BRICS

Basic Research in Computer Science

BRICS RS-95-13 T. Braüner: The Girard Translation Extended with Recursion

The Girard Translation Extended with Recursion

Torben Braüner

BRICS Report Series **RS-95-13**

ISSN 0909-0878 **February 1995**

See back inner page for a list of recent publications in the BRICS Report Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@daimi.aau.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.aau.dk/BRICS/`
`ftp ftp.brics.aau.dk (cd pub/BRICS)`

The Girard Translation Extended with Recursion*

Torben Braüner

BRICS†
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark
Internet: tor@daimi.aau.dk

Abstract

This paper extends Curry-Howard interpretations of Intuitionistic Logic (IL) and Intuitionistic Linear Logic (ILL) with rules for recursion. The resulting term languages, the λ^{rec}-calculus and the linear λ^{rec}-calculus respectively, are given sound categorical interpretations. The embedding of proofs of IL into proofs of ILL given by the Girard Translation is extended with the rules for recursion, such that an embedding of terms of the λ^{rec}-calculus into terms of the linear λ^{rec}-calculus is induced via the extended Curry-Howard isomorphisms. This embedding is shown to be sound with respect to the categorical interpretations.

*Full version of paper to appear in Proceedings of CSL '94, LNCS 933, 1995.

†Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

Contents

1 Introduction

Linear Logic was discovered by J.-Y. Girard in 1987 and published in the now famous paper [Gir87]. In the abstract of this paper, it is stated that "a completely new approach to the whole area between constructive logics and computer science is initiated". Since then, a lot of work has been done to corroborate this claim. This paper deals with a Curry-Howard interpretation of the intuitionistic fragment of Linear Logic, appropriate for recursion.

The original Curry-Howard isomorphism, [How80], relates the natural deduction formulation of IL to the λ-calculus; formulas correspond to types, proofs to terms, and normalisation of proofs to reduction of terms. This is dealt with in [GLT89] and in [Abr90]; the first emphasizes the logic side of the isomorphism, the second the computational side. The fundamental idea of categorical logic is that formulas are interpreted as objects, proof-rules as natural operations on maps, and proofs as maps. We can give a sound interpretation of IL in a cartesian closed category, so given the above mentioned Curry-Howard isomorphism, this induces a sound categorical interpretation of the λ-calculus; types are interpreted as objects, typing rules as natural operations on arrows, and derivations of type assignments as maps.

ILL can be given a Curry-Howard interpretation in the same way as the Curry-Howard interpretation of IL. In [Abr90] the first Curry-Howard interpretation of ILL is introduced. The resulting term language is essentially a refinement of the usual λ-calculus where copying and discarding of values is written explicitly in the terms. One of the rules of this system has the property that it forces ! to be isomorphic to !! in any reasonable categorical interpretation, as pointed out in [Wad91]. In 1992 this was remedied by the authors of [BBdPH92] (and by the author of this paper) by changing the rule in an appropriate way, and by discovering a natural deduction formulation equivalent to the Gentzen style formulation of ILL (the hitherto known natural deduction formulation, [Mac91] did not possess that property). This work settled the question about how to interpret ILL via a Curry-Howard isomorphism. The Curry-Howard interpretation of ILL, the linear λ-calculus, is in [BBdPH92] given a sound categorical interpretation.

The [Gir87] paper introduced the Girard Translation which embeds IL into ILL. This translation works at the level of formulas as well as at the level of proofs. The Girard Translation at the level of proofs induce an embedding of terms of the λ-calculus into terms of the linear λ-calculus via the Curry-Howard isomorphisms. The embedding preserves β reductions. At the semantic level, a categorical model for the linear λ-calculus, that is, a linear category $\mathcal{C}$ with finite products, induces a categorical model for the λ-calculus, that is, a cartesian closed category. This is the category of free coalgebras with respect to the comonad on $\mathcal{C}$. So, given a categorical model for the linear λ-calculus, we can interpret the linear λ-calculus as well as the λ-calculus. The embedding of terms is shown to be sound with respect to the categorical interpretations, that is, it is shown to correspond to the adjunction between $\mathcal{C}$ and the category of free coalgebras in a precise way, which is a categorical generalisation of a result in [Gir87] showing that the Girard Translation is sound with respect to interpretation in the category of coherence spaces and linear stable functions.

We can give a sound categorical interpretation of the linear λ-calculus extended with finite sums using a linear category with finite products and finite sums. The category of free coalgebras induced by a linear category with finite products and finite sums has weak finite sums, and moreover, the weak finite sums satisfy certain additional conditions making it possible to give a categorical interpretation of the λ-calculus extended with finite sums, which is sound with respect to β and η reductions. Note that we give a sound interpretation using weak finite sums satisfying certain additional conditions, and not by using finite sums. This is a categorical generalisation of the sound interpretation given in [GLT89], where the category of coherence spaces and stable functions is used; a category that has weak finite sums, but not finite sums. We also have that the Curry-Howard isomorphisms and the Girard Translation still work in a way such that the induced embedding of terms is shown to be sound with respect to the categorical interpretations.

The above mentioned Curry-Howard isomorphisms will in the present paper be extended to deal with the λ-calculus and the linear λ-calculus extended with rules for recursion, the λ^{rec}-calculus and the linear λ^{rec}-calculus. The Girard Translation at the level of proofs is extended with the rules for recursion such that an embedding of terms of the λ^{rec}-calculus into terms of the linear λ^{rec}-calculus is induced. The embedding preserves the reduction rule for recursion. At the semantic level, a sound categorical model for the linear λ^{rec}-calculus, that is, a linear category $\mathcal{C}$ with finite products and a parametrised linear fixpoint operator, induces a sound categorical model for the λ^{rec}-calculus, that is, a cartesian closed category with a parametrised fixpoint operator, and moreover, the embedding of terms is shown to be sound with respect to the categorical interpretations.

An appendix on linear fixpoint objects is included for the sake of completeness. A linear fixpoint object is essentially a linear version of a fixpoint object in a cartesian closed category with a strong monad, [CP90]. It is shown that a linear category with a linear fixpoint object induces a unique parametrised linear fixpoint operator satisfying a linear version of Plotkin's Axiom. It is moreover shown how certain axioms entails that the linear fixpoint of a map is the least upper bound of an appropriate increasing chain with respect to the intrinsic preorder. This makes the induced coKleisli category rational, [AJM95] with respect to the partial order obtained as the quotient of the intrinsic preorder. Thus, we have paved the way for an adequacy result; this topic will be dealt with elsewhere in details.

We have moreover included an appendix with some remarks on recursion at the level of types.

The extensions of IL and ILL with rules for recursion makes every formula provable, so they should not be understood as having logical meaning. Neither should the extension of the Girard Translation and the extensions of the Curry-Howard isomorphisms be understood as having logical meaning. The role of the extensions are to extend the translation from terms in the λ-calculus to terms in the linear λ-calculus to a translation from terms in the λ^{rec}-calculus to terms in the linear λ^{rec}-calculus.

It is shown in [HP90] that a cartesian closed category with finite sums and a fixpoint operator is *inconsistent*, that is, it is equivalent to the category consisting of one object and one arrow. But the category of CPOs and strict continuous functions is a consistent linear category with finite sums and a linear fixpoint operator; so the presence of a linear fixpoint operator in a linear category is consistent with the presence of finite sums. Thus, the inconsistency of recursion with this standard construct vanish when we go to a linear context, which is in accordance with [Plo93].

In [MRA93] a different approach to recursion in a linear context is taken. The $(I, \otimes, \multimap)$ fragment of the linear λ-calculus is extended with natural numbers, corresponding to a weak natural numbers object in the categorical model. The discussion above implies that this approach is consistent with ours.

In what follows we will frequently abuse the use of notation by applying the same notation to denote the same kind of structure in different categories.

2 The Categorical Picture

2.1 Previous Work and Related Results

In what follows, we need the notion of a categorical model for the $(I, \otimes, \multimap, !)$ fragment of ILL as defined in [BBdPH92]:

Definition 2.1 *A* !-category *is a symmetric monoidal category* $(\mathcal{C}, I, \otimes)$ *equipped with:*

1. *A symmetric monoidal comonad* $(!, \varepsilon, \delta, m_I, m)$.
2. *Monoidal natural transformations* $e :!(-) \to I$ *and* $d :!(-) \to !(-) \otimes !(-)$ *such that*
   (a) e_A *and* d_A *are maps of coalgebras,*
   (b) e_A *and* d_A *give the free coalgebra* $(!A, \delta)$ *structure of a cocommutative comonoid,*
   (c) *maps between free coalgebras are maps between cocommutative comonoids.*

A linear category *is a* !*-category where each functor* $(-) \otimes A$ *has a right adjoint* $A \multimap (-)$.

The assumption that the comonad is symmetric monoidal means that ! is a symmetric monoidal functor and ε and δ are monoidal natural transformations. When assuming the natural transformations e and d to be monoidal, we are assuming the functors I and $!(-)\otimes!(-)$ to have the obvious monoidal structure induced by the monoidal structure on $!$. Hence, the assumption that the natural transformation e is monoidal amounts to commutativity of the following diagrams:

$$\begin{tikzcd}
!A\otimes!B \arrow[r, "m_{A,B}"] \arrow[d, "e_A \otimes e_B"'] & !(A\otimes B) \arrow[d, "e_{A\otimes B}"] \\
I\otimes I \arrow[r, "\cong"] & I
\end{tikzcd}
\qquad
\begin{tikzcd}
I \arrow[r, "m_I"] \arrow[dr, "I"'] & !I \arrow[d, "e_I"] \\
 & I
\end{tikzcd}$$

The assumption that the natural transformation d is monoidal amounts to commutativity of the following diagrams:

$$\begin{tikzcd}
!A\otimes!B \arrow[rr, "m_{A,B}"] \arrow[d, "d_A \otimes d_B"'] & & !(A\otimes B) \arrow[d, "d_{A\otimes B}"] \\
!A\otimes!A\otimes!B\otimes!B \arrow[r, "Id\otimes\cong\otimes Id"] & !A\otimes!B\otimes!A\otimes!B \arrow[r, "m_{A,B}\otimes m_{A,B}"] & !(A\otimes B)\otimes!(A\otimes B)
\end{tikzcd}$$

$$\begin{tikzcd}
I \arrow[rr, "m_I"] \arrow[dr, "\cong"'] & & !I \arrow[dd, "d_I"] \\
 & I\otimes I \arrow[dr, "m_I\otimes m_I"'] & \\
 & & !I\otimes!I
\end{tikzcd}$$

It can be shown that (I, m_I) and $(!A\otimes!A, (\delta_A\otimes\delta_A); m_{!A,!A})$ are coalgebras. The assumption that e_A is a map of coalgebras amounts to e_A being a map from $(!A, \delta_A)$ to (I, m_I), and the assumption that d_A is a map of coalgebras amounts to d_A being a map from $(!A, \delta_A)$ to $(!A\otimes!A, (\delta_A\otimes\delta_A); m_{!A,!A})$.

We also need the notion of a generalised coKleisli operator, as defined in [BBdPH92]:

Definition 2.2 *Given a map $f : !A_1 \otimes ... \otimes !A_n \to B$ in a $!$-category, we define $\gamma(f)$ to be the composite*

$$!A_1 \otimes ... \otimes !A_n \xrightarrow{\delta_{A_1}\otimes ... \otimes\delta_{A_n}} !!A_1 \otimes ... \otimes !!A_n \xrightarrow{m_{A_1,...,A_n}} !(!A_1 \otimes ... \otimes !A_n) \xrightarrow{!f} !B$$

In case $n = 1$ this definition is consistent with the usual coKleisli operator. Note that the definition of γ does not assume the presence of finite products. In [See89] another notion of a categorical model for ILL is defined where the induced generalised coKleisli operator relies on the existence of finite products.

Given a category $\mathcal{C}$ equipped with a comonad $(!, \varepsilon, \delta)$, the coEilenberg-Moore category, $\mathcal{C}^!$, is the category of coalgebras. We have an adjunction $U^! \dashv F^!$ between $\mathcal{C}^!$ and $\mathcal{C}$; the functor $U^! : \mathcal{C}^! \to \mathcal{C}$ simply forgets the coalgebra structure, while the functor $F^! : \mathcal{C} \to \mathcal{C}^!$ takes an object B to the coalgebra $(!B, \delta)$, and a map $f : B \to C$ to the map of coalgebras $!f : (!B, \delta) \to (!C, \delta)$. The adjunction between the functors $U^!$ and $F^!$ is given by the following natural bijection between maps:

$$\phi_{(A,h),B} : \mathcal{C}^!((A, h), F^!B) \cong \mathcal{C}(U^!(A, h), B)$$

where $\phi(f) = f; \varepsilon_B : A \to B$ and $\phi^{-1}(g) = h; !g : (A, h) \to (!B, \delta)$. Note that if (A, h) is equal to $(!A_1, \delta) \otimes ... \otimes (!A_n, \delta)$, then ϕ^{-1} coincides with the generalised coKleisli operator γ. The category of free coalgebras is the full subcategory of $\mathcal{C}^!$ whose objects are free coalgebras, that is, coalgebras of the kind $(!B, \delta)$. We will denote the category of free coalgebras by $Free(\mathcal{C}^!)$ in what follows. The adjunction $U^! \dashv F^!$ can be restricted to an adjunction between $Free(\mathcal{C}^!)$ and $\mathcal{C}$. Note that $Free(\mathcal{C}^!)$ is equivalent to the coKleisli category; it is straightforward to check that the comparison functor from $\mathcal{C}_!$ to $\mathcal{C}^!$ is an equivalence of categories when considered as a functor from $\mathcal{C}_!$ to $Free(\mathcal{C}^!)$.

Definition 2.3 *Let $\mathcal{C}$ be a category equipped with a comonad $(!, \varepsilon, \delta)$. A map of free coalgebras is called* linear *iff it is in the image of $F^!$.*

It is shown in [Bie94] that a symmetric monoidal comonad $(!, \varepsilon, \delta, m_I, m)$ on a symmetric monoidal category $(\mathcal{C}, I, \otimes)$ induces a symmetric monoidal structure on $\mathcal{C}^!$; the unit I of the tensor product is given by the coalgebra (I, m_I), and given two coalgebras (A, k) and (B, h), their tensor product $(A, k) \otimes (B, h)$ is given by the coalgebra $(A \otimes B, (k \otimes h); m_{A,B})$. If moreover $\mathcal{C}$ is a !-category, then the symmetric monoidal structure on $\mathcal{C}^!$ is a finite product structure, that is, I is a terminal object, and $(A, k) \otimes (B, h)$ is a binary product of (A, k) and (B, h) when equipped with $(A \otimes (h; e_B)); \cong$ and $((k; e_A) \otimes B); \cong$ as projections. Given (A, k), a unique map from (A, k) to I is given by $k; e_A$, and given maps $f : (C, l) \to (A, k)$ and $g : (C, l) \to (B, h)$, a unique map from (C, l) to $(A, k) \otimes (B, h)$ making the appropriate diagrams commute is given by $l; d_C; (e_C \otimes e_C); (f \otimes g)$.

Now, assume that a symmetric monoidal closed category $(\mathcal{C}, I, \otimes, \multimap)$ is equipped with a symmetric monoidal comonad $(!, \varepsilon, \delta, m_I, m)$. Then $\mathcal{C}^!$ has a symmetric monoidal structure, as mentioned above, and moreover, it is shown in [Bie94] that every free coalgebra is exponentiable with respect to the monoidal structure on $\mathcal{C}^!$. The internal-hom object $(A, h) \Rightarrow (!B, \delta)$ of (A, h) and $(!B, \delta)$ is given by $(!(A \multimap B), \delta)$, and we have the following bijections between maps such that each is natural in (C,k):

$$
\begin{array}{rcll}
\mathcal{C}^!((C,k)\otimes(A,h),(!B,\delta)) & = & \mathcal{C}^!((C,k)\otimes(A,h),F^!B) & \\
& \cong & \mathcal{C}(U^!((C,k)\otimes(A,h)),B) & U^! \dashv F^! \\
& = & \mathcal{C}(U^!(C,k)\otimes A,B) & \\
& \cong & \mathcal{C}(U^!(C,k),A\multimap B) & (-)\otimes A \dashv A\multimap(-) \\
& \cong & \mathcal{C}^!((C,k),F^!(A\multimap B)) & U^! \dashv F^! \\
& = & \mathcal{C}^!((C,k),(!(A\multimap B),\delta)) & \\
& = & \mathcal{C}^!((C,k),(A,h)\Rightarrow(!B,\delta)) &
\end{array}
$$

Definition 2.4 *Let $\mathcal{C}$ be a !-category. We will say that $Free(\mathcal{C}^!)$ is* closed under finite products *iff the terminal object I in $\mathcal{C}^!$ is isomorphic to a free coalgebra, and given two free coalgebras $(!A,\delta)$ and $(!B,\delta)$, their product $(!A,\delta)\otimes(!B,\delta)$ in $\mathcal{C}^!$ is isomorphic to a free coalgebra.*

Proposition 2.5 *Let $\mathcal{C}$ be a !-category. The category $Free(\mathcal{C}^!)$ is closed under finite products iff $Free(\mathcal{C}^!)$ has finite products $(1,\times)$ such that 1 is a terminal object in $\mathcal{C}^!$ and such that $(!A,\delta)\times(!B,\delta)$ is a binary product of $(!A,\delta)$ and $(!B,\delta)$ in $\mathcal{C}^!$.*

Proof. Straightforward calculation. □

If $\mathcal{C}$ is a !-category such that $Free(\mathcal{C}^!)$ is closed under finite products, then Proposition 2.5 entails that we have an isomorphism $n_I : I \to 1$ and a natural isomorphism $n_{(!A,\delta),(!B,\delta)} : (!A,\delta)\otimes(!B,\delta)\to(!A,\delta)\times(!B,\delta)$ because diagrams of the finite products $(1,\times)$ on $Free(\mathcal{C}^!)$ are universal in $\mathcal{C}^!$.

If $\mathcal{C}$ is a linear category such that $Free(\mathcal{C}^!)$ is closed under finite products, then the internal-hom objects induce a cartesian closed structure on $Free(\mathcal{C}^!)$. The finite products $(1,\times)$ is given by the assumption that $Free(\mathcal{C}^!)$ is closed under finite products, cf. Proposition 2.5, and an appropriate bijection between maps is given by the following composite of bijections:

$$
\begin{array}{rcll}
\mathcal{C}^!((!C,\delta)\times(!A,\delta),(!B,\delta)) & \cong & \mathcal{C}^!((!C,\delta)\otimes(!A,\delta),(!B,\delta)) & \text{composition with } n \\
& \cong & \mathcal{C}^!((!C,\delta),(!A,\delta)\Rightarrow(!B,\delta)) & (!B,\delta) \text{ is exponentiable}
\end{array}
$$

Since the composite of bijections is natural in $(!C,\delta)$, it follows from the Pointwise Adjointness Theorem, [BW90] that $(!A,\delta)\Rightarrow(-)$ can be extended to a right adjoint to $(-)\times(!A,\delta)$. Moreover, we have that

Proposition 2.6 *Let $\mathcal{C}$ be a linear category such that $Free(\mathcal{C}^!)$ is closed under finite products. Given a map $f : A\to B$, then the map*

$$(!C,\delta)\Rightarrow F^!(f) : (!C,\delta)\Rightarrow F^!(A) \longrightarrow (!C,\delta)\Rightarrow F^!(B)$$

is equal to $F^!(Id\multimap f)$ and thus linear.

Proof. The following calculation proves the result:

$$
\begin{array}{rcll}
(!C,\delta) \Rightarrow F^!(f) & = & \lambda(eval; F^!(f)) & \text{def. of } \Rightarrow \\
& = & \phi^{-1}(\lambda(\phi(n; n^{-1}; \phi^{-1}(\lambda^{-1}(\phi(Id))); F^!(f)))) & \text{def. of } \lambda \text{ and } eval \\
& = & \phi^{-1}(\lambda(\phi(\phi^{-1}(\lambda^{-1}(\phi(Id))); F^!(f)))) & \\
& = & \phi^{-1}(\lambda(\lambda^{-1}(\phi(Id)); f)) & \\
& = & \phi^{-1}(\phi(Id); (Id \multimap f)) & \\
& = & F^!(Id \multimap f) &
\end{array}
$$

□

Proposition 2.7 *Let $\mathcal{C}$ be a linear category such that $Free(\mathcal{C}^!)$ is closed under finite products. Given a map of free coalgebras $f : (!A,\delta) \to (!B,\delta)$, then the map*

$$f \Rightarrow (!C,\delta) : (!B,\delta) \Rightarrow (!C,\delta) \longrightarrow (!A,\delta) \Rightarrow (!C,\delta)$$

is equal to $F^!(f \multimap Id)$ and thus linear.

Proof. The following calculation proves the result:

$$
\begin{array}{rcll}
f \Rightarrow (!C,\delta) & = & \lambda((Id \times f); eval) & \text{def. of } \Rightarrow \\
& = & \phi^{-1}(\lambda(\phi(n^{-1}; (Id \times f); n; \phi^{-1}(\lambda^{-1}(\phi(Id)))))) & \text{def. of } \lambda \text{ and } eval \\
& = & \phi^{-1}(\lambda(\phi((Id \otimes f); \phi^{-1}(\lambda^{-1}(\phi(Id)))))) & \\
& = & \phi^{-1}(\lambda((Id \otimes f); \lambda^{-1}(\phi(Id)))) & \\
& = & \phi^{-1}(\lambda((\phi(Id) \otimes f); eval)) & \\
& = & \phi^{-1}(\phi(Id); \lambda((Id \otimes f); eval)) & \\
& = & \phi^{-1}(\phi(Id); (f \multimap Id)) & \text{def. of } \multimap \\
& = & F^!(f \multimap Id) &
\end{array}
$$

□

Now, under which circumstances is $Free(\mathcal{C}^!)$ closed under finite products? The following observation induces a sufficient condition:

Proposition 2.8 *Let $\mathcal{C}$ be a category equipped with a comonad $(!,\varepsilon,\delta)$. If $\mathcal{C}$ has terminal object 1 then $(!1,\delta)$ is a terminal object in $\mathcal{C}^!$, and if $\mathcal{C}$ has binary products $\times$ then $(!(A \times B),\delta)$ is a binary product of $(!A,\delta)$ and $(!B,\delta)$ in $\mathcal{C}^!$.*

Proof. The free functor $F^! : \mathcal{C} \to \mathcal{C}^!$ preserves finite products because it has a left adjoint, so a terminal object 1 in $\mathcal{C}$ is sent into a terminal object $(!1,\delta)$ in $\mathcal{C}^!$, and a product diagram $A \leftarrow A \times B \to B$ in $\mathcal{C}$ is sent into a product diagram $(!A,\delta) \leftarrow (!(A \times B),\delta) \to (!B,\delta)$ in $\mathcal{C}^!$. □

We will for later use note a couple of facts following from the proof of Proposition 2.8. The following bijections

$$
\begin{array}{rcll}
\mathcal{C}^!((C,k),(!1,\delta)) & = & \mathcal{C}^!((C,k),F^!1) & \\
& \cong & \mathcal{C}(U^!(C,k),1) & U^! \dashv F^! \\
& = & \mathcal{C}(C,1) &
\end{array}
$$

make $(!1, \delta)$ a terminal object, and the following bijections

$$\begin{array}{rcll} \mathcal{C}^!((C,k),(!A,\delta)) \times \mathcal{C}^!((C,k),(!B,\delta)) & = & \mathcal{C}^!((C,k),F^!A) \times \mathcal{C}^!((C,k),F^!B) & \\ & \cong & \mathcal{C}(U^!(C,k),A) \times \mathcal{C}(U^!(C,k),B) & U^! \dashv F^! \\ & \cong & \mathcal{C}(U^!(C,k),A \times B) & \\ & \cong & \mathcal{C}^!((C,k),F^!(A \times B)) & U^! \dashv F^! \\ & = & \mathcal{C}^!((C,k),(!(A \times B),\delta)) & \end{array}$$

make $(!A, \delta) \leftarrow (!(A \times B), \delta) \rightarrow (!B, \delta)$ a product diagram.

