ELSEVIER

Theoretical Computer Science 227 (1999) 43–78

Theoretical Computer Science

www.elsevier.com/locate/tcs

A classical linear λ-calculus

G.M. Bierman

Gonville and Caius College, Cambridge CB2 1TA, UK

Abstract

This paper proposes and studies a particular typed λ-calculus for classical linear logic. I shall give an explanation of a natural deduction formulation of classical logic due to Parigot and compare it to more traditional treatments by Prawitz and others. I shall use Parigot's method to devise a natural deduction formulation of classical linear logic. This formulation is compared in detail to the sequent calculus formulation.

Keywords: Linear logic; λ-calculus; Type systems; Proof theory

1. Introduction

In the past classical logic (CL) has often been dismissed as having no interesting proof theory. However, following a rather pleasing interplay between theoretical computer science and practical computer science, there has been a renewed interest in CL and, in particular, the constructive content of classical proofs. This content appears to have links with, at the theoretical level, game theory [20] and at the practical level, certain control operators for functional programming languages [24]. To some extent Girard's linear logic [22] has also renewed interest in game theory and functional programming languages. The refined connectives of linear logic have helped shed new light on work on games [17]. The games models have proved useful for programming language semantics: the recent fully abstract models of PCF [2, 26] are good examples of this. In addition, intuitionistic linear logic (ILL) has been proposed as a resource-sensitive foundation of functional programming languages [7, 14]. Thus it would seem useful to reconsider the work on CL in a linear setting, i.e. to reconsider classical linear logic (CLL).

E-mail address: gmb@cl.cam.ac.uk (G.M. Bierman)

0304-3975/99/$ - see front matter
PII: S0304-3975(99)00048-1

Gentzen's natural deduction is a very suitable deduction system for intuitionistic logic (IL) but seems less so for classical logic.[1] One could say that classical logic is a logic of symmetry whereas natural deduction is, by its very nature, an asymmetric system. To that extent Gentzen's alternative system, the sequent calculus, seems better suited as the system for CL.

The Curry–Howard correspondence [25] allows us to annotate natural deductions with terms. For IL this yields the typed λ-calculus. For the sequent calculus it is not entirely clear what the appropriate annotations are. In fact, there are a number of choices and there is no real consensus on the best. It might seem prudent to revisit natural deduction, where the question of syntax is settled, and see if we might be able to produce a more symmetric system. Various people have proposed multiple- and sequence-conclusion formulations, including Shoesmith and Smiley [35], Boričić [18] and Celluci [19]. More recently Parigot [30] has introduced a variant of sequence-conclusion natural deduction which seems particularly well suited for handling CL.

In this paper I shall consider in detail Parigot's technique and, in particular, study its application to CLL. This paper is organised as follows. In Section 2 I recall Parigot's formulation of CL, which I call CL_μ. Via the Curry–Howard correspondence one derives a term calculus for CL_μ, which is called the (typed) $\lambda\mu$-calculus. Also in Section 2 I study the particularly tricky area of reduction. In Section 3 I follow a similar method to derive a natural deduction formulation of CLL, called CLL_μ. There are some surprises here concerning the exponential modality (!). Applying the Curry–Howard correspondence to CLL_μ yields the (typed) *linear* $\lambda\mu$-calculus. In Section 3.2 I consider the process of reduction for the linear $\lambda\mu$-calculus, proving strong normalisation in Section 4. In Section 5 I show how to map sequent proofs in CLL to deductions in CLL_μ and then use this to compare the process of cut-elimination with the term reduction process. In Section 6 I consider briefly the Q- and T-translations of Schellinx, at the level of terms, between CL_μ and CLL_μ. In Section 7 I briefly give another presentation of the linear $\lambda\mu$-calculus based on Benton's mixed presentation of the linear λ-calculus [5].

Before continuing I should perhaps clarify the rôle of this work. In his seminal paper [22], Girard presented *proof nets*, which are a succinct presentation of proofs in CLL. One important feature of proof nets is that formulae which are equivalent with respect to the dualities (for example, $\phi \otimes \psi$ and $(\phi^\perp \parr \psi^\perp)^\perp$) are considered to be equal. This cuts down considerably the number of proofs. What I am striving for here is a calculus which does *not* have these formula equivalences built-in. Consider an analogous situation for the λ-calculus with products and coproducts. We might consider the formulae $\phi \times (\psi + \upsilon)$ and $(\phi \times \psi) + (\phi \times \upsilon)$ to be equivalent and in (categorical) models they are *isomorphic*. However, we certainly do not insist on them being equal (they are distinct types!). Indeed it is hard to imagine a programming language where this is so. Thus I suggest that the linear $\lambda\mu$-calculus is a more realistic foundation for a

[1] "One may doubt that this is the proper way of analysing classical inferences" [33, pp. 244–245].

programming language based on CLL. A language based on proof nets would probably be some variant of Abramsky's proof expressions [1, Section 6].

2. Classical logic

In this paper I shall only consider *propositional* formulations of the various logics. Formulae of CL are given by the grammar

$$\phi ::= p \mid \phi \times \phi \mid \phi + \phi \mid \phi \to \phi,$$

where p is taken from a countable set of atomic formulae which includes a distinguished member, $\bot$, which denotes falsum.

Parigot's formulation of CL is a particular sequence-conclusion natural deduction. The conclusion is now a *sequence* of formulae, of which at most one is considered *active* and is labelled with a bullet; the rest are considered *passive* and are labelled with a variable. Parigot's original presentation (for just the implication fragment) is as follows.

$$\frac{\Gamma, \phi \vdash \psi^\bullet, \Sigma}{\Gamma \vdash (\phi \to \psi)^\bullet, \Sigma} \to_{\mathscr{I}} \qquad \frac{\Gamma \vdash (\phi \to \psi)^\bullet, \Sigma \quad \Gamma \vdash \phi^\bullet, \Sigma \quad \Gamma, \phi \vdash \phi^\bullet, \Sigma}{\Gamma \vdash \psi^\bullet, \Sigma} \to_{\mathscr{E}}$$

$$\frac{\Gamma \vdash \phi^\bullet, \Sigma}{\Gamma \vdash \phi^a, \Sigma} \text{ Passivate} \qquad \frac{\Gamma \vdash \phi^a, \Sigma}{\Gamma \vdash \phi^\bullet, \Sigma} \text{ Activate}$$

As it stands this formulation appears to be a conservative extension of the (implication fragment of the) natural deduction formulation of IL. But this is a slight illusion – in any instance of the rules the active formula need not exist. Thus,

$$\frac{\Gamma, \phi \vdash \Sigma}{\Gamma \vdash \phi \to \psi^\bullet, \Sigma} \to_{\mathscr{I}}$$

is a perfectly valid instance of the $\to_{\mathscr{I}}$ rule where there is no active formula in the upper sequent and ψ is a completely fresh formula. To make the rôle of active and passive formulae more precise I shall present the formulation so there is always exactly one active conclusion in any deduction. This necessitates the use of the falsum, $\bot$. The resulting system, which I shall call CL_μ, is given in Fig. 1. I have also added the conjunction and disjunction connectives to Parigot's original formulation.

Judgements in CL_μ are of the form $\Gamma \vdash \phi^\bullet, \Sigma$ where Γ denotes a set of formulae and Σ denotes a set of formulae labelled with μ-variables, which are written as ψ^a. The bullet annotation signifies that a formula is active. The Passivate rule is not permitted if ϕ is $\bot$. CL_μ is a sound and complete formulation of CL in the following sense.

Theorem 1 (Parigot). $\vdash_{\mathrm{CL}} \Gamma \vdash \Delta$ *iff* $\vdash_{\mathrm{CL}_\mu} \Gamma \vdash \Delta$.

$$\frac{}{\Gamma,\phi\vdash\phi^{\bullet},\Sigma}\ \text{Identity}$$

$$\frac{\Gamma,\phi\vdash\psi^{\bullet},\Sigma}{\Gamma\vdash(\phi\to\psi)^{\bullet},\Sigma}\to_{\mathscr{I}}\qquad\frac{\Gamma\vdash(\phi\to\psi)^{\bullet},\Sigma\qquad\Gamma\vdash\phi^{\bullet},\Sigma}{\Gamma\vdash\psi^{\bullet},\Sigma}\to_{\mathscr{E}}$$

$$\frac{\Gamma\vdash\phi^{\bullet},\Sigma\qquad\Gamma\vdash\psi^{\bullet},\Sigma}{\Gamma\vdash(\phi\times\psi)^{\bullet},\Sigma}\times_{\mathscr{I}}$$

$$\frac{\Gamma\vdash(\phi\times\psi)^{\bullet},\Sigma}{\Gamma\vdash\phi^{\bullet},\Sigma}\times_{\mathscr{E}-1}\qquad\frac{\Gamma\vdash(\phi\times\psi)^{\bullet},\Sigma}{\Gamma\vdash\psi^{\bullet},\Sigma}\times_{\mathscr{E}-2}$$

$$\frac{\Gamma\vdash\phi^{\bullet},\Sigma}{\Gamma\vdash(\phi+\psi)^{\bullet},\Sigma}+_{\mathscr{I}-1}\qquad\frac{\Gamma\vdash\psi^{\bullet},\Sigma}{\Gamma\vdash(\phi+\psi)^{\bullet},\Sigma}+_{\mathscr{I}-2}$$

$$\frac{\Gamma\vdash(\phi+\psi)^{\bullet},\Sigma\qquad\Gamma,\phi\vdash\varphi^{\bullet},\Sigma\qquad\Gamma,\psi\vdash\varphi^{\bullet},\Sigma}{\Gamma\vdash\varphi^{\bullet},\Sigma}+_{\mathscr{E}}$$

$$\frac{\Gamma\vdash\phi^{\bullet},\Sigma}{\Gamma\vdash\ \perp^{\bullet},\phi^{a},\Sigma}\ \text{Passivate}\qquad\frac{\Gamma\vdash\ \perp^{\bullet},\phi^{a},\Sigma}{\Gamma\vdash\phi^{\bullet},\Sigma}\ \text{Activate}$$

Fig. 1. Natural deduction formulation of CL: CL_{μ}.

To demonstrate the power of CL_{μ}, here is a derivation of Peirce's law

$$\dfrac{\dfrac{}{(\phi\to\psi)\to\phi,\phi\vdash\phi^{\bullet},\psi^{b}}\ \text{Identity}}{\dfrac{(\phi\to\psi)\to\phi,\phi\vdash\perp^{\bullet},\phi^{a},\psi^{b}}{(\phi\to\psi)\to\phi,\phi\vdash\psi^{\bullet},\phi^{a}}\ \text{Activate}}\ \text{Passivate}\qquad\dfrac{}{(\phi\to\psi)\to\phi\vdash\phi\to\psi^{\bullet},\phi^{a}}\to_{\mathscr{I}}}{\dfrac{\dfrac{\dfrac{}{(\phi\to\psi)\to\phi\vdash(\phi\to\psi)\to\phi^{\bullet},\phi^{a}}\ \text{Identity}\qquad\dfrac{(\phi\to\psi)\to\phi\vdash\phi^{\bullet},\phi^{a}}{(\phi\to\psi)\to\phi\vdash\perp^{\bullet},\phi^{a}}\ \text{Passivate}}{(\phi\to\psi)\to\phi\vdash\phi^{\bullet}}\ \text{Activate}}{\vdash((\phi\to\psi)\to\phi)\to\phi^{\bullet}}\to_{\mathscr{I}}}\to_{\mathscr{E}}$$

and of the law of the excluded middle (where $\neg\phi\stackrel{\text{def}}{=}\phi\to\perp$)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\phi\vdash\phi}\ \text{Identity}}{\phi\vdash(\phi+\neg\phi)^{\bullet}}(+_{\mathscr{I}-1})}{\phi\vdash\perp^{\bullet},(\phi+\neg\phi)^{a}}\ \text{Passivate}}{\vdash(\phi\to\perp)^{\bullet},(\phi+\neg\phi)^{a}}\to_{\mathscr{I}}}{\vdash(\phi+\neg\phi)^{\bullet},(\phi+\neg\phi)^{a}}(+_{\mathscr{I}-2})}{\vdash\perp^{\bullet},(\phi+\neg\phi)^{a}}\ \text{Passivate}}{\vdash(\phi+\neg\phi)^{\bullet}}\ \text{Activate}$$

It is quite instructive to compare these deductions to the corresponding sequent calculus derivations.

$$\frac{}{\Gamma, x:\phi \triangleright x:\phi, \Sigma}\ \text{Identity}$$

$$\frac{\Gamma, x:\phi \triangleright M:\psi, \Sigma}{\Gamma \triangleright \lambda x:\phi.M:\phi \to \psi, \Sigma} \to_{\mathscr{I}} \qquad \frac{\Gamma \triangleright M:\phi \to \psi, \Sigma \quad \Gamma \triangleright N:\phi, \Sigma}{\Gamma \triangleright MN:\psi, \Sigma} \to_{\mathscr{E}}$$

$$\frac{\Gamma \triangleright M:\phi, \Sigma \quad \Gamma \triangleright N:\psi, \Sigma}{\Gamma \triangleright \langle M, N\rangle : \phi \times \psi, \Sigma} \times_{\mathscr{I}}$$

$$\frac{\Gamma \triangleright M:\phi \times \psi, \Sigma}{\Gamma \triangleright \mathsf{fst}(M):\phi, \Sigma} \times_{\mathscr{E}-1} \qquad \frac{\Gamma \triangleright M:\phi \times \psi, \Sigma}{\Gamma \triangleright \mathsf{snd}(M):\psi, \Sigma} \times_{\mathscr{E}-2}$$

$$\frac{\Gamma \triangleright M:\phi, \Sigma}{\Gamma \triangleright \mathsf{inl}(M):\phi + \psi, \Sigma} +_{\mathscr{I}-1} \qquad \frac{\Gamma \triangleright M:\psi, \Sigma}{\Gamma \triangleright \mathsf{inr}(M):\phi + \psi, \Sigma} +_{\mathscr{I}-2}$$

$$\frac{\Gamma \triangleright M:\phi + \psi, \Sigma \quad \Gamma, x:\phi \triangleright N:\varphi, \Sigma \quad \Gamma, y:\psi \triangleright P:\varphi, \Sigma}{\Gamma \triangleright \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}(x) \to N \parallel \to \mid \mathsf{inr}(y) \to P:\varphi, \Sigma} +_{\mathscr{E}}$$

$$\frac{\Gamma \triangleright M:\phi, \Sigma}{\Gamma \triangleright [a:\phi]M:\bot, a:\phi, \Sigma}\ \text{Passivate} \qquad \frac{\Gamma \triangleright M:\bot, a:\phi, \Sigma}{\Gamma \triangleright \mu a:\phi.M:\phi, \Sigma}\ \text{Activate}$$

Fig. 2. The $\lambda\mu$-calculus

2.1. The (typed) $\lambda\mu$-calculus

We can apply the Curry–Howard (formulae-as-types) correspondence to CL_μ to get what Parigot calls the (typed) $\lambda\mu$-calculus. Raw $\lambda\mu$-terms are given by the grammar

$M ::=$	x	Variable
	$\lambda x:\phi.M$	Abstraction
	MM	Application
	$\langle M, M\rangle$	Pair
	$\mathsf{fst}(M)$	First Projection
	$\mathsf{snd}(M)$	Second Projection
	$\mathsf{inl}(M)$	Left Injection
	$\mathsf{inr}(M)$	Right Injection
	$\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}(x) \to M \parallel \mathsf{inr}(x) \to M$	Conditional
	$[a:\phi]M$	Passivate
	$\mu a:\phi.M$	Activate;

where x is taken from some countable set of λ-variables, ϕ is a well-formed type (formula) and a is taken from some other countable set of μ-variables.

