH.1)\\
[\![M'[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}\widehat{\sigma} \dagger \widehat{z}\langle z.\nu\rangle & \leftarrow & (=_{\alpha})\\
([\![M'[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\gamma}\widehat{\sigma} \dagger \widehat{w}\langle w.\tau\rangle)\widehat{\tau} \dagger \widehat{z}\langle z.\nu\rangle & \stackrel{\Delta}{=} & \\
[\![\mu\sigma.[\gamma]M'[N\cdot\gamma/\delta]N]\!]^{\lambda\mu}_{\tau}\widehat{\tau} \dagger \widehat{z}\langle z.\nu\rangle & \to & \\
[\![(\mu\sigma.[\gamma]M'[N\cdot\gamma/\delta]N)]\!]^{\lambda\mu}_{\nu} & = & \\
[\![(\mu\sigma.[\delta]M')[N\cdot\gamma/\delta]]\!]^{\lambda\mu}_{\nu}. &&
\end{array}$$

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

$$
\begin{array}{ll}
M = \mu\sigma.[\tau]M' & \\
\quad [\![\mu\sigma.[\tau]M']\!]^{\lambda\mu}_{\nu}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \stackrel{\Delta}{=} \\
\quad ([\![M']\!]^{\lambda\mu}_{\tau}\widehat{\sigma} \dagger \widehat{z}\langle z.v\rangle)\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle) & \to \\
\quad ([\![M']\!]^{\lambda\mu}_{\tau}\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle))\,\widehat{\sigma} \dagger \widehat{z} & \\
\qquad\qquad (\langle z.v\rangle\widehat{\delta} \not\prec \widehat{x}([\![N]\!]^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\gamma\rangle)) & \downarrow\ (IH.2) \\
\quad [\![M'[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\tau}\widehat{\sigma} \dagger \widehat{z}\langle z.v\rangle & \to \\
\quad [\![(\mu\sigma.[\tau]M')[N{\cdot}\gamma/\delta]]\!]^{\lambda\mu}_{\nu}. & \square
\end{array}
$$

Note that a number of the 'reverse' reduction steps in this proof could have been avoided by defining $[\![\mu\alpha.[\beta]M]\!]^{\lambda\mu}_{\gamma} \stackrel{\Delta}{=} [\![M]\!]^{\lambda\mu}_{\beta}[\gamma/\alpha]$. But, as is evident from case $M = M_1M_2$ in the proof of Part 1, this would not give a proof that the interpretation is preserved by reduction.

Note also that, for the (*renaming*) and (*erasure*) rules, we have

$$
\begin{array}{ll}
[\![\mu\alpha.[\beta](\mu\gamma.[\delta]M)]\!]^{\lambda\mu}_{\sigma} & \stackrel{\Delta}{=} \\
[\![\mu\gamma.[\delta]M]\!]^{\lambda\mu}_{\beta}\widehat{\alpha} \dagger \widehat{x}\langle x.\sigma\rangle & = \\
([\![M]\!]^{\lambda\mu}_{\delta}\widehat{\gamma} \dagger \widehat{x}\langle x.\beta\rangle)\widehat{\alpha} \dagger \widehat{x}\langle x.\sigma\rangle & \to \\
[\![M[\beta/\gamma]]\!]^{\lambda\mu}_{\delta}\widehat{\alpha} \dagger \widehat{x}\langle x.\sigma\rangle & = \\
[\![\mu\alpha.[\delta]M[\beta/\gamma]]\!]^{\lambda\mu}_{\sigma} &
\end{array}
$$

and

$$
\begin{array}{ll}
[\![\mu\alpha.[\alpha]M]\!]^{\lambda\mu}_{\sigma} & \stackrel{\Delta}{=} \\
[\![M]\!]^{\lambda\mu}_{\alpha}\widehat{\alpha} \dagger \widehat{x}\langle x.\sigma\rangle & \to \\
[\![M]\!]^{\lambda\mu}_{\sigma}. &
\end{array}
$$

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 $[\![M]\!]^{\lambda\mu}_{\alpha} \downarrow_{\mathrm{N}} [\![N]\!]^{\lambda\mu}_{\alpha}$.

Proof. As before, the proof is by induction on the length of the reductions sequence. We will just show the base case, and, by the above, we only need to focus on rules (β) and (μ). The proof for case (β) is as for Theorem 4.6, and for case (μ), we have the following cases:

— $(\mu\delta.[\delta]M)N \to \mu\alpha.([\delta]M)[N{\cdot}\alpha/\delta]$:

By Lemma 6.6, we already know that

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

(notice that x is introduced). Again, there are two cases to consider:

– δ is introduced:

So either:

- $[\![M]\!]^{\lambda\mu}_{\delta} = \langle z.\delta\rangle$:

Note that

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

and that

$$\begin{aligned}\llbracket \mu\alpha.[\alpha]zN \rrbracket^{\lambda\mu}_{\alpha} &\stackrel{\Delta}{=} \llbracket zN \rrbracket^{\lambda\mu}_{\alpha}\widehat{\alpha} \dagger \widehat{v}\langle v.\alpha\rangle \\ &\stackrel{\Delta}{=} (\llbracket z \rrbracket^{\lambda\mu}_{\gamma}\widehat{\gamma} \dagger \widehat{w}(\llbracket N \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [w]\ \widehat{a}\langle a.\alpha\rangle))\widehat{\alpha} \dagger \widehat{v}\langle v.\alpha\rangle \\ &\to \llbracket N \rrbracket^{\lambda\mu}_{\beta}\beta[z]y\langle y.\alpha\rangle.\end{aligned}$$

- $\llbracket M \rrbracket^{\lambda\mu}_{\delta} = \widehat{z}\llbracket M' \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\cdot\delta$:
  So $M = \lambda z.M'$ and δ does not occur in M'. Note that

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

  – δ is not introduced:
    So the latter reduces to $\llbracket M \rrbracket^{\lambda\mu}_{\delta}\widehat{\delta} \not\dagger \widehat{x}(\llbracket N \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle)$.
    By Lemma 6.7, $\llbracket M \rrbracket^{\lambda\mu}_{\delta}\widehat{\delta} \not\dagger \widehat{x}(\llbracket N \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle) \downarrow \llbracket M[N\cdot\alpha/\delta]N \rrbracket^{\lambda\mu}_{\alpha}$. Note that

$$\llbracket M[N\cdot\alpha/\delta]N \rrbracket^{\lambda\mu}_{\alpha} \leftarrow \llbracket M[N\cdot\alpha/\delta]N \rrbracket^{\lambda\mu}_{\alpha}\widehat{\alpha} \dagger \widehat{v}\langle v.\alpha\rangle \stackrel{\Delta}{=} \llbracket \mu\alpha.[\alpha]M[N\cdot\alpha/\delta]N \rrbracket^{\lambda\mu}_{\alpha}.$$

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

$$\llbracket (\mu\delta.[v]M))N \rrbracket^{\lambda\mu}_{\alpha} \to \llbracket M \rrbracket^{\lambda\mu}_{v}\widehat{\delta} \dagger \widehat{x}(\llbracket N \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle).$$

  Since δ is not introduced, the latter reduces to $\llbracket M \rrbracket^{\lambda\mu}_{v}\widehat{\delta} \not\dagger \widehat{x}(\llbracket N \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle)$.
  By Lemma 6.7, $\llbracket M \rrbracket^{\lambda\mu}_{v}\widehat{\delta} \not\dagger \widehat{x}(\llbracket N \rrbracket^{\lambda\mu}_{\beta}\widehat{\beta}\ [x]\ \widehat{y}\langle y.\alpha\rangle) \downarrow \llbracket M[N\cdot\alpha/\delta] \rrbracket^{\lambda\mu}_{v}$. Note that

$$\llbracket M[N\cdot\alpha/\delta] \rrbracket^{\lambda\mu}_{v} \leftarrow \llbracket M[N\cdot\alpha/\delta] \rrbracket^{\lambda\mu}_{v}\widehat{\alpha} \dagger \widehat{v}\langle v.\alpha\rangle \stackrel{\Delta}{=} \llbracket \mu\alpha.[v]M[N\cdot\alpha/\delta] \rrbracket^{\lambda\mu}_{\alpha}.$$ □

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

Theorem 6.9. If $\Gamma \vdash_{\lambda\mu} M{:}A \mid \Delta$, then $\llbracket M \rrbracket^{\lambda\mu}_{\alpha} \mathrel{:\cdot} \Gamma \vdash \alpha{:}A, \Delta$.

Proof. The proof is similar to that 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 $\llbracket M \rrbracket^{\lambda\mu}_{\alpha} \stackrel{\Delta}{=} \llbracket N \rrbracket^{\lambda\mu}_{\gamma}\widehat{\delta} \dagger \widehat{x}\langle x.\alpha\rangle$. If we assume $\Gamma \vdash_{\lambda\mu} M{:}A \mid \gamma{:}B, \Delta$, we also have $\Gamma \vdash_{\lambda\mu} N{:}B \mid \delta{:}A, \gamma{:}B, \Delta$, and then $\llbracket N \rrbracket^{\lambda\mu}_{\gamma} \mathrel{:\cdot} \Gamma \vdash \delta{:}A, \gamma{:}B$ follows by induction. Hence,

$$\dfrac{\dfrac{\boxed{\phantom{\llbracket N \rrbracket^{\lambda\mu}_{\gamma} \mathrel{:\cdot} \Gamma \vdash \delta{:}A, \gamma{:}B, \Delta}}}{\llbracket N \rrbracket^{\lambda\mu}_{\gamma} \mathrel{:\cdot} \Gamma \vdash \delta{:}A, \gamma{:}B, \Delta} \qquad \dfrac{}{\langle x.\alpha\rangle \mathrel{:\cdot} \Gamma, x{:}A \vdash \alpha{:}A, \gamma{:}B, \Delta}}{\llbracket N \rrbracket^{\lambda\mu}_{\gamma}\widehat{\delta} \dagger \widehat{x}\langle x.\alpha\rangle \mathrel{:\cdot} \Gamma \vdash \alpha{:}A, \gamma{:}B, \Delta}$$