If $\mathcal{C}$ is a !-category with finite products, then Proposition 2.8 entails that $Free(\mathcal{C}^!)$ is closed under finite products. This induce an isomorphism $n_1 : I \rightarrow !1$, and a natural isomorphism $n_{A,B} : !A \otimes !B \rightarrow !(A \times B)$. It can be shown that the map n_1 and the natural isomorphism n makes ! a symmetric monoidal functor from $(\mathcal{C}, 1, \times)$ to $(\mathcal{C}, I, \otimes)$. Given these isomorphisms, we can define a model for ILL as described in [See89]. Calculations show that the way the isomorphisms are defined and the universal properties of I and $(!A, \delta) \otimes (!B, \delta)$ in $\mathcal{C}^!$ forces ! to take the cocommutative comonoid structure with respect to the finite products to the cocommutative comonoid structure with respect to the symmetric monoidal structure, that is, e_A is equal to the composite

$$!A \xrightarrow{!<>} !1 \xrightarrow{n^{-1}} I$$

and d_A is equal to the composite

$$!A \xrightarrow{!\Delta} !(A \times A) \xrightarrow{n^{-1}} !A \otimes !A$$

If $\mathcal{C}$ is a category with a comonad $(!, \varepsilon, \delta)$, then an initial object 0 in $\mathcal{C}$ induces a weak initial object in $Free(\mathcal{C}^!)$ and binary sums $+$ in $\mathcal{C}$ induce weak binary sums in $Free(\mathcal{C}^!)$, as pointed out in [Bie94]. The weak initial object 0 is given by the free coalgebra $(!0, \delta)$, and given two free coalgebras $(!A, \delta)$ and $(!B, \delta)$, their weak binary sum $(!A, \delta) + (!B, \delta)$ is given by the free coalgebra $(!(!A + !B), \delta)$ when equipped with $\delta; !in_1$ and $\delta; !in_2$ as injections. Given a free coalgebra $(!C, \delta)$, a map from 0 to $(!C, \delta)$ is given by $![]$, and given maps $f : (!A, \delta) \rightarrow (!C, \delta)$ and $g : (!B, \delta) \rightarrow (!C, \delta)$, a map from $(!A, \delta) + (!B, \delta)$ to $(!C, \delta)$ making the appropriate diagrams commute is given by $![(f; \varepsilon), (g; \varepsilon)]$. One can actually show more than that:

Proposition 2.9 *Let $\mathcal{C}$ be a category with a comonad. If $\mathcal{C}$ has an initial object 0, then the family of maps of free coalgebras $[] : 0 \rightarrow (!C, \delta)$ induced by the weak initial object is natural in $(!C, \delta)$ with respect to linear maps. If $\mathcal{C}$ has binary sums $+$, then the operation on maps of free coalgebras*

$$[-,+] : \mathcal{C}^!((!A,\delta),(!C,\delta)) \times \mathcal{C}^!((!B,\delta),(!C,\delta)) \rightarrow \mathcal{C}^!((!A,\delta)+(!B,\delta),(!C,\delta))$$

induced by the weak binary sums is natural in $(!C, \delta)$ with respect to linear maps.

Proof. Let a map $h : C \to D$ be given. Then

$$\begin{array}{rcll} [];!h &=& ![];!h & \text{def. of } [] \text{ in } Free(\mathcal{C}^!) \\ &=& ![] & \\ &=& [] & \text{def. of } [] \text{ in } Free(\mathcal{C}^!) \end{array}$$

and given maps $f : (!A, \delta) \to (!C, \delta)$ and $g : (!B, \delta) \to (!C, \delta)$, we have

$$\begin{array}{rcll} [f, g];!h &=& ![(f;\varepsilon), (g;\varepsilon)];!h & \text{def. of } [-,+] \text{ in } Free(\mathcal{C}^!) \\ &=& ![(f;\varepsilon;h), (g;\varepsilon;h)] & \\ &=& ![(f;!h;\varepsilon), (g;!h;\varepsilon)] & \\ &=& [(f;!h), (g;!h)] & \text{def. of } [-,+] \text{ in } Free(\mathcal{C}^!) \end{array}$$

□

Proposition 2.10 *Let $\mathcal{C}$ be a category with a comonad. If $\mathcal{C}$ has an initial object 0, then the map of free coalgebras $[] : 0 \to 0$ induced by the weak initial object is equal to the identity. If $\mathcal{C}$ has binary sums $+$, then the map of free coalgebras*

$$[in_1, in_2] : (!A, \delta) + (!B, \delta) \to (!A, \delta) + (!B, \delta)$$

induced by the weak binary sums is equal to the identity.

Proof. Straightforward calculation. □

The last two results might seem ad hoc, but it turns out that they are suffucint to give a categorical interpretation of the λ-calculus extended with finite sums which is sound with respect to β and η reductions.

2.2 How to deal with parameters

Before considering fixpoints, we need some results enabling us to deal with parameters. Let $\mathcal{C}$ be a !-category. With any object A one may associate a comonad $(S^A, \varepsilon^A, \delta^A)$. The functor S^A is given by $!A \otimes (-)$, a component ε^A_B is given by

$$S^A B = !A \otimes B \xrightarrow{e_A \otimes B} I \otimes B \cong B$$

and a component δ^A_B is given by the composite

$$S^A B = !A \otimes B \xrightarrow{d_A \otimes B} (!A \otimes !A) \otimes B \cong !A \otimes (!A \otimes B) = S^A S^A B$$

We will in what follows show that the functor S^A together with the natural transformations ε^A and δ^A indeed is a comonad. Commutativity of the following diagram

$$\begin{tikzcd}
S^A \arrow[r, "\delta^A"] \arrow[d, "\delta^A"'] & S^A S^A \arrow[d, "\delta^A S^A"'] \\
S^A S^A \arrow[r, "S^A \delta^A"] & S^A S^A S^A
\end{tikzcd}$$

follows from commutativity of the following diagram

$$\begin{tikzcd}
!A \otimes B \arrow[r, "d \otimes B"] \arrow[dd, "d \otimes B"'] & (!A \otimes !A) \otimes B \arrow[r, "\cong"] \arrow[d, "(d \otimes Id) \otimes B"'] & !A \otimes (!A \otimes B) \arrow[d, "d \otimes (Id \otimes B)"'] \\
\arrow[r, phantom, "1."] & ((!A \otimes !A) \otimes !A) \otimes B \arrow[r, "\cong"] \arrow[d, "\cong \otimes B"'] & (!A \otimes !A) \otimes (!A \otimes B) \arrow[dd, "\cong"'] \\
(!A \otimes !A) \otimes B \arrow[r, "(Id \otimes d) \otimes B"] \arrow[d, "\cong"'] & (!A \otimes (!A \otimes !A)) \otimes B \arrow[d, "\cong"'] & \\
!A \otimes (!A \otimes B) \arrow[r, "Id \otimes (d \otimes B)"] & !A \otimes ((!A \otimes !A) \otimes B) \arrow[r, "Id \otimes \cong"] & !A \otimes (!A \otimes (!A \otimes B))
\end{tikzcd}$$

1. d is part of a comonoid.

Commutativity of the following diagram

$$\begin{tikzcd}
& S^A \arrow[dl, "Id"'] \arrow[d, "\delta^A"'] \arrow[dr, "Id"] & \\
S^A & S^A S^A \arrow[l, "S^A \varepsilon^A"'] \arrow[r, "\varepsilon^A S^A"] & S^A
\end{tikzcd}$$

follows from commutativity of the following two diagrams

$$\begin{tikzcd}
 & !A \otimes B \arrow[dl, "Id"'] \arrow[d, "\cong \otimes B"'] \arrow[dr, phantom, "1."] & !A \otimes B \arrow[l, "Id"'] \arrow[d, "d \otimes B"] \\
!A \otimes B \arrow[d, "Id"'] & (!A \otimes I) \otimes B \arrow[l, "\cong \otimes B"'] \arrow[d, "\cong"'] & (!A \otimes !A) \otimes B \arrow[l, "(Id \otimes e) \otimes B"'] \arrow[d, "\cong"'] \\
!A \otimes B & !A \otimes (I \otimes B) \arrow[l, "Id \otimes \cong"'] & !A \otimes (!A \otimes B) \arrow[l, "Id \otimes (e \otimes B)"']
\end{tikzcd}$$

1. e is part of a comonoid.

$$\begin{tikzcd}
!A \otimes B \arrow[r, "Id"] \arrow[d, "d \otimes B"'] \arrow[dr, phantom, "1."] & !A \otimes B \arrow[d, "\cong \otimes B"'] \arrow[dr, "Id"] & \\
(!A \otimes !A) \otimes B \arrow[r, "(e \otimes Id) \otimes B"] \arrow[d, "\cong"'] & (I \otimes !A) \otimes B \arrow[r, "\cong \otimes B"] \arrow[d, "\cong"'] & !A \otimes B \arrow[d, "Id"'] \\
!A \otimes (!A \otimes B) \arrow[r, "e \otimes (Id \otimes B)"] & I \otimes (!A \otimes B) \arrow[r, "\cong"] & !A \otimes B
\end{tikzcd}$$

1. e is part of a comonoid.

Moreover, we have a distributive law $\lambda^A : S^A! \to !S^A$ of the comonad $(!, \varepsilon, \delta)$ over the comonad $(S^A, \varepsilon^A, \delta^A)$, [BW85]. A component λ^A_B is given by the composite:

$$S^A!B = !A \otimes !B \xrightarrow{\delta_A \otimes Id} !!A \otimes !B \xrightarrow{m} !(!A \otimes B) = !S^A B$$

We will in what follows show that the natural transformation λ indeed is a distributive law. Commutativity of the following diagram

$$\begin{tikzcd}
S^A! \arrow[rr, "\lambda^A"] \arrow[dr, "\varepsilon^A !"'] & & !S^A \arrow[dl, "!\varepsilon^A"] \\
 & ! &
\end{tikzcd}$$

follows from commutativity of the following diagram

$$\begin{tikzcd}
!A\otimes !B \arrow[r, "\delta\otimes Id"] \arrow[d, "e\otimes Id"'] \arrow[dr, phantom, "1."] & !!A\otimes !B \arrow[r, "m"] \arrow[d, "!e\otimes Id"] & !(!A\otimes B) \arrow[d, "!(e\otimes B)"] \\
I\otimes !B \arrow[r, "m_I\otimes Id"] \arrow[dr, "\cong"'] & !I\otimes !B \arrow[r, "m"] \arrow[d, phantom, "2."] & !(I\otimes B) \arrow[dl, "!\cong"] \\
 & !B &
\end{tikzcd}$$

1. e is a map of coalgebras. 2. ! is monoidal.

Commutativity of the following diagram

$$\begin{tikzcd}
S^A! \arrow[rr, "\lambda^A"] \arrow[dr, "S^A\varepsilon"'] & & !S^A \arrow[dl, "\varepsilon S^A"] \\
 & S^A &
\end{tikzcd}$$

follows from commutativity of the following diagram

$$\begin{tikzcd}
!A\otimes !B \arrow[r, "\delta\otimes Id"] \arrow[dr, "Id\otimes\varepsilon"'] & !!A\otimes !B \arrow[r, "m"] \arrow[d, "\varepsilon\otimes\varepsilon"] \arrow[dr, phantom, "1."] & !(!A\otimes B) \arrow[dl, "\varepsilon"] \\
 & !A\otimes B &
\end{tikzcd}$$

1. ε is monoidal.

Commutativity of the following diagram

$$\begin{tikzcd}
S^A! \arrow[rr, "\lambda^A"] \arrow[d, "\delta^A!"'] & & !S^A \arrow[d, "!\delta^A"] \\
S^AS^A! \arrow[r, "S^A\lambda^A"] & S^A!S^A \arrow[r, "\lambda^AS^A"] & !S^AS^A
\end{tikzcd}$$

follows from commutativity of the following diagram

$$\begin{tikzcd}
!A\otimes !B \arrow[rr, "\delta\otimes Id"] \arrow[d, "d\otimes Id"'] \arrow[drr, phantom, "1."] & & !!A\otimes !B \arrow[r, "m"] \arrow[d, "!d\otimes Id"'] & !(!A\otimes B) \arrow[d, "!(d\otimes Id)"'] \\
(!A\otimes !A)\otimes !B \arrow[r, "(\delta\otimes\delta)\otimes Id"] \arrow[d, "\cong"'] & (!!A\otimes !!A)\otimes !B \arrow[r, "m\otimes Id"] \arrow[d, "\cong"'] & !(!A\otimes !A)\otimes !B \arrow[r, "m"] \arrow[d, phantom, "2."] & !((!A\otimes !A)\otimes B) \arrow[d, "!\cong"'] \\
!A\otimes(!A\otimes !B) \arrow[r, "\delta\otimes(\delta\otimes Id)"] & !!A\otimes(!!A\otimes !B) \arrow[r, "Id\otimes m"] & !!A\otimes !(!A\otimes B) \arrow[r, "m"] & !(!A\otimes(!A\otimes B))
\end{tikzcd}$$

1. d is a map of coalgebras. 2. ! is monoidal.

Commutativity of the following diagram

$$\begin{tikzcd}
S^A! \arrow[rr, "\lambda^A"] \arrow[d, "S^A\delta"'] & & !S^A \arrow[d, "\delta S^A"'] \\
S^A!! \arrow[r, "\lambda^A!"] & !S^A! \arrow[r, "!\lambda^A"] & !!S^A
\end{tikzcd}$$

follows from commutativity of the following diagram

$$\begin{tikzcd}
!A\otimes !B \arrow[r, "\delta\otimes Id"] \arrow[d, "\delta\otimes\delta"'] & !!A\otimes !B \arrow[r, "m"] \arrow[d, "\delta\otimes\delta"'] & !(!A\otimes B) \arrow[dd, "\delta"'] \\
!!A\otimes !!B \arrow[r, "!\delta\otimes Id"] \arrow[d, "m"'] & !!!A\otimes !!B \arrow[d, "m"'] \arrow[r, phantom, "1."] & \\
!(!A\otimes !B) \arrow[r, "!(\delta\otimes Id)"] & !(!!A\otimes !B) \arrow[r, "!m"] & !!(!A\otimes B)
\end{tikzcd}$$

1. δ is monoidal.

This induce a lifting of the comonad $(S^A, \varepsilon^A, \delta^A)$ to $\mathcal{C}^!$, [BW85], that is, a comonad $(S^{A*}, \varepsilon^{A*}, \delta^{A*})$ on $\mathcal{C}^!$ such that

1. $U^! S^{A*} = S^A U^!$

2. $U^!\varepsilon^{A*} = \varepsilon^A U^!$

3. $U^!\delta^{A*} = \delta^A U^!$

The only part of the comonad $(S^{A*}, \varepsilon^{A*}, \delta^{A*})$ that is not determined by these conditions is the structure map of $S^{A*}(B,h)$, which is given by the composite

$$S^A B \xrightarrow{S^A(h)} S^A !B \xrightarrow{\lambda^A_B} !S^A B$$

Moreover, we have an adjunction between $\mathcal{C}_{S^A}$ and $\mathcal{C}^!_{S^{A*}}$ such that the right (left) adjoints of the following diagram commute:

$$\begin{tikzcd}[row sep=large, column sep=large]
\mathcal{C}^!_{S^{A*}} \arrow[d, shift right=2, "\widetilde{U^!}"'] \arrow[d, phantom, "\dashv"] \arrow[r, shift right=2, "U_{S^{A*}}"'] \arrow[r, phantom, "\top"] & \mathcal{C}^! \arrow[l, shift right=2, "F_{S^{A*}}"'] \arrow[d, shift right=2, "U^!"'] \arrow[d, phantom, "\dashv"] \\
\mathcal{C}_{S^A} \arrow[u, shift right=2, "\widetilde{F^!}"'] \arrow[r, shift right=2, "U_{S^A}"'] \arrow[r, phantom, "\top"] & \mathcal{C} \arrow[l, shift right=2, "F_{S^A}"'] \arrow[u, shift right=2, "F^!"']
\end{tikzcd}$$

The functor $\widetilde{U^!}$ simply forgets the coalgebra structure. Given an object B in $\mathcal{C}_{S^A}$, which has the same objects as $\mathcal{C}$, then $\widetilde{F^!}B$ is given by $F^!C$. An appropriate bijection $\tilde{\phi}_{(B,h),C}$ between maps is given by the following composite of bijections

$$\begin{array}{rcll}
\mathcal{C}^!_{S^{A*}}((B,h), \widetilde{F^!}C) & = & \mathcal{C}^!_{S^{A*}}((B,h), F^!C) & \\
& = & \mathcal{C}^!(S^{A*}(B,h), F^!C) & U_{S^{A*}} \dashv F_{S^{A*}} \\
& = & \mathcal{C}(U^!S^{A*}(B,h), C) & U^! \dashv F^! \\
& = & \mathcal{C}(S^A U^!(B,h), C) & \text{condition 1.} \\
& = & \mathcal{C}_{S^A}(U^!(B,h), C) & U_{S^A} \dashv F_{S^A} \\
& = & \mathcal{C}_{S^A}(\widetilde{U^!}(B,h), C) &
\end{array}$$

Since the composite of bijections is natural in (B,h), it follows from the Pointwise Adjointness Theorem, [BW90] that $\widetilde{F^!}$ can be extended to a right adjoint to $\widetilde{U^!}$. Given a map $f : B \to C$ in $\mathcal{C}_{S^A}$, that is, a map $f : S^A B \to C$ in $\mathcal{C}$, then the map $\widetilde{F^!}(f)$ is equal to the composite

$$S^{A*}\widetilde{F^!}B = (S^A!B, S^A(\delta); \lambda^A) \xrightarrow{\lambda^A} (!S^A B, \delta) \xrightarrow{!f} (!C, \delta) = \widetilde{F^!}C$$

The comonad on $\mathcal{C}_{S^A}$ induced by the adjunction $\widetilde{U^!} \dashv \widetilde{F^!}$ will be denoted $(\tilde{!}, \tilde{\varepsilon}, \tilde{\delta})$; it has the following properties

1. $\tilde{!}F_{S^A} = F_{S^A}!$

2. $\tilde{\varepsilon} F_{S^A} = F_{S^A} \varepsilon$

3. $\tilde{\delta} F_{S^A} = F_{S^A} \delta$

Note that the construction of the adjunction $\widetilde{U^!} \dashv \widetilde{F^!}$ making the mentioned diagrams commute could have been done with an arbitrary distributive law of one comonad over another comonad.

Now, with any object D in a cartesian category $(\mathcal{D}, \times, 1)$ we may associate a comonad, [Sim92]; the functor part is given by $D \times (-)$, a component of the counit is given by

$$D \times B \xrightarrow{\pi_2} B$$

and a component of the comultiplication is given by

$$D \times B \xrightarrow{<\pi_1, Id>} D \times (D \times B)$$

which is equal to the composite

$$D \times B \xrightarrow{\Delta \times Id} (D \times D) \times B \cong D \times (D \times B)$$

If $\mathcal{C}$ is a !-category, then $\mathcal{C}^!$ has the cartesian structure $(\otimes, I)$, and it turns out that the comonad induced by the object $(!A, \delta)$ of $\mathcal{C}^!$ by using this construction is the same as $(S^{A*}, \varepsilon^{A*}, \delta^{A*})$.

The functor $F_{S^{A*}} : \mathcal{C}^! \to \mathcal{C}^!_{S^{A*}}$ preserves limits because it has a left adjoint, so I is a terminal object in $\mathcal{C}^!_{S^{A*}}$. We will not assume that the category $\mathcal{C}_{S^A}$ has a terminal object, but the following weaker notion will do:

Definition 2.11 *A* pre !-category *is a category $\mathcal{C}$ together with an object I, a comonad $(!, \varepsilon, \delta)$, and a map $m_I : I \to !I$ such that the following diagrams commute*

$$\begin{tikzcd} I \arrow[r, "m_I"] \arrow[dr, "Id"'] & !I \arrow[d, "\varepsilon"] \\ & I \end{tikzcd} \qquad \begin{tikzcd} ! \arrow[r, "m_I"] \arrow[d, "m_I"'] & !I \arrow[d, "\delta"] \\ !I \arrow[r, "!m_I"] & !!I \end{tikzcd}$$

This is simply the definition of a monoidal comonad on a monoidal category where the bifunctor part has been left out. This enables us to define the generalised coKleisli operator γ in the nullary and the unary case, Definition 2.2. A !-category is obviously a pre !-category, and moreover, it is straightforward to check that the category $\mathcal{C}_{S^A}$ together with the object I, the comonad $(\tilde{!}, \tilde{\varepsilon}, \tilde{\delta})$, and the map $F_{S^A}(m_I)$ constitutes a pre !-category. We will for later use note that given a map $g : I \to C$ in $\mathcal{C}$, then $\gamma(F_{S^A}(g)) = F_{S^A}(\gamma(g))$, and similarly, given a map $f : !B \to C$ in $\mathcal{C}$, then $\gamma(F_{S^A}(f)) = F_{S^A}(\gamma(f))$.

2.3 Fixpoints in Cartesian Categories

The main concern of this subsection will be a parametrised version of fixpoints in cartesian categories as introduced in [Law69].

Definition 2.12 *Let $\mathcal{C}$ be a category with a terminal object* 1*. A map $f^\sharp : 1 \to B$ is a* fixpoint *of the map $f : B \to B$ iff $f^\sharp = f^\sharp; f$. A* fixpoint operator *is an operation on maps*

$$(-)^\sharp_B : \mathcal{C}(B, B) \longrightarrow \mathcal{C}(1, B)$$

such that $f^\sharp$ is a fixpoint of f for any map $f : B \to B$.

Definition 2.13 *Let $(\mathcal{C}, \times, 1)$ be a cartesian category. A map $f^\sharp : A \to B$ is a* parametrised fixpoint *of the map $f : A \times B \to B$ iff $f^\sharp$ is equal to the composite*

$$A \xrightarrow{\Delta} A \times A \xrightarrow{A \times f^\sharp} A \times B \xrightarrow{f} B$$

A parametrised fixpoint operator *is an operation on maps*

$$(-)^\sharp_{A,B} : \mathcal{C}(A \times B, B) \longrightarrow \mathcal{C}(A, B)$$

such that $f^\sharp$ is a parametrised fixpoint of f for any map $f : A \times B \to B$. A parametrised fixpoint operator is natural *if the operation is natural in A.*

Note how the diagonal map $\Delta_A : A \to A \times A$ is used to copy parameters.

Proposition 2.14 *Let $(\mathcal{C}, \times, 1)$ be a cartesian category. A map $f^\sharp : A \to B$ is a parametrised fixpoint of the map $f : A \times B \to B$ iff the map $\cong; f^\sharp : A \times 1 \to B$ is a fixpoint of $f : A \times B \to B$ in $\mathcal{C}_{A\times(-)}$, thus, to give a parametrised fixpoint operator on $\mathcal{C}$ is equivalent to give a fixpoint operator on each category $\mathcal{C}_{A\times(-)}$.*

Proof. Note that 1 is a terminal object in $\mathcal{C}_{A\times(-)}$. The result follows from straightforward calculation. □

We can deal with internal fixpoint operators if the category is closed with respect to the cartesian structure:

Definition 2.15 *Let $(\mathcal{C}, \times, 1, \Rightarrow)$ be a cartesian closed category. An* internal fixpoint operator *is a family of maps $Y_B : B \Rightarrow B \to B$ such that $\lambda(f); Y_B$ is a parametrised fixpoint of f for any map $f : A \times B \to B$.*

Now, one can show that a cartesian closed category has a parametrised fixpoint operator iff it has an internal fixpoint operator, [Poi92], but we will show here a more informative result:

Proposition 2.16 *There is a bijective correspondence between natural parametrised fixpoint operators and internal fixpoint operators in a cartesian closed category.*

Proof. Assume that $\mathcal{C}$ has a natural parametrised fixpoint operator $(-)^\sharp$, and define a map $Y = eval^\sharp$ for every object. Let a map $f : A \times B \to B$ be given, then

$$\begin{array}{rcll} \lambda(f); Y & = & \lambda(f); eval^\sharp & \text{def. of } Y \\ & = & ((\lambda(f) \times B); eval)^\sharp & \\ & = & f^\sharp & \end{array}$$

which entails that $\lambda(f); Y$ is a parametrised fixpoint of f. The construction of internal fixpoint operators from natural parametrised fixpoint operators is clearly injective. Conversely, assume that Y is an internal fixpoint operator on $\mathcal{C}$, and define an operation on maps $\lambda(-); Y$. This is a natural parametrised fixpoint operator because $\lambda(f); Y$ is a parametrised fixpoint of f for any map $f : A \times B \to B$, and naturality of the operation follows from naturality of λ. The construction of natural parametrised fixpoint operators from internal fixpoint operators is injective because the following calculation

$$\begin{array}{rcl} \lambda(eval); Y & = & Id; Y \\ & = & Y \end{array}$$

shows that the internal fixpoint operator induced by $\lambda(-); Y$ is equal to Y. $\Box$

2.4 Linear Fixpoints

We will now consider fixpoints in a linear context. The previous definition of fixpoints can not be used because it assumes the presence of finite products.

Definition 2.17 *Let $\mathcal{C}$ be a pre !-category. A map $f^\sharp : I \to B$ is a* linear fixpoint *of the map $f : !B \to B$ iff $f^\sharp = \gamma(f^\sharp); f$. A* linear fixpoint operator *is an operation on maps*

$$(-)^\sharp_B : \mathcal{C}(!B, B) \longrightarrow \mathcal{C}(I, B)$$

such that $f^\sharp$ is a linear fixpoint of f for any map $f : !B \to B$.

We get back Definition 2.12 if we equip a category with a terminal object with the identity functor and the identity map.