Typing judgements are of the form, $\Gamma \triangleright M:\phi, \Sigma$, where Γ is a set of pairs of λ-variables and types written $x:\psi$, M is a term from the above grammar and Σ denotes a set of pairs of μ-variables and types written $a:\varphi$. For conciseness we drop the convention from the formulation of CL_μ that the active formula is annotated with a bullet, whence the convention that the non-labelled formula on the right of the turnstile is the active formula. The rules for forming valid typing judgements are given in Fig. 2.

Lemma 1. *The following rules are admissible in the λμ-calculus.*

$$\frac{\Gamma \rhd M : \phi, \Sigma}{\Gamma, x : \psi \rhd M : \phi, \Sigma}\ \text{Weakening} \qquad \frac{\Gamma \rhd M : \phi, \Sigma}{\Gamma \rhd M : \phi, \Sigma, a : \psi}\ \text{Weakening}_p$$

$$\frac{\Gamma, x : \psi, y : \psi \rhd M : \phi, \Sigma}{\Gamma, z : \psi \rhd M[x, y := z] : \phi, \Sigma}\ \text{Contraction} \qquad \frac{\Gamma \rhd M : \phi, \Sigma, a : \psi, b : \psi}{\Gamma \rhd M[a, b := c] : \phi, \Sigma, c : \psi}\ \text{Contraction}_p$$

$$\frac{\Gamma \rhd M : \phi, \Sigma \quad \Gamma, x : \phi \rhd N : \psi, \Sigma}{\Gamma \rhd N[x := M] : \psi, \Sigma}\ \text{Substitution}$$

2.2. Reduction rules

There are β-rules corresponding to the introduction–elimination pairs, along with commuting conversions for the disjunction (as these are quite well known, I shall not give them here), a renaming rule and commuting conversions for the Activate rule. These are as follows:

$$\begin{aligned}
(\lambda x : \phi.M)N &\rightsquigarrow_\beta M[x := N] \\
\mathsf{fst}\langle M, N\rangle &\rightsquigarrow_\beta M \\
\mathsf{snd}\langle M, N\rangle &\rightsquigarrow_\beta N \\
\mathsf{case}\ \mathsf{inl}\langle m\rangle\ \mathsf{of}\ \mathsf{inl}\langle x\rangle \to N \| \mathsf{inr}\langle y\rangle \to P &\rightsquigarrow_\beta N[x := M] \\
\mathsf{case}\ \mathsf{inr}\langle x\rangle\ \mathsf{of}\ \mathsf{inl}\langle x\rangle \to N \| \mathsf{inr}\langle y\rangle \to P &\rightsquigarrow_\beta P[y := M] \\
\mu a : \phi.[a : \phi]M &\rightsquigarrow_\beta M \qquad \text{where } a \notin \mu\mathrm{FV}(M) \\
[a : \phi]\mu b : \phi.M &\rightsquigarrow M[b := a] \\
(\mu a : \phi \to \psi.M)N &\rightsquigarrow_c \mu a : \psi.M[a : \phi \to \psi \Leftarrow [a : \psi] \bullet N] \qquad (*) \\
\mathsf{fst}(\mu a : \phi \times \psi.M) &\rightsquigarrow_c \mu a : \phi.M[a : \phi \times \psi \Leftarrow [a : \phi]\mathsf{fst}(\bullet)] \\
\mathsf{snd}(\mu a : \phi \times \psi.M) &\rightsquigarrow_c \mu a : \psi.M[a : \phi \times \psi \Leftarrow [a : \psi]\mathsf{snd}(\bullet)] \\
\mathsf{case}(\mu a : \phi + \psi.M)\ \mathsf{of} &\rightsquigarrow_c \mu a : \varphi.M[a : \phi + \psi \Leftarrow \\
\quad \mathsf{inl}\langle x\rangle \to N \,|\, \mathsf{inr}\langle y\rangle \to P &\qquad [a : \varphi]\mathsf{case} \bullet \mathsf{of}\ \mathsf{inl}\langle x\rangle \to N \| \mathsf{inr}\langle y\rangle \to P]
\end{aligned}$$

$\mu\mathrm{FV}(M)$ denotes the set of free μ-variables in a term M (I shall omit its rather obvious definition). The renaming rule is written $\rightsquigarrow$. In the commuting conversions ($\rightsquigarrow_c$) I have used the notation $M[a \Leftarrow P[\bullet]]$ to denote the term M where *all* occurrences of the subterm $[a]N$ have been (recursively) replaced by the term $P[N]$ (where $P[\bullet]$ is a term with a single hole in it, and $P[N]$ is the term P where the hole has been replaced by N). These complicated commuting conversions are often glossed over in other papers. In truth there is a special case of each conversion depending on the type of the μ-variable a. For example, the first commuting conversion (*) is defined as follows:

$$(\mu a : \phi \to \psi.M)N \rightsquigarrow_c \begin{cases} \mu b : \psi.M[a : \phi \to \psi \Leftarrow [b : \psi] \bullet N] & \text{where } \psi \neq \bot \\ M[a : \phi \to \psi \Leftarrow \bullet N] & \text{where } \psi = \bot \end{cases}$$

where

$$
\begin{aligned}
x[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} x\\
(\lambda x.M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \lambda x.(M[a:\phi \Leftarrow P[\bullet]])\\
(MN)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} (M[a:\phi \Leftarrow P[\bullet]])(N[a:\phi \Leftarrow P[\bullet]])\\
\langle M,N\rangle[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \langle M[a:\phi \Leftarrow P[\bullet]], N[a:\phi \Leftarrow P[\bullet]]\rangle\\
\mathsf{fst}(M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \mathsf{fst}(M[a:\phi \Leftarrow P[\bullet]])\\
\mathsf{snd}(M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \mathsf{snd}(M[a:\phi \Leftarrow P[\bullet]])\\
\mathsf{inl}(M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \mathsf{inl}(M[a:\phi \Leftarrow P[\bullet]])\\
\mathsf{inr}(M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \mathsf{inr}(M[a:\phi \Leftarrow P[\bullet]])\\
(\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}(x)\to N \parallel \mathsf{inr}(y)\to N')[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \mathsf{case}\ M[a:\phi \Leftarrow P[\bullet]]\ \mathsf{of}\\
&\qquad \mathsf{inl}(x)\to N[a:\phi \Leftarrow P[\bullet]] \parallel\\
&\qquad \mathsf{inr}(y)\to N'[a:\phi \Leftarrow P[\bullet]]\\
(\mu b:\psi.M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \mu b:\psi.(M[a:\phi \Leftarrow P[\bullet]])\\
([b:\psi]M)[a:\phi \Leftarrow P[\bullet]] &\stackrel{\text{def}}{=} \begin{cases} P[M[a:\phi \Leftarrow P[\bullet]]] & \text{if } a:\phi = b:\psi\\ [b:\psi](M[a:\phi \Leftarrow P[\bullet]]) & \text{o'wise}\end{cases}
\end{aligned}
$$

The commuting conversions are quite unusual as they involve a substitution of a term for a term, in contrast to the more familiar substitution of a term for a variable. Despite the complexities of the commuting conversions, Parigot has (impressively) shown the following results for this reduction system.

Theorem 2 (Parigot). (1) *The $\lambda\mu$-calculus is strongly normalising*; *and*
(2) *The $\lambda\mu$-calculus is confluent.*

Remark 1. The ability to activate a vacuous μ-variable allows some strange behaviour when considering the *untyped* $\lambda\mu$-calculus. As mentioned by Parigot, the term $\mu a.M$, where a is not a free μ-variable of M, can 'consume' any number of arguments, i.e.

$$(\mu a.M)N_1\cdots N_k \rightsquigarrow_c^k \mu a.M$$

for any number of terms $N_1,\ldots,N_k$.

Remark 2. An interesting question is whether there is a natural computational interpretation of the $\lambda\mu$-calculus. In another paper [15] it is proposed that the $\lambda\mu$-calculus can be thought of as a λ-calculus which is extended with control operators which allow the current continuation to be saved and restored. This can be expressed using a set of single-step reduction rules, from which an operational theory can be developed.

2.3. Comparison with cut-elimination

It is folklore that the sequent calculus formulation of CL has the undesirable feature of several disastrous critical pairs. A simple example of this is the following derivation [23, p. 151]:

$$\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma \vdash \Delta\end{array}}{\Gamma \vdash \Delta, \phi}\ \text{Weakening}_{\mathscr{R}} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\\ \Gamma \vdash \Delta\end{array}}{\Gamma, \phi \vdash \Delta}\ \text{Weakening}_{\mathscr{L}}}{\Gamma \vdash \Delta}\ \text{Cut}$$

Given the usual process of local cut-elimination, it is not clear whether to reduce this proof to π_1 or to π_2. It is interesting to note that this example translates (where I write $\mathscr{M}(\pi)$ to denote the translation of a sequent calculus derivation, π, to a deduction in CL_μ) to the following application of substitution in Parigot's formulation:

$$\mathscr{M}(\pi_2)[x := \mathscr{M}(\pi_1)],$$

where x is *not* a free λ-variable of $\mathscr{M}(\pi_2)$, and so by the definition of substitution, this is equal to

$$\mathscr{M}(\pi_2).$$

Thus Parigot's formulation resolves critical pairs essentially by its syntactic form.

Another important property of Parigot's formulation is that ϕ and $\phi^{\perp\perp}$ are not forced to be equal by the proof theory. Of course, we have the derived rules

$$\dfrac{\dfrac{}{\Gamma, x : \phi \to \perp \rhd x : \phi \to \perp} \quad \dfrac{\begin{array}{c}\vdots\\ \Gamma \rhd M : \phi\end{array}}{\Gamma, x : \phi \to \perp \rhd M : \phi}\ \text{Weakening}}{\dfrac{\Gamma, x : \phi \to \perp \rhd xM : \perp}{\Gamma \rhd \lambda x : \phi \to \perp . xM : (\phi \to \perp) \to \perp}\ \to_{\mathscr{I}}}\ \to_{\mathscr{E}}$$

and

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma \rhd M : (\phi \to \perp) \to \perp\end{array}}{\Gamma, x : \phi \rhd [a : \phi]\, x : \perp, a : \phi}\ \text{Weakening}_p \quad \dfrac{\dfrac{}{\Gamma, x : \phi \rhd x : \phi}\ \text{Passivate}}{\Gamma \rhd M : (\phi \to \perp) \to \perp, a : \phi}\ \to_{\mathscr{I}}}{\dfrac{\Gamma \rhd M(\lambda x.[a : \phi]\, x) : \perp, a : \phi}{\Gamma \rhd \mu a : \phi . M(\lambda x.[a : \phi]\, x) : \phi}\ \text{Activate}}\ \to_{\mathscr{E}}$$

Composing the first with the second gives

$$\begin{aligned}\mu a.(\lambda x.xM)(\lambda x.[a]x) &\rightsquigarrow_\beta \mu a.(\lambda x.[a]x)M\\ &\rightsquigarrow_\beta \mu a.[a]M\\ &\rightsquigarrow_\beta M;\end{aligned}$$

but composing the second with the first yields

$$\lambda y.y(\mu a.M(\lambda x.[a]x)$$

which is in (head) normal form.[2]

2.4. Further consideration on normal forms

One motivation for the commuting conversion given in Section 2.2 is that the Activate rule can act as a barrier between an introduction–elimination pair and so we add a reduction to remove it. This has both a familiar and unfamiliar feel to it. We are used to this notion of commuting conversions to permit β-reductions when considering the disjunction in IL. However in this case, it introduces a new, unfamiliar, form of substitution, *textual* substitution, where whole subterms are replaced.

One could take these ideas further. Gentzen [21] suggested adding the rule

$$\dfrac{\begin{array}{c}[\neg\phi]\\ \vdots\\ \bot\end{array}}{\phi}\ \text{RAA}$$

to the natural deduction formulation of IL to get a formulation of CL. However, Prawitz [33] noted that applications of this rule can be restricted to cases where ϕ is *atomic*. This is achieved by both factoring formulae through the de Morgan dualities (thus eliminating certain problematic connectives) and by transformation. For example, an application of the above rule where $\phi = \phi \to \psi$ is transformed to

$$\dfrac{\dfrac{\dfrac{\dfrac{[\phi\to\psi]\quad[\phi]}{\psi}\to_{\mathscr{E}}\qquad[\neg\psi]}{\bot}}{\neg(\phi\to\psi)}\to_{\mathscr{E}}\quad\begin{array}{c}\vdots\\ \bot\end{array}}{\dfrac{\psi}{\phi\to\psi}\ \text{RAA}}\to_{\mathscr{I}}$$

where clearly the size of the formula used in the application of the RAA rule has been reduced. Prawitz suggests transforming all applications of this rule until they involve only atomic formulae. However the use of the de Morgan dualities is vital here; Prawitz [32, Footnote 1, p. 50] mentions that this technique does not extend to all the connectives (the problematic one being the disjunction).

[2] This property enables Ong [29] to define a categorical model. It is well known that a CCC with an isomorphism $A^{\perp\perp} \cong A$ collapses to a boolean algebra.

Ong [29] suggests a similar strategy for the $\lambda\mu$-calculus by rewriting applications of the Activate rule until they are of atomic type, although his motivation is to ensure confluence when considering η-reduction. Given that this technique requires the use of the formula equivalences when considering all the connectives, it is not considered any further.

3. Classical linear logic

Linear logic is the logic obtained by removing the structural rules of Weakening and Contraction. This has the effect of refining the traditional connectives into two different kinds: multiplicative and additive. Of course what remains is a terribly weak logic. To regain full logical power the structural rules are re-introduced but in a controlled way, via the exponentials. A fuller introduction to linear logic can be found, for example, in Troelstra's book [36], the article by Lincoln [27] or the original article by Girard [22].

3.1. The linear $\lambda\mu$-calculus

Unlike the case for IL and CL, the grammars for intuitionistic linear and classical linear formulae are different. For CLL the grammar is

$$\phi ::= p \mid \phi \otimes \phi \mid \phi \multimap \phi \mid \phi \& \phi \mid \phi \oplus \phi \mid \phi \parr \phi \mid !\phi \mid ?\phi,$$

where p is taken from some countable set of atomic formulae which contains the distinguished elements I (the unit for $\otimes$), $\mathbf{t}$ (the unit for $\&$), $\mathbf{f}$ (the unit for $\oplus$) and $\bot$ (the unit for $\parr$).

It is possible to extend a natural deduction formulation of ILL [8, 10] using Parigot's methodology, outlined in Section 2. However this process is not entirely straightforward. Firstly, it has to be established what the unit is in the linear equivalent of the Passivate and Activate rules. It turns out that it is, $\bot$, the unit for Par ($\parr$).[3] Also, some of the connectives are difficult to handle directly and they shall be defined as follows (cf. Section 3.4).

$$\begin{aligned} \phi^{\bot} &\stackrel{\text{def}}{=} \phi \multimap \bot \\ ?\phi &\stackrel{\text{def}}{=} (!\phi^{\bot})^{\bot} \\ \phi \parr \psi &\stackrel{\text{def}}{=} ((\phi^{\bot}) \otimes (\psi^{\bot}))^{\bot} \end{aligned}$$

A surprise is that the Promotion rule has to be extended for the classical formulation. It seems that, rather, its ILL formulation is a particular instance of the full classical formulation.