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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

In its typed version, $\mathscr{X}$ is a proof-term syntax for a classical sequent calculus. Another proof-system was proposed for (a variant of) classical sequent calculus in Curien and Herbelin's $\overline{\lambda}\mu\tilde{\mu}$-calculus (Curien and Herbelin 2000). It is interesting to relate these two formalisms and to observe that $\overline{\lambda}\mu\tilde{\mu}$ can be interpreted in $\mathscr{X}$ as well.

For the $\overline{\lambda}\mu\tilde{\mu}$-calculus presented in Curien and Herbelin (2000), there are two sets of variables: $x, y, z \ldots$ label the types of the hypotheses and $\alpha, \beta, \gamma \ldots$ label the types of the conclusions. Moreover, the syntax of $\overline{\lambda}\mu\tilde{\mu}$ has three different categories: commands, terms and contexts or co-terms. Correspondingly, they are typed by three kinds of sequents: the usual sequents $\Gamma \vdash \Delta$ type commands, while the sequents typing terms (respectively, contexts) are of the form $\Gamma \vdash A \mid \Delta$ (respectively, $\Gamma \mid A \vdash \Delta$), marking the conclusion (respectively, hypothesis) A as *active*.

Definition 7.1 (Commands, Terms and Contexts).

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

Here $\mu\alpha.c$, $\tilde{\mu}x.c'$ and $\lambda y.v$ bind α in c, x in c' and y in v, respectively; 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 '$\rightarrow$' on the left-hand side of a sequent), and commands fill the hole of a context with a term.

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

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

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

$$(LI) : \frac{\Gamma \vdash_{\overline{\lambda}\mu\tilde{\mu}} v{:}A \mid \Delta \quad \Gamma \mid e{:}B \vdash \Delta}{\Gamma \mid v \cdot e{:}A{\rightarrow}B \vdash_{\overline{\lambda}\mu\tilde{\mu}} \Delta} \qquad (RI) : \frac{\Gamma, x{:}A \vdash_{\overline{\lambda}\mu\tilde{\mu}} v{:}B \mid \Delta}{\Gamma \vdash_{\overline{\lambda}\mu\tilde{\mu}} \lambda x.v{:}A{\rightarrow}B \mid \Delta}$$

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

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 $\mathscr{X}$. With the conventional notation for contexts, $v \cdot e$ is to be thought of as $e[[\,]\,v]$. We see here how a term (respectively, a context) is built either by introducing '$\rightarrow$' on the right-hand side (respectively, left-hand side) of a sequent, or just by activating one conclusion (respectively, hypothesis) from a sequent typing a command: $\mu\alpha.c$ is inherited from $\lambda\mu$, and $\tilde{\mu}x.c$ is to be thought of as $\mathsf{let}\ x = [\,]\ \mathsf{in}\ c$.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Proofs can sometimes be simplified, that is, commands can be computed by eliminating the cuts.

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

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

The system has a critical pair $\langle \mu\alpha.c_1 \parallel \tilde{\mu}x.c_2\rangle$, and applying rule (μ) in this case gives a call-by-value evaluation, whereas applying rule $(\tilde{\mu})$ gives a call-by-name evaluation. As 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 elegantly expresses 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 Wadler (2003)). Indeed, the call-by-value reduction $\rightarrow_{\mathrm{V}}$ is obtained by forbidding a $\tilde{\mu}$-reduction when the redex is also a μ-redex, whereas the call-by-name reduction $\rightarrow_{\mathrm{N}}$ forbids a μ-reduction when the redex is also a $\tilde{\mu}$-redex. We show here how $\mathscr{X}$ accounts for this duality.

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

$$\dfrac{\dfrac{\Gamma, A \vdash_{\mathrm{LK}} B, \Delta}{\Gamma \vdash_{\mathrm{LK}} A \rightarrow B, \Delta}(\rightarrow R) \qquad \dfrac{\Gamma \vdash_{\mathrm{LK}} A, \Delta \quad \Gamma, B \vdash_{\mathrm{LK}} \Delta}{\Gamma, A \rightarrow B \vdash_{\mathrm{LK}} \Delta}(\rightarrow L)}{\Gamma \vdash_{\mathrm{LK}} \Delta}(cut)$$

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

$$\dfrac{\dfrac{\Gamma, A \vdash_{\mathrm{LK}} B, \Delta}{\Gamma \vdash_{\mathrm{LK}} A, \Delta} \quad \dfrac{\Gamma, B \vdash_{\mathrm{LK}} \Delta \quad \Gamma, A, B \vdash_{\mathrm{LK}} \Delta}{\Gamma, A \vdash_{\mathrm{LK}} \Delta}}{\Gamma \vdash_{\mathrm{LK}} \Delta} \quad \text{and} \quad \dfrac{\dfrac{\Gamma \vdash_{\mathrm{LK}} A, \Delta}{\Gamma \vdash_{\mathrm{LK}} A, B, \Delta} \quad \Gamma, A \vdash_{\mathrm{LK}} B, \Delta}{\Gamma \vdash_{\mathrm{LK}} B, \Delta \quad \Gamma, B \vdash_{\mathrm{LK}} \Delta}}{\Gamma \vdash_{\mathrm{LK}} \Delta}$$

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 $\mathscr{X}$, 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 $\mathscr{X}$ into $\overline{\lambda}\mu\tilde{\mu}$.