Definition 2.18 *Let $\mathcal{C}$ be a !-category. A map $f^\sharp : !A \to B$ is a* parametrised linear fixpoint *of the map $f : !A \otimes !B \to B$ iff $f^\sharp$ is equal to the composite*

$$!A \xrightarrow{d} !A \otimes !A \xrightarrow{Id \otimes \gamma(f^\sharp)} !A \otimes !B \xrightarrow{f} B$$

A parametrised linear fixpoint operator *is an operation on maps*

$$(-)^{\sharp}_{A,B} : \mathcal{C}(!A \otimes !B, B) \longrightarrow \mathcal{C}(!A, B)$$

such that $f^{\sharp}$ *is a parametrised linear fixpoint of* f *for any map* $f : !A \otimes !B \to B$*. A parametrised linear fixpoint operator is* natural *if the operation is natural in* $!A$ *with respect to maps of free coalgebras.*

Note that maps of free coalgebras is the same as maps in the image of γ. We get back Definition 2.13 if we equip a cartesian category with the identity comonad and the natural transformation induced by the diagonal maps. The definition of a parametrised linear fixpoint is simply an extension of the definition of parametrised fixpoints in a cartesian category to a linear context, where we have only a "diagonal map" d_A for objects of the shape $!A$.

Proposition 2.19 *Let* $\mathcal{C}$ *be a !-category. A map* $f^{\sharp} : !A \to B$ *is a parametrised linear fixpoint of the map* $f : !A \otimes !B \to B$ *iff the map* $\cong; f^{\sharp} : !A \otimes I \to B$ *is a linear fixpoint of* $f : !A \otimes !B \to B$ *in* $\mathcal{C}_{S^A}$*, thus, to give a parametrised linear fixpoint operator on* $\mathcal{C}$ *is equivalent to give a linear fixpoint operator on each category* $\mathcal{C}_{S^A}$*.*

Proof. Note that we consider $\mathcal{C}_{S^A}$ as a pre !-category. The result follows from some straightforward calculation. □

The definition of a linear fixpoint in $\mathcal{C}_{S^A}$ is essentially the definition of a fixpoint in $\mathcal{C}^{!}_{S^{A*}}$ stated in terms of maps of $\mathcal{C}_{S^A}$, which gives us the following results:

Lemma 2.20 *Let* $\mathcal{C}$ *be a !-category. A map* $h : (!A, \delta) \otimes I \to (!B, \delta)$ *is a fixpoint of a map* $f : (!A, \delta) \otimes (!B, \delta) \to (!B, \delta)$ *in* $\mathcal{C}^{!}_{S^{A*}}$ *iff* $\tilde{\phi}(h) : !A \otimes I \to B$ *is a linear fixpoint of* $\tilde{\phi}(f) : !A \otimes !B \to B$ *in* $\mathcal{C}_{S^A}$*, thus, to give a linear fixpoint operator on* $\mathcal{C}_{S^A}$ *is equivalent to give a fixpoint operator on* $\mathcal{C}^{!}_{S^{A*}}$ *restricted to free coalgebras.*

Proof. Note that we consider $\mathcal{C}^{!}_{S^{A*}}$ together with the terminal object I and we consider $\mathcal{C}_{S^A}$ as a pre !-category. The following calculation, where composition is in $\mathcal{C}_{S^A}$, proves the result:

$$\begin{array}{lll}
h \text{ is a fixpoint of } f & \text{iff} & h = h; f \\
& \text{iff} & h = \tilde{\phi}^{-1}(\tilde{\phi}(h)); f \\
& \text{iff} & \tilde{\phi}(h) = \tilde{\phi}(\tilde{\phi}^{-1}(\tilde{\phi}(h)); f) \\
& \text{iff} & \tilde{\phi}(h) = \tilde{\phi}^{-1}(\tilde{\phi}(h)); \tilde{\phi}(f) \\
& \text{iff} & \tilde{\phi}(h) = \gamma(\tilde{\phi}(h)); \tilde{\phi}(f) \qquad\qquad * \\
& \text{iff} & \tilde{\phi}(h) \text{ is a linear fixpoint of } \tilde{\phi}(f)
\end{array}$$

* The underlying map of $\tilde{\phi}^{-1}(\tilde{\phi}(h))$ is equal to $S^A(m_I); \lambda^A; !\tilde{\phi}(h)$ because $S^A(m_I); \lambda^A$ is structure map for the coalgebra $S^{A*}I$. But $\lambda^A; !\tilde{\phi}(h)$ is equal to $\tilde{!}\tilde{\phi}(h)$, and $S^A(m_I); \tilde{!}\tilde{\phi}(h)$ is equal to $F_{S^A}(m_I); \tilde{!}\tilde{\phi}(h)$ where composition is in $\mathcal{C}_{S^A}$. Thus, $\tilde{\phi}^{-1}(\tilde{\phi}(h))$ is equal to $\gamma(\tilde{\phi}(h))$. □

Lemma 2.21 *Let $\mathcal{C}$ be a !-category such that $Free(\mathcal{C}^!)$ is closed under finite products. A map $h : (!A, \delta) \otimes I \to (!B, \delta)$ is a fixpoint of a map $f : (!A, \delta) \otimes (!B, \delta) \to (!B, \delta)$ in $\mathcal{C}^!_{SA}$ iff the composite*

$$(!A, \delta) \times 1 \xrightarrow{n^{-1}} (!A, \delta) \otimes 1 \xrightarrow{Id \otimes n_1^{-1}} (!A, \delta) \otimes I \xrightarrow{h} (!B, \delta)$$

is a fixpoint of the composite

$$(!A, \delta) \times (!B, \delta) \xrightarrow{n^{-1}} (!A, \delta) \otimes (!B, \delta) \xrightarrow{f} (!B, \delta)$$

in $Free(\mathcal{C}^!)_{(!A,\delta)\times(-)}$, thus, to give a fixpoint operator on $Free(\mathcal{C}^!)_{(!A,\delta)\times(-)}$ is equivalent to give a fixpoint operator on $\mathcal{C}^!_{SA*}$ restricted to free coalgebras.

Proof. Note that we consider $\mathcal{C}^!_{SA*}$ together with the terminal object I and note that $Free(\mathcal{C}^!)$ has finite products $(\times, 1)$, Proposition 2.5, which makes 1 a terminal object in $Free(\mathcal{C}^!)_{(!A,\delta)\times(-)}$. The following calculation proves the result:

$n^{-1}; (Id \otimes n_1^{-1}); h$ is a fixpoint of $n^{-1}; f$ in $Free(\mathcal{C}^!)_{(!A,\delta)\times(-)}$ iff
$n^{-1}; (Id \otimes n_1^{-1}); h =< \pi_1, Id >; (Id \times (n^{-1}; (Id \otimes n_1^{-1}); h)); n^{-1}; f$ iff
$h = (Id \otimes n_1); n; < \pi_1, Id >; (Id \times (n^{-1}; (Id \otimes n_1^{-1}); h)); n^{-1}; f$ iff $\quad *$
$h =< \pi_1, Id >; (Id \otimes h); f$
h is a fixpoint of f in $\mathcal{C}^!_{SA}$

* We obtain the map $< g, k >$ in $\mathcal{C}^!$ by composing the map $< g, k >$ in $Free(\mathcal{C}^!)$ with n^{-1}, and similarly, we obtain the projection map π_1 in $\mathcal{C}^!$ by composing the projection map π_1 in $Free(\mathcal{C}^!)$ with n. $\square$

Theorem 2.22 *Let $\mathcal{C}$ be a !-category such that $Free(\mathcal{C}^!)$ is closed under finite products. There is a bijective correspondence between (natural) parametrised fixpoint operators on $Free(\mathcal{C}^!)$ and (natural) parametrised linear fixpoint operators on $\mathcal{C}$.*

Proof. Note that $Free(\mathcal{C}^!)$ has finite products $(\times, 1)$, Proposition 2.5. To give a parametrised fixpoint operator on $Free(\mathcal{C}^!)$ is equivalent to give a fixpoint operator on each $Free(\mathcal{C}^!)_{(!A,\delta)\times(-)}$, Proposition 2.14, which is equivalent to give a fixpoint operator on each $\mathcal{C}^!_{SA*}$ restricted to free coalgebras, Lemma 2.21, which is equivalent to give a linear fixpoint operator on each category $\mathcal{C}_{SA}$, Lemma 2.20, which is equivalent to give a parametrised linear fixpoint operator on $\mathcal{C}$, Proposition 2.19. It can be shown that a parametrised fixpoint operator on $Free(\mathcal{C}^!)$ is natural iff the corresponding parametrised linear fixpoint operators on $\mathcal{C}$ is natural. $\square$

We can deal with internal linear fixpoint operators if our category is closed with respect to the monoidal structure:

Definition 2.23 *Let $\mathcal{C}$ be a linear category. An* internal linear fixpoint operator *is a family of maps $Y_B :!(!B \multimap B) \to B$ such that $\gamma(\lambda(f)); Y_B$ is a parametrised linear fixpoint of f for any map $f :!A \otimes !B \to B$.*

We get back Definition 2.15 if we equip a cartesian closed category with the identity comonad and the natural transformation induced by the diagonal maps. One can show that a linear category has a parametrised linear fixpoint operator iff it has an internal linear fixpoint operator, which would be a generalisation of the result saying that a cartesian closed category has a parametrised fixpoint operator iff it has an internal fixpoint operator. We will here show a more informative result, which is a generalisation of Proposition 2.16:

Proposition 2.24 *There is a bijective correspondence between natural parametrised linear fixpoint operators and internal linear fixpoint operators in a linear category.*

Proof. Assume that $\mathcal{C}$ has a natural parametrised linear fixpoint operator $(-)^\sharp$, and define a map $Y = ((\varepsilon \otimes Id); eval)^\sharp$ for every object. Let a map $f :!A \otimes !B \to B$ be given, then

$$\begin{array}{rcll} \gamma(\lambda(f)); Y & = & \gamma(\lambda(f)); ((\varepsilon \otimes Id); eval)^\sharp & \text{def. of } Y \\ & = & ((\gamma(\lambda(f)) \otimes Id); (\varepsilon \otimes Id); eval)^\sharp & \\ & = & ((\lambda(f) \otimes Id); eval)^\sharp & \\ & = & f^\sharp & \end{array}$$

which entails that $\gamma(\lambda(f)); Y$ is a parametrised linear fixpoint of f. The construction of internal linear fixpoint operators from natural parametrised linear fixpoint operators is clearly injective. Conversely, assume that Y is an internal linear fixpoint operator on $\mathcal{C}$, and define the operation on maps $\gamma(\lambda(-)); Y$. This is a natural parametrised linear fixpoint operator; the map $\gamma(\lambda(f)); Y$ is a parametrised linear fixpoint of f for any map $f :!A \otimes !B \to B$, and given a map $h :!A' \to A$, we have

$$\begin{array}{rcl} \gamma(h); \gamma(\lambda(f)); Y & = & \gamma(\gamma(h); \lambda(f)); Y \\ & = & \gamma(\lambda((\gamma(h) \otimes Id); f)); Y \end{array}$$

so the operation is natural in $!A$ with respect to maps in the image of γ. The construction of natural parametrised linear fixpoint operators from internal linear fixpoint operators is injective because the following calculation

$$\begin{array}{rcl} \gamma(\lambda((\varepsilon \otimes Id); eval)); Y & = & \gamma(\lambda((\varepsilon \otimes Id); \lambda^{-1}(Id))); Y \\ & = & \gamma(\lambda(\lambda^{-1}(\varepsilon))); Y \\ & = & \gamma(\varepsilon); Y \\ & = & Id; Y \\ & = & Y \end{array}$$

shows that the internal linear fixpoint operator induced by $\gamma(\lambda(-)); Y$ is equal to Y. $\Box$

Theorem 2.25 *Let $\mathcal{C}$ be a linear category such that $Free(\mathcal{C}^!)$ is closed under finite products. There is a bijective correspondence between internal fixpoint operators in $Free(\mathcal{C}^!)$ and internal linear fixpoint operators in $\mathcal{C}$.*

Proof. Follows from Proposition 2.16, Theorem 2.22 and Proposition 2.24. $\Box$

This result can also be derived from the following theorem, which is a consequence of the fact that the definition of an internal linear fixpoint operator in $\mathcal{C}$ is essentially the definition of an internal fixpoint operator in $Free(\mathcal{C}^!)$ stated in terms of maps in $\mathcal{C}$:

Theorem 2.26 *Let $\mathcal{C}$ be a linear category such that $Free(\mathcal{C}^!)$ is closed under finite products. A family of maps $Y : (!B, \delta) \Rightarrow (!B, \delta) \to (!B, \delta)$ is an internal fixpoint operator in $Free(\mathcal{C}^!)$ iff the family of maps $\phi(Y) :!(!B \multimap B) \to B$ is an internal linear fixpoint operator in $\mathcal{C}$.*

Proof. Note that $Free(\mathcal{C}^!)$ has finite products $(\times, 1)$, Proposition 2.5. We will use the observation that we have a bijection between $\mathcal{C}(!A \otimes !B, B)$ and $Free(\mathcal{C}^!)((!A, \delta) \times (!B, \delta), (!B, \delta))$, namely the operation $n^{-1}; \phi^{-1}(-)$. We will also use the result that a map $f : (!A, \delta) \otimes (!B, \delta) \to (!B, \delta)$ has the same parametrised fixpoints in $\mathcal{C}^!$ as the composite

$$(!A, \delta) \times (!B, \delta) \xrightarrow{n^{-1}} (!A, \delta) \otimes (!B, \delta) \xrightarrow{f} (!B, \delta)$$

has in $Free(\mathcal{C}^!)$, Lemma 2.21. Now, Y is an internal fixpoint operator in $Free(\mathcal{C}^!)$ iff for any map $f :!A \otimes !B \to B$ we have that $\lambda(n^{-1}; \phi^{-1}(f)); Y$ is a parametrised fixpoint of $\phi^{-1}(f)$. But we have

$\lambda(n^{-1}; \phi^{-1}(f)); Y$ is a par. fixpoint of $\phi^{-1}(f)$ iff	def. of λ in $Free(\mathcal{C}^!)$
$\phi^{-1}(\lambda(\phi(n; n^{-1}; \phi^{-1}(f)))); Y$ is a par. fixpoint of $\phi^{-1}(f)$ iff	
$\phi^{-1}(\lambda(f)); Y$ is a par. fixpoint of $\phi^{-1}(f)$ iff	*
$\phi(\phi^{-1}(\lambda(f)); Y)$ is a par. linear fixpoint of f iff	
$\gamma(\lambda(f)); \phi(Y)$ is a par. linear fixpoint of f iff	

* Proposition 2.14, Lemma 2.20, and Proposition 2.19. Thus, Y is an internal fixpoint operator in $Free(\mathcal{C}^!)$ iff for any map $f :!A \otimes !B \to B$ we have that $\gamma(\lambda(f)); \phi(Y)$ is a parametrised linear fixpoint of f, which is the definition of an internal linear fixpoint operator applied to $\phi(Y)$. $\Box$

2.5 Generalisation of Linear Fixpoints

The definition of linear fixpoints can be generalised to an arbitrary number of parameters. We first need a generalisation of the natural transformation d:

Definition 2.27 *Let $\mathcal{C}$ be a !-category. We define $D_{A_1,\ldots,A_n}$ to be the composite*

$$!A_1 \otimes \ldots \otimes !A_n \xrightarrow{d_{A_1} \otimes \ldots \otimes d_{A_n}} !A_1 \otimes !A_1 \otimes \ldots \otimes !A_n \otimes !A_n \xrightarrow{\cong} !A_1 \otimes \ldots \otimes !A_n \otimes !A_1 \otimes \ldots \otimes !A_n$$

Proposition 2.28 *Let $\mathcal{C}$ be a !-category. The underlying map of the diagonal map*

$$(!A_1, \delta) \otimes \ldots \otimes (!A_n, \delta) \xrightarrow{\Delta} (!A_1, \delta) \otimes \ldots \otimes (!A_n, \delta) \otimes (!A_1, \delta) \otimes \ldots \otimes (!A_n, \delta)$$

in $\mathcal{C}^!$ is equal to D.

Proof. The following calculation proves the result:

$$\begin{array}{rcll}
\Delta & = & (\otimes_i \delta); m; d; (\varepsilon \otimes \varepsilon) & \text{def. of } \Delta \text{ in } \mathcal{C}^! \\
& = & (\otimes_i \delta); (\otimes_i d); \cong; (m \otimes m); (\varepsilon \otimes \varepsilon) & d \text{ is monoidal} \\
& = & (\otimes_i \delta); (\otimes_i d); \cong; ((\otimes_i \varepsilon) \otimes (\otimes_i \varepsilon)) & \varepsilon \text{ is monoidal} \\
& = & (\otimes_i \delta); (\otimes_i d); (\otimes_i (\varepsilon \otimes \varepsilon)); \cong & \delta \text{ is a map of comonoids} \\
& = & (\otimes_i d); (\otimes_i (\delta \otimes \delta)); (\otimes_i (\varepsilon \otimes \varepsilon)); \cong & \\
& = & (\otimes_i d); \cong & \\
& = & D & \text{def. of } D
\end{array}$$

□

Definition 2.29 *(Generalisation) Let $\mathcal{C}$ be a !-category. A map $f^\sharp : !A_1 \otimes \ldots \otimes !A_n \to B$ is a* parametrised linear fixpoint *of the map $f : !A_1 \otimes \ldots \otimes !A_n \otimes !B \to B$ iff $f^\sharp$ is equal to the composite*

$$!A_1 \otimes \ldots \otimes !A_n \xrightarrow{D} !A_1 \otimes \ldots \otimes !A_n \otimes !A_1 \otimes \ldots \otimes !A_n \xrightarrow{Id \otimes \gamma(f^\sharp)} !A_1 \otimes \ldots \otimes !A_n \otimes !B \xrightarrow{f} B$$

A parametrised linear fixpoint operator *is an operation on maps $(-)^\sharp$ such that $f^\sharp$ is a parametrised linear fixpoint of f for any map $f : !A_1 \otimes \ldots \otimes !A_n \otimes !B \to B$. A parametrised linear fixpoint operator is* natural *if the operation is natural in $!A_1 \otimes \ldots \otimes !A_n$ with respect to maps of coalgebras.*

We get back the original definition, Definition 2.18, with the restriction that $n = 1$. The generalisation can be shown to be equivalent to the original definition in a !-category where $Free(\mathcal{C}^!)$ is closed under finite products.

Definition 2.30 *(Generalisation) Let $\mathcal{C}$ be a linear category. An* internal linear fixpoint operator *is a family of maps $Y_B : !(!B \multimap B) \to B$ such that $\gamma(\lambda(f)); Y_B$ is a parametrised linear fixpoint of f for any map $f : !A_1 \otimes \ldots \otimes !A_n \otimes !B \to B$.*

We get back the original definition, Definition 2.23, with the restriction that $n = 1$. This generalisation can be shown to be equivalent to the original definition in a linear category where $Free(\mathcal{C}^!)$ is closed under finite products.

2.6 Concrete Models

An example of a linear category with finite products and finite sums is the category of CPOs and strict continuous functions; the symmetric monoidal structure is given by the smash product, the internal-hom is given by the strict continuous function space, and the comonad is given by the lift operation. This category has an internal linear fixpoint operator because the induced category of free coalgebras is equivalent to the category of CPOs and continuous functions; a cartesian closed category with an internal fixpoint operator. The internal fixpoint operator

$$B \Rightarrow B \xrightarrow{Y} B$$

in the category of CPOs and continuous functions is defined as $Y(f) = \sqcup_{n\in\omega} f^n(\perp)$, which entails that the induced internal linear fixpoint operator

$$!(!B \multimap B) \xrightarrow{Y} B$$

in the category of CPOs and strict continuous functions works as follows: a map $f \in !(!B \multimap B)$ is either equal to $\perp$ or corresponds to a map in $!B \multimap B$. If $f = \perp$, then $Y(f) = \perp$. If f corresponds to a map in $!B \multimap B$, then this map corresponds to a map $f' \in B \Rightarrow B$ and $Y(f) = Y(f') = \sqcup_{n\in\omega} f'^n(\perp)$. It should be noted that the category of CPOs and strict continuous functions is a model for Intuitionistic Relevant Logic in the sense of [Jac], that is, there is a natural transformation $d : (-) \to (-) \otimes (-)$ making appropriate diagrams commute.

Another example of a linear category with finite products and finite sums is the category of dI domains and linear stable functions. This category has an internal linear fixpoint operator because the induced category of free coalgebras is equivalent to the category of dI domains and stable functions; a cartesian closed category with an internal fixpoint operator.

In [Bra94b] a third example of a linear category with finite products is given; namely the category of dI domains and non-empty-join preserving stable functions. This category has an internal linear fixpoint operator because the induced category of free coalgebras is equivalent to the category of dI domains and stable functions; a cartesian closed category with an internal fixpoint operator. The category of dI domains and non-empty-join preserving stable functions does not have finite sums, but if we extend it to the category of bottomless dI domains and non-empty-join preserving stable functions, then we obtain a linear category with finite products and finite sums, but lose the internal linear fixpoint operator. This is so because the induced category of free coalgebras is equivalent to the category of bottomless dI domains and stable functions; a cartesian closed category with finite sums, which is easily seen not to have an internal fixpoint operator. The reader is invited to figure out how a fixpoint of the twist map $[in_2, in_1] : 1+1 \to 1+1$ might look in this category. The result that a cartesian closed category with finite sums and a fixpoint operator is inconsistent, [HP90], also implies that the category of bottomless dI domains and non-empty-join preserving stable functions can not have a fixpoint operator. It should be noted that the category of dI domains and non-empty-join preserving stable functions as well as the category of bottomless dI domains and non-empty-join preserving stable functions are models of Intuitionistic Affine Logic in the sense of [Jac], that is, the units of the tensor products are terminal objects.

A fourth example of a linear category with finite products and finite sums is the category of coherence spaces and linear stable functions, [GLT89]. This category has an internal linear fixpoint operator because the induced category of free coalgebras is equivalent to the category of coherence spaces and stable functions; a cartesian closed category with an internal fixpoint operator.

3 The λ-Calculus

3.1 Definition

Types are given by the grammar $s \ ::= \ \mathrm{t} \mid s \wedge s \mid s \Rightarrow s$. The metavariables A, B, C, D will range over types. Terms are given by the grammar

$$
\begin{array}{rll}
t & ::= & x \mid \\
 & & <> \ \mid \ < t, t > \ \mid \mathrm{fst}(t) \mid \mathrm{snd}(t) \mid \\
 & & \lambda x^A.t \mid tt
\end{array}
$$

where x is a variable that ranges over terms. The metavariables f, u, v, w will range over terms. Rules for assignment of types to terms are given in Appendix A (which also contains the rules for assignment of categorical semantics). Type assignments have the form of sequents $x_1 : A_1, ..., x_n : A_n \vdash u : A$ where $x_1, ..., x_n$ are pairwise distinct variables. We will frequently write Γ instead of $x_1 : A_1, ..., x_n : A_n$ or $A_1, ..., A_n$. The notation will be abused when necessary; $\Gamma \vdash u : A$ can mean either the sequent itself or a certain derivation of the sequent, and the name of a rule, for example (Ax), can mean either the rule itself or a certain instance of the rule. The actual interpretation is to be decided by the context. From now on, we will consider only typable terms. The expression $v[u/x]$ denotes the term v where u has been substituted for every free occurence of x, and where bound variables of v have been renamed to avoid capture of free variables in u. The λ-calculus satisfies the following properties:

Lemma 3.1 *If $\Gamma \vdash u : A$ is derived by a certain derivation, then for any derivable $\Gamma \vdash u : B$ we have $A = B$.*

Proof. Induction in the derivation of $\Gamma \vdash u : A$. $\square$

Proposition 3.2 *If* $\Gamma \vdash u : A$ *is derivable, then the rule instance above the sequent is uniquely determined.*

Proof. Use Lemma 3.1 to check each case. $\Box$

Lemma 3.3 *If* $\Gamma_1, \Gamma_2 \vdash u : A$ *is derived by a certain derivation and the variables in* Γ_1, Γ_2 *and* Λ *are pairwise distinct, then* $\Gamma_1, \Lambda, \Gamma_2 \vdash u : A$ *is derivable too.*

Proof. Induction in the derivation of $\Gamma_1, \Gamma_2 \vdash u : A$. $\Box$

Lemma 3.4 *(Substitution Property) If* $\Gamma \vdash u : A$ *and* $\Gamma, x : A, \Lambda \vdash v : B$ *are derived by certain derivations, then* $\Gamma, \Lambda \vdash v[u/x] : B$ *is derivable too.*

Proof. Induction in the derivation of $\Gamma, x : A, \Lambda \vdash v : B$. We need Lemma 3.3 to the (Ax) case. $\Box$

3.2 The Curry-Howard Isomorphism

The Curry-Howard isomorphism says that formulas corresponds to types, proofs to terms, and normalisation of proofs to reduction of terms. In what follows, we need a natural deduction formulation of IL as given in Appendix B. The relation between formulas of IL and types of the λ-calculus is obvious. The idea of the Curry-Howard isomorphism on the level of proofs is that proof-rules can be "decorated" with terms such that the term induced by a proof encodes the proof. An appropriate term language for this purpose is in the case of IL the λ-calculus. It turns out that we get the rules for assigning types to terms in the λ-calculus if we decorate the proof-rules of IL with terms in an appropriate way. We can obviously recover the proof-rules if we take the typing rules of the λ-calculus and remove the terms. We get the Curry-Howard isomorphism on the level of proofs as follows: given a proof of $A_1, ..., A_n \vdash A$, that is, a proof of the formula A, one can inductively construct a derivation of a sequent $x_1 : A_1, ..., x_n : A_n \vdash u : A$, that is, a term u of type A. Conversely, if one has a derivable sequent $x_1 : A_1, ..., x_n : A_n \vdash u : A$, there is an easy way to get a proof of $A_1, ..., A_n \vdash A$; erase all terms in the derivation of the type assignment. The two processes are each other's inverses modulo renaming of variables. The isomorphism on the level of proofs is essentially given by Proposition 3.2. The Curry-Howard isomorphism on the level of normalisation says that a normalisation step followed by application of the Curry-Howard isomorphism on the level of proofs, yields the same term as application of the Curry-Howard isomorphism on the level of proofs followed by the reduction step corresponding to the normalisation step. This is dealt with in details in [GLT89].