[3] As the Passivate rule really introduces the par unit and the Activate rule eliminates it, they shall be considered as normal introduction and elimination rules.

$$\frac{}{\phi \vdash \phi^{\bullet}}\ \text{Identity}$$

$$\frac{\Gamma, \phi \vdash \psi^{\bullet}, \Sigma}{\Gamma \vdash (\phi \multimap \psi)^{\bullet}, \Sigma} \multimap_{\mathscr{I}} \qquad \frac{\Gamma \vdash (\phi \multimap \psi)^{\bullet}, \Sigma \qquad \Delta \vdash \phi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash \psi^{\bullet}, \Sigma, \Sigma'} \multimap_{\mathscr{E}}$$

$$\frac{}{\vdash I^{\bullet}} I_{\mathscr{I}} \qquad \frac{\Gamma \vdash I^{\bullet}, \Sigma \qquad \Delta \vdash \phi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash \phi^{\bullet}, \Sigma, \Sigma'} I_{\mathscr{E}}$$

$$\frac{\Gamma \vdash \phi^{\bullet}, \Sigma \qquad \Delta \vdash \psi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash (\phi \otimes \psi)^{\bullet}, \Sigma, \Sigma'} \otimes_{\mathscr{I}}$$

$$\frac{\Gamma \vdash (\phi \otimes \psi)^{\bullet}, \Sigma \qquad \Delta, \phi, \psi \vdash \varphi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash \varphi^{\bullet}, \Sigma, \Sigma'} \otimes_{\mathscr{E}}$$

$$\frac{\Gamma \vdash \phi^{\bullet}, \Sigma \qquad \Gamma \vdash \psi^{\bullet}, \Sigma}{\Gamma \vdash (\phi \,\&\, \psi)^{\bullet}, \Sigma} \,\&_{\mathscr{I}}$$

$$\frac{\Gamma \vdash (\phi \,\&\, \psi)^{\bullet}, \Sigma}{\Gamma \vdash \phi^{\bullet}, \Sigma} \,\&_{\mathscr{E}-1} \qquad \frac{\Gamma \vdash (\phi \& \psi)^{\bullet}, \Sigma}{\Gamma \vdash \psi^{\bullet}, \Sigma} \,\&_{\mathscr{E}-2}$$

$$\frac{\Gamma \vdash \phi^{\bullet}, \Sigma}{\Gamma \vdash (\phi \oplus \psi)^{\bullet}, \Sigma} \oplus_{\mathscr{I}-1} \qquad \frac{\Gamma \vdash \psi^{\bullet}, \Sigma}{\Gamma \vdash (\phi \oplus \psi)^{\bullet}, \Sigma} \oplus_{\mathscr{I}-2}$$

$$\frac{\Gamma \vdash (\phi \oplus \psi)^{\bullet}, \Sigma \qquad \Delta, \phi \vdash \varphi^{\bullet}, \Sigma' \qquad \Delta, \psi \vdash \varphi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash \varphi^{\bullet}, \Sigma, \Sigma'} \oplus_{\mathscr{E}}$$

$$\frac{\begin{array}{cc} \Gamma_1 \vdash !\phi_1^{\bullet}, \Sigma_1 & \Delta_1 \vdash !\varphi_1^{\bullet}, \Upsilon_1 \\ \Gamma_n \vdash !\phi_n^{\bullet}, \Sigma_n & \Delta_m \vdash !\varphi_m^{\bullet}, \Upsilon_m \\ !\phi_1, \ldots, !\phi_n \vdash \psi^{\bullet}, (!\varphi_1 \multimap \bot)^{a_1}, \ldots, (!\varphi_m \multimap \bot)^{a_m} & \end{array}}{\vec{\Gamma}, \vec{\Delta} \vdash !\psi^{\bullet}, \vec{\Sigma}, \vec{\Upsilon}}\ \text{Promotion}$$

$$\frac{\Gamma \vdash !\phi^{\bullet}, \Sigma}{\Gamma \vdash \phi^{\bullet}, \Sigma}\ \text{Dereliction}$$

$$\frac{\Gamma \vdash !\phi^{\bullet}, \Sigma \qquad \Delta \vdash \psi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash \psi^{\bullet}, \Sigma, \Sigma'}\ \text{Weakening}$$

$$\frac{\Gamma \vdash !\phi^{\bullet}, \Sigma \qquad \Delta, !\phi, !\phi \vdash \psi^{\bullet}, \Sigma'}{\Gamma, \Delta \vdash \psi^{\bullet}, \Sigma, \Sigma'}\ \text{Contraction}$$

$$\frac{\Gamma \vdash \phi^{\bullet}, \Sigma}{\Gamma \vdash \bot^{\bullet}, \phi^{a}, \Sigma} \bot_{\mathscr{I}} \qquad \frac{\Gamma \vdash \bot^{\bullet}, \phi^{a}, \Sigma}{\Gamma \vdash \phi^{\bullet}, \Sigma} \bot_{\mathscr{E}}$$

Fig. 3. Natural deduction formulation of CLL: CLL_{μ}

The natural deduction formulation of CLL, CLL_{μ}, is given in Fig. 3. Again the $\bot_{\mathscr{I}}$ rule is only permitted if the formula being passivated is not $\bot$. This formulation is sound and complete in the usual sense.

Theorem 3. $\vdash_{\mathrm{CLL}} \Gamma \vdash \Delta$ *iff* $\vdash_{\mathrm{CLL}_\mu} \Gamma \vdash \Delta$.

Applying the Curry–Howard correspondence to CLL_μ yields the (typed) linear $\lambda\mu$-calculus – an extension of the linear λ-calculus [7]. Raw terms are given by the grammar

$M ::= x$		Variable
	$\lambda x{:}\,\phi.M$	Abstraction
	MM	Application
	$M \otimes M$	Multiplicative Pair
	$\mathsf{let}\ M\ \mathsf{be}\ x \otimes x\ \mathsf{in}\ M$	Split
	$\langle M, M\rangle$	Additive Pair
	$\mathsf{fst}(M)$	First Projection
	$\mathsf{snd}(M)$	Second Projection
	$\mathsf{inl}(M)$	Left Injection
	$\mathsf{inr}(M)$	Right Injection
	$\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}(x) \to M \parallel \mathsf{inr}(x) \to M$	Conditional
	$\mathsf{promote}\ \vec{M} \mid \vec{M}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ M$	Promote
	$\mathsf{derelict}(M)$	Derelict
	$\mathsf{discard}\ M\ \mathsf{in}\ M$	Discarding
	$\mathsf{copy}\ M\ \mathsf{as}\ x,\ x\ \mathsf{in}\ M$	Duplication
	$[a{:}\,\phi]\,M$	Passivate
	$\mu a{:}\,\phi.M$	Activate,

where, as for $\lambda\mu$-calculus, x is taken from some countable set of λ-variables, ϕ is a well-formed type (formula) and a is taken from some countable set of μ-variables.

Typing judgements are now of the form, $\Gamma \triangleright M{:}\,\phi, \Sigma$, where Γ is a *multiset* of pairs of λ-variables and types, written $x{:}\,\psi$, M is a term from the above grammar and Σ denotes a *multiset* of pairs of μ- variables and types, written $a{:}\,\varphi$. As is the case for ILL, in well-typed terms of the multiplicative-exponential fragment $(\otimes, \multimap, \parr, !, ?)$ variables occur exactly once. The rules for forming a valid typing judgement are given in Fig. 4.

Lemma 2. *The following is an admissible rule*:

$$\frac{\Gamma \triangleright M{:}\,\phi, \Sigma \qquad \Gamma', x{:}\,\phi \triangleright N{:}\,\psi, \Sigma'}{\Gamma, \Gamma' \triangleright N[x := M]{:}\,\psi, \Sigma, \Sigma'}\ \text{Substitution}$$

3.2. Reduction rules

From the linear λ-calculus there are both β-rules and commuting conversions. Of course, the reductions for the Promotion rule have to be suitably extended. The β-rules

$$\frac{}{x\colon \phi \rhd x\colon \phi}\ \text{Identity}$$

$$\frac{\Gamma, x\colon \phi \rhd M\colon \psi, \Sigma}{\Gamma \rhd \lambda x\colon \phi.M\colon \phi \multimap \psi, \Sigma}\multimap_{\mathscr{I}} \qquad \frac{\Gamma \rhd M\colon \phi \multimap \psi, \Sigma \quad \Delta \rhd N\colon \phi, \Sigma'}{\Gamma, \Delta \rhd MN\colon \psi, \Sigma, \Sigma'}\multimap_{\mathscr{E}}$$

$$\frac{}{\rhd *\colon I}I_{\mathscr{I}} \qquad \frac{\Gamma \rhd M\colon I, \Sigma \quad \Delta \rhd N\colon \phi, \Sigma'}{\Gamma, \Delta \rhd \ \text{let}\ M\ \text{be}\ *\ \text{in}\ N\colon \phi, \Sigma, \Sigma'}I_{\mathscr{E}}$$

$$\frac{\Gamma \rhd M\colon \phi, \Sigma \quad \Delta \rhd N\colon \psi, \Sigma'}{\Gamma, \Delta \rhd M \otimes N\colon \phi \otimes \psi, \Sigma, \Sigma'}\otimes_{\mathscr{I}}$$

$$\frac{\Gamma \rhd M\colon \phi \otimes \psi, \Sigma \quad \Delta, x\colon \phi, y\colon \psi \rhd N\colon \varphi, \Sigma'}{\Gamma, \Delta \rhd \text{let}\ M\ \text{be}\ x \otimes y\ \text{in}\ N\colon \varphi, \Sigma, \Sigma'}\otimes_{\mathscr{E}}$$

$$\frac{\Gamma \rhd M\colon \phi, \Sigma \quad \Gamma \rhd N\colon \psi, \Sigma}{\Gamma \rhd \langle M, N\rangle\colon \phi \,\&\, \psi, \Sigma}\,\&_{\mathscr{I}}$$

$$\frac{\Gamma \rhd M\colon \phi \,\&\, \psi, \Sigma}{\Gamma \rhd \text{fst}(M)\colon \phi, \Sigma}\,\&_{\mathscr{E}-1} \qquad \frac{\Gamma \rhd M\colon \phi \,\&\, \psi, \Sigma}{\Gamma \rhd \text{snd}(M)\colon \psi, \Sigma}\,\&_{\mathscr{E}-2}$$

$$\frac{\Gamma \rhd M\colon \phi, \Sigma}{\Gamma \rhd \text{inl}(M)\colon \phi \oplus \psi, \Sigma}\oplus_{\mathscr{I}-1} \qquad \frac{\Gamma \rhd M\colon \psi, \Sigma}{\Gamma \rhd \text{inr}(M)\colon \phi \oplus \psi, \Sigma}\oplus_{\mathscr{I}-2}$$

$$\frac{\Gamma \rhd M\colon \phi \oplus \psi, \Sigma \quad \Delta, x\colon \phi \rhd N\colon \varphi, \Sigma' \quad \Delta, y\colon \psi \rhd P\colon \varphi, \Sigma'}{\Gamma, \Delta \rhd \text{case}\ M\ \text{of}\ \ \text{inl}(x) \to N \,\|\, \text{inr}(y) \to P\colon \varphi, \Sigma, \Sigma'}\oplus_{\mathscr{E}}$$

$$\frac{\begin{array}{ll}\Gamma_1 \rhd M_1\colon !\phi_1, \Sigma_1 & \Delta_1 \rhd P_1\colon !\varphi_1, \Upsilon_1\\ \Gamma_n \rhd M_n\colon !\phi_n, \Sigma_n & \Delta_m \rhd P_m\colon !\varphi_m, \Upsilon_m\\ x_1\colon !\phi_1, \ldots, x_n\colon !\phi_n \rhd N\colon \psi, a_1\colon !\varphi_1 \multimap \bot, \ldots, a_m\colon !\varphi_m \multimap \bot & \end{array}}{\vec{\Gamma}, \vec{\Delta} \rhd \text{promote}\ \vec{M} \mid \vec{P}\ \text{for}\ \vec{x} \mid \vec{a}\ \text{in}\ N\colon !\psi, \vec{\Sigma}, \vec{\Upsilon}}\ \text{Promotion}$$

$$\frac{\Gamma \rhd M\colon !\phi, \Sigma}{\Gamma \rhd \text{derelict}(M)\colon \phi, \Sigma}\ \text{Dereliction}$$

$$\frac{\Gamma \rhd M\colon !\phi, \Sigma \quad \Delta \rhd N\colon \psi, \Sigma'}{\Gamma, \Delta \rhd \text{discard}\, M\, \text{in}\ N\colon \psi, \Sigma, \Sigma'}\ \text{Weakening}$$

$$\frac{\Gamma \rhd M\colon !\phi, \Sigma \quad \Delta, x\colon !\phi, y\colon !\phi \rhd N\colon \psi, \Sigma'}{\Gamma, \Delta \rhd \ \text{copy}\, M\ \text{as}\ x, y\ \text{in}\ N\colon \psi, \Sigma, \Sigma'}\ \text{Contraction}$$

$$\frac{\Gamma \rhd M\colon \phi, \Sigma}{\Gamma \rhd [a\colon \phi]M\colon \bot, a\colon \phi, \Sigma}\bot_{\mathscr{I}} \qquad \frac{\Gamma \rhd M\colon \bot, a\colon \phi, \Sigma}{\Gamma \rhd \mu a\colon \phi.M\colon \phi, \Sigma}\bot_{\mathscr{E}}$$

Fig. 4. The linear $\lambda\mu$-calculus.

are as follows:

$$(\lambda x{:}\,\phi.M){:}N \rightsquigarrow_\beta M[x := N]$$

$$\mathsf{let}\ M\otimes N\ \mathsf{be}\ x\otimes y\ \mathsf{in}\ P \rightsquigarrow_\beta P[x := M, y := N]$$

$$\mathsf{fst}(\langle M,N\rangle) \rightsquigarrow_\beta M$$

$$\mathsf{snd}(\langle M,N\rangle) \rightsquigarrow_\beta N$$

$$\mathsf{case}\ \mathsf{inl}(M)\ \mathsf{of}\ \mathsf{inl}(x)\to N \parallel \mathsf{inr}(y)\to P \rightsquigarrow_\beta N[x := M]$$

$$\mathsf{case}\ \mathsf{inr}(M)\ \mathsf{of}\ \mathsf{inl}(x)\to N \parallel \mathsf{inr}(y)\to P \rightsquigarrow_\beta P[y := M]$$

$$\mathsf{derelict}(\mathsf{promote}\ \vec{M} \mid \vec{P}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ N) \rightsquigarrow_\beta N\,[x_i := M_i, a_j{:}\,!\varphi_j \multimap \bot \Leftarrow (\bullet P_j)]$$

$$\mathsf{discard}\ (\mathsf{promote}\ \vec{M} \mid \vec{P}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ N)\ \mathsf{in}\ R \rightsquigarrow_\beta \mathsf{discard}\ \vec{M}, \vec{P}\ \mathsf{in}\ R$$

$$\begin{aligned}\mathsf{copy}\ (\mathsf{promote}\ \vec{M} \mid \vec{P}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ N)\ \mathsf{as}\ y,z\ \mathsf{in}\ R \rightsquigarrow_\beta\ & \mathsf{copy}\ \vec{M}\ \mathsf{as}\ \vec{x'}, \vec{x''}\ \mathsf{in}\\ & \mathsf{copy}\ \vec{P}\ \mathsf{as}\ \vec{w'}, \vec{w''}\ \mathsf{in}\\ & R\,[y := \mathsf{promote}\ \vec{x'} \mid \vec{w'}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ N,\\ & z := \mathsf{promote}\ \vec{x''} \mid \vec{w''}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ N]\end{aligned}$$