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

Definition 7.4 (Translation of $\mathscr{X}$ into $\overline{\lambda}\mu\tilde{\mu}$ (Lengrand 2003)).

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

In fact, this is the origin of $\mathscr{X}$: Curien and Herbelin gave a hint in Curien and Herbelin (2000, Remark 4.1) of 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 $\bar{\lambda}\mu\tilde{\mu}$:

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

where the typing rules correspond to the axiom rule, the right introduction of '$\Rightarrow$', the left introduction of '$\Rightarrow$' and the rule for a cut, respectively. Later it was discovered that this corresponded closely to Urban's approach in Urban (2000); however, as discussed above following Definition 4.3, the approaches differ.

Although $\mathscr{X}$ is in fact a sub-syntax of $\bar{\lambda}\mu\tilde{\mu}$, it is no less expressive. On the contrary, on the logical side, two proofs of the same sequent might be considered to be different in $\mathsf{LK}_{\mu\tilde{\mu}}$ just because the naming of formulae, or the activation/deactivation of formulae, has not been done in 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 $\mathscr{X}$: the one eagerly naming all formulae it deals with.

In concrete terms, here is a translation of $\bar{\lambda}\mu\tilde{\mu}$ into $\mathscr{X}$ that preserves the typing.

Definition 7.5 (Translation of $\bar{\lambda}\mu\tilde{\mu}$ into $\mathscr{X}$ (Lengrand 2003)).

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

Example 7.6. Ghilezan and Lescanne (2004) showed that Peirce's Law $((A{\to}B){\to}A){\to}A$ can be inhabited in $\bar{\lambda}\mu\tilde{\mu}$ by the term

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

which is a term in normal form that is itself the reduction of the translation of the $\lambda\mu$-term given by Ong and Stewart (1997) (see Section 6) and is typeable as follows (where $\vdash = \vdash_{\bar{\lambda}\mu\tilde{\mu}}$):

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

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

When we remove all term information from this derivation, we obtain a semi-proof for Peirce's Law in classical logic

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

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

Moreover, the two (activation) steps, that is, 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 $\llbracket\cdot\rrbracket^{\bar\lambda\mu\tilde\mu}$ on the $\bar\lambda\mu\tilde\mu$-term, we obtain

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

Notice that the process obtained through translation has four cuts – removing these produces the following (inside out) reduction sequence:

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

The final term is exactly that of Example 3.5.

The interpretation function preserves typeability.

Lemma 7.7.

1 If $\Gamma\vdash_{\bar\lambda\mu\tilde\mu} v{:}A\mid\Delta$, then $\llbracket v\rrbracket^{\bar\lambda\mu\tilde\mu}_{\alpha}\ {:}\cdot\ \Gamma\vdash\alpha{:}A,\Delta$.
2 If $\Gamma\mid e{:}A\vdash_{\bar\lambda\mu\tilde\mu}\Delta$, then $\llbracket e\rrbracket^{\bar\lambda\mu\tilde\mu}_{x}\ {:}\cdot\ \Gamma,x{:}A\vdash\Delta$.
3 If $c:\Gamma\vdash_{\bar\lambda\mu\tilde\mu}\Delta$, then $\llbracket c\rrbracket^{\bar\lambda\mu\tilde\mu}\ {:}\cdot\ \Gamma\vdash\Delta$.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Proof. The proof is straightforward by simultaneous induction on the structure of derivations. □

Lengrand (2003) showed that we can simulate the implicit substitution of $\bar{\lambda}\mu\tilde{\mu}$ in $\mathscr{X}$.

Lemma 7.8.

1 $[\![c]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\widehat{\alpha}} \not\curlywedge \widehat{x}[\![e]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{x} \to [\![c[e/\alpha]]\!]^{\bar{\lambda}\mu\tilde{\mu}}$.
2 $[\![v]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} \not\curlywedge \widehat{x}[\![e]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{x} \to [\![v[e/\alpha]]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta}$.
3 $[\![e']\!]^{\bar{\lambda}\mu\tilde{\mu}}_{z} \not\curlywedge \widehat{x}[\![e]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{x} \to [\![e'[e/\alpha]]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{z}$.

Proof. The proof is by simultaneous induction on the structure of nets. □

Similarly, we can show the following results.