3.3 Categorical Semantics

Given a cartesian closed category, we define a categorical interpretation of the λ-calculus; types are interpreted as objects, typing rules as natural operations on arrows, and derivations of type assignments as maps. A derivable sequent

$$x_1 : A_1, ..., x_n : A_n \vdash u : B$$

is interpreted as a map

$$[\![A_1]\!] \times ... \times [\![A_n]\!] \xrightarrow{[\![x_1:A_1,...,x_n:A_n \vdash u:B]\!]} [\![B]\!]$$

by induction in its derivation cf. the operations on arrows (corresponding to the typing rules) given in Appendix A. Note that the derivation of the sequent is uniquely determined, Proposition 3.2, so it makes sense to speak of the interpretation of a derivable sequent without mentioning its derivation explicitly. It can be shown that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the sequents. In what follows, we will omit the $[\![-]\!]$ brackets when appropriate.

Lemma 3.5 *If $\Gamma_1, \Gamma_2 \vdash u : A$ is derived by a certain derivation and the variables in Γ_1, Γ_2 and Λ are pairwise distinct, then $\Gamma_1, \Lambda, \Gamma_2 \vdash u : A$ is derivable too with the interpretation*

$$\Gamma_1 \times \Lambda \times \Gamma_2 \xrightarrow{\pi_1 \times \Gamma_2} \Gamma_1 \times \Gamma_2 \xrightarrow{[\![\Gamma_1,\Gamma_2 \vdash u:A]\!]} A$$

Proof. Induction in the derivation of $\Gamma_1, \Gamma_2 \vdash u : A$. All cases except (Ax) are covered by the observation that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the left hand sides of the sequents. We need direct calculation to the (Ax) case. $\Box$

The following extension of Lemma 3.4 essentially says how substitution relates to composition:

Lemma 3.6 *If $\Gamma \vdash u : A$ and $\Gamma, x : A, \Lambda \vdash v : B$ are derived by certain derivations, then $\Gamma, \Lambda \vdash v[u/x] : B$ is derivable too with the interpretation*

$$\Gamma \times \Lambda \xrightarrow{\Delta \times \Lambda} \Gamma \times \Gamma \times \Lambda \xrightarrow{\Gamma \times [\![\Gamma \vdash u:A]\!] \times \Lambda} \Gamma \times A \times \Lambda \xrightarrow{[\![\Gamma,x:A,\Lambda \vdash v:B]\!]} B$$

Proof. Induction in the derivation of $\Gamma, x : A, \Lambda \vdash v : B$. All cases except (Ax) are covered by the observation that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the left hand sides of the sequents. We need Lemma 3.5 to the (Ax) case. $\Box$

It can be shown by using Lemma 3.6 that the interpretation is sound with respect to the usual reduction rules for terms of the λ-calculus, that is, interpretation is preserved by application of reduction rules.

4 The Linear λ-Calculus

4.1 Definition

Types are given by the grammar $s \ ::= \ I \mid s \otimes s \mid \top \mid s \& s \mid s \multimap s \mid\ !s$ and terms by the grammar

$$
\begin{array}{lll}
t & ::= & x \mid \\
 & & * \mid \text{let } t \text{ be } * \text{ in } t \mid t \otimes t \mid \text{let } t \text{ be } x \otimes y \text{ in } t \mid \\
 & & <> \mid \ < t, t > \ \mid \text{fst}(t) \mid \text{snd}(t) \mid \\
 & & \lambda x^A . t \mid tt \mid \\
 & & \text{let } t, ..., t \text{ be } x_1, ..., x_n \text{ in } !t \mid \text{derelict}(t) \mid \text{discard } t \text{ in } t \mid \text{copy } t \text{ as } x, y \text{ in } t
\end{array}
$$

where x is a variable that ranges over terms, and $t, ..., t$ means a sequence of n occurrences of t. Rules for assignment of types to terms are given in Appendix C. Type assignments have the form of sequents $x_1 : A_1, ..., x_n : A_n \Vdash u : A$ where $x_1, ..., x_n$ are pairwise distinct variables. A Girardian turnstyle $\Vdash$ is used to distinguish sequents in the linear λ-calculus from sequents in the λ-calculus, where the usual turnstyle $\vdash$ is used. Note that the definition of sequents implicitly restricts use of the rules. It is for example not possible to use the $(\otimes_I)$ rule if Γ and Δ have common variables. The linear λ-calculus is essentially the same as the calculus given in [BBdPH92]. From now on, we will consider only typable terms. The expression $\overline{w}$ means $w_1, ..., w_n$, the expression

$$\text{copy } \overline{w} \text{ as } \overline{x}, \overline{y} \text{ in } u$$

mean

$$\text{copy } w_1 \text{ as } x_1, y_1 \text{ in } (...\text{copy } w_n \text{ as } x_n, y_n \text{ in } u...)$$

and the expression

$$\text{discard } \overline{w} \text{ in } u$$

mean

$$\text{discard } w_1 \text{ in } (...\text{discard } w_n \text{ in } u...)$$

respectively. The linear λ-calculus satisfies the following properties:

Lemma 4.1 *If $\Gamma \Vdash u : A$ is derived by a certain derivation, then for any derivable $\Gamma' \Vdash u : B$, where the assumptions in Γ' is a permutation of the assumptions in Γ, we have $A = B$.*

Proof. Induction in the derivation of $\Gamma \Vdash u : A$. $\square$

Proposition 4.2 *If $\Gamma \Vdash u : A$ is derivable, then the first rule instance above the sequent which is different from an instance of (Ex) is uniquely determined up to permutation of assumptions.*

Proof. Use Lemma 4.1 to check each case. □

Lemma 4.3 *(Substitution Property) If* $\Gamma \vdash u : A$ *and* $\Delta, x : A, \Lambda \vdash v : B$ *are derived by certain derivations and the variables in* Γ *and* Δ, Λ *are pairwise distinct, then* $\Delta, \Gamma, \Lambda \vdash v[u/x] : B$ *is derivable too.*

Proof. Induction in the derivation of $\Delta, x : A, \Lambda \vdash v : B$. □

4.2 The Curry-Howard Isomorphism

In what follows, we need a natural deduction formulation of ILL as given in Appendix D. ILL corresponds to the linear λ-calculus via a Curry-Howard isomorphism in the same way as IL corresponds to the λ-calculus; we get the rules for assigning types to terms in the linear λ-calculus if we decorate the proof-rules of ILL with terms in an appropriate way, and conversely, we recover the proof-rules if we remove the terms from the typing rules. The isomorphism on the level of proofs is essentially given by Proposition 4.2.

4.3 Categorical Semantics

Given a linear category with finite products, we define a categorical interpretation of the linear λ-calculus; a derivable sequent

$$x_1 : A_1, ..., x_n : A_n \vdash u : B$$

is interpreted as a map

$$[\![A_1]\!] \otimes ... \otimes [\![A_n]\!] \xrightarrow{[\![x_1 : A_1, ..., x_n : A_n \vdash u : B]\!]} [\![B]\!]$$

by induction in its derivation cf. the operations on arrows given in Appendix C. Note that the derivation of the sequent is uniquely determined up to permutation of assumptions, Proposition 4.2, so it makes sense to speak of the interpretation of a derivable sequent without mentioning its derivation explicitly. It can be shown that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the sequents. The interpretation of the linear λ-calculus is the same as the interpretation given in [BBdPH92]. The following extension of Lemma 4.3 does essentially say that substitution corresponds to composition:

Lemma 4.4 *If* $\Gamma \vdash u : A$ *and* $\Delta, x : A, \Lambda \vdash v : B$ *are derived by certain derivations and the variables in* Γ *and* Δ, Λ *are pairwise distinct, then* $\Delta, \Gamma, \Lambda \vdash v[u/x] : B$ *is derivable too with the interpretation*

$$\Delta \otimes \Gamma \otimes \Lambda \xrightarrow{\Delta \otimes [\![\Gamma \vdash u:A]\!] \otimes \Lambda} \Delta \otimes A \otimes \Lambda \xrightarrow{[\![\Delta, x:A, \Lambda \vdash v:B]\!]} B$$

Proof. Induction in the derivation of $\Delta, x : A, \Lambda \vdash v : B$. All cases except (Ax) are covered by the observation that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the left hand sides of the sequents. The (Ax) case is trivial. $\square$

It can be shown by using Lemma 4.4 that the interpretation is sound with respect to the usual reduction rules for terms of the linear λ-calculus, [BBdPH92].

5 The Girard Translation

5.1 Definition

The [Gir87] paper introduced the Girard Translation which embeds IL into ILL. We will state the Girard Translation in terms of the natural deduction formulations of IL and ILL given in Appendix B and Appendix D. At the level of formulas the Girard Translation is defined inductively as follows:

- $\mathrm{t}^o = \top$
- $(A \wedge B)^o = A^o \& B^o$
- $(A \Rightarrow B)^o = !A^o \multimap B^o$

At the level of proofs the Girard Translation translates a proof of $A_1, ..., A_n \vdash B$ into a proof of $!A_1^o, ..., !A_n^o \vdash B^o$. The Girard Translation at the level of proofs is stated in Appendix E. The translation is sound with respect to provability in the sense that $A_1, ..., A_n \vdash B$ is provable (in IL) iff $!A_1^o, ..., !A_n^o \vdash B^o$ is provable (in ILL), [Gir87], and moreover, the translation preserves β reductions, [Bie94]. The translation induces a translation from types and derivable sequents in the λ-calculus to types and derivable sequents in the linear λ-calculus cf. the Curry-Howard isomorphisms; a sequent $x_1 : A_1, ..., x_n : A_n \vdash u : A$ is translated into a sequent $(x_1 : A_1, ..., x_n : A_n \vdash u : A)^o$ of the shape $x_1 :!A_1^o, ..., x_n :!A_n^o \vdash u' : A^o$ where u' encodes the translation of the proof encoded by u. An inductive definition of the translation from derivable sequents in the λ-calculus to derivable sequents in the linear λ-calculus can be obtained simply by decorating the rules in the definition of the extended Girard Translation (in Appendix E) appropriately with terms.

5.2 Soundness with Respect to Categorical Semantics

If $\mathcal{C}$ is a linear category with finite products, then $Free(\mathcal{C}^!)$ is a cartesian closed category, as previous results show. We can therefore interpret types and derivable sequents in the

λ-calculus as objects and arrows in $Free(\mathcal{C}^!)$. It turns out that the interpretation of a type can be written in a simple way using the Girard Translation at the level of types:

Proposition 5.1 *Let $\mathcal{C}$ be a linear category with finite products. If a type A of the λ-calculus is interpreted in $Free(\mathcal{C}^!)$, and the type A^o of the linear λ-calculus is interpreted in $\mathcal{C}$, then $[\![A]\!] = (![\![A^o]\!], \delta)$.*

Proof. Induction in A, we proceed case by case.

$$\begin{array}{rcll}
[\![\mathrm{t}]\!] & = & (!1, \delta) & \\
 & = & (![\![\top]\!], \delta) & \\
 & = & (![\![\mathrm{t}^o]\!], \delta) & \\
[\![B \wedge C]\!] & = & [\![B]\!] \times [\![C]\!] & \\
 & = & (![\![B^o]\!], \delta) \times (![\![C^o]\!], \delta) & \text{IH} \\
 & = & (!([\![B^o]\!] \times [\![C^o]\!]), \delta) & \\
 & = & (![\![B^o \& C^o]\!], \delta) & \\
 & = & (![\![(B \wedge C)^o]\!], \delta) & \\
[\![B \Rightarrow C]\!] & = & [\![B]\!] \Rightarrow [\![C]\!] & \\
 & = & (![\![B^o]\!], \delta) \Rightarrow (![\![C^o]\!], \delta) & \text{IH} \\
 & = & (!(![\![B^o]\!] \multimap [\![C^o]\!]), \delta) & \\
 & = & (![\![!B^o \multimap C^o]\!], \delta) & \\
 & = & (![\![(B \Rightarrow C)^o]\!], \delta) &
\end{array}$$

$\square$

Let the following composite of bijections be denoted by lin:

$$\begin{array}{rcll}
\mathcal{C}^!([\![A_1]\!] \times ... \times [\![A_n]\!], [\![B]\!]) & = & \mathcal{C}^!((![\![A_1^o]\!], \delta) \times ... \times (![\![A_n^o]\!], \delta), (![\![B^o]\!], \delta)) & \text{Prop. 5.1} \\
 & \cong & \mathcal{C}^!((![\![A_1^o]\!], \delta) \otimes ... \otimes (![\![A_n^o]\!], \delta), (![\![B^o]\!], \delta)) & \text{comp. with } n \\
 & = & \mathcal{C}^!((![\![A_1^o]\!], \delta) \otimes ... \otimes (![\![A_n^o]\!], \delta), F^![\![B^o]\!]) & \\
 & \cong & \mathcal{C}(U^!((![\![A_1^o]\!], \delta) \otimes ... \otimes (![\![A_n^o]\!], \delta)), [\![B^o]\!]) & U^! \dashv F^! \\
 & = & \mathcal{C}(![\![A_1^o]\!] \otimes ... \otimes ![\![A_n^o]\!], [\![B^o]\!]) &
\end{array}$$

We are now ready to state a result showing that the Girard Translation is sound with respect to the above mentioned categorical interpretations. The result essentially says that the Girard Translation corresponds to the adjunction between $Free(\mathcal{C}^!)$ and $\mathcal{C}$, or to be precise, to the function lin. Recall that $(x_1 : A_1, ..., x_n : A_n \vdash u : B)^o$ is a derivable sequent in the linear λ-calculus of the shape $x_1 :!A_1^o, ..., x_n :!A_n^o \vdash u' : B^o$.

Theorem 5.2 *(Soundness) Let $\mathcal{C}$ be a linear category with finite products. If*

$$x_1 : A_1, ..., x_n : A_n \vdash u : B$$

is a derivable sequent in the λ-calculus, then

$$lin([\![x_1 : A_1, ..., x_n : A_n \vdash u : B]\!]) = [\![(x_1 : A_1, ..., x_n : A_n \vdash u : B)^o]\!]$$

Proof. Induction in the derivation of $x_1 : A_1, ..., x_n : A_n \vdash u : B$, we proceed case by case. In what follows we will disregard terms, and only consider the underlying proofs.

The (Ax) case. The proof

$$\frac{}{A_1, ..., A_n \vdash A_q}\,(Ax)$$

is translated into the proof

$$\dfrac{\dfrac{\dfrac{}{!A_q^o \vdash !A_q^o}\,(Ax)}{!A_q^o \vdash A_q^o}\,(Der)}{!A_1^o, ..., !A_n^o \vdash A_q^o}\,(Weak) \text{ and } (Ex)$$

The following calculation shows the wanted result:

$$\begin{array}{rcll} lin([\![A_1, ..., A_n \vdash A_q]\!]) & = & n; \pi_q; \varepsilon & \\ & = & \pi_q; \varepsilon & * \\ & = & (e \otimes ... \otimes e \otimes !A_q^o \otimes e \otimes ... \otimes e); \cong; \varepsilon & \text{def. of } \pi_q \text{ in } \mathcal{C}^! \\ & = & [\![!A_1^o, ..., !A_n^o \vdash A_q^o]\!] & \end{array}$$

* We obtain the projection map π_q in $\mathcal{C}^!$ by composing the projection map π_q in $Free(\mathcal{C}^!)$ with n.

The (t_I) case. The proof

$$\frac{}{\Gamma \vdash \mathrm{t}}\,(\mathrm{t}_I)$$

is translated into the proof

$$\frac{}{!\Gamma^o \vdash \top}\,(\top_I)$$

The following calculation shows the wanted result:

$$\begin{array}{rcll} lin([\![\Gamma \vdash \mathrm{t}]\!]) & = & n; <>; \varepsilon & \\ & = & n; \gamma(<>); \varepsilon & \text{def. of } <> \text{ in } Free(\mathcal{C}^!) \\ & = & <> & \\ & = & [\![!\Gamma^o \vdash \top]\!] & \end{array}$$

The $(\wedge_I)$ case. The proof

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,(\wedge_I)$$

is translated into the proof

$$\frac{!\Gamma^o \vdash A^o \quad !\Gamma^o \vdash B^o}{!\Gamma^o \vdash A^o \& B^o}\,(\&_I)$$

The following calculation shows the wanted result:

$$\begin{array}{rcll} lin([\![\Gamma \vdash A \wedge B]\!]) & = & n; < [\![\Gamma \vdash A]\!], [\![\Gamma \vdash B]\!] >; \varepsilon & \\ & = & n; \phi^{-1}(< \phi([\![\Gamma \vdash A]\!]), \phi([\![\Gamma \vdash B]\!]) >); \varepsilon & \text{def. of } < -, + > \\ & = & < (n; [\![\Gamma \vdash A]\!]; \varepsilon), (n; [\![\Gamma \vdash B]\!]; \varepsilon) > & \\ & = & < lin([\![\Gamma \vdash A]\!]), lin([\![\Gamma \vdash B]\!]) > & \\ & = & < [\![!\Gamma^o \vdash A^o]\!], [\![!\Gamma^o \vdash B^o]\!] > & \text{IH} \\ & = & [\![!A_1^o, ..., !A_n^o \vdash A^o \& B^o]\!] & \end{array}$$

The $(\wedge_{E1})$ case. The proof

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} (\wedge_{E1})$$

is translated into the proof

$$\frac{!\Gamma^o \vdash A^o \& B^o}{!\Gamma^o \vdash A^o} (\&_{E1})$$

The following calculation shows the wanted result:

$$\begin{array}{rcll} lin([\![\Gamma \vdash A]\!]) & = & n; [\![\Gamma \vdash A \wedge B]\!]; \pi_1; \varepsilon & \\ & = & n; [\![\Gamma \vdash A \wedge B]\!]; !\pi_1; \varepsilon & \text{def. of } \pi_1 \text{ in } Free(\mathcal{C}^!) \\ & = & n; [\![\Gamma \vdash A \wedge B]\!]; \varepsilon; \pi_1 & \\ & = & lin([\![\Gamma \vdash A \wedge B]\!]); \pi_1 & \\ & = & [\![!\Gamma^o \vdash A^o \& B^o]\!]; \pi_1 & \text{IH} \\ & = & [\![!\Gamma^o \vdash A^o]\!] & \end{array}$$

The $(\wedge_{E2})$ case is analogous to the $(\wedge_{E1})$ case.

The $(\Rightarrow_I)$ case. The proof

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

is translated into the proof

$$\frac{!\Gamma^o, !A^o \vdash B^o}{!\Gamma^o \vdash !A^o \multimap B^o} (\multimap_I)$$

The following calculation shows the wanted result:

$$\begin{array}{rcll} lin([\![\Gamma \vdash A \Rightarrow B]\!]) & = & n; \lambda([\![\Gamma, A \vdash B]\!]); \varepsilon & \\ & = & n; \phi^{-1}(\lambda(\phi(n; [\![\Gamma, A \vdash B]\!]))); \varepsilon & \text{def. of } \lambda \text{ in } Free(\mathcal{C}^!) \\ & = & n; \lambda(n; [\![\Gamma, A \vdash B]\!]; \varepsilon) & \\ & = & \lambda((n \otimes Id); n; [\![\Gamma, A \vdash B]\!]; \varepsilon) & \\ & = & \lambda(n; [\![\Gamma, A \vdash B]\!]; \varepsilon) & * \\ & = & \lambda(lin([\![\Gamma, A \vdash B]\!])) & \\ & = & \lambda([\![!\Gamma^o, !A^o \vdash B^o]\!]) & \text{IH} \\ & = & [\![!\Gamma^o \vdash !A^o \multimap B^o]\!] & \end{array}$$

* Because n makes ! a monoidal functor from $(\mathcal{C}, 1, \times)$ to $(\mathcal{C}, I, \otimes)$.

The $(\Rightarrow_E)$ case. The derivation

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

is translated into the proof

$$\dfrac{\dfrac{!\Gamma^o \vdash !A^o \multimap B^o \qquad \dfrac{!\Gamma^o \vdash A^o}{!\Gamma^o \vdash !A^o}\,(!_I)}{!\Gamma^o, !\Gamma^o \vdash B^o}\,(\multimap_E)}{!\Gamma^o \vdash B^o}\,(Con) \text{ and } (Ex)$$

The following calculation shows the wanted result:

$$\begin{array}{rcll}
lin([\![\Gamma \vdash B]\!]) & = & n; < [\![\Gamma \vdash A \Rightarrow B]\!], [\![\Gamma \vdash A]\!] >; eval; \varepsilon & \\
& = & n; < [\![\Gamma \vdash A \Rightarrow B]\!], [\![\Gamma \vdash A]\!] >; n^{-1}; \phi^{-1}(\lambda^{-1}(\phi(Id))); \varepsilon & \text{def. of } eval \\
& = & < (n; [\![\Gamma \vdash A \Rightarrow B]\!]), (n; [\![\Gamma \vdash A]\!]) >; n^{-1}; (\varepsilon \otimes Id); eval & \\
& = & < (n; [\![\Gamma \vdash A \Rightarrow B]\!]), (n; [\![\Gamma \vdash A]\!]) >; (\varepsilon \otimes Id); eval & * \\
& = & D; ((n; [\![\Gamma \vdash A \Rightarrow B]\!]) \otimes (n; [\![\Gamma \vdash A]\!])); (\varepsilon \otimes Id); eval & ** \\
& = & D; ((n; [\![\Gamma \vdash A \Rightarrow B]\!]; \varepsilon) \otimes \gamma(n; [\![\Gamma \vdash A]\!]; \varepsilon)); eval & \\
& = & D; (lin([\![\Gamma \vdash A \Rightarrow B]\!]) \otimes \gamma(lin([\![\Gamma \vdash A]\!]))); eval & \\
& = & D; ([\![!\Gamma^o \vdash !A^o \multimap B^o]\!] \otimes \gamma([\![!\Gamma^o \vdash A^o]\!])); eval & \text{IH} \\
& = & [\![!\Gamma^o \vdash B^o]\!] &
\end{array}$$

* We obtain a map $< f, g >$ in $\mathcal{C}^!$ by composing the map $< f, g >$ in $Free(\mathcal{C}^!)$ with n^{-1}.
** Definition of $< -, + >$ in $\mathcal{C}^!$, Definition 2.27 and Proposition 2.28. □

Note that if we consider proofs instead of derivable sequents, then we get a soundness result in the usual proof-theoretic sense. Recall that $(A_1, ..., A_n \vdash B)^o$ is a proof of $!A_1^o, ..., !A_n^o \vdash B^o$ in ILL.

Corollary 5.3 *Let $\mathcal{C}$ be a linear category with finite products. If $A_1, ..., A_n \vdash B$ is a proof in IL, then*

$$lin([\![A_1, ..., A_n \vdash B]\!]) = [\![(A_1, ..., A_n \vdash B)^o]\!]$$

Proof. Use Theorem 5.2 and the Curry-Howard isomorphism. □

This Corollary is a categorical generalisation of a result in [Gir87] showing that the Girard Translation is sound with respect to interpretation in a concrete category, namely the category of coherence spaces and linear stable functions.

6 Extensions with Finite Sums

6.1 The λ-Calculus Extended with Finite Sums

The λ-calculus can be extended with finite sums, that is, we have additional types f $\mid s \vee s$, additional terms

$$[]^{C}(t) \mid \mathrm{inl}^{A \vee B}(t) \mid \mathrm{inr}^{A \vee B}(t) \mid \text{case } t \text{ of } \mathrm{inl}(x) => t \mid \mathrm{inr}(y) => t$$

and typing rules as given in Appendix A. We still have Lemma 3.1, Proposition 3.2, Lemma 3.3, Lemma 3.4, and the Curry-Howard isomorphism.