Rather than give all the commuting conversions for the ILL connectives (they are given in full in [10]), I shall only give the 'promote-of-promote' one, which for ease of typesetting I shall present as two reduction rules depending on where the inner Promotion rule occurs. First, I need to define an important term, written $\mathscr{P}_a$, which is given by the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x{:}\,!\phi \triangleright x{:}\,!\phi}}{x{:}\,!\phi \triangleright [b{:}\,!\phi]x{:}\,\bot, b{:}\,!\phi}\bot_{\mathscr{I}}}{\emptyset \triangleright \lambda x{:}\,!\phi.[b{:}\,!\phi]\,x{:}\,!\phi\multimap \bot, b{:}\,!\phi}\multimap_{\mathscr{I}}}{\emptyset \triangleright [a{:}\,!\phi\multimap \bot]\,\lambda x{:}\,!\phi.[b{:}\,!\phi]\,x{:}\bot, b{:}\,!\phi, a{:}\,!\phi\multimap \bot}\bot_{\mathscr{I}}}{\emptyset \triangleright \mu b{:}\,!\phi\,.[a{:}\,!\phi\multimap \bot]\,\lambda x{:}\,!\phi.[b{:}\,!\phi]\,x{:}\,!\phi, a{:}\,!\phi\multimap \bot}\bot_{\mathscr{E}}$$

where a is the final passive variable. I use the shorthand $\mathscr{P}_{\vec{a}}$ to represent the obvious extension of the above term to multiple passive variables. The commuting conversions for the Promotion rule are then

$$\begin{aligned}&\mathsf{promote}\ (\mathsf{promote}\ \vec{M} \mid \vec{N}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ P) \mid \vec{Q}\ \mathsf{for}\ y \mid \vec{b}\ \mathsf{in}\ R\\ &\quad \rightsquigarrow_c \mathsf{promote}\ \vec{M} \mid \vec{N}, \vec{Q}\ \mathsf{for}\ \vec{x'} \mid \vec{a'}, \vec{b}\ \mathsf{in}\ R[y := (\mathsf{promote}\ \vec{x'} \mid \mathscr{P}_{\vec{a'}}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ P)]\end{aligned}$$

and

$$\begin{aligned}&\mathsf{promote}\ \vec{M} \mid (\mathsf{promote}\ \vec{N} \mid \vec{P}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ Q)\ \mathsf{for}\ \vec{y} \mid b\ \mathsf{in}\ R\\ &\quad \rightsquigarrow_c \mathsf{promote}\ \vec{M},\vec{N} \mid \vec{P}\ \mathsf{for}\ \vec{y},\vec{x'} \mid \vec{a'}\ \mathsf{in}\\ &\qquad R[b{:}\,!\phi \multimap \bot \Leftarrow \bullet\,(\mathsf{promote}\ \vec{x'} \mid \mathscr{P}_{\vec{a'}}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ Q)].\end{aligned}$$

There is a β-rule corresponding to the introduction-elimination pair for the $\bot$,[4] and a number of commuting conversions for this unit (as per the discussion in Section 2.2). These are as follows:

$$\begin{aligned}\mu a{:}\,\phi\,.[a{:}\,\phi]\,M &\rightsquigarrow_\beta M\\ (\mu a{:}\,\phi \multimap \psi\,.M)N &\rightsquigarrow_c \mu a{:}\,\psi\,.M[a{:}\,\phi \multimap \psi \Leftarrow [a{:}\,\psi] \bullet N]\\ \mathsf{let}\ \mu a{:}\,\phi \otimes \psi\,.M\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ N &\rightsquigarrow_c \mu a{:}\,\varphi\,.M[a{:}\,\phi \otimes \psi \Leftarrow [a{:}\,\varphi]\ \mathsf{let}\ \bullet\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ N]\\ \mathsf{fst}(\mu a{:}\,\phi \,\&\, \psi\,.M) &\rightsquigarrow_c \mu a{:}\,\phi\,.M[a{:}\,\phi \,\&\, \psi \Leftarrow [a{:}\,\phi]\ \mathsf{fst}(\bullet)]\\ \mathsf{snd}(\mu a{:}\,\phi \,\&\, \psi\,.M) &\rightsquigarrow_c \mu a{:}\,\psi\,.M[a{:}\,\phi \,\&\, \psi \Leftarrow [a{:}\,\psi]\ \mathsf{snd}(\bullet)]\\ \mathsf{case}\ (\mu a{:}\,\phi \oplus \psi\,.M)\ \mathsf{of}&\\ \mathsf{inl}(x) \to N \parallel \mathsf{inr}(y) \to P &\rightsquigarrow_c \mu a{:}\,\varphi.Ma{:}\,\phi \oplus \psi \Leftarrow [a{:}\,\varphi]\ \mathsf{case}\ \bullet\ \mathsf{of}\ \mathsf{inl}(x) \to N \parallel \mathsf{inr}(y) \to P\\ \mathsf{derelict}(\mu a{:}\,!\phi\,.M) &\rightsquigarrow_c \mu a{:}\,\phi\,.M[a{:}\,!\phi \Leftarrow [a{:}\,\phi]\ \mathsf{derelict}(\bullet)]\\ \mathsf{copy}\ (\mu a{:}\,!\phi\,.M)\ \mathsf{as}\ x, y\ \mathsf{in}\ N &\rightsquigarrow_c \mu a{:}\,\varphi\,.M[a{:}\,!\phi \Leftarrow [a{:}\,\varphi]\ \mathsf{copy}\ \bullet\ \mathsf{as}\ x, y\ \mathsf{in}\ N]\\ \mathsf{discard}\ (\mu a{:}\,!\psi\,.M)\ \mathsf{in}\ N &\rightsquigarrow_c \mu a{:}\,\varphi\,.M[a{:}\,!\phi \Leftarrow [a{:}\,\varphi]\ \mathsf{discard}\ \bullet\ \mathsf{in}\ N]\end{aligned}$$

Of course there are η-rules for the connectives. Those for the ILL connectives have appeared elsewhere [10, Fig. 4.3] and the new one, for the $\bot$, is

$$[b{:}\,\phi]\,\mu a{:}\,\phi\,.M \rightsquigarrow_\eta M[a := b]$$

A vital property of this formulation is the so-called subject reduction property.

Theorem 4. *If* $\Gamma \triangleright M{:}\,\phi, \Sigma$ *and* $M \rightsquigarrow_{\beta,c} N$ *then* $\Gamma \triangleright N{:}\,\phi, \Sigma$.

3.3. Subformula property

It is worth pointing out that for the obvious definition of subformula, this formulation fails to satisfy the subformula property, i.e. in a (β, c)-normal derivation it need not be the case that all formulae are subformulae of the open hypotheses or conclusions.

[4] Here is an advantage of linearity: we need no side-condition for this rule as we do for the non-linear system.

An example of this failure is the following (β, c)-normal derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{!(\phi\multimap\bot)\vdash !(\phi\multimap\bot)^\bullet}{!(\phi\multimap\bot)\vdash \phi\multimap\bot^\bullet}\text{Der.} \quad \dfrac{!\phi\vdash !\phi^\bullet}{!\phi\vdash\phi^\bullet}\text{Der.}}{!(\phi\multimap\bot),!\phi\vdash\bot^\bullet}\multimap_{\mathscr{E}}}{!(\phi\multimap\bot)\vdash !\phi\multimap\bot^\bullet}\multimap_{\mathscr{I}}}{!(\phi\multimap\bot)\vdash\bot^\bullet,!\phi\multimap\bot^a}\bot_{\mathscr{I}} \quad !\psi\vdash !\psi^\bullet}{!(\phi\multimap\bot),!\psi\vdash\bot^\bullet,!\phi\multimap\bot^a}\text{Weak.}}{!(\phi\multimap\bot)\vdash !\psi\multimap\bot^\bullet,!\phi\multimap\bot^a}\multimap_{\mathscr{I}} \quad !(\phi\multimap\bot)\vdash !(\phi\multimap\bot)^\bullet !\phi\vdash !\phi^\bullet}{!(\phi\multimap\bot),!\phi\vdash !(!\psi\multimap\bot)^\bullet}\text{Promotion}$$

Unfortunately, the formula $!\phi\multimap\bot$ is not a subformula of the hypotheses or the conclusion. Devising an appropriate notion of subformula remains future work.

3.4. The Par connective

It is possible to extend the $\lambda\mu$-calculus to allow the Activate rule to bind multiple formulae (and similarly for the Passivate rule). In the linear setting this enables a direct formulation of the par ('$\parr$') connective, as follows:

$$\dfrac{\Gamma\triangleright M:\bot,a:\phi,b:\psi,\Sigma}{\Gamma\triangleright \parr\mu(a:\phi,b:\psi).M:\phi\parr\psi,\Sigma}\parr_{\mathscr{I}} \qquad \dfrac{\Gamma\triangleright M:\phi\parr\psi,\Sigma}{\Gamma\triangleright \parr[a:\phi,b:\psi]\,M:\bot,a:\phi,b:\psi,\Sigma}\parr_{\mathscr{E}}$$

A fuller discussion of this alternative formulation will appear in [37].

4. Strong normalisation

It is possible to prove strong normalisation for linear $\lambda\mu$-calculus directly by reworking Parigot's original proof [31]. However, I shall prove it by demonstrating a translation from linear $\lambda\mu$-calculus to the second-order $\lambda\mu$-calculus ($\lambda\mu 2$-calculus), which allows the result to be proved by appealing to the strong normalisation property of $\lambda\mu 2$-calculus. The translation is an adaptation of that used by Benton [6] to prove strong normalisation for the linear λ-calculus.

The method is as follows: If we wish to prove strong normalisation for a calculus $\mathscr{C}_1$, knowing already that strong normalisation holds for a calculus $\mathscr{C}_2$, then it suffices to exhibit a translation $M\mapsto M^\circ$ from $\mathscr{C}_1$ to $\mathscr{C}_2$ with the property that if $M\rightsquigarrow N$ then $M^\circ\rightsquigarrow^+ N^\circ$. Thus if there were an infinite reduction sequence

$$M_0\rightsquigarrow M_1\rightsquigarrow M_2\rightsquigarrow\cdots$$

in $\mathscr{C}_1$, then there would be an infinite sequence

$$M_0^\circ \rightsquigarrow^+ M_1^\circ \rightsquigarrow^+ M_2^\circ \rightsquigarrow^+ \cdots$$

in $\mathscr{C}_2$, contradicting strong normalisation for that calculus.

The $\lambda\mu 2$-calculus is given by extending the $\lambda\mu$-calculus of Section 2 with the rules

$$\frac{\Gamma \triangleright M : \phi, \Sigma}{\Gamma \triangleright \Lambda X.M : \forall X.\phi, \Sigma}\forall_{\mathscr{I}} \quad \text{and} \quad \frac{\Gamma \triangleright M : \forall X.\phi, \Sigma}{\Gamma \triangleright M\psi : \phi[\psi/X], \Sigma}\forall_{\mathscr{E}}$$

and the reduction rule

$$(\forall X.M)\phi \rightsquigarrow_\beta M[\phi/X].$$

Parigot [31] has shown that the reduction of terms always terminates (including, importantly, the commuting conversion marked $(*)$ in Section 2.2).

Theorem 5 (Parigot). *All well-typed terms of $\lambda\mu 2$-calculus are strongly normalising.*

An important feature of the translation is an encoding of coinductive types. What follows is taken from Benton's paper, to which the reader is referred. Let $\Phi(X)$ be a $\lambda\mu 2$-type where X appears positively; its greatest fixed point is given by

$$\nu_\Phi = \exists X.(X \to \Phi(X)) \times X,$$

where

$$\exists X.\psi \stackrel{\text{def}}{=} \forall Y.(\forall X.\psi \to Y) \to Y.$$

Expanding this out gives

$$\nu_\Phi \stackrel{\text{def}}{=} \forall Y.(\forall X.(X \to \Phi(X)) \to X \to Y) \to Y.$$

Terms of type μ_Φ are built using

$$\begin{aligned} build_\Phi \;&:\; \forall X.(X \to \Phi(X)) \to X \to \nu_\Phi \\ &\stackrel{\text{def}}{=} \Lambda X.\lambda f.\lambda x.\Lambda C.\lambda h.(hXfx) \end{aligned}$$

along with its associated destructor

$$\begin{aligned} out_\Phi : \nu_\Phi &\to \Phi(\nu_\Phi) \\ &\stackrel{\text{def}}{=} \lambda m.m\Phi(\nu_\Phi)(\Lambda X.\lambda f.\lambda x.(\Phi[build_\Phi Xf](fx))). \end{aligned}$$

The translation of linear types to $\lambda\mu 2$-types is given inductively as follows:[5]

$$
\begin{aligned}
(\phi \multimap \psi)^\circ &\stackrel{\text{def}}{=} \phi^\circ \to \psi^\circ \\
(\phi \otimes \psi)^\circ &\stackrel{\text{def}}{=} \forall X.(\phi^\circ \to \psi^\circ \to X) \to X \\
\bot^\circ &\stackrel{\text{def}}{=} \bot \\
(\phi \,\&\, \psi) &\stackrel{\text{def}}{=} \phi^\circ \times \psi^\circ \\
(\phi \oplus \psi) &\stackrel{\text{def}}{=} \phi^\circ + \psi^\circ \\
(!\varphi)^\circ &\stackrel{\text{def}}{=} v_{\Phi^\varphi \circ}
\end{aligned}
$$

where

$$\Phi^\varphi(X) \stackrel{\text{def}}{=} (\forall Z.Z \to Z) \times \varphi \times (\forall Z.(X \to X \to Z) \to Z).$$

This translation can be lifted to terms in a straightforward manner. The details are essentially the same as in Benton's paper. The important new case is the Promotion rule.

$$\frac{\begin{array}{c}\Gamma_1 \rhd M_1 : !\phi_1, \Sigma_1 \quad \Delta_1 \rhd P_1 : !\varphi_1, \Upsilon_1 \\ \Gamma_n \rhd M_n : !\phi_n, \Sigma_n \quad \Delta_m \rhd P_m : !\varphi_m, \Upsilon_m \\ x_1 : !\phi_1, \ldots, x_n : !\phi_n \rhd N : \psi, a_1 : !\varphi_1 \multimap \bot, \ldots, a_m : !\varphi_m \multimap \bot\end{array}}{\vec{\Gamma}, \vec{\Delta} \rhd \text{promote } \vec{M} \mid \vec{P} \text{ for } \vec{x} \mid \vec{a} \text{ in } N : !\psi, \vec{\Sigma}, \vec{\Upsilon}} \text{ Promotion}$$

which is translated to

$$(\vec{\Gamma}, \vec{\Delta})^\circ \rhd build_{\Phi^\psi \circ} \left(\prod_{i=1}^{n} (!\phi_i)^\circ \times \prod_{j=1}^{m} (!\varphi_j)^\circ \right) \; hx : (!\psi)^\circ, (\vec{\Sigma}, \vec{\Upsilon})^\circ$$

where

$$
\begin{aligned}
x &\stackrel{\text{def}}{=} \langle M_1^\circ, \ldots, M_n^\circ, \; P_1^\circ, \ldots, P_m^\circ \rangle \\
h &\stackrel{\text{def}}{=} \lambda p.\langle A, B, C \rangle \\
A &\stackrel{\text{def}}{=} \Lambda Z.\lambda z.discard\,(\pi_1 p)\; in \cdots discard\,(\pi_n p)\; in \\
&\qquad discard\,(\pi_{n+1} p)\; in \cdots discard\,(\pi_{n+m} p)\; in\, z \\
B &\stackrel{\text{def}}{=} \mu b : \psi^\circ.(\mu a_m : (!\varphi_m \multimap \bot)^\circ. \cdots \mu a_1 : (!\varphi_1 \multimap \bot)^\circ.[b : \psi^\circ] B'(\pi_{n+1} p))(\pi_{n+m} p) \\
B' &\stackrel{\text{def}}{=} (\lambda x_1. \cdots \lambda x_n.N^\circ)(\pi_1 p) \cdots (\pi_n p) \\
C &\stackrel{\text{def}}{=} \Lambda Z.\lambda g.copy\,(\pi_1 p)\; as\; x_1', x_1''\; in \cdots \\
&\qquad copy\,(\pi_{n+m} p)\; as\; x_{n+m}', x_{n+m}''\; in\, (g\langle x_1', \ldots, x_{n+m}' \rangle \langle x_1'', \ldots, x_{n+m}'' \rangle)
\end{aligned}
$$

[5] Obviously, this translation could be extended to the *second-order* linear $\lambda\mu$-calculus, i.e.

$$(\forall X.\phi)^\circ \stackrel{\text{def}}{=} \forall X.\phi^\circ$$

and *discard* and *copy* are defined as in Benton's paper and π_k is the obvious kth projection term. We are then in a position to consider the various cases of reduction in the linear $\lambda\mu$-calculus. I shall present only one case, the rest are essentially as given in Benton's paper.

