Appeared in: *Logic from Computer Science*, MSRI Publications,
Y. Moschovakis, ed., Vol. 21, Springer-Verlag, 1992, pp. 217-241.

Normal Forms and Cut-Free Proofs as Natural Transformations

Jean-Yves Girard Andre Scedrov * Philip J. Scott†

Dedicated to the Memory of E.S. Bainbridge

Abstract

What equations can we *guarantee* that simple functional programs must satisfy, irrespective of their obvious defining equations? Equivalently, what non-trivial identifications must hold between lambda terms, thought-of as encoding appropriate natural deduction proofs ? We show that the usual syntax guarantees that certain naturality equations from category theory are necessarily provable. At the same time, our categorical approach addresses an equational meaning of cut-elimination and asymmetrical interpretations of cut-free proofs. This viewpoint is connected to Reynolds' relational interpretation of parametricity ([27], [2]), and to the Kelly-Lambek-Mac Lane-Mints approach to coherence problems in category theory.

1 Introduction

In the past several years, there has been renewed interest and research into the interconnections of proof theory, typed lambda calculus (as a functional programming paradigm) and category theory. Some of these connections can be surprisingly subtle. Here we address some relationships of three fundamental notions:

*Research supported by ONR contract N00014-88K0635, NSF Grant CCR-87-05596, and a Young Faculty Award of the Natural Sciences Association of the University of Pennsylvania. Scedrov would like to thank John C. Mitchell, the Computer Science Department, and the Center for the Study of Language and Information at Stanford University for their hospitality during his Sabbatical leave.

†Research supported by an operating grant from the Natural Sciences and Engineering Research Council of Canada. We thank Narciso Martí-Oliet and Robert Seely for helpful comments on a preliminary version of this work.

- Cut-elimination from proof theory.
- Natural Transformations from category theory.
- Parametricity from the foundations of polymorphism.

Familiar work of Curry, Howard, Lambek and others [12, 15, 17] has shown how we may consider constructive proofs as programs. For example, Gentzen's intuitionistic sequents $A_1, \ldots, A_k \vdash B$ may be interpreted as functional programs mapping k inputs of types A_i, $1 \leq i \leq k$, to outputs of type B. More precisely, proofs are interpreted as certain terms of the typed lambda calculus.

The cut rule

$$\frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}$$

has a special status; the cut formula A appears simultaneously covariantly and contravariantly (i.e. to the right and the left of $\vdash$, resp.). In the functional formalism, cut corresponds to *composition*, i.e. substitution or "plugging together" of one program into another. One meaning of cut-elimination is that general substitution is already definable from the special instances implicit in the other Gentzen rules.

In this paper we show how to associate to natural deduction proofs in normal form certain families of normal lambda terms indexed by types. These families necessarily satisfy certain naturality conditions from category theory in any cartesian closed category. In the case of the syntax, we obtain as corollary that appropriate lambda terms *provably* satisfy, in addition to the usual beta eta equations, also these new equations.

We begin with two examples from simply typed lambda calculus. Recall, the types are built inductively from a countably infinite set of ground types (which can be considered as type variables) by the two type constructors $\Rightarrow$ and $\times$.

Example 1.1 Let r be any closed simple typed lambda term of type

$$\alpha \times \alpha \Rightarrow \alpha \times \alpha,$$

where α is a type variable. What can we say about such r in general? We shall show that for any two types A and B and any closed term f of type $A \Rightarrow B$ it must be the case that:

$$(1) \qquad (f \times f) \circ r_A = r_B \circ (f \times f)$$

where r_A and r_B are instances of r, where $=$ is beta-eta conversion, and where $g \circ f$ denotes function composition.

The equation can be restated in cartesian closed categories (= ccc's). Recall that simply typed lambda calculi correspond to ccc's and that in any ccc morphisms $A \to B$ uniquely correspond to morphisms $1 \to A \Rightarrow B$ [23, 19] . We shall blur this latter distinction and abuse the notation accordingly. In any ccc, then, the above equation says that for any morphism $f : A \to B$ the following diagram commutes.

$$\begin{tikzcd}
A \times A \arrow[r, "r_A"] \arrow[d, "f \times f"'] & A \times A \arrow[d, "f \times f"] \\
B \times B \arrow[r, "r_B"'] & B \times B
\end{tikzcd}$$

This means that the instantiations of r at all values of type variable α form a natural transformation from the covariant endofunctor $F(A) = A \times A$ to itself.

In fact, in the example above there is a functional dependency on the map f, this dependency itself being natural. This is further explored in the following example.