We could give a categorical interpretation using a cartesian closed category with finite sums such that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the sequents, such that Lemma 3.6 still holds, and such that the interpretation is sound with respect to the usual reduction rules, [GLT89]. The $(\vee_E)$ rule should then be interpreted using a natural isomorphism $\Gamma \times (A + B) \cong (\Gamma \times A) + (\Gamma \times B)$ coming from the observation that the functor $\Gamma \times (-)$ has a right adjoint, and thus preserves sums. But a cartesian closed category with finite sums together with a fixpoint operator, [HP90], so we will instead assume that we have a cartesian closed category $\mathcal{C}$ with weak finite sums $(0, +)$ such that

1. the operation $[-, +] : \mathcal{C}(A, \Gamma \Rightarrow C) \times \mathcal{C}(B, \Gamma \Rightarrow C) \to \mathcal{C}(A + B, \Gamma \Rightarrow C)$ is natural in Γ,
2. the map $[] : 0 \to 0$ induced by the weak initial object is equal to the identity, and the map $[in_1, in_2] : (!A, \delta) + (!B, \delta) \to (!A, \delta) + (!B, \delta)$ induced by the weak binary sum is equal to the identity.

We are then able to give a categorical interpretation, cf. Appendix A, such that the operations on arrows induced by the typing rules are natural in the interpretation of Γ, and such that Lemma 3.5 and Lemma 3.6 still hold. If the weak finite sums are finite sums in the usual sense, then the above mentioned conditions 1. and 2.are satisfied, and the interpretation will coincide with the one given by the isomorphism $\Gamma \times (A + B) \cong (\Gamma \times A) + (\Gamma \times B)$. It takes little calculation to see that the operation on arrows induced by the $(\vee_E)$ rule is natural in Γ. Let a map $h : \Delta \to \Gamma$ be given, then

$$\begin{aligned}
&h; < \Gamma, w >; (\Gamma \times [\lambda(u), \lambda(v)]); eval = \\
&< \Delta, (h; w) >; (\Delta \times [\lambda(u), \lambda(v)]); (h \times (\Gamma \Rightarrow C)); eval = \\
&< \Delta, (h; w) >; (\Delta \times [\lambda(u), \lambda(v)]); (\Delta \times (h \Rightarrow C)); eval = \\
&< \Delta, (h; w) >; (\Delta \times [(\lambda(u); (h \Rightarrow C)), (\lambda(v); (h \Rightarrow C))]); eval = \\
&< \Delta, (h; w) >; (\Delta \times [\lambda((h \times A); u), \lambda((h \times B); v)]); eval
\end{aligned}$$

This categorical interpretation can be shown to be sound with respect to β and η reductions, [GLT89]. It might be a surprise that we do not need finite sums to give a sound categorical interpretation with respect to η reductions, but the two statements in the above mentioned condition 2. corresponds case by case to the two η reductions

$$[]^C(w) \;\rightsquigarrow\; w \qquad\qquad \text{case } w \text{ of inl}(x) => \text{inl}(x) \mid \text{inr}(y) => \text{inr}(y) \;\rightsquigarrow\; w$$

The operation on arrows induced by the $(\vee_E)$ rule is not necessarily natural in C. This entails that the categorical interpretation will be sound only with respect to commuting conversions with the property that the operation on arrows induced by the $(\vee_E)$ rule is natural in C with respect to the interpretation of the typing rule corresponding to the conversion, [GLT89].

6.2 The linear λ-Calculus Extended with Finite Sums

The linear λ-calculus can be extended with finite sums, that is, we have additional types $0 \mid s \oplus s$, additional terms

$$[]^C(t) \mid \text{inl}^{A\oplus B}(t) \mid \text{inr}^{A\oplus B}(t) \mid \text{case } t \text{ of inl}(x) => t \mid \text{inr}(y) => t$$

and rules for assignment of types to terms as given in Appendix C. We still have Lemma 4.1, Proposition 4.2, Lemma 4.3, and the Curry-Howard isomorphism.

We can give a categorical interpretation using a linear category with finite products and finite sums $(0, +)$, cf. Appendix C, such that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the sequents, and such that Lemma 4.4 still holds. The natural isomorphism $\Gamma \otimes (A + B) \cong (\Gamma \otimes A) + (\Gamma \otimes B)$ comes from the observation that the functor $\Gamma \otimes (-)$ has a right adjoint, and thus preserves sums. The categorical interpretation is sound with respect to the usual reduction rules.

6.3 The Girard Translation Extended with Finite Sums

The Girard Translation still works if we extend it with finite sums. At the level of formulas, the extension is given by:

- $\mathrm{f}^o = 0$
- $(A \vee B)^o = !A^o \oplus !B^o$

At the level of proofs, the extended Girard Translation is stated in Appendix E. The translation is still sound with respect to provability, [Gir87], and it still preserves β reductions, [Bie94].

If $\mathcal{C}$ is a linear category with finite products and finite sums, then $Free(\mathcal{C}^!)$ is a cartesian closed category with weak finite sums such that the above mentioned conditions 1. and 2.

are satisfied, cf. Proposition 2.9, Proposition 2.7, and Proposition 2.10. We can therefore interpret types and derivable sequents in the λ-calculus extended with finite sums as objects and arrows in $Free(\mathcal{C}^!)$.

The categorical interpretation is sound with respect to β and η reductions, and it is sound with respect to commuting conversions with the property that the operation on arrows induced by the $(\vee_E)$ rule is natural in C with respect to the interpretation of the typing rule corresponding to the conversion, as mentioned above. It follows from Proposition 2.9 and Proposition 2.6 that the operation on arrows induced by the $(\vee_E)$ rule is natural in C with respect to linear maps, so the categorical interpretation is sound with respect to commuting conversions with the property that the interpretation of the typing rule corresponding to the conversion is linear.

We still have Proposition 5.1, as the following addition to the proof shows:

$$\begin{aligned} [\![\mathrm{f}]\!] &= (!0, \delta) \\ &= (![\![0]\!], \delta) \\ &= (![\![\mathrm{f}^o]\!], \delta) \end{aligned}$$

$$\begin{aligned} [\![B \vee C]\!] &= [\![B]\!] + [\![C]\!] \\ &= (![\![B^o]\!], \delta) + (![\![C^o]\!], \delta) && \text{IH} \\ &= (!(![\![B^o]\!] + ![\![C^o]\!]), \delta) \\ &= (![\![!B^o \oplus !C^o]\!], \delta) \\ &= (![\![(B \vee C)^o]\!], \delta) \end{aligned}$$

□

We also still have Theorem 5.2 (Soundness), as the following addition to the proof shows:

The (f_E) case. The proof

$$\frac{\Gamma \vdash \mathrm{f}}{\Gamma \vdash C}\,(\mathrm{f}_E)$$

is translated into the proof

$$\frac{!\Gamma^o \vdash 0}{!\Gamma^o \vdash C^o}\,(0_E)$$

The following calculation shows the wanted result:

$$\begin{aligned} lin([\![\Gamma \vdash C]\!]) &= n; [\![\Gamma \vdash \mathrm{f}]\!]; []; \varepsilon \\ &= n; [\![\Gamma \vdash \mathrm{f}]\!]; ![]; \varepsilon && \text{def. of } [] \text{ in } Free(\mathcal{C}^!) \\ &= n; [\![\Gamma \vdash \mathrm{f}]\!]; \varepsilon; [] \\ &= lin([\![\Gamma \vdash \mathrm{f}]\!]); [] \\ &= [\![!\Gamma^o \vdash 0]\!]; [] && \text{IH} \\ &= [\![!\Gamma^o \vdash C^o]\!] \end{aligned}$$

The $(\vee_{I1})$ case. The proof

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,(\vee_{I1})$$

is translated into the proof

$$\cfrac{\cfrac{!\Gamma^o \vdash A^o}{!\Gamma^o \vdash !A^o}\,(!_I)}{!\Gamma^o \vdash !A^o \oplus !B^o}\,(\oplus_{I1})$$

The following calculation shows the wanted result:

$$\begin{array}{rcll} lin([\![\Gamma \vdash A \vee B]\!]) & = & n; [\![\Gamma \vdash A]\!]; in_1; \varepsilon & \\ & = & n; [\![\Gamma \vdash A]\!]; \delta; !in_1; \varepsilon & \text{def. of } in_1 \text{ in } Free(\mathcal{C}^!) \\ & = & n; [\![\Gamma \vdash A]\!]; \delta; !\varepsilon; in_1 & \\ & = & \gamma(n; [\![\Gamma \vdash A]\!]); !\varepsilon; in_1 & \\ & = & \gamma(n; [\![\Gamma \vdash A]\!]; \varepsilon); in_1 & \\ & = & \gamma(lin([\![\Gamma \vdash A]\!])); in_1 & \\ & = & \gamma([\![!\Gamma^o \vdash A^o]\!]); in_1 & \text{IH} \\ & = & [\![!\Gamma^o \vdash !A^o \oplus !B^o]\!] & \end{array}$$

The $(\vee_{I1})$ case is analogous to the $(\vee_{I1})$ case.

The $(\vee_E)$ case. The proof

$$\cfrac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\,(\vee_E)$$

is translated into the proof

$$\cfrac{\cfrac{!\Gamma^o \vdash !A^o \oplus !B^o \quad !\Gamma^o, !A^o \vdash C^o \quad !\Gamma^o, !B^o \vdash C^o}{!\Gamma^o, !\Gamma^o \vdash C^o}\,(\oplus_E)}{!\Gamma^o \vdash C^o}\,(Con) \text{ and } (Ex)$$

It can be shown by using IH and a lot of algebraic manipulations that $lin([\![\Gamma \vdash C]\!]) = [\![!\Gamma^o \vdash C^o]\!]$. $\square$

We still have Corollary 5.3 which corresponds to the fact that the result in [Gir87] showing that the Girard Translation is sound with respect to interpretation in the category of coherence spaces and linear stable functions includes finite sums.

7 Extensions with Recursion

7.1 The λ^{rec}-Calculus

The λ-calculus can be extended with recursion, that is, we have an additional term

$$\mathrm{rec}x^A.t$$

and a typing rule as given in Appendix A. The extension with recursion will be called the λ^{rec}-calculus. The term and the typing rule for recursion given here can also be found in [Win93]. We still have Lemma 3.1, Proposition 3.2, Lemma 3.3, and Lemma 3.4.

The usual reduction rules for terms of the λ-calculus, [GLT89] can be extended with a reduction rule for the term corresponding to recursion:

$$\mathrm{rec}x^A.u \;\rightsquigarrow\; u[\mathrm{rec}x^A.u/x]$$

By using Lemma 3.4, we see that the rule satisfies "Subject Reduction", that is, typing is preserved by an application of the reduction rule. Instead of equipping the λ^{rec}-calculus with the mentioned reduction rules, one could define an operational semantics in natural semantics style. This is done in [Win93].

The Curry-Howard isomorphism on the level of proofs can be extended to include the term and the rule for recursion, that is, we have a bijective correspondence between proofs in IL, extended with the rule for recursion as given in Appendix B, and derivable sequents of the λ^{rec}-calculus.

Given a cartesian closed category with an internal fixpoint operator as in Definition 2.15, we extend the categorical interpretation of the λ-calculus to include the rule for recursion, cf. Appendix A, such that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the sequents (where we consider naturality of the operation on arrows corresponding to recursion as naturality in the interpretation of Γ), and such that Lemma 3.5 and Lemma 3.6 still hold.

By using Lemma 3.6, it can be shown that the interpretation is sound with respect to the reduction rule for the term corresponding to recursion. This is so because the reduction rule is essentially a syntactic restatement of the defining equation of a parametrised fixpoint in Definition 2.13.

7.2 The linear λ^{rec}-Calculus

The linear λ-calculus can be extended with recursion, that is, we have an additional term

$$\mathrm{let}\ t,...,t\ \mathrm{be}\ x_1,...,x_n\ \mathrm{in}\ \mathrm{rec}z^{!A}.t$$

and a typing rule as given in Appendix C. The extension with recursion will be called the linear λ^{rec}-calculus. We still have Lemma 4.1, Proposition 4.2, and Lemma 4.3.

The reduction rules for terms of the linear λ-calculus, [BBdPH92], can be extended with a reduction rule for the term corresponding to recursion:

$$\mathrm{let}\ \overline{w}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ \mathrm{rec}z^{!A}.u \rightsquigarrow \mathrm{copy}\ \overline{w}\ \mathrm{as}\ \overline{x}, \overline{x''}\ \mathrm{in}\ (u[\mathrm{let}\ \overline{x''}\ \mathrm{be}\ \overline{x'}\ \mathrm{in}\ !(\mathrm{let}\ \overline{x'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ \mathrm{rec}z^{!A}.u)/z])$$

By using Lemma 4.3, we see that the rule satisfies "Subject Reduction". Instead of equipping the linear λ^{rec}-calculus with the mentioned reduction rules, one could define an operational semantics in natural semantics style. This is dealt with in [Bra94a].

The Curry-Howard isomorphism on the level of proofs can also be extended in the ILL case to include the term and the rule for recursion, that is, we have a bijective correspondence between proofs in ILL, extended with the rule for recursion as given in Appendix D, and derivable sequents of the linear λ^{rec}-calculus.

Given a linear category with finite products and a (generalised) internal linear fixpoint operator as in Definition 2.30, we extend the categorical interpretation of the linear λ-calculus to include the rule for recursion, cf. Appendix C, such that the operations on arrows induced by the typing rules are natural in the interpretation of the unchanged components of the sequents (where we consider naturality of the operation on arrows corresponding to recursion as naturality in the interpretation of $\Gamma_1, ..., \Gamma_n$), and such that Lemma 4.4 still holds.

By using Lemma 4.4, it can be shown that the interpretation is sound with respect to the reduction rule for the term corresponding to recursion. This is so because the reduction rule is essentially a syntactic restatement of the defining equation of a (generalised) parametrised linear fixpoint in Definition 2.29.

7.3 The Girard Translation Extended with Recursion

The Girard Translation still works if we extend it with the rules for recursion given in Appendix B and Appendix D. The extended Girard Translation at the level of proofs is stated in Appendix E. The translation preserves the reduction rule for recursion. The translation induces a translation from types and derivable sequents in the λ^{rec}-calculus to types and derivable sequents in the linear λ^{rec}-calculus cf. the extended Curry-Howard isomorphisms.

If $\mathcal{C}$ is a linear category with finite products and a (generalised) internal linear fixpoint operator, then $Free(\mathcal{C}^!)$ is a cartesian closed category with an internal fixpoint operator, as previous results show. We can therefore interpret types and derivable sequents in the λ^{rec}-calculus as objects and arrows in $Free(\mathcal{C}^!)$. We obviously still have Proposition 5.1; and Theorem 5.2 (Soundness) also still holds, as the following addition to the proof shows:

The proof

$$\frac{\Gamma, B \vdash B}{\Gamma \vdash B}\,(Rec)$$

is translated into the proof

$$\frac{!\Gamma^o, !B^o \vdash B^o}{!\Gamma^o \vdash B^o}\,(Rec)$$

The following calculation shows the wanted result:

$$\begin{array}{rcll}
lin(\llbracket \Gamma \vdash B \rrbracket) & = & n; \lambda(\llbracket \Gamma, B, \vdash B \rrbracket); Y; \varepsilon & \\
& = & n; \phi^{-1}(\lambda(\phi(n; \llbracket \Gamma, B, \vdash B \rrbracket))); \phi^{-1}(Y); \varepsilon & \text{def. of } \lambda \text{ and } Y \text{ in } Free(\mathcal{C}^!) \\
& = & \gamma(n; \lambda(n; \llbracket \Gamma, B, \vdash B \rrbracket; \varepsilon)); Y & \\
& = & \gamma(\lambda((n \otimes Id); n; \llbracket \Gamma, B, \vdash B \rrbracket; \varepsilon)); Y & \\
& = & \gamma(\lambda(n; \llbracket \Gamma, B, \vdash B \rrbracket; \varepsilon)); Y & * \\
& = & \gamma(\lambda(lin(\llbracket \Gamma, B, \vdash B \rrbracket))); Y & \\
& = & \gamma(\lambda(\llbracket !\Gamma^o, !B^o \vdash B^o \rrbracket)); Y & \text{IH} \\
& = & \llbracket !\Gamma^o \vdash B^o \rrbracket &
\end{array}$$

* Because n makes ! a monoidal functor from $(\mathcal{C}, 1, \times)$ to $(\mathcal{C}, I, \otimes)$. $\Box$

7.4 The Choice of Rule for Recursion in the linear λ^{rec}-Calculus

The first demand to an extension of the linear λ-calculus with recursion is that it must enable the existing Curry-Howard isomorphism to be extended, as is the case with the extension of the λ-calculus with recursion. Secondly, the Girard Translation must still work with the extensions, and moreover, the translation must preserve the reduction rule for recursion.

The the choice of rules for recursion is best explained in terms of Gentzen style formulations of IL and ILL. In natural deduction style we have introduction and elimination rules for each type constructor. In Gentzen style we have two introduction rules for each type constructor; a constructor can be introduced on each side of the sequent, and moreover, we have the cut rule. The Girard Translation translation for Gentzen style formulations of IL and ILL is similar to the translation for the natural deduction style formulations given in Appendix E, except the case with the cut rule:

$$\dfrac{\Gamma \vdash A \quad \Gamma, A \vdash C}{\Gamma \vdash C}\,(Cut) \quad \mapsto \quad \dfrac{\dfrac{\dfrac{!\Gamma^o \vdash A^o}{!\Gamma^o \vdash !A^o}\,(!_R) \qquad !\Gamma^o, !A^o \vdash C^o}{!\Gamma^o, !\Gamma^o \vdash C^o}\,(Cut)}{!\Gamma^o \vdash C^o}\,(Con) \text{ and } (Ex)$$

Now, we will ensure that the Girard Translation still works with the rules for recursion by taking the rule for recursion in the extended ILL in Gentzen style to be the image under the translation of the rule for recursion in the extended IL in Gentzen style (note that the rule for recursion in the extended IL in Gentzen style is the same as in the natural deduction formulation, the λ^{rec}-Calculus). We thus obtain the extended translation as follows:

$$\dfrac{\Gamma, B \vdash B}{\Gamma \vdash B}\,(Rec) \quad \mapsto \quad \dfrac{!\Gamma^o, !B^o \vdash B^o}{!\Gamma^o \vdash B^o}\,(Rec)$$

When decorated appropriately with terms, it is easy to see that these rules enable the existing Curry-Howard isomorphisms for IL and ILL to be extended. We also want the translation to preserve the reduction rule for recursion, which in the extended IL looks as follows:

$$\frac{\Gamma, B \vdash B}{\Gamma \vdash B}\,(Rec) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Gamma, B \vdash B}{\Gamma \vdash B}\,(Rec) \qquad \Gamma, B \vdash B}{\Gamma \vdash B}\,(Cut)$$

corresponding to

$$\mathrm{rec}x.u \;\rightsquigarrow\; u[\mathrm{rec}x.u/x]$$

when the rules are decorated with terms. We will ensure that the Girard Translation preserves the reduction rule for recursion by taking the reduction rule for recursion in the extended ILL to be the image under the translation of the reduction rule for recursion in the extended IL. We thus obtain the reduction rule for recursion in the extended ILL as follows:

$$\frac{!\Gamma^o, !B^o \vdash B^o}{!\Gamma^o \vdash B^o}\,(Rec) \quad \rightsquigarrow \quad \cfrac{\cfrac{\dfrac{\dfrac{!\Gamma^o, !B^o \vdash B^o}{!\Gamma^o \vdash B^o}\,(Rec)}{!\Gamma^o \vdash !B^o}\,(!_R) \qquad !\Gamma^o, !B^o \vdash B^o}{!\Gamma^o, !\Gamma^o \vdash B^o}\,(Cut)}{!\Gamma^o \vdash B^o}\,(Con) \text{ and } (Ex)$$

corresponding to

$$\text{let } \overline{y} \text{ be } \overline{x} \text{ in } \mathrm{rec}z.u \;\rightsquigarrow\; \text{copy } \overline{y} \text{ as } \overline{x}, \overline{x''} \text{ in } (u[\text{let } \overline{x''} \text{ be } \overline{x'} \text{ in } !(\text{let } \overline{x'} \text{ be } \overline{x} \text{ in } \mathrm{rec}z.u)/z])$$

when the rules are decorated with terms.

The above mentioned rule for recursion in the extended ILL is appropriate for a Gentzen style formulation, but we lose the Substitution Property if we add it to a natural deduction formulation of ILL. This problem is similar to the problem with the $(!_R)$ rule, [Wad91]. The solution is similar too; we generalise the rule for recursion to the one given in Appendix D. The above mentioned reduction rule for recursion in the extended ILL in Gentzen style corresponds to the one given in the natural deduction formulation, the linear λ^{rec}-calculus.

Recursion in the linear λ^{rec}-calculus is essentially the image under the Girard Translation of recursion in the λ^{rec}-calculus, that is, it is essentially recursion in the IL fragment of ILL. Speaking in semantical terms, then parametrised linear fixpoints are parametrised fixpoints in $Free(\mathcal{C}^!)$. When looking at particular models of ILL, for example the category of CPOs and strict continuous functions, there does not seem to be a general method to find non-trivial fixpoints of arbitrary maps; there only seems to be a general method for maps corresponding to maps of free coalgebras. This is evidence that it is not possible to define recursion in the linear λ-calculus in a way that is more general that the one given in the present paper.

Acknowledgements. I am grateful to my supervisor, Glynn Winskel, for his guidance and support. Thanks to Valeria de Paiva for discussions on linear logic and for comments on this paper at various stages of its history. Thanks also to Gavin Bierman for comments on an early version of this paper and for a Girardian turnstyle macro. Thanks to Claudio Hermida for advice on categorical matters and thanks to Francois Lamarche for a discussion on linear fixpoint objects. I have used Paul Taylor's macros to produce the diagrams and proof-rules.

References

[Abr90] S. Abramsky. Computational interpretations of linear logic. Technical Report 90/20, Department of Computing, Imperial College, 1990.

[AJM95] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. Manuscript, 1995.

[AM94] S. Abramsky and G. McCusker. Games for recursive types. Manuscript, 1994.

[BBdPH92] N. Benton, G. Bierman, V. de Paiva, and M. Hyland. Term assignment for intuitionistic linear logic. Technical Report 262, Computer Laboratory, University of Cambridge, 1992.

[Bie94] G. Bierman. *On Intuitionistic Linear Logic*. PhD thesis, Computer Laboratory, University of Cambridge, 1994.

[Bra94a] T. Braüner. A general adequacy result for a linear functional language. Technical Report BRICS-RS-94-22, BRICS, Department of Computer Science, University of Aarhus, aug 1994. Manuscript presented at MFPS '94.

[Bra94b] T. Braüner. A model of intuitionistic affine logic from stable domain theory. In *Proceedings of ICALP '94, LNCS*, volume 820. Springer-Verlag, 1994.

[BW85] M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer-Verlag, 1985.

[BW90] M. Barr and C. Wells. *Category Theory for Computing Science*. Prentice Hall, 1990.

[CP90] R. L. Crole and A. M. Pitts. New foundations for fixpoint computations. In *5th LICS Conference*. IEEE, 1990.

[Fio94] M. P. Fiore. First steps on the representation of domains. Manuscript, 1994.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge University Press, 1989.

[HO94] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF. Manuscript, 1994.

[How80] W. A. Howard. The formulae-as-type notion of construction. In J. R. Hindley and J. P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, 1980.

[HP90] H. Huwig and A. Poigne. A note on inconsistencies caused by fixpoints in a cartesian closed category. *Theoretical Computer Science*, 73, 1990.

[Jac] B. Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic (to appear)*.

[Lam] F. Lamarche. Dialectics: a model of linear logic and PCF. Submitted to MSCS.

[Law69] F. W. Lawvere. Diagonal arguments and cartesian closed categories. In P. Hilton, editor, *Category Theory, Homology Theory and their Applications II, LNM*, volume 92. Springer-Verlag, 1969.

[Mac91] I. Mackie. *Lilac : A Functional Programming Language Based on Linear Logic*. M.Sc. thesis, Imperial College, 1991.

[MRA93] I. Mackie, L. Román, and S. Abramsky. An internal language for autonomous categories. *Journal of Applied Categorical Structures*, 1, 1993.

[Mul92] P. S. Mulry. Strong monads, algebras and fixed points. In M. P. Fourman, P. T. Johnstone, and A. M. Pitts, editors, *Application of Categories in Computer Science*, volume 177. London Mathematical Society Lecture Notes Series, 1992.

[Plo93] G. D. Plotkin. Type theory and recursion (extended abstract). In *8th LICS Conference*. IEEE, 1993.

[Poi92] A. Poigne. Basic category theory. In S. Abramsky et al, editor, *Handbook of Logic in Computer Science*. Oxford University Press, 1992.