Consider the reduction

$$\mathsf{derelict}(\text{promote } \vec{M} \mid \vec{P} \text{ for } \vec{x} \mid \vec{a} \text{ in } N) \rightsquigarrow_\beta N\,[\vec{x} := \vec{M}, a_i\colon !\varphi_i \multimap \perp \Leftarrow (\bullet P_i)]$$

which translates as

$$\begin{aligned}
&\pi_2 \left(out_{\Phi^{\psi\circ}} \left(build_{\Phi^{\psi\circ}} \left(\prod_{i=1}^{n} (!\phi_i)^\circ \times \prod_{j=1}^{m} (!\varphi_j)^\circ \right) \; h\, x \right) \right) \\
&\rightsquigarrow_\beta^+ \pi_2((\lambda w.\langle(\pi_1 w), \ldots, (\pi_{n+w}\, p)\rangle)(h\, x)) \\
&\rightsquigarrow_\beta{}^2 \pi_2(h\, x) \\
&\rightsquigarrow_\beta^+ \mu b\colon \psi^\circ.(\mu a_m\colon (!\varphi_m^\perp)^\circ. \cdots \mu a_1\colon (!\varphi_1^\perp)^\circ.[b\colon \psi^\circ](\lambda x_1. \cdots \lambda x_n.N^\circ)(M_1^\circ) \\
&\cdots (M_n^\circ)(P_1^\circ)) \cdots (P_m^\circ) \\
&\rightsquigarrow_\beta^+ \mu b\colon \psi^\circ.(\mu a_m\colon (!\varphi_m^\perp)^\circ. \cdots \mu a_1\colon (!\varphi_1^\perp)^\circ.[b\colon \psi^\circ]N^\circ[\vec{x} := \vec{M}^\circ](P_1^\circ)) \cdots (P_m^\circ) \\
&\rightsquigarrow_c^+ \mu b\colon \psi^\circ.[b\colon \psi^\circ]N^\circ[\vec{x} := \vec{M}^\circ, a_i \Leftarrow (\bullet P_i^\circ)] \\
&\rightsquigarrow_\beta N^\circ[\vec{x} := \vec{M}^\circ, a_i \Leftarrow (\bullet P_i^\circ)] \\
&\equiv (N[\vec{x} := \vec{M}, a_i \Leftarrow (\bullet P_i)])^\circ
\end{aligned}$$

Thus we can conclude the following.

Theorem 6. *If* $M \rightsquigarrow_\beta N$ *then* $M^\circ \rightsquigarrow_{\beta,c}^+ N^\circ$.

Corollary 1. *The linear* $\lambda\mu$*-calculus is strongly normalising.*

5. Comparison with sequent calculus formulation

In this section I shall first show how to translate derivations in the sequent calculus formulation to deductions in the CLL_μ. Given this translation I shall consider the principal steps in the cut-elimination process for CLL and show how they are reflected in CLL_μ.[6]

I shall show how to translate sequent derivations in CLL to deductions in CLL_μ by defining a procedure, $\mathscr{M}$, inductively over the sequent derivation (where I shall use the shorthand $\phi^\perp$ to denote $\phi \multimap \perp$).

- A proof of the form

$$\frac{}{\phi \vdash \phi}\text{Identity}$$

[6] In fact, for ease of reference, I shall use the term annotations, i.e. the linear $\lambda\mu$-calculus.

is translated to

$$\frac{}{x\colon \phi \rhd x\colon \phi}\,\text{Identity}$$

- A proof of the form

$$\frac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma, \phi, \psi \vdash \Delta\end{array}}{\Gamma, \phi \otimes \psi \vdash \Delta}\otimes_{\mathscr{L}}$$

is translated to

$$\frac{\overline{z\colon \phi \otimes \psi \rhd z : \phi \otimes \psi} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma, x\colon \phi, y\colon \psi \rhd M\colon \Delta\end{array}}{\Gamma, z\colon \phi \otimes \psi \rhd \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ M\colon \Delta}\otimes_{\mathscr{E}}$$

- A proof of the form

$$\frac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma \vdash \phi, \Delta\end{array} \qquad \begin{array}{c}\pi_2\\ \vdots\\ \Gamma' \vdash \psi, \Delta'\end{array}}{\Gamma, \Gamma' \vdash \phi \otimes \psi, \Delta, \Delta'}\otimes_{\mathscr{R}}$$

is translated to

$$\frac{\begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma \rhd M\colon \phi, \Delta\end{array} \qquad \begin{array}{c}\mathscr{M}(\pi_2)\\ \vdots\\ \Gamma' \rhd N\colon \psi, \Delta'\end{array}}{\Gamma, \Gamma' \rhd M \otimes N\colon \phi \otimes \psi, \Delta, \Delta'}\otimes_{\mathscr{I}}$$

- A proof of the form

$$\frac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma \vdash \phi, \Delta\end{array} \qquad \begin{array}{c}\pi_2\\ \vdots\\ \Gamma', \psi \vdash \Delta'\end{array}}{\Gamma, \phi \multimap \psi, \Gamma' \vdash \Delta, \Delta'}\multimap_{\mathscr{L}}$$

is translated to

$$\frac{\dfrac{\overline{f\colon \phi \multimap \psi \rhd f\colon \phi \multimap \psi} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma \rhd M\colon \phi, \Delta\end{array}}{\Gamma, f\colon \phi \multimap \psi \rhd fM\colon \psi, \Delta}\multimap_{\mathscr{E}} \qquad \begin{array}{c}\mathscr{M}(\pi_2)\\ \vdots\\ \Gamma', x\colon \psi \rhd N\colon \Delta'\end{array}}{\Gamma, \Gamma' \rhd N[x := fM]\colon \Delta', \Delta}\,\text{Substitution}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma, \phi \vdash \psi, \Delta\end{array}}{\Gamma \vdash \phi \multimap \psi, \Delta}\ \multimap_{\mathscr{R}}$$

is translated to

$$\dfrac{\begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma, x\colon \phi \rhd M\colon \psi, \Delta\end{array}}{\Gamma \rhd \lambda x.M\colon \phi \multimap \psi, \Delta}\ \multimap_{\mathscr{I}}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma, \phi \vdash \Delta\end{array} \qquad \begin{array}{c}\pi_2\\ \vdots\\ \Gamma', \psi \vdash \Delta'\end{array}}{\Gamma, \Gamma', \phi \parr \psi \vdash \Delta, \Delta'}\ \parr_{\mathscr{L}}$$

is translated to

$$\dfrac{\dfrac{\dfrac{z : \phi \parr \psi \rhd z : \phi \parr \psi \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma, x\colon \phi \rhd M : \Delta\end{array}}{\Gamma, x\colon \phi \rhd [a]M :\ \bot, \Delta}\ \bot_{\mathscr{I}}}{\Gamma \rhd \lambda x.[a]M : \phi^{\bot}, \Delta}\ \multimap_{\mathscr{I}} \qquad \dfrac{\dfrac{\begin{array}{c}\mathscr{M}(\pi_2)\\ \vdots\\ \Gamma', y\colon \psi \rhd N : \Delta'\end{array}}{\Gamma', y\colon \psi \rhd [b]N :\bot, \Delta'}\ \bot_{\mathscr{I}}}{\Gamma' \rhd \lambda y.[b]N : \psi^{\bot}, \Delta'}\ \multimap_{\mathscr{I}}}{\Gamma, \Gamma' \rhd (\lambda x.[a]M) \otimes (\lambda y.[b]N) : (\phi^{\bot}) \otimes (\psi^{\bot}), \Delta, \Delta'}\ \otimes_{\mathscr{I}}}{\Gamma, \Gamma', z\colon \phi \parr \psi \rhd z((\lambda x.[a]M) \otimes (\lambda y.[b]N)) :\bot, \Delta, \Delta'}\ \multimap_{\mathscr{E}}}{\Gamma, \Gamma', z\colon \phi \parr \psi \rhd \mu c.z((\lambda x.[a]M) \otimes (\lambda y.[b]N)) : \Delta, \Delta'}\ \bot_{\mathscr{E}}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma \vdash \phi, \psi, \Delta\end{array}}{\Gamma \vdash \phi \parr \psi, \Delta}\ \parr_{\mathscr{R}}$$

is translated to

$$
\dfrac{
  \dfrac{}{z\colon \phi^{\perp}\otimes\psi^{\perp} \rhd z\colon \phi^{\perp}\otimes\psi^{\perp}}
  \qquad
  \dfrac{
    \dfrac{}{y\colon \psi^{\perp} \rhd y\colon \psi^{\perp}}
    \quad
    \dfrac{
      \dfrac{
        \dfrac{}{x\colon \phi^{\perp} \rhd x : \phi^{\perp}}
        \quad
        \dfrac{
          \dfrac{
            \begin{matrix}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma \rhd M\colon \varphi, c\colon \phi, b\colon \psi, \Delta\end{matrix}
          }{\Gamma \rhd [a]M\colon \perp, c\colon \phi, b\colon \psi, a\colon \varphi, \Delta}\perp_{\mathscr{I}}
        }{\Gamma \rhd \mu c.[a]M\colon \phi, b\colon \psi, a\colon \varphi, \Delta}\perp_{\mathscr{E}}
      }{\Gamma, x\colon \phi^{\perp} \rhd x(\mu c.[a]M) : \perp, b\colon \psi, a\colon \varphi, \Delta}\multimap_{\mathscr{E}}
    }{\Gamma, x\colon \phi^{\perp} \mu b.x(\mu c.[a]M) : \psi, a\colon \varphi, \Delta}\perp_{\mathscr{E}}
  }{\Gamma, x\colon \phi^{\perp}, y\colon \psi^{\perp} \rhd y(\mu b.x(\mu c.[a]M))\colon \perp, a\colon \varphi, \Delta}\multimap_{\mathscr{E}}
}{
  \dfrac{\Gamma, z\colon \phi^{\perp}\otimes\psi^{\perp} \rhd \mathsf{let}\ z\ \mathsf{be}\ x\otimes y\ \mathsf{in}\ y(\mu b.x(\mu c.[a]M))\colon \perp, a\colon \varphi, \Delta}{\Gamma \rhd \lambda z.\mathsf{let}\ z\ \mathsf{be}\ x\otimes y\ \mathsf{in}\ y(\mu b.x(\mu c.[a]M))\colon \phi \parr \psi, a\colon \varphi, \Delta}\multimap_{\mathscr{I}}
}\otimes_{\mathscr{E}}
$$

- A proof of the form

$$
\dfrac{\begin{matrix}\pi_1\\ \vdots\\ \Gamma \vdash \phi, \Delta\end{matrix}}{\Gamma, \phi^{\perp} \vdash \Delta}(\perp_{\mathscr{L}})
$$

is translated to

$$
\dfrac{
  \dfrac{
    \dfrac{}{x\colon \phi^{\perp} \rhd x\colon \phi^{\perp}}
    \quad
    \begin{matrix}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma \rhd M\colon \phi, \Delta\end{matrix}
  }{\Gamma, x\colon \phi^{\perp} \rhd xM\colon \perp, \Delta}\multimap_{\mathscr{E}}
}{\Gamma, x\colon \phi^{\perp} \rhd \mu a.xM\colon \Delta}\perp_{\mathscr{E}}
$$

- A proof of the form

$$
\dfrac{\begin{matrix}\pi_1\\ \vdots\\ \Gamma, \phi \vdash \Delta\end{matrix}}{\Gamma \vdash \phi^{\perp}, \Delta}(\perp_{\mathscr{R}})
$$

is translated to

$$
\dfrac{
  \dfrac{\begin{matrix}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma, x : \phi \rhd M : \Delta\end{matrix}}{\Gamma, x : \phi \rhd [a]M : \perp, \Delta}\perp_{\mathscr{I}}
}{\Gamma \rhd \lambda x : \phi.[a]M : \phi^{\perp}, \Delta}\multimap_{\mathscr{I}}
$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma, \phi \vdash \Delta\end{array}}{\Gamma, !\phi \vdash \Delta}\ \text{Dereliction}_{\mathscr{L}}$$

is translated to

$$\dfrac{\dfrac{}{x :!\phi \triangleright x :!\phi}\ \text{Derelict} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma, y : \phi \triangleright M : \Delta\end{array}}{\Gamma, x :!\phi \triangleright M[y := \mathsf{derelict}(x)] : \Delta}\ \text{Substitution}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma \vdash \phi, \Delta\end{array}}{\Gamma \vdash ?\phi, \Delta}\ \text{Dereliction}_{\mathscr{R}}$$

is translated to

$$\dfrac{\dfrac{\dfrac{}{x :!(\phi^{\perp}) \triangleright x :!(\phi^{\perp})}\ \text{Dereliction} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots\\ \Gamma \triangleright M : \phi, \Delta\end{array}}{x :!(\phi^{\perp}) \triangleright \mathsf{derelict}(x) : \phi^{\perp}}\ \multimap_{\mathscr{E}}}{\Gamma, x :!(\phi^{\perp}) \triangleright (\mathsf{derelict}(x))M :\perp, \Delta}\ \multimap_{\mathscr{I}}$$
$$\Gamma \triangleright \lambda x.(\mathsf{derelict}(x))M :?\phi, \Delta$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma \vdash \Delta\end{array}}{\Gamma, !\phi \vdash \Delta}\ \text{Weakening}_{\mathscr{L}}$$

is translated to

$$\dfrac{\dfrac{}{x :!\phi \triangleright x :!\phi} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots \\ \Gamma \triangleright M : \Delta\end{array}}{\Gamma, x :!\phi \triangleright \mathsf{discard}\ x\ \mathsf{in}\ M : \Delta}\ \text{Weakening}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots \\ \Gamma \vdash \Delta\end{array}}{\Gamma \vdash ?\phi, \Delta}\ \text{Weakening}_{\mathscr{R}}$$

is translated to

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x :!(\phi^{\perp}) \triangleright x :!(\phi^{\perp})} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots \\ \Gamma \triangleright M : \Delta\end{array}}{\Gamma, x :!(\phi^{\perp}) \triangleright \mathsf{discard}\ x\ \mathsf{in}\ M : \Delta}\ \text{Weakening}}{\Gamma, x :!(\phi^{\perp}) \triangleright [a]\ \mathsf{discard}\ x\ \mathsf{in}\ M :\perp, \Delta}\ \perp_{\mathscr{I}}}{\Gamma \triangleright \lambda x.[a]\ \mathsf{discard}\ x\ \mathsf{in}\ M :?\phi, \Delta}\ \multimap_{\mathscr{I}}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots \\ \Gamma, !\phi, !\phi \vdash \Delta\end{array}}{\Gamma, !\phi \vdash \Delta}\ \text{Contraction}_{\mathscr{L}}$$

is translated to

$$\dfrac{\dfrac{}{z :!\phi \triangleright x :!\phi} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots \\ \Gamma, x :!\phi, y :!\phi \triangleright M : \Delta\end{array}}{\Gamma, z :!\phi \triangleright \mathsf{copy}\ z\ \mathsf{as}\ x, y\ \mathsf{in}\ M : \Delta}\ \text{Contraction}$$