Example 1.2 Let m be any closed simply typed lambda term of type

$$(\alpha \Rightarrow \beta) \Rightarrow (\alpha \times \alpha \Rightarrow \beta \times \beta),$$

where α and β are type variables. What can m possibly be? A first guess is that , for any argument f of appropriate type, $m(f) = f \times f$. We shall confirm this intuition by showing that for any two types A, B, and any closed term f of type $A \Rightarrow B$ it is the case that:

$$(2) \qquad m_{AB}(f) = (f \times f) \circ m_{AA}(1_A) = m_{BB}(1_B) \circ (f \times f)$$

in the sense of beta-eta conversion. In otherwords, in any ccc, the equation holds for any morphism $f : A \longrightarrow B$. Thus $m_{AB}(f)$ "is" $f \times f$, up to composition with a distinguished endomorphism of the domain or codomain.

In the next section, these examples will be seen in their "natural" setting of multivariant functors and multivariant (dinatural) natural transformations.

Notation: In what follows, $\mathcal{C}$ denotes a category (usually cartesian closed). Objects of $(\mathcal{C}^\circ)^n \times \mathcal{C}^n$ are denoted $(\mathbf{A}; \mathbf{B})$, where $\mathbf{A}$, $\mathbf{B}$ are objects of $\mathcal{C}^n$

and semicolon separates the contravariant and covariant arguments. For notational ease, we often write the value of a functor $F : (\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$ at objects $(\mathbf{A};\mathbf{B})$ by $F\mathbf{AB}$ rather than $F(\mathbf{A};\mathbf{B})$.

For much of this paper, we purposely abuse notation between cartesian closed categories (= ccc's) and typed lambda calculus and blur, for example, the distinction (in the associated typed lambda calculus) between closed terms $m : A \Rightarrow B$, open terms $m(x) : B$ containing at most one free variable $x : A$, and arrows $m : A \longrightarrow B$, ([19], pp. 75-78, and [23]). Substitution vs composition will be equally abused. Readers may easily restate results in the notation of their choice.

2 The Dinatural Calculus

The basic view that led to the work on *functorial polymorphism* in [2] is that we may interpret polymorphic type expressions $\sigma(\alpha_1, \ldots, \alpha_n)$, with type variables α_i, as certain kinds of multivariant "definable" functors

$$F : (\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$$

over an appropriate ccc $\mathcal{C}$; moreover, terms $t(x)$ of type τ, with variable x of type σ , are then interpreted as certain multivariant (= *dinatural*) transformations between (the interpretations of) the types. The resultant calculus has many interesting properties [2, 7], despite the fact that in general dinatural transformations do not compose. Previous studies [2, 3, 8] emphasized semantical (compositional) models for this calculus; in this paper we prove that the pure (cut-free) syntax itself admits a compositional dinatural interpretation.

We briefly review this functorial interpretation, following [2], Section 3. In this paper we emphasize the ccc connectives $\times$ and $\Rightarrow$; hence we omit the functorial interpretation of universal type quantification $\forall\alpha$, emphasized in [2]. In fact, we can now interpret the syntax over an *arbitrary* ccc, as we shall see.

Let $\mathcal{C}$ be any category, and $n \geq 1$ an integer.

Definition 2.1 A *dinatural transformation* between functors $F, G : (\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$ is a family of morphisms $\theta = \{\theta_{\mathbf{A}} : F\mathbf{AA} \longrightarrow G\mathbf{AA} \mid \mathbf{A} \in \mathcal{C}^n\}$ satisfying the following: for any vector of arrows $\mathbf{f} : \mathbf{A} \longrightarrow \mathbf{B}$, the following diagram commutes:

$$
\begin{tikzcd}
 & F\mathbf{AA} \arrow[rr, "\theta\mathbf{A}"] & & G\mathbf{AA} \arrow[dr, "G\mathbf{Af}"] & \\
F\mathbf{BA} \arrow[ur, "F\mathbf{fA}"] \arrow[dr, "F\mathbf{Bf}"'] & & & & G\mathbf{AB} \\
 & F\mathbf{BB} \arrow[rr, "\theta\mathbf{B}"'] & & G\mathbf{BB} \arrow[ur, "G\mathbf{fB}"'] &
\end{tikzcd}
$$

We write $\theta : F \longrightarrow G$ to denote that θ is a dinatural transformation from F to G, as above. Many examples of such transformations arise in practice, cf. [2, 20]; in particular, among other things, dinatural transformations include ordinary natural transformations as a special case. It is useful to note that the definition of dinaturality is an *equational* condition on the family θ in the category $\mathcal{C}$. Moreover, dinaturality is, on the face of it, a *semantical* condition: it must be verified for all objects $\mathbf{A}, \mathbf{B} \in \mathcal{C}^n$ and all morphisms $\mathbf{f} : \mathbf{A} \longrightarrow \mathbf{B}$.

Consider a simply typed lambda calculus whose types are freely generated from appropriate ground constants, including type variables, and whose terms are freely generated from infinitely many individual variables by the usual term-formation rules (appropriate to the ground types) [19, 12]. We impose familiar lambda calculus equations, including beta and eta, and product types with surjective pairing [1].