[See89] R. A. G. Seely. Linear logic, $*$-autonomous categories, and cofree coalgebras. In *Contemporary Mathematics, Categories in Computer Science and Logic*, volume 92. American Mathematical Society, 1989.

[Sim92] A. Simpson. Recursive types in kleisli categories. Manuscript, 1992.

[Wad91] P. Wadler. There's no substitute for linear logic. Manuscript, 1991.

[Win93] G. Winskel. *The Formal Semantics of Programming Languages.* The MIT Press, 1993.

A Appendix, The λ^{rec}-Calculus with Categorical Semantics

Axiom

$$\frac{}{x_1 : A_1, ..., x_n : A_n \vdash x_q : A_q}(Ax) \qquad \frac{}{A_1 \times ... \times A_n \xrightarrow{\pi_q} A_q}$$

Logical Rules, the $(\mathbf{t}, \wedge, \Rightarrow)$ Fragment

$$\frac{}{\Gamma \vdash <>: \mathrm{t}}(\mathrm{t}_I) \qquad \frac{}{\Gamma \xrightarrow{<>} 1}$$

$$\frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B}{\Gamma \vdash < u, v >: A \wedge B}(\wedge_I) \qquad \frac{\Gamma \xrightarrow{u} A \qquad \Gamma \xrightarrow{v} B}{\Gamma \xrightarrow{<u,v>} A \times B}$$

$$\frac{\Gamma \vdash u : A \wedge B}{\Gamma \vdash \mathrm{fst}(u) : A}(\wedge_{E1}) \qquad \frac{\Gamma \xrightarrow{u} A \times B}{\Gamma \xrightarrow{u} A \times B \xrightarrow{\pi_1} A}$$

$$\frac{\Gamma \vdash u : A \wedge B}{\Gamma \vdash \mathrm{snd}(u) : B}(\wedge_{E2}) \qquad \frac{\Gamma \xrightarrow{u} A \times B}{\Gamma \xrightarrow{u} A \times B \xrightarrow{\pi_2} B}$$

$$\frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x^A.u : A \Rightarrow B}(\Rightarrow_I) \qquad \frac{\Gamma \times A \xrightarrow{u} B}{\Gamma \xrightarrow{\lambda(u)} A \Rightarrow B}$$

$$\frac{\Gamma \vdash f : A \Rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash fu : B}(\Rightarrow_E) \qquad \frac{\Gamma \xrightarrow{f} A \Rightarrow B \qquad \Gamma \xrightarrow{u} A}{\Gamma \xrightarrow{<f,u>} (A \Rightarrow B) \times A \xrightarrow{eval} B}$$

Logical Rules, the $(\mathbf{f}, \vee)$ Fragment

$$\frac{\Gamma \vdash w : \mathrm{f}}{\Gamma \vdash []^C(w) : C}\,(\mathrm{f}_E) \qquad \frac{\Gamma \xrightarrow{w} 0}{\Gamma \xrightarrow{w} 0 \xrightarrow{[]} C}$$

$$\frac{\Gamma \vdash u : A}{\Gamma \vdash \mathrm{inl}^{A\vee B}(u) : A \vee B}\,(\vee_{I1}) \qquad \frac{\Gamma \xrightarrow{u} A}{\Gamma \xrightarrow{u} A \xrightarrow{in_1} A + B}$$

$$\frac{\Gamma \vdash u : B}{\Gamma \vdash \mathrm{inr}^{A\vee B}(u) : A \vee B}\,(\vee_{I2}) \qquad \frac{\Gamma \xrightarrow{u} B}{\Gamma \xrightarrow{u} B \xrightarrow{in_2} A + B}$$

$$\frac{\Gamma \vdash w : A \vee B \qquad \Gamma, x : A \vdash u : C \qquad \Gamma, y : B \vdash v : C}{\Gamma \vdash \text{case } w \text{ of } \mathrm{inl}(x) => u \mid \mathrm{inr}(y) => v : C}\,(\vee_E)$$

$$\frac{\Gamma \xrightarrow{w} A + B \qquad \Gamma \times A \xrightarrow{u} C \qquad \Gamma \times B \xrightarrow{v} C}{\Gamma \xrightarrow{<\Gamma, w>} \Gamma \times (A + B) \xrightarrow{\Gamma \times [\lambda(u), \lambda(v)]} \Gamma \times (\Gamma \Rightarrow C) \xrightarrow{eval} C}$$

Rule for Recursion

$$\frac{\Gamma, x : B \vdash u : B}{\Gamma \vdash \mathrm{rec}x^B.u : B}\,(Rec) \qquad \frac{\Gamma \times B \xrightarrow{u} B}{\Gamma \xrightarrow{\lambda(u)} B \Rightarrow B \xrightarrow{Y} B}$$

B Appendix, Natural Deduction Formulation of IL (Extended with Recursion)

Axiom

$$\frac{}{A_1, ..., A_n \vdash A_q}\,(Ax)$$

Logical Rules, the $(\mathbf{t}, \wedge, \Rightarrow)$ Fragment

$$\frac{}{\Gamma \vdash \mathrm{t}}\,(\mathrm{t}_I)$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,(\wedge_I) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,(\wedge_{E1}) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,(\wedge_{E2})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\,(\Rightarrow_I) \qquad \frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\Rightarrow_E)$$

Logical Rules, the $(\mathbf{f}, \vee)$ Fragment

$$\frac{\Gamma \vdash \mathrm{f}}{\Gamma \vdash C}\,(\mathrm{f}_E)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,(\vee_{I1}) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,(\vee_{I2}) \quad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\,(\vee_E)$$

Rule for Recursion

$$\frac{\Gamma, B \vdash B}{\Gamma \vdash B}\,(Rec)$$

C Appendix, The linear λ^{rec}-Calculus with Categorical Semantics

Axiom

$$\frac{}{x : A \vdash x : A}\,(Ax) \qquad \frac{}{A \xrightarrow{A} A}$$

Structural Rule

$$\frac{\Gamma, x : A, y : B, \Delta \vdash u : C}{\Gamma, y : B, x : A, \Delta \vdash u : C}\,(Ex) \qquad \frac{\Gamma \otimes A \otimes B \otimes \Delta \xrightarrow{u} C}{\Gamma \otimes B \otimes A \otimes \Delta \xrightarrow{\Gamma \otimes \cong \otimes \Delta} \Gamma \otimes A \otimes B \otimes \Delta \xrightarrow{u} C}$$

Logical Rules, the $(I, \otimes, \top, \&, \multimap, !)$ Fragment

$$\frac{}{\vdash * : I}\,(I_I) \qquad \frac{}{I \xrightarrow{I} I}$$

$$\frac{\Lambda \vdash w : I \qquad \Gamma \vdash u : A}{\Gamma, \Lambda \vdash \text{let } w \text{ be } * \text{ in } u : A}\,(I_E) \qquad \frac{\Lambda \xrightarrow{w} I \qquad \Gamma \xrightarrow{u} A}{\Gamma \otimes \Lambda \xrightarrow{\Gamma \otimes w} \Gamma \otimes I \cong \Gamma \xrightarrow{u} A}$$

$$\frac{\Gamma \vdash u : A \qquad \Delta \vdash v : B}{\Gamma, \Delta \vdash u \otimes v : A \otimes B}\,(\otimes_I) \qquad \frac{\Gamma \xrightarrow{u} A \qquad \Delta \xrightarrow{v} B}{\Gamma \otimes \Delta \xrightarrow{u \otimes v} A \otimes B}$$

$$\frac{\Lambda \vdash w : A \otimes B \qquad \Gamma, x : A, y : B \vdash u : C}{\Gamma, \Lambda \vdash \text{let } w \text{ be } x \otimes y \text{ in } u : C}\,(\otimes_E) \qquad \frac{\Lambda \xrightarrow{w} A \otimes B \qquad \Gamma \otimes A \otimes B \xrightarrow{u} C}{\Gamma \otimes \Lambda \xrightarrow{\Gamma \otimes w} \Gamma \otimes A \otimes B \xrightarrow{u} C}$$

$$\frac{}{\Gamma \vdash \ <>: \top}\,(\top_I) \qquad \frac{}{\Gamma \xrightarrow{<>} 1}$$

$$\frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B}{\Gamma \vdash \ < u, v >: A \& B}\,(\&_I) \qquad \frac{\Gamma \xrightarrow{u} A \qquad \Gamma \xrightarrow{v} B}{\Gamma \xrightarrow{<u,v>} A \times B}$$

$$\frac{\Lambda \vdash u : A \& B}{\Lambda \vdash \text{fst}(u) : A}\,(\&_{E1}) \qquad \frac{\Lambda \xrightarrow{u} A \times B}{\Lambda \xrightarrow{u} A \times B \xrightarrow{\pi_1} A}$$

$$\frac{\Lambda \vdash u : A \& B}{\Lambda \vdash \text{snd}(u) : B}\,(\&_{E2}) \qquad \frac{\Lambda \xrightarrow{u} A \times B}{\Lambda \xrightarrow{u} A \times B \xrightarrow{\pi_2} B}$$

$$\frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x^A . u : A \multimap B}\,(\multimap_I) \qquad \frac{\Gamma \otimes A \xrightarrow{u} B}{\Gamma \xrightarrow{\lambda(u)} A \multimap B}$$

$$\frac{\Lambda \vdash f : A \multimap B \qquad \Delta \vdash u : A}{\Lambda, \Delta \vdash fu : B}\,(\multimap_E) \qquad \frac{\Lambda \xrightarrow{f} A \multimap B \qquad \Delta \xrightarrow{u} A}{\Lambda \otimes \Delta \xrightarrow{f \otimes u} (A \multimap B) \otimes A \xrightarrow{eval} B}$$

$$\frac{\Gamma_1 \vdash w_1 :!A_1 \quad ,\ldots, \quad \Gamma_n \vdash w_n :!A_n \qquad x_1 :!A_1, \ldots, x_n :!A_n \vdash u : A}{\Gamma_1, \ldots, \Gamma_n \vdash \text{let } w_1, \ldots, w_n \text{ be } x_1, \ldots, x_n \text{ in } !u :!A}\,(!_I)$$

$$\frac{\Gamma_1 \xrightarrow{w_1} !A_1 \quad ,\ldots, \quad \Gamma_n \xrightarrow{w_n} !A_n \qquad !A_1 \otimes \ldots \otimes !A_n \xrightarrow{u} A}{\Gamma_1 \otimes \ldots \otimes \Gamma_n \xrightarrow{w_1 \otimes \ldots \otimes w_n} !A_1 \otimes \ldots \otimes !A_n \xrightarrow{\gamma(u)} !A}$$

$$\frac{\Lambda \vdash u :!A}{\Lambda \vdash \mathrm{derelict}(u) : A}\,(Der) \qquad \frac{\Lambda \xrightarrow{u} !A}{\Lambda \xrightarrow{u} !A \xrightarrow{\varepsilon} A}$$

$$\frac{\Lambda \vdash w :!A \qquad \Gamma, x :!A, y :!A \vdash u : B}{\Gamma, \Lambda \vdash \text{copy } w \text{ as } x, y \text{ in } u : B}\,(Con) \qquad \frac{\Lambda \xrightarrow{w} !A \qquad \Gamma \otimes !A \otimes !A \xrightarrow{u} B}{\Gamma \otimes \Lambda \xrightarrow{\Gamma \otimes w} \Gamma \otimes !A \xrightarrow{\Gamma \otimes d} \Gamma \otimes !A \otimes !A \xrightarrow{u} B}$$

$$\frac{\Lambda \vdash w :!A \qquad \Gamma \vdash u : B}{\Gamma, \Lambda \vdash \text{discard } w \text{ in } u : B}\,(Weak) \qquad \frac{\Lambda \xrightarrow{w} !A \qquad \Gamma \xrightarrow{u} B}{\Gamma \otimes \Lambda \xrightarrow{\Gamma \otimes w} \Gamma \otimes !A \xrightarrow{\Gamma \otimes e} \Gamma \otimes I \xrightarrow{\cong} \Gamma \xrightarrow{u} B}$$

Logical Rules, the $(0, \oplus)$ Fragment

$$\frac{\Lambda \vdash w : 0}{\Lambda \vdash []^{C}(w) : C}\,(0_E) \qquad \frac{\Lambda \xrightarrow{w} 0}{\Lambda \xrightarrow{w} 0 \xrightarrow{[]} C}$$

$$\frac{\Gamma \vdash u : A}{\Gamma \vdash \mathrm{inl}^{A \oplus B}(u) : A \oplus B}\,(\oplus_{I1}) \qquad \frac{\Gamma \xrightarrow{u} A}{\Gamma \xrightarrow{u} A \xrightarrow{in_1} A + B}$$

$$\frac{\Gamma \vdash u : B}{\Gamma \vdash \mathrm{inr}^{A \oplus B}(u) : A \oplus B}\,(\oplus_{I2}) \qquad \frac{\Gamma \xrightarrow{u} B}{\Gamma \xrightarrow{u} B \xrightarrow{in_2} A + B}$$

$$\frac{\Lambda \vdash w : A \oplus B \qquad \Gamma, x : A \vdash u : C \qquad \Gamma, y : B \vdash v : C}{\Gamma, \Lambda \vdash \text{case } w \text{ of } \mathrm{inl}(x) => u \mid \mathrm{inr}(y) => v : C}\,(\oplus_E)$$

$$\frac{\Lambda \xrightarrow{w} A + B \qquad \Gamma \otimes A \xrightarrow{u} C \qquad \Gamma \otimes B \xrightarrow{v} C}{\Gamma \otimes \Lambda \xrightarrow{\Gamma \otimes w} \Gamma \otimes (A + B) \cong (\Gamma \otimes A) + (\Gamma \otimes B) \xrightarrow{[u,v]} C}$$

Rule for Recursion

$$\frac{\Gamma_1 \vdash w_1 :!A_1 \quad ,..., \quad \Gamma_n \vdash w_n :!A_n \qquad x_1 :!A_1, ..., x_n :!A_n, z :!B \vdash u : B}{\Gamma_1, ..., \Gamma_n \vdash \text{let } w_1, ..., w_n \text{ be } x_1, ..., x_n \text{ in } \mathrm{rec} z^{!B}.u : B}\,(Rec)$$

$$\frac{\Gamma_1 \xrightarrow{w_1} !A_1 \quad ,..., \quad \Gamma_n \xrightarrow{w_n} !A_n \qquad !A_1 \otimes ... \otimes !A_n \otimes !B \xrightarrow{u} B}{\Gamma_1 \otimes ... \otimes \Gamma_n \xrightarrow{w_1 \otimes ... \otimes w_n} !A_1 \otimes ... \otimes !A_n \xrightarrow{\gamma(\lambda(u))} !(!B \multimap B) \xrightarrow{Y} B}$$

D Appendix, Natural Deduction Formulation of ILL (Extended with Recursion)

Axiom

$$\frac{}{A \vdash A}\,(Ax)$$

Structural Rule

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\,(Ex)$$

Logical Rules, the $(I, \otimes, \top, \&, \multimap, !)$ Fragment

$$\frac{}{\vdash I}\,(I_I) \qquad \frac{\Lambda \vdash I \quad \Gamma \vdash A}{\Gamma, \Lambda \vdash A}\,(I_E)$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,(\otimes_I) \qquad \frac{\Lambda \vdash A \otimes B \quad \Gamma, A, B \vdash C}{\Gamma, \Lambda \vdash C}\,(\otimes_E)$$

$$\frac{}{\Gamma \vdash \top}\,(\top_I)$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\,(\&_I) \qquad \frac{\Lambda \vdash A\&B}{\Lambda \vdash A}\,(\&_{E1}) \quad \frac{\Lambda \vdash A\&B}{\Lambda \vdash B}\,(\&_{E2})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\,(\multimap_I) \qquad \frac{\Lambda \vdash A \multimap B \quad \Delta \vdash A}{\Lambda, \Delta \vdash B}\,(\multimap_E)$$

$$\frac{\Gamma_1 \vdash !A_1 \;, ..., \; \Gamma_n \vdash !A_n \quad !A_1, ..., !A_n \vdash A}{\Gamma_1, ..., \Gamma_n \vdash !A}\,(!_I) \qquad \frac{\Lambda \vdash !A}{\Lambda \vdash A}\,(Der)$$

$$\frac{\Lambda \vdash !A \quad \Gamma, !A, !A \vdash B}{\Gamma, \Lambda \vdash B}\,(Con)$$

$$\frac{\Lambda \vdash !A \quad \Gamma \vdash B}{\Gamma, \Lambda \vdash B}\,(Weak)$$

Logical Rules, the $(0, \oplus)$ Fragment

$$\frac{\Lambda \vdash 0}{\Lambda \vdash C}\,(0_E)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}\,(\oplus_{I1}) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}\,(\oplus_{I2}) \quad \frac{\Lambda \vdash A \oplus B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, \Lambda \vdash C}\,(\oplus_E)$$

Rule for Recursion

$$\frac{\Gamma_1 \vdash !A_1 \;\; ,...,\;\; \Gamma_n \vdash !A_n \quad !A_1, ..., !A_n, !B \vdash B}{\Gamma_1, ..., \Gamma_n \vdash B}\,(Rec)$$

E Appendix, The Girard Translation (Extended with Recursion)

A proof of $A_1, ..., A_n \vdash B$ is translated into a proof of $!A_1^o, ..., !A_n^o \vdash B^o$ by induction in the proof of $A_1, ..., A_n \vdash B$. Special cases of rules will be used in the definition when appropriate (for example in the case of (Rec)). A double bar means a number of applications of a rule.

Axiom

$$\frac{}{A_1, ..., A_n \vdash A_q}\,(Ax) \quad \mapsto \quad \genfrac{}{}{1.2pt}{}{\dfrac{\dfrac{}{!A_q^o \vdash !A_q^o}\,(Ax)}{!A_q^o \vdash A_q^o}\,(Der)}{!A_1^o, ..., !A_n^o \vdash A_q^o}\,(Weak) \text{ and } (Ex)$$

Logical Rules, the $(\mathbf{t}, \wedge, \Rightarrow)$ Fragment

$$\frac{}{\Gamma \vdash \mathrm{t}}\,(\mathrm{t}_I) \quad \mapsto \quad \frac{}{!\Gamma^o \vdash \top}\,(\top_I)$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,(\wedge_I) \quad \mapsto \quad \frac{!\Gamma^o \vdash A^o \quad !\Gamma^o \vdash B^o}{!\Gamma^o \vdash A^o \& B^o}\,(\&_I)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,(\wedge_{E1}) \quad \mapsto \quad \frac{!\Gamma^o \vdash A^o \& B^o}{!\Gamma^o \vdash A^o}\,(\&_{E1})$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,(\wedge_{E2}) \quad \mapsto \quad \frac{!\Gamma^o \vdash A^o \& B^o}{!\Gamma^o \vdash B^o}\,(\&_{E2})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\,(\Rightarrow_I) \quad \mapsto \quad \frac{!\Gamma^o, !A^o \vdash B^o}{!\Gamma^o \vdash !A^o \multimap B^o}\,(\multimap_I)$$

$$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\Rightarrow_E) \quad \mapsto \quad \genfrac{}{}{1.2pt}{}{\dfrac{!\Gamma^o \vdash !A^o \multimap B^o \qquad \dfrac{!\Gamma^o \vdash A^o}{!\Gamma^o \vdash !A^o}\,(!_I)}{!\Gamma^o, !\Gamma^o \vdash B^o}\,(\multimap_E)}{!\Gamma^o \vdash B^o}\,(Con) \text{ and } (Ex)$$

Logical Rules, the $(\mathbf{f}, \oplus)$ Fragment

$$\frac{\Gamma \vdash \mathrm{f}}{\Gamma \vdash C}\,(\mathrm{f}_E) \quad\mapsto\quad \frac{!\Gamma^o \vdash 0}{!\Gamma^o \vdash C^o}\,(0_E)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,(\vee_{I1}) \quad\mapsto\quad \frac{\dfrac{!\Gamma^o \vdash A^o}{!\Gamma^o \vdash !A^o}\,(!_I)}{!\Gamma^o \vdash !A^o \oplus !B^o}\,(\oplus_{I1})$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,(\vee_{I2}) \quad\mapsto\quad \frac{\dfrac{!\Gamma^o \vdash B^o}{!\Gamma^o \vdash !B^o}\,(!_I)}{!\Gamma^o \vdash !A^o \oplus !B^o}\,(\oplus_{I2})$$

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\,(\vee_E) \quad\mapsto$$

$$\cfrac{!\Gamma^o \vdash !A^o \oplus !B^o \quad !\Gamma^o, !A^o \vdash C^o \quad !\Gamma^o, !B^o \vdash C^o}{\cfrac{!\Gamma^o, !\Gamma^o \vdash C^o}{!\Gamma^o \vdash C^o}\,(Con) \text{ and } (Ex)}\,(\oplus_E)$$

Rule for Recursion

$$\frac{\Gamma, B \vdash B}{\Gamma \vdash B}\,(Rec) \quad\mapsto\quad \frac{!\Gamma^o, !B^o \vdash B^o}{!\Gamma^o \vdash B^o}\,(Rec)$$

F Appendix, (Linear) Fixpoint Objects

F.1 Fixpoint Objects

Let $\mathcal{C}$ be a linear category such that $Free(\mathcal{C}^!)$ is closed under finite products. The category $Free(\mathcal{C}^!)$ is cartesian closed as shown above, and moreover, it has a strong monad (T, η, μ, t) induced by the adjunction $U^! \dashv F^!$. The functor T is given by $F^!U^!$, a component $\eta_{(!A,\delta)}$ is given by

$$(!A,\delta) \xrightarrow{\delta_A} (!!A,\delta) = T(!A,\delta)$$

and a component $\mu_{(!A,\delta)}$ is given by

$$TT(!A,\delta) = (!!!A,\delta) \xrightarrow{!\varepsilon_{!A}} (!!A,\delta) = T(!A,\delta)$$

A cartesian closed category with a strong monad enables us to define a fixpoint object, [CP90]. We will not give the original definition here, but instead the following equivalent definition, [Mul92].

Definition F.1 *A* fixpoint object *in a cartesian closed category with a strong monad is*

1. *an initial T-algebra $\sigma_Z : TZ \to Z$,*
2. *a map $\infty : 1 \to Z$ which is an equaliser of $\eta_Z; \sigma_Z$ and Id_Z.*

It is shown in [Mul92] that there is a parametrised fixpoint operator restricted to underlying objects of Eilenberg-Moore T-algebras in a cartesian closed category with a strong monad and a fixpoint object.

If we consider $Free(\mathcal{C}^!)$, then every object $(!A,\delta)$ can be equipped with a map $h_{(!A,\delta)}$ making it an Eilenberg-Moore T-algebra, namely the map

$$T(!A,\delta) = (!!A,\delta) \xrightarrow{!\varepsilon_A} (!A,\delta)$$

It is obviously a map of free coalgebras, and it is straightforward to check that the the following diagrams commute

$$\begin{tikzcd}
(!A,\delta) \arrow[r, "\eta"] \arrow[dr, "Id"'] & T(!A,\delta) \arrow[d, "h"] \\
 & (!A,\delta)
\end{tikzcd}
\qquad
\begin{tikzcd}
TT(!A,\delta) \arrow[r, "\mu"] \arrow[d, "Th"'] & T(!A,\delta) \arrow[d, "h"] \\
T(!A,\delta) \arrow[r, "h"] & (!A,\delta)
\end{tikzcd}$$

so it is an Eilenberg-Moore T-algebra. We conclude that if $Free(\mathcal{C}^!)$ has a fixpoint object, then it has a parametrised fixpoint operator, which is equivalent to $\mathcal{C}$ having a parametrised linear fixpoint operator.

It should be remarked that the category of CPOs and continuous functions, which is equivalent to the category of free coalgebras induced by category of CPOs and strict continuous functions, has a fixpoint object, namely the vertical natural numbers.

F.2 Linear Fixpoint Objects

We have above given an account af a parametrised linear fixpoint operator in $\mathcal{C}$ in terms of a fixpoint object in $Free(\mathcal{C}^!)$. It is actually possible to give an account af a parametrised linear fixpoint operator in terms of something similar to a fixpoint object that lives in the same category as where the parametrised linear fixpoint operator lives, namely $\mathcal{C}$. The definition of a linear fixpoint object exploits an initial !-algebra as suggested in [Lam].

Definition F.2 *A* linear fixpoint object *in a pre !-category $\mathcal{C}$ is*

1. *an initial !-algebra $\sigma_Z : !Z \to Z$,*
2. *a map $\infty : I \to Z$ such that $\gamma(\infty)$ is an equaliser of $\gamma(\sigma_Z)$ and $Id_{!Z}$.*

Note that $\gamma(f) = \gamma(f); \gamma(\sigma_Z)$ iff $f = \gamma(f); \sigma_Z$ for any map $f : I \to Z$. Given a !-algebra $g : !A \to A$, then we define $It(g) : Z \to A$ to be a unique map making

$$\begin{tikzcd} !Z \arrow[r, "\sigma"] \arrow[d, "!It(g)"'] & Z \arrow[d, "It(g)"'] \\ !A \arrow[r, "g"] & A \end{tikzcd}$$

commute.