Lemma 7.9.

1 $[\![v]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\alpha} \not\curlyvee \widehat{x}[\![c]\!]^{\bar{\lambda}\mu\tilde{\mu}} \to [\![c[v/x]]\!]^{\bar{\lambda}\mu\tilde{\mu}}$.
2 $[\![v]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\alpha} \not\curlyvee \widehat{x}[\![v']\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} \to [\![v'[v/x]]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta}$.
3 $[\![v]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\alpha} \not\curlyvee \widehat{x}[\![e]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{y} \to [\![e[v/x]]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{y}$.

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

Theorem 7.10 (Simulation of $\bar{\lambda}\mu\tilde{\mu}$). If $c \to c'$, then $[\![c]\!]^{\bar{\lambda}\mu\tilde{\mu}} \downarrow [\![c']\!]^{\bar{\lambda}\mu\tilde{\mu}}$.

Proof. We use induction of the length of the reduction sequence. We will only show the base cases:

($\to$) We have $c = \langle \lambda x.v_1 \,\|\, v_2 \cdot e\rangle$ and $c' = \langle v_2 \,\|\, \tilde{\mu}x.\langle v_1 \,\|\, e\rangle\rangle$. Hence,

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

(μ) This follows by Lemma 7.8.
($\tilde{\mu}$) This follows by Lemma 7.9. □

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

In fact, we can even show the following results.

Theorem 7.11.

1 If $c \to_{\mathrm{N}} c'$, then $[\![c]\!]^{\bar{\lambda}\mu\tilde{\mu}} \downarrow_{\mathrm{N}} [\![c']\!]^{\bar{\lambda}\mu\tilde{\mu}}$.
2 If $c \to_{\mathrm{V}} c'$, then $[\![c]\!]^{\bar{\lambda}\mu\tilde{\mu}} \downarrow_{\mathrm{V}} [\![c']\!]^{\bar{\lambda}\mu\tilde{\mu}}$.

Curien and Herbelin define two encodings.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Definition 7.12 (Curien and Herbelin 2000)). The interpretation $\cdot^<$ and interpretation $\cdot^>$ of $\lambda\mu$ into $\bar{\lambda}\mu\tilde{\mu}$ are defined as follows:

$$\begin{array}{rcl@{\qquad}rcl}
x^< & \stackrel{\Delta}{=} & x & x^> & \stackrel{\Delta}{=} & x \\
(\lambda x.M)^< & \stackrel{\Delta}{=} & \lambda x.M^< & (\lambda x.M)^> & \stackrel{\Delta}{=} & \lambda x.M^> \\
(MN)^< & \stackrel{\Delta}{=} & \mu\alpha.\langle N^< \| \tilde{\mu}x.\langle M^< \| x \cdot \alpha\rangle\rangle & (MN)^> & \stackrel{\Delta}{=} & \mu\alpha.\langle M^> \| N^> \cdot \alpha\rangle \\
(\mu\beta.c)^< & \stackrel{\Delta}{=} & \mu\beta.c^< & (\mu\beta.c)^> & \stackrel{\Delta}{=} & \mu\beta.c^> \\
([\alpha]M)^< & \stackrel{\Delta}{=} & \langle M^< \| \alpha\rangle & ([\alpha]M)^> & \stackrel{\Delta}{=} & \langle M^> \| \alpha\rangle\,.
\end{array}$$

$\cdot^<$ corresponds to right-to-left call-by-value, and $\cdot^>$ is related to left-to-right call-by-value.

Curien and Herbelin's result is given by the following theorem.

Theorem 7.13 (Simulation of $\lambda\mu$ in $\bar{\lambda}\mu\tilde{\mu}$ (Curien and Herbelin 2000)).

1 If $M \to_{\mathrm{V}} N$, then $M^< \to_{\mathrm{V}} N^<$ up to μ-expansion.
2 If $M \to_{\mathrm{N}} N$, then $M^> \to_{\mathrm{N}} N^>$ 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 and CBN evaluations relies on two *distinct* translations of terms.

As above, the CBV/CBN distinction is accounted for by the encodings themselves, and the distinction between CBV and CBN relies mostly on Curien and Herbelin's two distinct encodings rather than the features of $\bar{\lambda}\mu\tilde{\mu}$. The same holds for Wadler (2003). While their CBN translation seems intuitive, in order to give an accurate interpretation of the CBV-λ-calculus, they apparently need to twist it in a more complex way than we do since we can show the following result.

Lemma 7.14. $M^< \to M^>$.

Proof. We use induction on the structure of terms. The interesting case is

$$\begin{array}{lcl}
(NP)^< & \stackrel{\Delta}{=} & \\
\mu\alpha.\langle N^< \| \tilde{\mu}x.\langle M^< \| x \cdot \alpha\rangle\rangle & \to & (IH) \\
\mu\alpha.\langle N^> \| \tilde{\mu}x.\langle M^> \| x \cdot \alpha\rangle\rangle & \to_{\tilde{\mu}} & \\
\mu\alpha.\langle M^> \| N^> \cdot \alpha\rangle & \stackrel{\Delta}{=} & \\
(NP)^>\,. & & \qquad \square
\end{array}$$

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 the following result.