- A proof of the form

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots \\ \Gamma \vdash ?\phi, ?\phi, \Delta\end{array}}{\Gamma \vdash ?\phi, \Delta}\ \text{Contraction}_{\mathscr{R}}$$

is translated to

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathscr{M}(\pi_1)}{\vdots}}{\Gamma \triangleright M :?\phi, c :?\phi, \Delta} \quad \dfrac{}{x :!(\phi^{\perp}) \triangleright x :!(\phi^{\perp})}}{\Gamma, x :!(\phi^{\perp}) \triangleright Mx :\perp, c :?\phi, \Delta}\multimap_{\mathscr{E}}}{\Gamma, x :!(\phi^{\perp}) \triangleright \mu c.Mx :?\phi, \Delta}\perp_{\mathscr{E}} \quad \dfrac{}{y :!(\phi^{\perp}) \triangleright y :!(\phi^{\perp})}}{\Gamma, x :!(\phi^{\perp}), y :!(\phi^{\perp}) \triangleright (\mu c.Mx)y :\perp, \Delta}\multimap_{\mathscr{E}} \quad \dfrac{}{z :!(\phi^{\perp}) \triangleright z :!(\phi^{\perp})}}{\Gamma, z :!(\phi^{\perp}) \triangleright \mathsf{copy}\ z\ \mathsf{as}\ x, y\ \mathsf{in}\ (\mu c.Mx)y :\perp, \Delta}\mathrm{Contr}}{\Gamma \triangleright \lambda z.\mathsf{copy}\ z\ \mathsf{as}\ x, y\ \mathsf{in}\ (\mu c.Mx)y :?\phi, \Delta}\multimap_{\mathscr{I}}$$

- A proof of the form

$$\dfrac{\dfrac{\pi_1}{\vdots}}{\dfrac{!\Gamma, \phi \vdash ?\Delta}{!\Gamma, ?\phi \vdash ?\Delta}\,\mathrm{Promotion}_{\mathscr{L}}}$$

(where $?\Delta = ?\varphi, ?\Delta'$) is translated to

$$\dfrac{\dfrac{\dfrac{}{z :?\phi \triangleright z :?\phi} \quad \dfrac{\dfrac{}{\vec{x} :!\Gamma \triangleright \vec{x} :!\Gamma} \quad \dfrac{}{\triangleright \mathscr{P}_{\vec{a}} :!(\Delta^{\perp}), \vec{a} :?\Delta} \quad \dfrac{\dfrac{\dfrac{\mathscr{M}(\pi_1)}{\vdots}}{\vec{x} :!\Gamma, y : \phi \triangleright M :?\varphi, ?\Delta'}\perp_{\mathscr{I}}}{\vec{x} :!\Gamma, y : \phi \triangleright [a_1]M :\perp, \vec{a} :?\Delta}\multimap_{\mathscr{I}}}{\vec{x} :!\Gamma \triangleright \lambda y.[a_1]M : \phi^{\perp}, \vec{a} :?\Delta}\,\mathrm{Prom.}}{\vec{x} :!\Gamma \triangleright \mathsf{promote}\ \vec{x} \mid \mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ \lambda y.[a_1]M :!(\phi^{\perp}), \vec{a} :?\Delta}\multimap_{\mathscr{E}}}{\vec{x} :!\Gamma, z :?\phi \triangleright z(\mathsf{promote}\ \vec{x} \mid \mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ \lambda y.[a_1]M) :\perp, \vec{a} :?\Delta}\perp_{\mathscr{E}}}{\vec{x} :!\Gamma, z :?\phi \triangleright \mu a_1.z(\mathsf{promote}\ \vec{x} \mid \mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{x} \mid \vec{a}\ \mathsf{in}\ \lambda y.[a_1]M) :?\varphi, ?\Delta'}$$

- A proof of the form

$$\dfrac{\dfrac{\pi_1}{\vdots}}{\dfrac{!\Gamma \vdash \phi, ?\Delta}{!\Gamma \vdash !\phi, ?\Delta}\,\mathrm{Promotion}_{\mathscr{R}}}$$

is translated to

$$\dfrac{\dfrac{}{\vec{x}:!\Gamma \triangleright \vec{x}:!\Gamma} \quad \dfrac{}{\triangleright \mathscr{P}_{\vec{a}}:!(\Delta^{\perp}),\vec{a}:?\Delta} \qquad \begin{array}{c}\mathscr{M}(\pi_1)\\ \vdots \\ \vec{x}:!\Gamma \triangleright M:\phi,\vec{a}:?\Delta\end{array}}{\vec{x}:!\Gamma \triangleright \text{promote } \vec{x}\,|\,\mathscr{P}_{\vec{a}} \text{ for } \vec{x}\,|\,\vec{a} \text{ in } M:!\phi,\vec{a}:?\Delta}\ \text{Promotion}$$

One of the features of CLL is that the cut-elimination process is much better behaved than it is for CL. For example, trying to construct the critical pair of Section 2.3 founders, i.e.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta,?\phi}\ \text{Weakening}_{\mathscr{R}} \qquad \dfrac{\Gamma' \vdash \Delta'}{\Gamma',!\phi \vdash \Delta'}\ \text{Weakening}_{\mathscr{L}}}{\Gamma,\Gamma' \vdash \Delta,\Delta'}\ \text{?Cut?}}{}$$

The instance of the Cut rule is not even valid! Hence the problematic critical pairs from CL are removed by moving to the linear framework with its more refined connectives.

It is now possible to reconsider the (better behaved) process of cut-elimination for CLL, by translating the steps across to CLL_{μ}. One might hope that there is a correspondence between (at least) the principal cut reductions and reduction in the linear $\lambda\mu$-calculus. For non-exponential rules there is an exact correspondence but the situation is not so nice for the exponential rules. Below are four instances of principal cuts.

- $(\parr_{\mathscr{R}}, \parr_{\mathscr{L}})$-cut.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \phi,\psi,\Delta}{\Gamma \vdash \phi \parr \psi,\Delta}\parr_{\mathscr{R}} \qquad \dfrac{\Gamma',\phi \vdash \Delta' \quad \Gamma'',\psi \vdash \Delta''}{\Gamma',\Gamma'',\phi \parr \psi \vdash \Delta',\Delta''}\parr_{\mathscr{L}}}{\Gamma,\Gamma',\Gamma'' \vdash \Delta,\Delta',\Delta''}\ \text{Cut}}{}$$

$$\rightsquigarrow_{\text{cut}} \dfrac{\dfrac{\Gamma \vdash \phi,\psi,\Delta,\ \Gamma',\phi \vdash \Delta'}{\Gamma,\Gamma' \vdash \Delta,\Delta',\psi}\ \text{Cut} \quad \Gamma'',\psi \vdash \Delta''}{\Gamma,\Gamma',\Gamma'' \vdash \Delta,\Delta',\Delta''}\ \text{Cut}$$

The former deduction is translated to

$$\begin{aligned}
&\mu c.z((\lambda x.[a]M) \otimes (\lambda y.[b]N))[z:=\lambda u.\text{let } u \text{ be } v \otimes w \text{ in } w(\mu e.v(\mu d.[g]P))]\\
&\quad \equiv \mu c.(\lambda u.\ \text{let } u \text{ be } v \otimes w \text{ in } w(\mu e.v(\mu d.[g]P)))((\lambda x.[a]M) \otimes (\lambda y.[b]N))\\
&\quad \rightsquigarrow_{\beta} \mu c.\ \text{let } (\lambda x.[a]M) \otimes (\lambda y.[b]N) \text{ be } v \otimes w \text{ in } w(\mu e.v(\mu d.[g]P))\\
&\quad \rightsquigarrow_{\beta} \mu c.(\lambda y,[b]N)(\mu e.(\lambda x.[a]M)(\mu d.[g]P))\\
&\quad \rightsquigarrow_{\beta}^{2} \mu c.[b]N[y:=\mu e.[a]M[x:=\mu d.[g]P]]
\end{aligned}$$

which is the translation of the latter.

- $({}^{\perp}\mathscr{R}, {}^{\perp}\mathscr{L})$-cut.

$$\dfrac{\dfrac{\Gamma, \phi \vdash \Delta}{\Gamma' \vdash \Delta, \phi^{\perp}}\ ({}^{\perp}\mathscr{R}) \qquad \dfrac{\Gamma' \vdash \phi, \Delta'}{\Gamma,' \phi^{\perp} \vdash \Delta'}\ ({}^{\perp}\mathscr{L})}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \text{Cut}$$

$$\rightsquigarrow_{\text{cut}} \dfrac{\Gamma' \vdash \phi, \Delta' \quad \Gamma, \phi \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \text{Cut}$$

The former deduction is translated to

$$\begin{aligned} & (\mu b.zN)[z{:=}\lambda x.[a]M] \\ \equiv\ & \mu b.(\lambda x.[a]M)N \\ \rightsquigarrow_{\beta}\ & \mu b.[a]M[x{:=}N] \end{aligned}$$

which is the translation of the latter.

- (Promotion$_{\mathscr{R}}$, Weakening$_{\mathscr{L}}$)-cut.

$$\dfrac{\dfrac{!\Gamma \vdash \phi, ?\Delta}{!\Gamma \vdash !\phi, ?\Delta}\ \text{Promotion}_{\mathscr{R}} \qquad \dfrac{\Gamma' \vdash \Delta'}{\Gamma', \phi \vdash \Delta'}\ \text{Weakening}_{\mathscr{L}}}{!\Gamma, \Gamma' \vdash ?\Delta, \Delta'}\ \text{Cut}$$

$$\rightsquigarrow_{\text{cut}} \dfrac{\dfrac{\Gamma' \vdash \Delta'}{\Gamma', !\Gamma \vdash \Delta'}\ \text{Weakening}^{*}_{\mathscr{L}}}{\Gamma', !\Gamma \vdash \Delta', ?\Delta}\ \text{Weakening}^{*}_{\mathscr{R}}$$

The former deduction is translated to

$$\begin{aligned} & \mathsf{discard}\ (\mathsf{promote}\ \vec{x}|\mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{x}|\vec{a}\ \mathsf{in}\ M)\ \mathsf{in}\ N \\ \rightsquigarrow^{*}_{\beta}\ & \mathsf{discard}\ \mathscr{P}_{\vec{a}}\ \mathsf{in}\ \mathsf{discard}\ \vec{x}\ \mathsf{in}\ N \end{aligned}$$

which is the translation of the latter.

Unfortunately, this nice relationship between the cut-elimination process and term reduction breaks down when considering the (Promotion$_{\mathscr{R}}$, Dereliction$_{\mathscr{L}}$)-cut. Here the cut reduction is of the form

$$\dfrac{\dfrac{!\Gamma \vdash \phi, ?\Delta}{!\Gamma \vdash !\phi, ?\Delta}\ \text{Promotion}_{\mathscr{R}} \qquad \dfrac{\Gamma', \phi \vdash \Delta'}{\Gamma', !\phi \vdash \Delta'}\ \text{Dereliction}_{\mathscr{L}}}{!\Gamma, \Gamma' \vdash ?\Delta, \Delta'}\ \text{Cut}$$

$$\rightsquigarrow_{\text{cut}} \dfrac{!\Gamma \vdash \phi_i ?\Delta \quad \Gamma', \phi \vdash \Delta'}{!\Gamma, \Gamma' \vdash ?\Delta, \Delta'}\text{Cut.}$$

The former deduction is translated to

$$\begin{aligned}
&N[z := \mathsf{derelict}(w)][w := \mathsf{promote}\,\vec{y} \mid \mathscr{P}_{\vec{a}} \text{ for } \vec{y} \mid \vec{a} \text{ in } M] \\
&\equiv N[z := \mathsf{derelict}(\mathsf{promote}\,\vec{y} \mid \mathscr{P}_{\vec{a}} \text{ for } \vec{y} \mid \vec{a} \text{ in } M)] \\
&\rightsquigarrow_{\beta} N[z := M[\vec{y} := \vec{y}, a_i : ?\delta_i \Leftarrow (\bullet \mathscr{P}_{a_i})]] \\
&\equiv N[z := M[a_i : ?\delta_i \Leftarrow (\bullet \mathscr{P}_{a_i})]].
\end{aligned}$$

Unfortunately, the inner textual substitution need not be equivalent to the identity substitution and so the latter term is not the translation of the latter deduction. What appears to be the problem is the asymmetry between the types ϕ and $\phi^{\perp\perp}$ as discussed in Section 2.3. When considering the refined connectives of CLL this entails certain connectives being available only indirectly, via the negation. In this light maybe it is too much to ask for there to be an exact match with the highly symmetric sequent calculus.

Remark 3. In previous accounts of this work [13, 12] I tried to find a formulation whose reduction behaviour *did* correspond to the cut elimination relation. This is possible given a different formulation of the Promotion rule, i.e.

$$\frac{\begin{array}{ll} \Gamma_1 \rhd M_1 : !\phi_1, \Sigma_1 & \Delta_1 \rhd P_1 : !((!\varphi_1 \multimap \perp) \multimap \perp), \Upsilon_1 \\ \Gamma_n \rhd M_n : !\phi_n, \Sigma_n & \Delta_m \rhd P_m : !((!\varphi_m \multimap \perp) \multimap \perp), \Upsilon_m \\ x_1 : !\phi_1, \ldots, x_n : !\phi_n \rhd N : \psi, a_1 : !\varphi_1 \multimap \perp, \ldots, a_m : !\varphi_m \multimap \perp & \end{array}}{\vec{\Gamma}, \vec{\Delta} \rhd \mathsf{promote}\ \vec{M} \mid \vec{P} \text{ for } \vec{x} \mid \vec{a} \text{ in } N : !\psi, \vec{\Sigma}, \vec{\Upsilon}} \text{ Promotion}$$

where the P_i terms have a double-negated type (which additionally has to be an exponential type to satisfy subject reduction). The associated reduction rule is

$$\begin{aligned}
&\mathsf{derelict}(\mathsf{promote}\ \vec{M} \mid \vec{P} \text{ for } \vec{x} \mid \vec{a} \text{ in } N) \\
&\quad \rightsquigarrow_{\beta} N[\vec{x} := \vec{M}, a_j : !\varphi_j \multimap \perp \Leftarrow (\mathsf{derelict}(P_j)\bullet)] \qquad (\dagger)
\end{aligned}$$

and the other reduction rules involving the Promotion rule remain the same. For this formulation we find that if we permit only η-reduction of the form

$$\lambda x.(M x) \rightsquigarrow_{\eta} M$$

then we can prove the desired correspondence theorem.

Theorem 7. *If* $\Gamma \overset{\pi_1}{\vdash} \Delta \rightsquigarrow_{cut} \Gamma \overset{\pi_2}{\vdash} \Delta$ *then* $\mathscr{M}(\pi_1) \rightsquigarrow^{*}_{\beta,c,\eta} \mathscr{M}(\pi_2)$.