It should be remarked that the category of CPOs and strict continuous functions as well as the category of dI domains and linear stable functions has a linear fixpoint object, namely the vertical natural numbers.

We will use the rest of this subsection to show that a linear fixpoint object on a linear category induces a unique parametrised linear fixpoint operator satisfying a linear version of Plotkin's Axiom. This is essentially a linear version of an analogous result given in [Sim92] which says that a fixpoint object under certain conditions induces a unique parametrised fixpoint operator satisfying Plotkin's Axiom. Now, Plotkin's Axiom is especially simple in the context of a pre !-category:

Definition F.3 *A linear fixpoint operator $(-)^\sharp$ on a pre !-category $\mathcal{C}$ satisfies* Plotkin's Axiom *iff whenever we have maps $f :!B \to B$, $g :!C \to C$, and $h : B \to C$ such that the following diagram commutes*

$$\begin{tikzcd}
!B \arrow[r, "f"] \arrow[d, "!h"'] & B \arrow[d, "h"] \\
!C \arrow[r, "g"] & C
\end{tikzcd}$$

then the following diagram commutes too

$$\begin{tikzcd}
I \arrow[r, "f^\sharp"] \arrow[dr, "g^\sharp"'] & B \arrow[d, "h"] \\
 & C
\end{tikzcd}$$

Lemma F.4 *Let $\mathcal{C}$ be a pre !-category with a linear fixpoint operator $(-)^\sharp$ satisfying Plotkin's Axiom. The diagram*

$$\begin{tikzcd}
I \arrow[r, "\varepsilon_B^\sharp"] \arrow[dr, "\varepsilon_C^\sharp"'] & B \arrow[d, "h"] \\
 & C
\end{tikzcd}$$

commutes for any map $h : B \to C$.

Proof. The result follows from commutativity of

$$\begin{tikzcd}
!B \arrow[r, "\varepsilon_B"] \arrow[d, "!h"'] & B \arrow[d, "h"] \\
!C \arrow[r, "\varepsilon_C"] & C
\end{tikzcd}$$

cf. Plotkin's Axiom. $\Box$

Lemma F.5 *Let $\mathcal{C}$ be a pre !-category with a linear fixpoint object (σ_Z, ∞) with the property that whenever we have a map $f : I \to Z$ such that $\gamma(f)$ equalises $\gamma(\sigma_Z)$ and $Id_{!Z}$ then $f = \infty$. Then $\mathcal{C}$ has a unique linear fixpoint operator $(-)^\sharp$ satisfying Plotkin's Axiom.*

Proof. Given a map $f :!B \to B$ define $f^\sharp = \infty; It(f)$. Commutativity of the following diagram shows that $f^\sharp$ indeed is a linear fixpoint of f

$$\begin{tikzcd}
!I \arrow[d, "!\infty"'] & I \arrow[l, "m_I"'] \arrow[d, "\infty"] \\
!Z \arrow[r, "\sigma"] \arrow[d, "!It(f)"'] & Z \arrow[d, "It(f)"] \\
!B \arrow[r, "f"] & B
\end{tikzcd}$$

We will now show that the linear fixpoint operator $(-)^\sharp$ satisfies Plotkin's Axiom, so assume that we have maps $f :!B \to B$, $g :!C \to C$, and $h : B \to C$ making the first diagram in Definition F.3 commute. The diagram

$$\begin{tikzcd}
!Z \arrow[r, "\sigma"] \arrow[d, "!It(f)"'] & Z \arrow[d, "It(f)"] \\
!B \arrow[r, "f"] \arrow[d, "!h"'] & B \arrow[d, "h"] \\
!C \arrow[r, "g"] & C
\end{tikzcd}$$

commutes, so we have $It(g) = It(f); h$, which entails that the last diagram in Definition F.3 commutes.

We will now show that the linear fixpoint operator $(-)^\sharp$ is the unique one which satisfies Plotkin's Axiom, so assume that $(-)^{\sharp'}$ is an arbitrary linear fixpoint operator satisfying Plotkin's Axiom and let a map $f :!B \to B$ be given. We will then show that $f^{\sharp'} = f^\sharp$.

The following diagram commutes

$$\begin{tikzcd}
!Z \arrow[r, "\sigma"] \arrow[d, "!It(f)"'] & Z \arrow[d, "It(f)"] \\
!B \arrow[r, "f"] & B
\end{tikzcd}$$

which entails that the following diagram commutes

$$\begin{tikzcd}
I \arrow[r, "\sigma^{\sharp'}"] \arrow[dr, "f^{\sharp'}"'] & Z \arrow[d, "It(f)"] \\
& B
\end{tikzcd}$$

because $(-)^{\sharp'}$ is assumed to satisfy Plotkin's Axiom. But $\sigma^{\sharp'} = \gamma(\sigma^{\sharp'}); \sigma$ which entails that $\sigma^{\sharp'} = \infty$ cf. the assumption, and we conclude that $f^{\sharp'} = f^{\sharp}$. $\square$

Let $\mathcal{C}$ be a pre !-category with a linear fixpoint object (σ_Z, ∞). If the pre !-category $\mathcal{C}$ is part of a !-category, then it is straightforward to check that whenever we have a map $f : I \to Z$ such that $\gamma(f)$ equalises $\gamma(\sigma_Z)$ and $Id_{!Z}$ then we automatically have $f = \infty$. Thus, the corresponding extra condition of Lemma F.5 vanishes.

We will show an analogous result for parametrised linear fixpoint operators, so here is a parametrised version of Plotkin's Axiom:

Definition F.6 *A parametrised linear fixpoint operator* $(-)^{\sharp}$ *on a* !*-category* $\mathcal{C}$ *satisfies* Plotkin's Axiom *iff whenever we have maps* $f : !A \otimes !B \to B$, $g : !A \otimes !C \to C$, *and*

$h :!A \otimes B \to C$ such that the following diagram commutes

$$\begin{tikzcd}
!A\otimes !B \arrow[r, "d \otimes Id"] \arrow[d, "d \otimes Id"] & (!A\otimes !A)\otimes !B \cong !A \otimes (!A\otimes !B) \arrow[r, "Id \otimes f"] & !A \otimes B \arrow[dddddd, "h"'] \\
(!A\otimes !A)\otimes !B \arrow[d, "\cong"] & & \\
!A \otimes (!A\otimes !B) \arrow[d, "Id \otimes (\delta \otimes Id)"] & & \\
!A \otimes (!!A\otimes !B) \arrow[d, "Id \otimes m"] & & \\
!A\otimes !(!A \otimes B) \arrow[d, "Id\otimes !h"] & & \\
!A\otimes !C \arrow[rr, "g"] & & C
\end{tikzcd}$$

then the following diagram commutes too

$$\begin{tikzcd}
!A \arrow[r, "d"] \arrow[dr, "g^\sharp"'] & !A\otimes !A \arrow[r, "Id \otimes f^\sharp"] & !A \otimes B \arrow[dl, "h"] \\
& C &
\end{tikzcd}$$

Lemma F.7 *Let $\mathcal{C}$ be a linear category with an initial !-algebra $\sigma_Z :!Z \to Z$. Given a map $f :!A\otimes !B \to B$ there is a unique map $IT(f) :!A \otimes Z \to B$ making the following* *diagram commute*

$$\begin{tikzcd}
!A\otimes !Z \arrow[r, "Id\otimes\sigma"] \arrow[d, "d\otimes Id"'] & !A\otimes Z \arrow[ddddd, "IT(f)"'] \\
(!A\otimes !A)\otimes !Z \arrow[d, "\cong"'] & \\
!A\otimes(!A\otimes !Z) \arrow[d, "Id\otimes(\delta\otimes Id)"'] & \\
!A\otimes(!!A\otimes !Z) \arrow[d, "Id\otimes m"'] & \\
!A\otimes !(!A\otimes Z) \arrow[d, "Id\otimes !IT(f)"'] & \\
!A\otimes !B \arrow[r, "f"] & B
\end{tikzcd}$$

Proof. Let a map $f :!A\otimes !B \to B$ be given. We have to show that there is a unique map $IT(f) :!A \otimes Z \to B$ making the wanted diagram commute. We first define H_1 to be the composite

$$!(!A \multimap B)\otimes !A \xrightarrow{Id\otimes d} !(!A \multimap B) \otimes (!A\otimes !A) \cong (!(!A \multimap B)\otimes !A)\otimes !A$$

we then define H_2 to be the composite

$$!(!A \multimap B)\otimes !A \xrightarrow{Id\otimes\delta} !(!A \multimap B)\otimes !!A \xrightarrow{m} !((!A \multimap B)\otimes !A) \xrightarrow{!eval} !B$$

and finally, we define $IT(f)$ to be a unique map making the following diagram commute

$$\begin{tikzcd}
!Z \arrow[rr, "\sigma"] \arrow[d, "!\lambda(\cong;IT(f))"'] & & Z \arrow[d, "\lambda(\cong;IT(f))"'] \\
!(!A \multimap B) \arrow[rr, "\lambda(H_1;(H_2 \otimes Id);\cong;f)"] & & !A \multimap B
\end{tikzcd}$$

where we use that the operation $\lambda(\cong;(-))$ from $\mathcal{C}(!A \otimes Z, B)$ to $\mathcal{C}(Z, !A \multimap B)$ is a bijection. We will now show that $IT(f)$ is a unique map making the wanted diagram commute by showing that the operation $\cong;\lambda^{-1}(-)$ from $\mathcal{C}(!Z, !A \multimap B)$ to $\mathcal{C}(!A \otimes !Z, B)$, which is a bijection, applied to each of the two paths in the defining diagram for $IT(f)$ yields each of the two paths in the wanted diagram. Firstly, we have that

$$\cong;\lambda^{-1}(\sigma;\lambda(\cong;IT(f))) = (Id \otimes \sigma);IT(f)$$

Secondly, we have that

$$\begin{array}{ll}
\cong;\lambda^{-1}(!\lambda(\cong;IT(f));\lambda(H_1;(H_2 \otimes Id);\cong;f)) = & \\
\cong;(!\lambda(\cong;IT(f)) \otimes Id);H_1;(H_2 \otimes Id);\cong;f = & \\
\cong;(Id \otimes d);\cong;((Id \otimes \delta) \otimes Id);(m \otimes Id);(! \cong \otimes Id);(!IT(f) \otimes Id);\cong;f = & * \\
\cong;(Id \otimes d);\cong;(\cong \otimes Id);\cong;(Id \otimes (\delta \otimes Id));(Id \otimes m);(Id \otimes !IT(f));f = & ** \\
(d \otimes Id);\cong;(Id \otimes (\delta \otimes Id));(Id \otimes m);(Id \otimes !IT(f));f &
\end{array}$$

* cf. commutativity of the following diagram

$$\begin{tikzcd}
(!Z \otimes !A) \otimes !A \arrow[r, "(Id \otimes \delta) \otimes Id"] \arrow[d, "\cong \otimes Id"'] & (!Z \otimes !!A) \otimes !A \arrow[r, "m \otimes Id"] \arrow[d, "\cong \otimes Id"'] \arrow[dr, phantom, "1."] & !(Z \otimes !A) \otimes !A \arrow[d, "! \cong \otimes Id"'] \\
(!A \otimes !Z) \otimes !A \arrow[r, "(\delta \otimes Id) \otimes Id"] \arrow[d, "\cong"'] & (!!A \otimes !Z) \otimes !A \arrow[r, "m \otimes Id"] \arrow[d, "\cong"'] & !(!A \otimes Z) \otimes !A \arrow[d, "!IT(f) \otimes Id"'] \\
!A \otimes (!A \otimes !Z) \arrow[r, "Id \otimes (\delta \otimes Id)"] & !A \otimes (!!A \otimes !Z) \arrow[d, "Id \otimes m"'] & !B \otimes !A \arrow[d, "\cong"'] \\
 & !A \otimes !(!A \otimes Z) \arrow[r, "Id \otimes !IT(f)"] & !A \otimes !B
\end{tikzcd}$$

1. ! is symmetric monoidal.

** cf. commutativity of the following diagram

$$
\begin{tikzcd}
 & !A \otimes !Z \arrow[r, "\cong"] \arrow[dl, "d \otimes Id"', "1."] \arrow[d, "d \otimes Id"] & !Z \otimes !A \arrow[d, "Id \otimes d"'] \\
(!A \otimes !A) \otimes !Z \arrow[ddd, "\cong"'] & (!A \otimes !A) \otimes !Z \arrow[l, "\cong \otimes Id"'] \arrow[r, "\cong"] \arrow[d, "\cong"'] & !Z \otimes (!A \otimes !A) \arrow[ddd, "\cong"'] \\
 & !A \otimes (!A \otimes !Z) \arrow[d, "Id \otimes \cong"'] & \\
 & !A \otimes (!Z \otimes !A) \arrow[d, "\cong"'] & \\
!A \otimes (!A \otimes !Z) & (!A \otimes !Z) \otimes !A \arrow[l, "\cong"'] & (!Z \otimes !A) \otimes !A \arrow[l, "\cong \otimes Id"']
\end{tikzcd}
$$

1. d is part of a commutative comonoid. $\Box$

Lemma F.8 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object (σ_Z, ∞). Then $(F_{S^A}(\sigma_Z), F_{S^A}(\infty))$ is a linear fixpoint object on $\mathcal{C}_{S^A}$ with the property that whenever we have a map $f : !A \otimes I \to Z$ in $\mathcal{C}_{S^A}$ such that $\gamma(f)$ equalises $\gamma(\sigma_Z)$ and $Id_{!Z}$ in $\mathcal{C}_{S^A}$ then $f = F_{S^A}(\infty)$.*

Proof. Note that we consider $\mathcal{C}_{S^A}$ as a pre !-category. The map $F_{S^A}(\sigma_Z)$ is an initial $\tilde{!}$-algebra, Lemma F.7. The functor F_{S^A} has a left adjoint, which entails that it preserves limits, so $F_{S^A}(\infty)$ is an equaliser of $F_{S^A}(\gamma(\sigma_Z))$ and $Id_{\tilde{!}Z}$ in $\mathcal{C}_{S^A}$. But $F_{S^A}(\gamma(\sigma_Z)) = \gamma(F_{S^A}(\sigma_Z))$, so $(F_{S^A}(\sigma_Z), F_{S^A}(\infty))$ is a linear fixpoint object in $\mathcal{C}_{S^A}$. We moreover have to show that whenever we have a map $f : !A \otimes I \to Z$ such that $\gamma(f)$ equalises $\gamma(\sigma_Z)$ and $Id_{!Z}$ in $\mathcal{C}_{S^A}$ then $f = F_{S^A}(\infty)$, which amounts to show that the unique map $h : !A \otimes I \to I$ such that $\gamma(f) = h; \gamma(F_{S^A}(\infty))$ in $\mathcal{C}_{S^A}$ is equal to Id_I in $\mathcal{C}_{S^A}$. Now, $\gamma(f) = h; \gamma(F_{S^A}(\infty))$ in $\mathcal{C}_{S^A}$, which entails that $\gamma(f); F_{S^A}(e) = h; \gamma(F_{S^A}(\infty)); F_{S^A}(e)$ in

$\mathcal{C}_{S^A}$. The following calculation shows that RHS is equal to h

$$\begin{array}{rcll} \gamma(F_{S^A}(\infty)); F_{S^A}(e) & = & F_{S^A}(\gamma(\infty)); F_{S^A}(e) & \\ & = & F_{S^A}(\gamma(\infty); e) & \\ & = & F_{S^A}(Id) & e \text{ is monoidal} \\ & = & Id & \end{array}$$

and the following calculation shows that LHS, that is, $\gamma(f); F_{S^A}(e)$, where composition is in $\mathcal{C}_{S^A}$, is equal to Id_I in $\mathcal{C}_{S^A}$

$$\begin{array}{rcll} \delta^A; S^A(\gamma(f)); \varepsilon^A; e & = & \gamma(f); e & \\ & = & \delta^A; S^A(\varepsilon^A; m_I); \lambda; !f; e & \\ & = & S^A(m_I); \lambda; e & \\ & = & (\delta \otimes m_I); m; e & \\ & = & ((\delta; e) \otimes I); \cong & e \text{ is monoidal} \\ & = & (e \otimes I); \cong & \delta \text{ is a map of comonoids} \\ & = & \varepsilon^A & \end{array}$$

We conclude that h is equal to Id_I in $\mathcal{C}_{S^A}$, and thus, $f = F_{S^A}(\infty)$. □

Theorem F.9 *A linear category $\mathcal{C}$ with a linear fixpoint object (σ_Z, ∞) has a unique parametrised linear fixpoint operator satisfying Plotkin's Axiom. The parametrised linear fixpoint operator is natural.*

Proof. Each category $\mathcal{C}_{S^A}$, considered as a pre !-category, has a linear fixpoint object satisfying the extra conditions of Lemma F.5, cf. Lemma F.8. This entails that each category $\mathcal{C}_{S^A}$ has a unique linear fixpoint operator satisfying Plotkin's Axiom, Lemma F.5. But to give a linear fixpoint operator on each category $\mathcal{C}_{S^A}$ is equivalent to give a parametrised linear fixpoint operator on $\mathcal{C}$, Proposition 2.19, and moreover, it is straightforward to see that the linear fixpoint operator on each category $\mathcal{C}_{S^A}$ satisfies Plotkin's Axiom iff the corresponding parametrised linear fixpoint operator on $\mathcal{C}$ satisfies the parametrised version of Plotkin's Axiom. The parametrised linear fixpoint operator on $\mathcal{C}$ can be shown to be natural. □

F.3 Linear Fixpoint Objects and the Intrinsic Preorder

This subsection should be seen in light of recent work in Axiomatic Domain Theory, [Fio94] where an objective has been to find non-order-theoretic axioms inducing CPO-enrichment on a cartesian closed category. It is, however, the case that a weaker assumption than CPO-enrichment will do if one wants to give an adequate semantics to PCF, namely that of rationality, [AJM95].

Definition F.10 *A poset-enriched cartesian closed category $\mathcal{C}$ is* pointed *if each poset $\mathcal{C}(B, C)$ has a bottom element $\perp_{B,C}$ such that $f; \perp_{B,C} = \perp_{A,C}$ for any map $f : A \to B$.*

Definition F.11 *Let $\mathcal{C}$ be a pointed poset-enriched cartesian closed category. Given a map $f : A \times B \to B$, we define a map $f^n : A \to B$ inductively as*

$$\begin{aligned} f^0 &= \perp_{A,B} \\ f^{n+1} &= \Delta; (A \times f^n); f \end{aligned}$$

Definition F.12 *A poset-enriched cartesian closed category $\mathcal{C}$ is* rational *if it is pointed and moreover for every map $f : A \times B \to B$ the increasing chain $\{f^n | n \in \omega\}$ has a least upper bound $f^\sharp : A \to B$ with the property that $g; f^\sharp; h$ is a least upper bound for the increasing chain $\{g; f^n; h | n \in \omega\}$ for any maps $g : C \to A$ and $h : B \to D$.*

The difference is essentially that CPO-enrichment assumes the existence of least upper bounds of arbitrary increasing chains whereas only certain increasing chains are assumed to have least upper bounds in a rational category. It is in this subsection shown how certain axioms entails that the linear fixpoint of a map is a least upper bound of an appropriate increasing chain with respect to the intrinsic preorder. This entails that the induced coKleisli category is rational with respect to the partial order obtained as the quotient of the intrinsic preorder. We first need a couple of definitions from [HO94].

Definition F.13 *Let $\mathcal{C}$ be a symmetric monoidal closed category. A* notion of observables *$\mathcal{O}$ associates to each object A a set $\mathcal{O}_A$ of subsets of $\mathcal{C}(I, A)$ with the property that if $S \in \mathcal{O}_B$ and $g : A \to B$, then $\{h : I \to A | h; g \in \mathcal{O}_B\} \in \mathcal{O}_A$.*

Given a map $f : A \to B$ in a symmetric monoidal closed category, then $\ulcorner f \urcorner : I \to A \multimap B$ is equal to $\lambda(\cong; f)$.

Definition F.14 *Let $\mathcal{C}$ be a symmetric monoidal closed category with a notion of observables $\mathcal{O}$. The* observational preorder *$\lesssim$ on each $\mathcal{C}(A, B)$ is defined as follows*

$$f \lesssim g \Leftrightarrow \forall R \in \mathcal{O}_{A \multimap B}. \ulcorner f \urcorner \in R \Rightarrow \ulcorner g \urcorner \in R$$

The following result can be found in [HO94].

Proposition F.15 *A symmetric monoidal closed category $\mathcal{C}$ with a notion of observables is preorder-enriched with respect to the induced observational preorder.*

Proof. [HO94]. □

In what follows, we will be dealing with an object Σ and a map $\top : I \to \Sigma$ in a symmetric monoidal closed category. The object Σ should be thought of as the Sierpinsky space with the map $\top$ representing the top element. One should in most concrete cases take Σ to be equal to I and $\top$ to be equal to Id_I; this is for example the case in the category of CPOs and strict continuous functions and in the category of dI domains and linear stable functions.

Definition F.16 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. For each object A we define*

$$\mathcal{O}_A = \{\mathcal{O}_f | f :!A \to \Sigma\}$$

where

$$\mathcal{O}_f = \{h : I \to A | \gamma(h); f = \top\}$$

This is a notion of observables because

$$\begin{array}{rcl} \{h : I \to A | h; g \in \mathcal{O}_k\} & = & \{h : I \to A | \gamma(h; g); k = \top\} \\ & = & \{h : I \to A | \gamma(h); !g; k = \top\} \\ & = & \mathcal{O}_{!g;k} \end{array}$$

where $g : A \to B$ and $k :!B \to \Sigma$

Definition F.17 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. The* intuitionistic intrinsic preorder $\lesssim$ *on each $\mathcal{C}(A, B)$ is defined as the observational preorder induced by $\mathcal{O}$, that is,*

$$f \lesssim g \Leftrightarrow \forall h :!(A \multimap B) \to \Sigma . \gamma(\ulcorner f \urcorner); h = \top \Rightarrow \gamma(\ulcorner g \urcorner); h = \top$$

Proposition F.18 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. If $f, g : I \to A$, then*

$$f \lesssim g \Leftrightarrow \forall h :!A \to \Sigma . \gamma(f); h = \top \Rightarrow \gamma(g); h = \top$$

Proof. Note that the two maps

$$A \xrightarrow{\lambda(\cong)} I \multimap A$$

and

$$I \multimap A \xrightarrow{\cong} (I \multimap A) \otimes I \xrightarrow{eval} A$$

makes A and $I \multimap A$ isomorphic. The result follows from the observation that $f = \ulcorner f \urcorner; \cong; eval$, and thus $\gamma(f) = \gamma(\ulcorner f \urcorner); !(\cong; eval)$, for any map $f : I \to A$. □

Proposition F.19 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. The symmetric monoidal closed category $(\mathcal{C}, I, \otimes, \multimap)$ is preorder-enriched with respect to the intuitionistic intrinsic preorder.*

Proof. Proposition F.15. □

Proposition F.20 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. The functor $!$ is preorder-enriched with respect to the intuitionistic intrinsic preorder.*

Proof. Let maps $f, g : A \to B$ such that $f \lesssim g$ be given. Consider the map

$$!(A \multimap B) \otimes !A \xrightarrow{m} !((A \multimap B) \otimes A) \xrightarrow{!eval} !B$$

We have that

$$\begin{aligned} \gamma(\ulcorner f \urcorner); \lambda(m, !eval) &= \lambda((\gamma(\ulcorner f \urcorner) \otimes !A); m, !eval) \\ &= \lambda((m_I \otimes !A); m, ! \cong; !f) \\ &= \lambda(\cong; !f) && ! \text{ is monoidal} \\ &= \ulcorner !f \urcorner \end{aligned}$$

and similarly for g. Thus

$$\begin{array}{ll} \gamma(\ulcorner !f \urcorner); h = \top & \Leftrightarrow \\ \gamma(\ulcorner f \urcorner); \gamma(\lambda(m, !eval)); h = \top & \Rightarrow \\ \gamma(\ulcorner g \urcorner); \gamma(\lambda(m, !eval)); h = \top & \Leftrightarrow \\ \gamma(\ulcorner !f \urcorner); h = \top & \end{array}$$

where $h : !(!A \multimap !B) \to \Sigma$ and we conclude that $!f \lesssim !g$. $\square$

The following notion of observables is in [HO94] called *termination to specified value.*

Definition F.21 *Let $\mathcal{C}$ be a symmetric monoidal closed category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. For each object A we define*

$$\mathcal{O}'_A = \{\mathcal{O}'_f | f : A \to \Sigma\}$$

where

$$\mathcal{O}'_f = \{h : I \to A | h; f = \top\}$$

This is a notion of observables, as mentioned in [HO94], because

$$\begin{aligned} \{h : I \to A | h; g \in \mathcal{O}'_k\} &= \{h : I \to A | h; g; k = \top\} \\ &= \mathcal{O}'_{g;k} \end{aligned}$$

where $g : A \to B$ and $k : B \to \Sigma$

Definition F.22 *Let $\mathcal{C}$ be a symmetric monoidal closed category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. The* linear intrinsic preorder $\lesssim'$ *on each $\mathcal{C}(A, B)$ is defined as the observational preorder induced by $\mathcal{O}'$, that is,*

$$f \lesssim' g \Leftrightarrow \forall h : A \multimap B \to \Sigma. \ulcorner f \urcorner; h = \top \Rightarrow \ulcorner g \urcorner; h = \top$$

Proposition F.23 *Let $\mathcal{C}$ be a symmetric monoidal closed category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. If $f, g : I \to A$, then*

$$f \lesssim' g \Leftrightarrow \forall h : A \to \Sigma. f; h = \top \Rightarrow g; h = \top$$

Proof. Analogous to the proof of Proposition F.18. □

Proposition F.24 *A symmetric monoidal closed category $\mathcal{C}$ with a given object Σ and a given map $\top : I \to \Sigma$ is preorder-enriched with respect to the linear intrinsic preorder.*

Proof. Proposition F.15. □

Definition F.25 *Let $\mathcal{C}$ be a linear category with a linear fixpoint operator. For any pair of objects A and B we define a map $\perp_{A,B}$: $A \to B$ as*

$$A \xrightarrow{\cong} I \otimes A \xrightarrow{\lambda^{-1}(\varepsilon^{\sharp}_{A \multimap B})} B$$

Proposition F.26 *Let $\mathcal{C}$ be a linear category with a linear fixpoint operator satisfying Plotkin's Axiom. For any object A we have $\perp_{I,A}= \varepsilon^{\sharp}_{A}$.*

Proof. The result follows from Lemma F.4. □

So $\perp_{I,A}$ is a linear fixpoint of ε_A, that is, a fixpoint of Id_A in the coKleisli category.