Lemma 7.15. $[\![M^<]\!]_\beta^{\bar{\lambda}\mu\tilde{\mu}} \downarrow [\![M^>]\!]_\beta^{\bar{\lambda}\mu\tilde{\mu}}$.

But we now show that we can take the simplest translation (that is, $[\![\cdot^>]\!]_\beta^{\bar{\lambda}\mu\tilde{\mu}}$) for both CBV and CBN, and that the evaluation strategies of $\mathscr{X}$ reflects the distinction between them, in showing that the interpretation $[\![\cdot]\!]_\alpha^{\lambda}$ is more refined that the composition of $\cdot^>$ and $[\![\cdot]\!]_\alpha^{\bar{\lambda}\mu\tilde{\mu}}$.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Lemma 7.16. $[\![M^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} \to [\![M]\!]^{\lambda}_{\beta}$.

Proof. By induction on the structure of terms:

$M = x$:

$[\![x^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} \stackrel{\Delta}{=} [\![x]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} \stackrel{\Delta}{=} \langle x.\beta\rangle \stackrel{\Delta}{=} [\![x]\!]^{\lambda}_{\beta}$.

$M = \lambda x.M'$:

$$
\begin{array}{ll}
[\![(\lambda x.M')^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} & \stackrel{\Delta}{=} \\
[\![\lambda x.M'^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} & \stackrel{\Delta}{=} \\
\widehat{x}[\![M'^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\alpha}\widehat{\alpha}\cdot\beta & \to \ (IH) \\
\widehat{x}[\![M']\!]^{\lambda}_{\alpha}\widehat{\alpha}\cdot\beta & \stackrel{\Delta}{=} \\
[\![\lambda x.M']\!]^{\lambda}_{\beta}. &
\end{array}
$$

$M = PQ$:

$$
\begin{array}{ll}
[\![(PQ)^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} & \stackrel{\Delta}{=} \\
[\![\mu\alpha\langle P^{>} \parallel Q^{>}\cdot\alpha\rangle]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\beta} & \stackrel{\Delta}{=} \\
[\![\langle P^{>} \parallel Q^{>}\cdot\alpha\rangle]\!]^{\bar{\lambda}\mu\tilde{\mu}}\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle & \to \ (ren\text{-R}) \\
[\![\langle P^{>} \parallel Q^{>}\cdot\beta\rangle]\!]^{\bar{\lambda}\mu\tilde{\mu}} & \stackrel{\Delta}{=} \\
[\![P^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\gamma}\widehat{\gamma} \dagger \widehat{x}[\![Q^{>}\cdot\beta]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{x} & \stackrel{\Delta}{=} \\
[\![P^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\gamma}\widehat{\gamma} \dagger \widehat{x}([\![Q^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\alpha}\widehat{\alpha}\ [x]\ \widehat{y}[\![\beta]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{y}) & \stackrel{\Delta}{=} \\
[\![P^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\gamma}\widehat{\gamma} \dagger \widehat{x}([\![Q^{>}]\!]^{\bar{\lambda}\mu\tilde{\mu}}_{\alpha}\widehat{\alpha}\ [x]\ \widehat{y}\langle y.\beta\rangle) & \to \ (IH) \\
[\![P]\!]^{\lambda}_{\gamma}\widehat{\gamma} \dagger \widehat{x}([\![Q]\!]^{\lambda}_{\alpha}\widehat{\alpha}\ [x]\ \widehat{y}\langle y.\beta\rangle) & \stackrel{\Delta}{=} \\
[\![PQ]\!]^{\lambda}_{\beta}. &
\end{array}
$$

□

8. Conclusions and future work

We have seen that $\mathscr{X}$ is an application- and substitution-free formal language that provides a Curry–Howard isomorphism for a sequent calculus for implicative classical logic. But, of greater interest, we have seen that $\mathscr{X}$ is very well suited to being a generic abstract machine for the running of (applicative) programming languages by building not only an interpretation for λ, $\lambda\mu$ and $\bar{\lambda}\mu\tilde{\mu}$, 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 and the confluence of the CBN and CBV strategies, to extend $\mathscr{X}$ so that we can represent the other logical connectives, to study the relationship with linear logic and proofnets (both typed and untyped), and how to express recursion, functions, and so on.

Details of the implementation of the tool for $\mathscr{X}$ can be found in van Bakel and Raghunandan (2005; 2006). Polymorphism is introduced in Summers and van Bakel (2006), the encoding of other connectors in Raghunandan and Summers (2006), and the interpretation of $\mathscr{X}$ in $\lambda\mu$ in Audebaud and van Bakel (2006). The relation between $\mathscr{X}$ and Π is currently under investigation.

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

Appendix A. The rules of $\mathscr{X}$

In this appendix we collect together all the rules of $\mathscr{X}$, and also give the admissible rules.