However this formulation is quite cumbersome and the principle reduction (†) contains an unusual form of textual substitution (one which is clearly admissible but not derivable from the normal form of textual substitution). However it is worth pointing out that the problems of Section 3.3 concerning the subformula property are resolved with this formulation.

6. Translations

Just as there are translations from IL into ILL [10, Chapter 2, Section 5], there are also translations from CL into CLL. These have been studied by Schellinx [34] in his thesis. As he points out they are (necessarily) quite complicated, requiring a large number of exponentials. Interestingly there is no unique 'optimal' solution as is the case for IL. Rather there are two candidates. The T-translation which is based on a linear decomposition of $\phi \to \psi$ as $!?\phi \multimap ?\psi$; and the Q-translation which interprets $\phi \to \psi$ as $!\phi \multimap ?!\psi$. They are defined as follows:

$$
\begin{aligned}
p^{\mathsf{Q}} &\stackrel{\text{def}}{=} p \qquad (p \text{ atomic})\\
(\phi \to \psi)^{\mathsf{Q}} &\stackrel{\text{def}}{=} !\phi^{\mathsf{Q}} \multimap ?!\phi^{\mathsf{Q}}\\
(\phi \times \psi)^{\mathsf{Q}} &\stackrel{\text{def}}{=} !\phi^{\mathsf{Q}} \otimes !\psi^{\mathsf{Q}}\\
(\phi + \psi)^{\mathsf{Q}} &\stackrel{\text{def}}{=} ?!\phi^{\mathsf{Q}} \parr ?!\psi^{\mathsf{Q}};
\end{aligned}
$$

$$
\begin{aligned}
p^{\mathsf{T}} &\stackrel{\text{def}}{=} p \qquad (p \text{ atomic})\\
(\phi \to \psi)^{\mathsf{T}} &\stackrel{\text{def}}{=} !?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}\\
(\phi \times \psi)^{\mathsf{T}} &\stackrel{\text{def}}{=} !?\phi^{\mathsf{T}} \otimes !?\psi^{\mathsf{T}}\\
(\phi + \psi)^{\mathsf{T}} &\stackrel{\text{def}}{=} ?\phi^{\mathsf{T}} \parr ?\psi^{\mathsf{T}}.
\end{aligned}
$$

Theorem 8 (Schellinx). $\vdash_{\mathrm{CLL}} !\Gamma^{\mathsf{Q}} \vdash ?!\Delta^{\mathsf{Q}}$ *iff* $\vdash_{\mathrm{CL}} \Gamma \vdash \Delta$ *iff* $\vdash_{\mathrm{CLL}} !?\Gamma^{\mathsf{T}} \vdash ?\Delta^{\mathsf{T}}$.

These translations can be lifted to translations between the $\lambda\mu$-calculus and the linear $\lambda\mu$-calculus. Rather than give all the details I shall show how the implication rules for CL_{μ} are translated using both the Q and T strategies.

The implication introduction rule

$$\frac{\Gamma, x:\phi \triangleright M:\psi, :\Delta}{\Gamma \triangleright \lambda x.M:\phi \to \psi, :\Delta} \to_{\mathscr{I}}$$

is T-translated to

$$
\dfrac{\dfrac{\dfrac{}{z:!((!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}) \multimap \perp) \triangleright z:!((!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}) \multimap \perp)}}{z:!((!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}) \multimap \perp) \triangleright \mathsf{derelict}(z):(!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}) \multimap \perp} \qquad \dfrac{}{!?\Gamma^{\mathsf{T}}, x:!?\phi^{\mathsf{T}} \triangleright M^{\mathsf{T}}:?\psi^{\mathsf{T}}, \vec{a}:?\Delta^{\mathsf{T}}}}{!?\Gamma^{\mathsf{T}} \triangleright \lambda x.M^{\mathsf{T}}:!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}, \vec{a}:?\Delta^{\mathsf{T}}}
$$

$$
\dfrac{!?\Gamma^{\mathsf{T}}, z:!((!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}) \multimap \perp) \triangleright \mathsf{derelict}(z)(\lambda x.M^{\mathsf{T}}):\perp, \vec{a}:?\Delta}{!?\Gamma^{\mathsf{T}} \triangleright \lambda z.\mathsf{derelict}(z)(\lambda x.M^{\mathsf{T}}):?(!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}), \vec{a}:?\Delta^{\mathsf{T}}}
$$

The implication elimination rule

$$\frac{\Gamma \triangleright M:\phi \to \psi, \Delta \quad \Gamma' \triangleright N:\phi, \Delta'}{\Gamma, \Gamma' \triangleright MN:\psi, \Delta, \Delta'} \to_{\mathscr{E}}$$

is T-translated to

$$\frac{!?\Gamma^{\mathsf{T}} \rhd M^{\mathsf{T}}: ?(!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}), ?\Delta^{\mathsf{T}} \quad \vec{y}: !?\Gamma'^{\mathsf{T}} \rhd \mathscr{A}: !((!?\phi^{\mathsf{T}} \multimap ?\psi^{\mathsf{T}}) \multimap \bot), \vec{b}: ?\Delta'^{\mathsf{T}}, c: ?\psi^{\mathsf{T}}}{\dfrac{!?\Gamma^{\mathsf{T}}, !?\Gamma'^{\mathsf{T}} \rhd M^{\mathsf{T}}\mathscr{A}: \bot, c: ?\psi^{\mathsf{T}}, ?\Delta^{\mathsf{T}}, ?\Delta'^{\mathsf{T}}}{!?\Gamma^{\mathsf{T}}, !?\Gamma'^{\mathsf{T}} \rhd \mu c.M^{\mathsf{T}}\mathscr{A}: ?\psi^{\mathsf{T}}, ?\Delta^{\mathsf{T}}, ?\Delta'^{\mathsf{T}}}}$$

where $\mathscr{A}$ denotes the term

$$\mathsf{promote}\ \vec{y} \mid \mathscr{P}_{\vec{b},c}\ \mathsf{for}\ \vec{y} \mid \vec{b},\ c\ \mathsf{in}\ \lambda z.[c]\ \ z(\mathsf{promote}\ \vec{y} \mid \mathscr{P}_{\vec{b}}\ \mathsf{for}\ \vec{y} \mid \vec{b}\ \mathsf{in}\ N^{\mathsf{T}})$$

The implication introduction rule

$$\frac{\vec{y}: \Gamma, x: \phi \rhd M: \psi, \vec{a}: \Delta}{\vec{y}: \Gamma \rhd \lambda x.M: \phi \to \psi, \vec{a}: \Delta} \to_{\mathscr{I}}$$

is Q-translated to

$$\dfrac{\dfrac{}{z \rhd z: !(!(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}) \multimap \bot)} \quad \dfrac{\vec{y}: !\Gamma^{\mathsf{Q}}, x: !\phi^{\mathsf{Q}} \rhd M^{\mathsf{Q}}: ?!\psi^{\mathsf{Q}}, \vec{a}: ?!\Delta^{\mathsf{Q}}}{\vec{y}: !\Gamma^{\mathsf{Q}} \rhd \lambda x.M^{\mathsf{Q}}: !\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}, \vec{a}: ?!\Delta^{\mathsf{Q}}}}{\dfrac{z \rhd \mathsf{derelict}(z): !(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}) \multimap \bot \quad \vec{y}: !\Gamma^{\mathsf{Q}} \rhd \mathsf{promote}\ \vec{y} \mid \mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{y} \mid \vec{a}\ \mathsf{in}\ \lambda x.M^{\mathsf{Q}}: !(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}), \vec{a}: ?!\Delta^{\mathsf{Q}}}{!\Gamma^{\mathsf{Q}}, z: !(!(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}) \multimap \bot) \rhd \mathsf{derelict}(z)(\mathsf{promote}\ \vec{y} \mid \mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{y} \mid \vec{a}\ \mathsf{in}\ \lambda x.M^{\mathsf{Q}}): \bot, \vec{a}: ?!\Delta}}{!\Gamma^{\mathsf{Q}} \rhd \lambda z.\mathsf{derelict}(z)(\mathsf{promote}\ \vec{y} \mid \mathscr{P}_{\vec{a}}\ \mathsf{for}\ \vec{y} \mid \vec{a}\ \mathsf{in}\ \lambda x.M^{\mathsf{Q}}): ?!(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}), \vec{a}: ?!\Delta^{\mathsf{Q}}}$$

Finally, the implication elimination rule

$$\frac{\Gamma \rhd M: \phi \to \psi, \Delta \qquad \Gamma' \rhd N: \phi, \Delta'}{\Gamma, \Gamma' \rhd MN: \psi, \Delta, \Delta'} \to_{\mathscr{E}}$$

is Q-translated to

$$\frac{!\Gamma^{\mathsf{Q}} \rhd M: ?!(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}), ?!\Delta^{\mathsf{Q}} \qquad !\Gamma'^{\mathsf{Q}} \rhd \mathscr{A}: !(!(!\phi^{\mathsf{Q}} \multimap ?!\psi^{\mathsf{Q}}) \multimap \bot), a: ?!\psi^{\mathsf{Q}}, ?!\Delta'^{\mathsf{Q}}}{\dfrac{!\Gamma^{\mathsf{Q}}, !\Gamma'^{\mathsf{Q}} \rhd M^{\mathsf{Q}}\mathscr{A}: \bot, a: ?!\psi^{\mathsf{Q}}, ?!\Delta^{\mathsf{Q}}, ?!\Delta'^{\mathsf{Q}}}{!\Gamma^{\mathsf{Q}}, !\Gamma'^{\mathsf{Q}} \rhd \mu a.M^{\mathsf{Q}}\mathscr{A}: ?!\psi^{\mathsf{Q}}, ?!\Delta^{\mathsf{Q}}, ?!\Delta'^{\mathsf{Q}}}}$$

where $\mathscr{A}$ denotes the term

$$\mathsf{promote}\ \vec{z} \mid \mathscr{P}_{\vec{b},a}\ \mathsf{for}\ \vec{z} \mid \vec{b},\ a\ \mathsf{in}\ (\lambda x.N^{\mathsf{Q}})\ \mathsf{promote}\ x \mid \mathscr{P}_a\ \mathsf{for}\ x \mid a\ \mathsf{in}\ \lambda y.[a]\ \mathsf{derelict}\ (x)y$$

Filling in all the details gives the following theorem.

Theorem 9. $\vdash_{\mathrm{CLL}_\mu} !\Gamma^{\mathsf{Q}} \vdash ?!\Delta^{\mathsf{Q}}$ *iff* $\vdash_{\mathbf{CL}_\mu} \Gamma \vdash \Delta$ *iff* $\vdash_{\mathrm{CLL}_\mu} !?\Gamma^{\mathsf{T}} \vdash ?\Delta^{\mathsf{T}}$.

In fact, these translations preserve reductions as well, although I shall not give any details here. Unlike the case for the various translations of IL into ILL [9], it is quite hard to determine computational interpretations of these two translation strategies.

7. A linear/non-linear formulation

As is the case for the linear λ-calculus it is possible to present the linear $\lambda\mu$-calculus in a number of ways. It is fairly easy to see how one could split the contexts into linear

$$\frac{}{\Theta; x:\phi \rhd_l x:\phi;\ \Upsilon}\,\mathrm{Identity}_l \qquad \frac{}{\Theta, x:\alpha \rhd_n x:\alpha,\ \Upsilon}\,\mathrm{Identity}_n$$

$$\frac{\Theta; \Gamma, x:\phi \rhd_l M:\psi, \Sigma;\ \Upsilon}{\Theta; \Gamma \rhd_l \lambda x:\phi.M:\phi \multimap \psi, \Sigma;\ \Upsilon}\,\multimap_{\mathscr{I}} \qquad \frac{\Theta, x:\alpha \rhd_n M:\beta, \Upsilon}{\Theta \rhd_n \lambda x:\alpha.M:\alpha \to \beta, \Upsilon}\,\to_{\mathscr{I}}$$

$$\frac{\Theta; \Gamma \rhd_l M:\phi \multimap \psi, \Sigma;\ \Upsilon \qquad \Theta; \Gamma' \rhd_l N:\phi, \Sigma';\ \Upsilon}{\Theta; \Gamma, \Gamma' \rhd_l MN:\psi, \Sigma, \Sigma';\ \Upsilon}\,\multimap_{\mathscr{E}} \qquad \frac{\Theta \rhd_n M:\alpha \to \beta,\ \Upsilon \qquad \Theta \rhd_n N:\beta,\ \Upsilon}{\Theta \rhd_n MN:\beta,\ \Upsilon}\,\to_{\mathscr{E}}$$

$$\frac{\Theta; \Gamma \rhd_l M:\phi, \Sigma;\ \Upsilon}{\Theta; \Gamma \rhd_l [a:\phi]M:\bot, a:\phi, \Sigma;\ \Upsilon}\,\mathrm{Passivate}_l \qquad \frac{\Theta \rhd_n M:\alpha,\ \Upsilon}{\Theta \rhd_n [a:\alpha]M:\mathbf{f}, a:\alpha,\ \Upsilon}\,\mathrm{Passivate}_n$$

$$\frac{\Theta; \Gamma \rhd_l M:\bot, a:\phi, \Sigma;\ \Upsilon}{\Theta; \Gamma \rhd_l \mu a:\phi.M:\phi, \Sigma;\ \Upsilon}\,\mathrm{Activate}_l \qquad \frac{\Theta \rhd_n M:\mathbf{f}, a:\alpha,\ \Upsilon}{\Theta \rhd_n \mu a:\alpha.M:\alpha,\ \Upsilon}\,\mathrm{Activate}_n$$

$$\frac{\Theta \rhd_n M:\alpha,\ \Upsilon}{\Theta; - \rhd_l \mathsf{F}M:\mathsf{F}\alpha;\ \Upsilon}\,\mathsf{F}_{\mathscr{I}} \qquad \frac{\Theta; - \rhd_l M:\phi;\ \Upsilon}{\Theta \rhd_n \mathsf{G}M:\mathsf{G}\phi,\ \Upsilon}\,\mathsf{G}_{\mathscr{I}}$$

$$\frac{\Theta; \Gamma \rhd_l M:\mathsf{F}\alpha, \Sigma;\ \Upsilon \qquad \Theta, x:\alpha; \Gamma' \rhd_l N:\phi, \Sigma';\ \Upsilon}{\Theta; \Gamma, \Gamma' \rhd_l \mathsf{let}\ M\ \mathsf{be}\ \mathsf{F}x\ \mathsf{in}\ N:\phi, \Sigma, \Sigma';\ \Upsilon}\,\mathsf{F}_{\mathscr{E}}$$

$$\frac{\Theta \rhd_n M:\mathsf{G}\phi,\ \Upsilon}{\Theta; - \rhd_l \mathsf{derelict}(M):\phi;\ \Upsilon}\,\mathsf{G}_{\mathscr{E}}$$

Fig. 5. A linear/non-linear formulation of the linear $\lambda\mu$-calculus.

and non-linear zones with rules for moving between them. Instead I shall sketch briefly how one can apply the ideas of Benton [5] to the linear $\lambda\mu$-calculus (for brevity I shall only discuss the $(\multimap, !)$-fragment). Benton proposed (following a categorical insight) to present ILL in three parts: a linear subsystem, a non-linear subsystem and a third part containing operations to move between the two subsystems. The exponential can then be thought of as a composite of these operations. I shall not go into any real detail here, the reader is referred to Benton's paper (and further details will appear in [37]).