Proposition F.27 *Let $\mathcal{C}$ be a linear category with a linear fixpoint operator satisfying Plotkin's Axiom. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon^{\sharp}_{\Sigma}$. The map $\perp_{A,B}$ is a bottom element of $\mathcal{C}(A, B)$ with respect to the linear intrinsic preorder.*

Proof. We have $\varepsilon^{\sharp}_{A \multimap B}; h = \varepsilon^{\sharp}_{\Sigma}$ cf. Lemma F.4, and therefore $\varepsilon^{\sharp}_{A \multimap B}; h \neq \top$, for any map $h : A \multimap B \to \Sigma$. □

Proposition F.28 *Let $\mathcal{C}$ be a linear category with a linear fixpoint operator satisfying Plotkin's Axiom. We have $\perp_{A,B}; f =\perp_{A,C}$ for any map $f : B \to C$.*

Proof. The result follows from Lemma F.4. □

Proposition F.29 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. We have $f; \perp_{B,C}=\perp_{A,C}$ for any map $f : A \to B$.*

Proof. Let a map $f : A \to B$ be given. We have commutativity of

$$\begin{tikzcd}
!Z \arrow[r, "\sigma"] \arrow[d, "!It(\varepsilon_{B \multimap C})"'] & Z \arrow[d, "It(\varepsilon_{B \multimap C})"'] \\
!(B \multimap C) \arrow[r, "\varepsilon_{B \multimap C}"] \arrow[d, "!(f \multimap C)"'] & B \multimap C \arrow[d, "f \multimap C"'] \\
!(A \multimap C) \arrow[r, "\varepsilon_{A \multimap C}"] & A \multimap C
\end{tikzcd}$$

which entails that $It(\varepsilon_{B \multimap C}); (f \multimap C) = It(\varepsilon_{A \multimap C})$, so

$$\begin{array}{rcl}
f; \perp_{B,C} & = & f; \cong; \lambda^{-1}(\infty; It(\varepsilon_{B \multimap C})) \\
& = & \cong; (I \otimes f); ((\infty; It(\varepsilon_{B \multimap C})) \otimes B); eval \\
& = & \cong; ((\infty; It(\varepsilon_{B \multimap C})) \otimes A); ((f \multimap C) \otimes A); eval \\
& = & \cong; \lambda^{-1}(\infty; It(\varepsilon_{B \multimap C}); (f \multimap C)) \\
& = & \cong; \lambda^{-1}(\infty; It(\varepsilon_{A \multimap C})) \\
& = & \perp_{A,C}
\end{array}$$

□

Proposition F.30 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given. We have $\lesssim \subseteq \lesssim'$.*

Proof. The result follows from the observation that
$\ulcorner f \urcorner = \gamma(\ulcorner f \urcorner); \varepsilon$ for any map $f : A \to B$. □

Now, ! is preorder-enriched with respect to $\lesssim$, and $\perp$ is a bottom element with respect to $\lesssim'$, so we would like the two orders to coinside.

Definition F.31 *Let $\mathcal{C}$ be a linear category. Assume that an object Σ and a map $\top : I \to \Sigma$ is given.* Axiom 1 *is satisfied iff we for any maps $f, g : I \to A$ have that*

$$\forall h : A \to \Sigma. f; h = \top \Rightarrow g; h = \top$$

implies

$$\forall h : !A \to \Sigma. \gamma(f); h = \top \Rightarrow \gamma(g); h = \top$$

It should be remarked that the category of CPOs and strict continuous functions as well as the category of dI domains and linear stable functions satisfies Axiom 1.

Proposition F.32 *A linear category with a given object Σ and a given map $\top : I \to \Sigma$ satisfies* Axiom 1 *iff* $\lesssim = \lesssim'$.

Proof. Proposition F.23, Proposition F.18, and Proposition F.30. $\Box$

This proposition enables us to blur the distinction between the intuitionistic intrinsic preorder and the linear intrinsic preorder whenever Axiom 1 is satisfied.

Definition F.33 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. For any number n we define a map $\underline{n} : I \to Z$ inductively as*

$$\begin{aligned} \underline{0} &= \perp_{I,Z} \\ \underline{n+1} &= \gamma(\underline{n}); \sigma \end{aligned}$$

and for any number n we define a map $\overline{n} : !I \to Z$ inductively as

$$\begin{aligned} \overline{0} &= \varepsilon_I; \infty \\ \overline{n+1} &= \gamma(\overline{n}); \sigma \end{aligned}$$

The underlying idea of the following lemma stems from [Fio94], where it is used to show that ∞ is greater that or equal to $\underline{n}$ with respect to the *path preorder*.

Lemma F.34 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. The diagram*

$$\begin{tikzcd} I \arrow[d, "\gamma(Id)"'] \arrow[dr, "\infty"] & \\ !I \arrow[r, "\overline{n}"] & Z \\ I \arrow[u, "\gamma(\perp)"] \arrow[ur, "\underline{n}"'] & \end{tikzcd}$$

commutes for any number n.

Proof. Induction in n. The case $n = 0$ is clearly true. Assume the lemma is true for n, then commutativity of

$$\begin{tikzcd}
I \arrow[r, "Id"] \arrow[d, "\gamma(Id)"'] \arrow[dr, "\gamma(\infty)"] & I \arrow[dr, "\infty"] & \\
!I \arrow[r, "\gamma(\overline{n})"] & !Z \arrow[r, "\sigma"] & Z \\
I \arrow[u, "\gamma(\bot)"] \arrow[ur, "\gamma(\underline{n})"'] & &
\end{tikzcd}$$

shows that the lemma is true for $n + 1$. $\Box$

Lemma F.35 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon^{\sharp}_{\Sigma}$. Assume Axiom 1 is satisfied. The map ∞ is an upper bound for the increasing chain $\{\underline{n} | n \in \omega\}$ in $\mathcal{C}(I, Z)$ with respect to the intrinsic preorder.*

Proof. First note that the chain is increasing because $\underline{0} \lesssim \underline{1}$ and monotenicity of $\gamma(-); \sigma$. We have $\gamma(\bot) \lesssim \gamma(Id)$ because $\bot \lesssim Id$ and monotenicity of γ; this entails that $\underline{n} \lesssim \infty$ cf. Lemma F.34. $\Box$

Definition F.36 *Let $\mathcal{C}$ be a linear category with a linear fixpoint operator. Given a map $f : !A \to A$, we define a map $f^n : I \to A$ inductively as*

$$\begin{aligned} f^0 &= \bot_{I,A} \\ f^{n+1} &= \gamma(f^n); f \end{aligned}$$

Lemma F.37 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Given a map $f : !A \to A$, we have $f^n = \underline{n}; It(f)$.*

Proof. Induction in n. The case $n = 0$ is clearly true. Assume the lemma is true for n, then

$$\begin{aligned} \underline{n+1}; It(f) &= \gamma(\underline{n}); \sigma; It(f) \\ &= \gamma(\underline{n}); !It(f); f \\ &= \gamma(\underline{n}; It(f)); f \\ &= \gamma(f^n); f \\ &= f^{n+1} \end{aligned}$$

shows that the lemma is true for $n + 1$. $\Box$

Theorem F.38 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon_\Sigma^\sharp$. Assume Axiom 1 is satisfied. Given maps $f : !A \to A$ and $g : !A \to B$, then $\gamma(f^\sharp); g$ is an upper bound for the increasing chain $\{\gamma(f^n); g | n \in \omega\}$ in $\mathcal{C}(I, B)$ with respect to the intrinsic preorder.*

Proof. First note that the chain is increasing because $f^n = \underline{n}; It(f)$ and the observation that $\gamma((-); It(f)); g$ is monotone. The same observation makes $\gamma(f^\sharp); g$ an upper bound because $f^\sharp = \infty; It(f)$. $\Box$

Corollary F.39 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon_\Sigma^\sharp$. Assume Axiom 1 is satisfied. Given maps $f : !A \to A$ and $g : A \to B$, then $f^\sharp; g$ is an upper bound for the increasing chain $\{f^n; g | n \in \omega\}$ in $\mathcal{C}(I, B)$ with respect to the intrinsic preorder.*

Proof. Immediate from Theorem F.38. $\Box$

Definition F.40 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given.* Axiom 2 *is satisfied iff for any $h : Z \to \Sigma$ with the property that $\infty; h = \top$ there is a number n such that $\underline{n}; h = \top$.*

It should be remarked that the category of CPOs and strict continuous functions as well as the category of dI domains and linear stable functions satisfies Axiom 2.

Theorem F.41 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon_\Sigma^\sharp$. Assume Axiom 1 and Axiom 2 are satisfied. Given maps $f : !A \to A$ and $g : !A \to B$, then $\gamma(f^\sharp); g$ is a least upper bound for the increasing chain $\{\gamma(f^n); g | n \in \omega\}$ in $\mathcal{C}(I, B)$ with respect to the intrinsic preorder.*

Proof. Theorem F.38 shows that $\gamma(f^\sharp); g$ is an upper bound. Now, let an arbitrary upper bound $k : I \to B$ and a map $h : B \to \Sigma$ such that $\gamma(f^\sharp); g; h = \top$ be given. We have

$$\begin{array}{rcll} \gamma(f^\sharp); g; h & = & \gamma(\infty); !It(f); g; h & \\ & = & \gamma(\infty); \sigma; \sigma^{-1}; !It(f); g; h & \quad \text{Lambek's Lemma} \\ & = & \infty; \sigma^{-1}; !It(f); g; h & \end{array}$$

which entails that there is a number n such that $\underline{n}; \sigma^{-1}; !It(f); g; h = \top$ cf. Axiom 2. The case $n = 0$ is precluded by the assumption that $\top \neq \varepsilon_\Sigma^\sharp$ and Proposition F.26, so we can assume that $n > 0$. We then have

$$\begin{array}{rcl} \underline{n}; \sigma^{-1}; !It(f); g; h & = & \gamma(\underline{n-1}); \sigma; \sigma^{-1}; !It(f); g; h \\ & = & \gamma(f^{n-1}); g; h \end{array}$$

so $k; h = \top$ because $\gamma(f^{n-1}); g \lesssim k$. We conclude that $\gamma(f^\sharp); g \lesssim k$. $\Box$

Corollary F.42 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon^{\sharp}_{\Sigma}$. Assume Axiom 1 and Axiom 2 are satisfied. Given maps $f :!A \to A$ and $g : A \to B$, then $f^{\sharp}; g$ is a least upper bound for the increasing chain $\{f^n; g | n \in \omega\}$ in $\mathcal{C}(I, B)$ with respect to the intrinsic preorder.*

Proof. Immediate from Theorem F.41. □

Definition F.43 *Let $\mathcal{C}$ be a linear category with a linear fixpoint operator. Given a map $f :!A \otimes !B \to B$, we define a map $f^n :!A \to B$ inductively as*

$$\begin{aligned} f^0 &= \perp_{!A,B} \\ f^{n+1} &= d; (Id \otimes \gamma(f^n)); f \end{aligned}$$

Theorem F.44 *Let $\mathcal{C}$ be a linear category with a linear fixpoint object. Assume that an object Σ and a map $\top : I \to \Sigma$ is given such that $\top \neq \varepsilon^{\sharp}_{\Sigma}$. Assume Axiom 1 and Axiom 2 are satisfied. Given maps $f :!A \otimes !B \to B$, $g :!C \to A$ and $h :!B \to D$, then $\gamma(g); \gamma(f^{\sharp}); h$ is a least upper bound for the increasing chain $\{\gamma(g); \gamma(f^n); h | n \in \omega\}$ in $\mathcal{C}(!C, D)$ with respect to the intrinsic preorder.*

Proof. Follows from Theorem F.41.

□

Now, the preorder-enrichment on $\mathcal{C}$ induces a preorder-enrichment on the coKleisli category, and it is straightforward to check that Theorem F.44 entails that the quotient of coKleisli category with respect to the preorder is rational.

G Appendix, Remarks on Recursion at the Level of Types

The λ-calculus can be extended with recursion at the level of types, that is, we have additional types $X \mid \mu X.s$, where X is a variable that ranges over types, additional terms

$$\mathrm{abs}(t) \mid \mathrm{rep}(t)$$

and the following typing rules in natural deduction style

$$\frac{\Gamma \vdash t : A[\mu X.A/X]}{\Gamma \vdash \mathrm{abs}(t) : \mu X.A}\,(\mu_I) \qquad \frac{\Gamma \vdash t : \mu X.A}{\Gamma \vdash \mathrm{rep}(t) : A[\mu X.A/X]}\,(\mu_E)$$

The above mentioned terms and typing rules can also be found in [Win93]. The usual reduction rules for terms of the λ-calculus can be extended with a reduction rule for the term corresponding to recursion at the level of types:

$$\mathrm{rep}(\mathrm{abs}(t)) \ \leadsto \ t$$

We are then able to define recursion at the level of terms in the λ^{rec}-calculus, as given in the present paper. Given a derivation

$$\Gamma, z : B \vdash u : B$$

we let C be an abbreviation for the type $\mu X.(X \Rightarrow B)$, and

$$\Gamma \vdash \alpha : C$$

an abbreviation for the derivation

$$\dfrac{\dfrac{\dfrac{\Gamma, z : B \vdash u : B}{\Gamma \vdash \lambda z.u : B \Rightarrow B} \qquad \dfrac{\dfrac{\dfrac{}{x : C \vdash x : C}}{x : C \vdash \mathrm{rep}(x) : C \Rightarrow B} \qquad \dfrac{}{x : C \vdash x : C}}{x : C \vdash \mathrm{rep}(x)x : B}}{\dfrac{\Gamma, x : C \vdash (\lambda z.u)(\mathrm{rep}(x)x) : B}{\Gamma \vdash \lambda x.((\lambda z.u)(\mathrm{rep}(x)x)) : C \Rightarrow B}}\,*}{\Gamma \vdash \mathrm{abs}(\lambda x.((\lambda z.u)(\mathrm{rep}(x)x))) : C}$$

* We have here used a multiplicative version of the $(\Rightarrow_E)$ rule, which is admissible in the λ-calculus. We define

$$\Gamma \vdash \mathrm{rec} z.u : B$$

to be an abbreviation for the derivation

$$\dfrac{\dfrac{\Gamma \vdash \alpha : C}{\Gamma \vdash \mathrm{rep}(\alpha) : C \Rightarrow B} \qquad \Gamma \vdash \alpha : C}{\Gamma \vdash \mathrm{rep}(\alpha)\alpha : B}$$

and we get the reduction rule for recursion at the level of terms in the λ^{rec}-calculus as follows:

$$\begin{array}{rcl} \mathrm{rec}z.u & = & \mathrm{rep}(\alpha)\alpha \\ & = & \mathrm{rep}(\mathrm{abs}(\lambda x.((\lambda z.u)(\mathrm{rep}(x)x))))\alpha \\ & \rightsquigarrow & (\lambda x.((\lambda z.u)(\mathrm{rep}(x)x)))\alpha \\ & \rightsquigarrow & (\lambda z.u)(\mathrm{rep}(\alpha)\alpha) \\ & \rightsquigarrow & u[(\mathrm{rep}(\alpha)\alpha)/z] \\ & = & u[\mathrm{rec}z.u/z] \end{array}$$

by using a lazy reduction strategy. This is essentially a syntactic restatement of a result in [Law69] saying that if $f : C \to C \Rightarrow B$ is a weakly point-surjective map in a cartesian closed category, then every endomap on B has a fixpoint.

Now, we want to have recursion at the level of types in the linear λ-calculus. Certain models of ILL have the property that they admit solution of recursive domain equations corresponding to type constructs. The category of CPOs ond strict continuous functions, and the category of dI domains and join preserving stable functions both have that property. Another example is the category of games given in [AM94]. The fact that one can solve recursive domain equations corresponding to type constructs in such models, suggests that we should take recursion at the level of types in the linear λ-calculus to have the same typing and reduction rules as in the λ-calculus case. Note that recursion at the level of types takes place in the full ILL, and not only in the IL fragment of ILL, as it is the case with recursion at the level of terms. The Girard Translation still works when IL and ILL both are extended with recursion at the level of types, and moreover, the translation preserves the reduction rule for recursion.

We are able to define recursion at the level of terms in the linear λ^{rec}-calculus, as given in the present paper, when we take recursion at the level of types to have the same typing and reduction rules as in the λ-calculus case. Given a derivation

$$\overline{x :!A}, z :!B \vdash u : B$$

where $\overline{x :!A}$ is an abbreviation for $x_1 :!A_1, ..., x_n :!A_n$, we let C be an abbreviation for the type $\mu X.(!X \multimap B)$, and

$$\overline{x :!A} \vdash \alpha : C$$

an abbreviation for the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{x :!A}, z :!B \vdash u : B}{\overline{x :!A} \vdash \lambda z.u :!B \multimap B} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x :!C \vdash x :!C}}{x :!C \vdash \mathrm{derelict}(x) : C}}{x :!C \vdash \mathrm{rep}(\mathrm{derelict}(x)) :!C \multimap B} \qquad \dfrac{}{y :!C \vdash y :!C}}{x :!C, y :!C \vdash \mathrm{rep}(\mathrm{derelict}(x))y : B}}{x' :!C \vdash \mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y) : B}}{x'' :!C \vdash \mathrm{let}\ x''\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y)) :!B}}{\overline{x :!A}, x'' :!C \vdash (\lambda z.u)\mathrm{let}\ x''\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y)) : B}}{\overline{x :!A} \vdash \lambda x''.((\lambda z.u)\mathrm{let}\ x''\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y))) :!C \multimap B}}{\overline{x :!A} \vdash \mathrm{abs}(\lambda x''.((\lambda z.u)\mathrm{let}\ x''\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y)))) : C}$$

Now, we define

$$\overline{y'' :!A} \vdash \mathrm{let}\ \overline{y''}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ \mathrm{rec}z.u : B$$

to be an abbreviation for the derivation

$$\dfrac{\dfrac{\dfrac{\overline{x :!A} \vdash \alpha : C}{\overline{x :!A} \vdash \mathrm{rep}(\alpha) :!C \multimap B} \qquad \dfrac{\overline{x :!A} \vdash \alpha : C}{\overline{y' :!A} \vdash \mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha :!C}}{\overline{x :!A}, \overline{y' :!A} \vdash \mathrm{rep}(\alpha)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha : B}}{\overline{y'' :!A} \vdash \mathrm{copy}\ \overline{y''}\ \mathrm{as}\ \overline{x}, \overline{y'}\ \mathrm{in}\ (\mathrm{rep}(\alpha)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha) : B}$$

and we have

$$\begin{array}{l}
\mathrm{rep}(\alpha)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha \ = \\
\mathrm{rep}(\mathrm{abs}(\lambda x''.((\lambda z.u)\mathrm{let}\ x''\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y)))))\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha \ = \\
(\lambda x''.((\lambda z.u)\mathrm{let}\ x''\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y))))\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha \ \leadsto \\
(\lambda z.u)\mathrm{let}\ (\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha)\ \mathrm{be}\ x'\ \mathrm{in}\ !(\mathrm{copy}\ x'\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y)) \ \leadsto \\
(\lambda z.u)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{y''}\ \mathrm{in}\ !(\mathrm{copy}\ (\mathrm{let}\ \overline{y''}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha)\ \mathrm{as}\ x, y\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(x))y)) \ \leadsto \\
(\lambda z.u)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{y''}\ \mathrm{in}\ !(\mathrm{copy}\ \overline{y''}\ \mathrm{as}\ \overline{x}, \overline{y'}\ \mathrm{in}\ (\mathrm{rep}(\mathrm{derelict}(\mathrm{let}\ \overline{x}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha))\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha)) \ \leadsto \\
(\lambda z.u)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{y''}\ \mathrm{in}\ !(\mathrm{copy}\ \overline{y''}\ \mathrm{as}\ \overline{x}, \overline{y'}\ \mathrm{in}\ (\mathrm{rep}(\alpha)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha)) \ \leadsto \\
u[\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{y''}\ \mathrm{in}\ !(\mathrm{copy}\ \overline{y''}\ \mathrm{as}\ \overline{x}, \overline{y'}\ \mathrm{in}\ (\mathrm{rep}(\alpha)\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ !\alpha))/z]
\end{array}$$

which entails that we get the reduction rule for recursion at the level of terms in the linear λ^{rec}-calculus:

$$\mathrm{let}\ \overline{y''}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ \mathrm{rec}z.u \ \leadsto\ \mathrm{copy}\ \overline{y''}\ \mathrm{as}\ \overline{x}, \overline{y'}\ \mathrm{in}\ (u[\mathrm{let}\ \overline{y'}\ \mathrm{be}\ \overline{y''}\ \mathrm{in}\ !(\mathrm{let}\ \overline{y''}\ \mathrm{be}\ \overline{x}\ \mathrm{in}\ \mathrm{rec}z.u)/z])$$

One might note that if we extend the !-free fragment of the linear λ-calculus with the above mentioned typing and reduction rules for recursion, then we get a system which is strongly normalising because the underlying proof of a term shrinks during each reduction step (so reduction will actually be linear in the size of the underlying proof).

Recent Publications in the BRICS Report Series

RS-95-13 Torben Braüner. *The Girard Translation Extended with Recursion*. February 1995. 79 pp. Full version of paper to appear in Proceedings of CSL '94, LNCS 933, 1995.

RS-95-12 Gerth Stølting Brodal. *Fast Meldable Priority Queues*. February 1995. 12 pp.

RS-95-11 Alberto Apostolico and Dany Breslauer. *An Optimal $O(\log\log n)$ Time Parallel Algorithm for Detecting all Squares in a String*. February 1995. 18 pp. To appear in SIAM Journal on Computing.

RS-95-10 Dany Breslauer and Devdatt P. Dubhashi. *Transforming Comparison Model Lower Bounds to the Parallel-Random-Access-Machine*. February 1995. 11 pp.

RS-95-9 Lars R. Knudsen. *Partial and Higher Order Differentials and Applications to the DES*. February 1995. 24 pp.

RS-95-8 Ole I. Hougaard, Michael I. Schwartzbach, and Hosein Askari. *Type Inference of Turbo Pascal*. February 1995. 19 pp.

RS-95-7 David A. Basin and Nils Klarlund. *Hardware Verification using Monadic Second-Order Logic*. January 1995. 13 pp.

RS-95-6 Igor Walukiewicz. *A Complete Deductive System for the μ-Calculus*. January 1995. 39 pp.

RS-95-5 Luca Aceto and Anna Ingólfsdóttir. *A Complete Equational Axiomatization for Prefix Iteration with Silent Steps*. January 1995. 27 pp.

RS-95-4 Mogens Nielsen and Glynn Winskel. *Petri Nets and Bisimulations*. January 1995. 36 pp. To appear in TCS.

RS-95-3 Anna Ingólfsdóttir. *A Semantic Theory for Value–Passing Processes, Late Approach, Part I: A Denotational Model and Its Complete Axiomatization*. January 1995. 37 pp.

RS-95-2 François Laroussinie, Kim G. Larsen, and Carsten Weise. *From Timed Automata to Logic - and Back*. January 1995. 21 pp.