Logical Reduction (applicable only if α and x are introduced).

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

Activations

$$\begin{array}{rl}
(act\text{-}\mathrm{L}): & P\widehat{\alpha}\dagger\widehat{x}Q \to P\widehat{\alpha}\not\leftarrow\widehat{x}Q, \textit{ if } P \textit{ does not introduce } \alpha \\
(act\text{-}\mathrm{R}): & P\widehat{\alpha}\dagger\widehat{x}Q \to P\widehat{\alpha}\not\rightarrow\widehat{x}Q, \textit{ if } Q \textit{ does not introduce } x.
\end{array}$$

Left Propagation

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

Right Propagation

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

Admissible rules

$$\begin{array}{rrcl}
(\not\leftarrow gc): & P\widehat{\alpha}\dagger\widehat{x}Q & \to & P, \quad \text{if } \alpha\notin fp(P),\ P \text{ pure} \\
(\not\rightarrow gc): & P\widehat{\alpha}\dagger\widehat{x}Q & \to & Q, \quad \text{if } x\notin fs(Q),\ Q \text{ pure} \\
(ren\text{-}\mathrm{R}): & P\widehat{\delta}\dagger\widehat{z}\langle z.\alpha\rangle & \to & P[\alpha/\delta],\ P \text{ pure} \\
(ren\text{-}\mathrm{L}): & \langle z.\alpha\rangle\widehat{\alpha}\dagger\widehat{x}P & \to & P[z/x],\ P \text{ pure}.
\end{array}$$

Modified activations for cbv and cbn

$$\begin{array}{rl}
(act\text{-}\mathrm{R}): & P\widehat{\alpha}\dagger\widehat{x}Q \to P\widehat{\alpha}\not\rightarrow\widehat{x}Q, \textit{ if } P \textit{ introduces } \alpha \textit{ and } Q \textit{ does not introduce } x. \\
(act\text{-}\mathrm{L}): & P\widehat{\alpha}\dagger\widehat{x}Q \to P\widehat{\alpha}\not\leftarrow\widehat{x}Q, \textit{ if } P \textit{ does not introduce } \alpha \textit{ and } Q \textit{ introduces } x.
\end{array}$$

Acknowledgements

We would like to thank Alexander Summers, Daniel Hirschkoff, Dragiša Žunić, Harry Mairson, Jamie Gabbay, Jayshan Raghunandan, Luca Cardelli, Luca Roversi, Maria

https://doi.org/10.1017/S0960129508006762 Published online by Cambridge University Press

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 was there at the origins of this research, and has kindly let us integrate his papers in ours.

We would also like to thank Hugo Herbelin for carefully reading our paper, and for suggesting many improvements and corrections.

References

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

Aho, A. V., Sethi, R. and Ullman, J. D. (1988) *Compilers: Principles, Techniques and Tools*, Addison-Wesley.

Appel, A. W. and Jim, T. (1989) Continuation-passing, closure-passing style. In: *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of Programming languages*, ACM Press 293–302.

Ariola, Z. M. and Herbelin, H. (2003) Minimal classical logic and control operators. In: Baeten, J. C. M., Lenstra, J. K., Parrow, J. and Woeginger, G. J. (eds.) ICALP. *Springer-Verlag Lecture Notes in Computer Science* **2719** 871–885.

Ariola, Z. M., Herbelin, H. and Sabry, A. (2004) A type-theoretic foundation of continuations and prompts. In: *Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming (ICFP'04)*, ACM 40–53.

Audebaud, P. and van Bakel, S. (2006) Understanding $\mathscr{X}$ with $\lambda\mu$. Consistent interpretations of the implicative sequent calculus in natural deduction (submitted).

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

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

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

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

Barendregt, H. (1984) *The Lambda Calculus: its Syntax and Semantics*, revised edition, North-Holland.

Barendregt, H. and Ghilezan, S. (2000) Lambda terms for natural deduction, sequent calculus and cut-elimination. *Journal of Functional Porgramming* **10** (1) 121–134.

Bloo, R. and Geuvers, J. H. (1996) Explicit substitution: on the edge of strong normalisation. Computing Science Reports 96–10, Eindhoven University of Technology, The Netherlands. (To appear in *Theoretical Computer Science*.)

Bloo, R. and Rose, K. (1995) Preservation of strong normalisation in named lambda calculi with explicit substitution and garbage collection. *CSN'95 – Computer Science in the Netherlands* 62–72. (Available at `ftp://ftp.diku.dk/diku/semantics/papers/D-246.ps`.)

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.

Church, A. (1940) A formulation of the simple theory of types. *The journal of Symbolic Logic* **5** 56–68.

Clement, D. (1986) A Simple Applicative Language: MINI-ML. Technical Report 529, INRIA centre Sophia Antipolis (France).

Coquand, T. and Huet, G. P. (1985) Constructions: A higher order proof system for mechanizing mathematics. *European Conference on Computer Algebra* (1) 151–184.