I shall use the following conventions: α to range over non-linear formulae, ϕ to range over linear formulae, Γ to range over linear contexts, Θ to range over non-linear contexts, Σ to range over linear passive contexts and Υ to range over non-linear passive contexts. Formulae are then defined by the grammars

$$\alpha ::= p \mid \alpha \to \alpha \mid \mathsf{G}\phi$$

$$\phi ::= q \mid \phi \multimap \phi \mid \mathsf{F}\alpha$$

where p ranges over some countable set of non-linear atomic formulae including a distinguished member $\mathbf{f}$, and q ranges over some countable set of linear atomic formulae including the distinguished member $\bot$.

We have two forms of typing judgement, linear and non-linear, which are of the form $\Theta; \Gamma \rhd_l M:\phi, \Sigma;\ \Upsilon$ and $\Theta \rhd_n M:\alpha,\ \Upsilon$, respectively. The rules for forming valid

typing judgements are given in Fig. 5. The resulting system then consists of both the $\lambda\mu$-calculus and (part of) the linear $\lambda\mu$-calculus along with operations to move between the systems.

An important feature of this formulation is that the structural rules are *admissible*, i.e.

$$\frac{\Theta;\Gamma \rhd_l M:\phi,\Sigma;\Upsilon}{x:\alpha,\Theta;\Gamma \rhd_l M:\phi,\Sigma;\Upsilon}\text{Weakening} \quad \text{and}$$

$$\frac{x:\alpha,y:\alpha,\Theta;\Gamma \rhd_l M:\phi,\Sigma;\Upsilon}{z:\alpha,\Theta;\Gamma \rhd_l M[x,y:=z]:\phi,\Sigma;\Upsilon}\text{Contraction}$$

The β-rules for this formulation are then quite succinct

$$\begin{aligned}
(\lambda x:\phi.M)N &\rightsquigarrow_\beta M[x:=N]\\
(\lambda x:\alpha.M)N &\rightsquigarrow_\beta M[x:=N]\\
\mu a:\phi.[a:\phi]M &\rightsquigarrow_\beta M\\
\mu a:\alpha.[a:\alpha]M &\rightsquigarrow_\beta M, \quad \text{where } a\notin \mu FV(M)\\
\mathsf{let}\ \mathsf{F}(M)\ \mathsf{be}\ \mathsf{F}(x)\ \mathsf{in}\ N &\rightsquigarrow_\beta N[x:=M]\\
\mathsf{derelict}(\mathsf{G}(M)) &\rightsquigarrow_\beta M.
\end{aligned}$$

There are also commuting conversions, which are to the reader. It is possible to translate between this linear/non-linear formulation and the linear $\lambda\mu$-calculus. First we need some translations between types.

$$\begin{aligned}
p^\circ &\stackrel{\text{def}}{=} p,\\
(\phi \multimap \psi)^\circ &\stackrel{\text{def}}{=} \phi^\circ \multimap \psi^\circ,\\
(!\phi)^\circ &\stackrel{\text{def}}{=} \mathsf{F}(\mathsf{G}(\phi^\circ)),\\
q^\star &\stackrel{\text{def}}{=} q\\
(\phi \multimap \psi)^\star &\stackrel{\text{def}}{=} \phi^\star \multimap \psi^\star\\
(\mathsf{F}(\alpha))^\star &\stackrel{\text{def}}{=} !(\alpha^\star)\\
p^\star &\stackrel{\text{def}}{=} \begin{cases} \bot & \text{if } p=f\\ p & \text{otherwise}\end{cases}\\
(\alpha\to\beta)^\star &\stackrel{\text{def}}{=} !\alpha^\star \multimap \beta^\star\\
(\mathsf{G}(\phi))^\star &\stackrel{\text{def}}{=} \phi^\star
\end{aligned}$$

Theorem 10. (1) *If* $\Gamma \rhd M:\phi,\Sigma$ *then there is a term* M° *such that* $-;\Gamma^\circ \rhd M^\circ:\phi^\circ,\Sigma^\circ;-$.

(2) *If* $\Theta;\Gamma \rhd M:\phi,\Sigma;\Upsilon$ *then there is a linear* $\lambda\mu$*-term* $M^\star$ *such that* $!\Theta^\star,\Gamma^\star \rhd M^\star:\phi^\star,\Sigma^\star,!\Upsilon^\star$.

(3) *If* $\Theta \rhd M:\alpha,\Upsilon$ *then there is a linear* $\lambda\mu$*-term* $M^\star$ *such that* $!\Theta^\star \rhd M^\star:\alpha^\star,!\Upsilon^\star$.

However, as is the case for ILL, it is not immediately clear how much of an improvement this formulation is. A smaller set of reduction rules has been gained at

the expense of a loss of information about Weakening and Contraction, which surely are the *raison d'être* of linear proof theory. Of course, at the level of a programming language, explicit duplication and erasure of data structures would be quite tiresome and it seems that this mixed presentation might be of some practical value.

8. Conclusions and future work

In this paper I have demonstrated how Parigot's techniques can be applied to CLL to yield a classical linear λ-calculus. I would claim that the linear $\lambda\mu$-calculus, considered as a programming language, is of more use than one based on proof nets. As mentioned earlier, proof nets rely on equivalent types being considered equal – this would present an unusual programming paradigm where, for example, the type inference mechanism would have to be adapted to factor all types by the various equivalences. In the linear $\lambda\mu$-calculus there are explicit coercion terms.

There are other proposals for natural deduction formulations of CLL. Troelstra [36] presents linear versions of Gentzen's original proposals. Martini and Masini [28] present a different formulation with the motivation of having the $\parr$ connective as fundamental and not, as it is in this paper, derived. Albrecht et al. [3] give yet another formulation which is very compact and appears to be closely related to a proof net formulation (in particular, the formulae equivalences are essential and implicit).

In the formulations of CL_μ and CLL_μ, I have included all of the connectives separately. Of course the formulae equivalences of both logics mean that, in fact, we could trim this down. For CL_μ both Parigot [30] and Ong [29] restrict their attention to a fragment with just the implication connective (and the Activate and Passivate rules). For CLL_μ the most obvious fragment includes just the linear implication ($\multimap$) and the exponential ($!$). As an illustration of how this might work, I shall show how the tensor ($\otimes$) and its unit (I) can be simulated with just these connectives. Term formation for these connectives is then defined as

$$\begin{aligned} M\otimes N &\stackrel{\text{def}}{=} \lambda x.(xM)N,\\ \mathsf{let}\ M\ \mathsf{be}\ x\otimes y\ \mathsf{in}\ N &\stackrel{\text{def}}{=} \mu a{:}\varphi.M(\lambda x.\lambda y.[a{:}\varphi]N),\\ * &\stackrel{\text{def}}{=} \lambda x{:}\bot.x,\\ \mathsf{let}\ M\ \mathsf{be}\ *\ \mathsf{in}\ N &\stackrel{\text{def}}{=} \mu a{:}\varphi.M([a{:}\varphi]N). \end{aligned}$$

The β-rules are preserved by this translation, i.e.

$$\begin{aligned} \mathsf{let}\ M\otimes N\ \mathsf{be}\ x\otimes y\ \mathsf{in}\ P &\stackrel{\text{def}}{=} \mu a{:}\varphi.(\lambda z.(zM)N)(\lambda x.\lambda y.[a{:}\varphi]P)\\ &\rightsquigarrow_\beta \mu a{:}\varphi.(\lambda x.\lambda y.[a{:}\varphi]P)M\,N\\ &\rightsquigarrow_\beta{}^* \mu a{:}\varphi.[a{:}\varphi]P[x:=M,y:=N]\\ &\rightsquigarrow_\beta P[x:=M,y:=N]; \end{aligned}$$

$$\begin{aligned}\textsf{let } * \textsf{ be } * \textsf{ in } M &\stackrel{\text{def}}{=} \mu a{:}\varphi.(\lambda x.x)[a{:}\varphi]M \\ &\rightsquigarrow_\beta \mu a{:}\varphi.[a{:}\varphi]M \\ &\rightsquigarrow_\beta M.\end{aligned}$$

If either the $\lambda\mu$-calculus or the linear $\lambda\mu$-calculus were to be made into a programming language, a design decision would have to be made as to which connectives were built-in and which ones were defined. Although experience might tell otherwise, it would seem likely that any programming language would be as verbose as possible, whereas an intermediate language might well profit for having only a few connectives. This is clearly future work.

A semantic study would also be desirable. Ong [29] has proposed a categorical semantics and a class of game-theoretic models for CL_μ. It would be interesting to see if a similar extension of linear categories [11] would produce some sort of (weak) ★-autonomous category [4]. I should also like to investigate to what extent this work can be adapted to give a natural deduction formulation of classical **S4**, in the same way that work on ILL can be adapted for *intuitionistic* **S4** [16].

This study has revealed some weaknesses with Parigot's approach. In particular, the treatment of the Promotion rule (or any S4-like modality) is quite poor. The problem appears to be that the inherent asymmetry of the system forces certain connectives to be treated rather weakly (⅋ and ?). As demonstrated in Section 3.4, this can be alleviated to some extent by mild extensions to the system. Alternatively, one could move to a full sequence-conclusion natural deduction, similar to Boričić's formulation [18]. Both these possibilities will be explored in full in [37].

Acknowledgements

I am grateful to Martin Hyland for many interesting discussions and for his suggestion to investigate a linear version of Parigot's work. I should also like to take this opportunity to thank Nick Benton for his friendship and critical advice. I am grateful to Luke Ong for many conversations about this work and for his patient explanation of his work. Nick Benton, Luke Ong, Valeria de Paiva, Thomas Streicher, A.S. Troelstra, Christian Urban and an anonymous referee all gave helpful comments on various versions of this paper.

References

[1] S. Abramsky, Computational interpretations of linear logic, Theoret. Comput. Sci. 111(1–2) (1993) 3–57. Previously Available as Department of Computing, Imperial College Technical Report 90/20, 1990.

[2] S. Abramsky, R. Jagadeesan, P. Malacaria, Full abstraction for PCF (Extended Abstract), Proc. Conf. on Theoretical Aspects of Computer Software, Lecture Notes in Computer Science, vol. 789, 1994, pp. 1–5.

[3] D. Albrecht, J.N. Crossley, J.S. Jeavons, New Curry–Howard terms for full linear logic, unpublished Manuscript, May 1996.

[4] M. Barr, ★-autonomous categories and linear logic, Math. Struct. Comput. Sci. 1 (1991) 159–178.
[5] P.N. Benton, A mixed linear and non-linear logic: proofs, terms and models, Technical Report 352, Computer Laboratory, University of Cambridge, 1994.
[6] P.N. Benton, Strong normalisation for the linear term calculus, J. Funct. Programming 5(1) (1995) 65–80.
[7] P.N. Benton, G.M. Bierman, V.C.V. de Paiva, J.M.E. Hyland, Term assignment for intuitionistic linear logic, Technical Report 262, Computer Laboratory, University of Cambridge, August 1992.
[8] P.N. Benton, G.M. Bierman, V.C.V. de Paiva, J.M.E. Hyland, A term calculus for intuitionistic linear logic, in: M. Bezem, J.F. Groote (Eds.), Proc. Ist Int. Conf. on Typed λ-calculi and applications, Lecture Notes in Computer Science, vol. 664, 1993, pp. 75–90.
[9] P.N. Benton, P. Wadler, Linear logic, monads and the lambda calculus, Proc. Symp. on Logic in Computer Science, 1996, pp. 420–431.
[10] G.M. Bierman, On intuitionistic linear logic, Ph.D. Thesis, Computer Laboratory, University of Cambridge, December 1993. Published as Computer Laboratory Technical Report 346, August 1994.
[11] G.M. Bierman, What is a categorical model of intuitionistic linear logic?, Proc. 2nd Int. Conf. on Typed λ-calculi and applications, vol. 902 of Lecture Notes in Computer Science, April 1995, pp. 78–93. Previously available as Technical Report 333, University of Cambridge Computer Laboratory, March 1994.
[12] G.M. Bierman, A classical linear λ-calculus, Technical Report 401, Computer Laboratory, University of Cambridge, July 1996.
[13] G.M. Bierman, Towards a classical linear λ-calculus, Proc. Tokyo Conf. on Linear Logic, Electronic Notes in Computer Science, vol. 3, Elsevier, Amsterdam, 1996.
[14] G.M. Bierman, Observations on a linear PCF, Technical Report 412, Computer Laboratory, University of Cambridge, January 1997.
[15] G.M. Bierman, A computational interpretation of the $\lambda\mu$-calculus, in: L. Brim, J. Gruska, J. Zlatuška (Eds.), Proc. Symp. on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, vol. 1450, August 1998, pp. 336–345.
[16] G.M. Bierman, V.C.V. de Paiva, Intuitionistic necessity revisited, Technical Report CSR-96-10, School of Computer Science, University of Birmingham, June 1996.
[17] A. Blass, A game semantics for linear logic, Ann. Pure Appl. Logic 56 (1992) 183–220.
[18] B.R. Boričić, On sequence-conclusion natural deduction systems, J. Philos. Logic 14 (1985) 359–377.
[19] C. Cellucci, Existential instantiation and normalization in sequent natural deduction, Ann. Pure Appl. Logic 58(2) (1992) 111–148.
[20] Th. Coquand, A semantics of evidence of classical arithmetic, J. Symbolic Logic 60 (1995) 325–337.
[21] G. Gentzen, Investigations into logical deduction, in: M.E. Szabo (Eds.), The Collected Papers of Gerhard Gentzen, North-Holland, Amsterdam, 1969, pp. 68–131. English Translation of 1935 German original.
[22] J.-Y. Girard, Linear logic, Theoret. Comput. Sci. 50 (1987) 1–101.
[23] J.-Y. Girard, Y. Lafont, P. Taylor, Proofs and Types, Cambridge Tracts in Theoretical Computer Science, vol. 7, Cambridge University Press, Cambridge, 1989.
[24] T.G. Griffin, A formulae-as-types notion of control, Proc. Symp. on Principles of Programming Languages, 1990, pp. 47–58.
[25] W.A. Howard, The formulae-as-types notion of construction, in: J.R. Hindley, J.P. Seldin (Eds.), To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and formalism, Academic Press, New York, 1980.
[26] J.M.E. Hyland, C.-H.L. Ong, Pi-calculus, dialogue games and PCF, Proc. Conf. on Functional Programming Languages and Computer Architecture, 1995, pp. 96–107.
[27] P. Lincoln, Linear logic, ACM SIGACT News 23(2) (1992) 29–37.
[28] S. Martini, A. Masini, Experiments in linear natural deduction, unpublished Manuscript, March 1995.
[29] C.-H.L. Ong, A semantic view of classical proofs: type-theoretic, categorical and denotational characterizations, Proc. Symp. on Logic in Computer Science, 1996, pp. 230–241.
[30] M. Parigot, $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction, Proc. Conf. on Logic Programming and Automated Reasoning, Lecture Notes in Computer Science, vol. 624, 1992, pp. 190–201.
[31] M. Parigot, Strong normalisation for second order classical natural deduction, Proc. Symp. on Logic in Computer Science, 1993, pp. 39–46.

[32] D. Prawitz, Natural Deduction, volume 3 of Stockholm Studies in Philosophy, Almqvist and Wiksell, Stockholm, 1965.
[33] D. Prawitz, Ideas and results in proof theory, in: J.E. Fenstad (Ed.), Proc. 2nd Scandinavian Logic Symp., 1971, 235–307.
[34] H. Schellinx, The noble art of linear decorating, Ph.D. Thesis, University of Amsterdam, February 1994.
[35] D.J. Shoesmith, T.J. Smiley, Multiple-conclusion logic, Cambridge University Press, Cambridge, 1978.
[36] A.S. Troelstra, Lectures on Linear Logic, Lecture Notes, vol. 29, CSLI, 1992.
[37] C. Urban, Ph.D. Thesis, 199x.