Curien, P.-L. and Herbelin, H. (2000) The Duality of Computation. In: *Proc. of the 5th ACM SIGPLAN International Conference on Functional Programming (ICFP'00)*, ACM 233–243. (A corrected version is available from `http://yquem.inria.fr/ herbelin/publis/icfp-CuHer00-duality+errata.ps.gz`)

Curry, H. B., Feys, R. and Craig, W. (1958) *Combinatory Logic*, Vol. 1, Elsevier Science Publishers B.V.

van Dalen, D. (1994) *Logic and Structure*, Springer-Verlag.

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

Danos, V., Joinet, J.-B. and Schellinx, H. (1997) A new deconstructive logic: Linear logic. *The journal of Symbolic Logic* **62**.

Gentzen, G. (1935) Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift* **39** 176–210 and 405–431. (English translation in Szabo (1969, Pages 68–131)).

Ghilezan, S. and Lescanne, P. (2004) Classical proofs, typed processes and intersection types. In: Berardi, S., Coppo, M. and Damiani, F. (eds.) TYPES'03 (to appear in *Springer-Verlag Lecture Notes in Computer Science*).

Girard, J. Y. (1986) The system F of variable types, fifteen years later. *Theoretical Computer Science* **45** 159–192.

Girard, J.-Y. (1987) Linear logic. *Theoretical Computer Science* **50** 1–102.

Girard, J.-Y. (1991) A new constrcutive logic: classical logic. *Mathematical Structures in Computer Science* **1** (3) 255–296.

Griffin, T. (1990) A formulae-as-types notion of control. In: *Proceedings of the 17th Annual ACM Symposium on Principles Of Programming Languages* 47–58.

de Groote, P. (1994) On the relation between the $\lambda\mu$-calculus and the syntactic theory of sequential control. In: Proceedings of the 5th International Conference on Logic Programming and Automated Reasoning, (LPAR'94). *Springer-Verlag Lecture Notes in Computer Science* **822** 31–43.

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.

Herbelin, H. (2005) *C'est maintenant qu'on calcule: au cœur de la dualité*, Mémoire d'habilitation, Université Paris 11.

Krivine, J.-L. (1994) Classical logic, storage operators and second-order lambda-calculus. *Ann. Pure Appl. Logic* **68**(1) 53–78.

Lengrand, S. (2002) A computational interpretation of the *cut*-rule in classical sequent calculus, Master's thesis, Mathematical Institute and Computing Laboratory, University of Oxford.

Lengrand, S. (2003) Call-by-value, call-by-name, and strong normalization for the classical sequent calculus. In: Gramlich, B. and Lucas, S. (eds.) Post-proceedings of the 3rd Workshop on Reduction Strategies in Rewriting and Programming (WRS 2003). *Electronic Notes in Theoretical Computer Science* **86**.

Lengrand, S. (2006) *Normalisation & Equivalence in Proof Theory & Type Theory*, Ph.D. thesis, University of Paris VII and University of St Andrews.

Lengrand, S., Lescanne, P., Dougherty, D., Dezani-Ciancaglini, M. and van Bakel, S. (2004) Intersection types for explicit substitutions. *Information and Computation* **189** (1) 17–42.

Lescanne, P. (1994) From $\lambda\sigma$ to $\lambda\upsilon$, a journey through calculi of explicit substitutions. In: Boehm, H. (ed.) *Proceedings of the 21st Annual ACM Symposium on Principles Of Programming Languages*, ACM 60–69.

Ong, C.-H. L. and Stewart, C. A. (1997) A Curry-Howard foundation for functional computation with control. In: *Proceedings of the 24th Annual ACM Symposium on Principles Of Programming Languages* 215–227.

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

Parigot, M. (1993) Classical proofs as programs. *Kurt Gödel Colloquium* 263–276. (Presented at TYPES Workshop, at Bǎstad, June 1992.)

Raghunandan, J. and Summers, A. (2006) On the Computational Representation of Classical Logical Connectives. (Presented at 2nd International Workshop on Developments in Computational Models (DCM 2006), Venice, Italy.)

Summers, A. and van Bakel, S. (2006) Approaches to polymorphism in classical sequent calculus. In: Sestoft, P. (ed.) Proceedings of 15th European Symposium on Programming (ESOP'06). *Springer-Verlag Lecture Notes in Computer Science* **3924** 84–99.

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

Urban, C. (2000) *Classical Logic and Computation*, Ph.D. thesis, University of Cambridge.

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

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

Wadler, P. (2003) Call-by-Value is Dual to Call-by-Name. In: *Proceedings of the eighth ACM SIGPLAN international conference on Functional programming* 189 – 201.

Whitehead, A. N. and Russell, B. (1925) *Principia Mathematica*, 2nd edition, Cambridge University Press.

Whitehead, A. N. and Russell, B. (1997) *Principia Mathematica to *56*, 2nd edition, Cambridge Mathematical Library, Cambridge University Press.