Let $\mathcal{C}$ be any ccc. For each type $\sigma(\alpha_1, \ldots, \alpha_n)$, with type variables α_i, we define its interpretation $\|\sigma\| : (\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$ by induction, as follows:

1. If $\sigma = \alpha_i$, $\|\sigma\|(\mathbf{A};\mathbf{B}) = B_i$, the projection onto the ith component of $\mathbf{B}$.

2. If σ = C, a ground constant interpreted as an object of $\mathcal{C}$, then $\|\sigma\|(\mathbf{A};\mathbf{B}) = K_C$, the constant functor with value C.

3. (Product) If $\sigma = \tau_1 \times \tau_2$, then $\|\sigma\|(\mathbf{A};\mathbf{B}) = \|\tau_1\|(\mathbf{A};\mathbf{B}) \times \|\tau_2\|(\mathbf{A};\mathbf{B})$.

4. (Twisted Exponential) If $\sigma = \tau_1 \Rightarrow \tau_2$ then $\|\sigma\|(\mathbf{A};\mathbf{B}) = \|\tau_1\|(\mathbf{B};\mathbf{A}) \Rightarrow \|\tau_2\|(\mathbf{A};\mathbf{B})$.

Having interpreted types as certain multivariant functors over an arbitrary ccc $\mathcal{C}$ we now sketch how to view the usual interpretation of typed lambda terms in ccc's as dinatural transformations.

[1] This theory is actually a fragment of second order lambda calculus (e.g. [12]) in which types are (implicitly universally quantified) schema, closed under type substitution. The language is closely related to the "Core-ML" language of implicit polymorphism [24].

Consider a legal typing judgement

$$x_1 : \sigma_1, \ldots, x_k : \sigma_k \rhd t : \tau \quad .$$

Let $\mathcal{C}$ be any ccc. As we've seen above, any assignment of all the type variables $\alpha_1, \ldots, \alpha_n$ (in the typing judgement) to objects of $\mathcal{C}$ induces functors $\|\sigma_i\|, \|\tau\| : (\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$. We now show that the familiar interpretation of typed lambda terms in ccc's induces a family by substituting objects for type variables. Hence every term t induces a family $\|t\|_{\mathbf{A}} : (\|\sigma_1\| \times \cdots \times \|\sigma_k\|)\ \mathbf{AA} \longrightarrow \|\tau\|\mathbf{AA}$, where $\mathbf{A} = (A_1, \ldots, A_n)$ and $A_i = \|\alpha_i\|$, as follows:

1. If $t : \sigma_i$ is the variable x_i, then $\|t\| : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \longrightarrow \|\sigma_i\|$ is the ith projection.

2. If $t : \sigma \Rightarrow \tau$ is $\lambda x : \sigma . r$, where $\|r\| : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \times \|\sigma\| \longrightarrow \|\tau\|$, then $\|t\| : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \longrightarrow \|\sigma \Rightarrow \tau\|$ is defined by: $\|t\|\mathbf{A} = (\|r\|\mathbf{A})^*$, the curryfication (or exponential transpose) of the arrow $\|r\|\mathbf{A} : \|\sigma_1\|\mathbf{AA} \times \cdots \times \|\sigma_k\|\mathbf{AA} \times \|\sigma\|\mathbf{AA} \longrightarrow \|\tau\|\ \mathbf{AA}$ in the ccc $\mathcal{C}$ (cf. [19], p.61).

3. If $t : \tau$ is ua, where $u : \sigma \Rightarrow \tau$ and $a : \sigma$, then $\|t\|$ is $ev \circ < \|u\|, \|a\| >$, where ev denotes the appropriate evaluation map and $\circ$ denotes composition in $\mathcal{C}$.

4. If $t : \tau_1 \times \tau_2$ is $t =< t_1, t_2 >$ and $\|t_i\| : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \longrightarrow \|\tau_i\|$ for $i = 1, 2$, then $\|t\|\mathbf{A} =< \|t_1\|\mathbf{A}, \|t_2\|\mathbf{A} >$.

5. If $t : \tau_i$ is $\Pi_i(t')$ ($i = 1, 2$) where $\|t'\| : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \longrightarrow \| \tau_1 \times \tau_2 \|$ then $\|\Pi_i(t')\|$ is $\Pi_i \circ \|t'\|$, where Π_i is the ith projection.

One of our main points is that the above family $\|t\| = \{\|t\|_{\mathbf{A}} \mid \mathbf{A} \in \mathcal{C}^n\} : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \longrightarrow \|\tau\|$,which arises from a single term by plugging in objects as values for type variables, is actually dinatural; moreover, all compositions of dinaturals in the interpretation are well-defined.

We summarize the results in the next two theorems, whose proof we shall discuss later.

Theorem 2.2 *Let $\mathcal{L}$ be simply typed lambda calculus with type variables, let $\mathcal{C}$ be any ccc , and let $\alpha_1, \ldots, \alpha_n$ be a list of type variables. Then any type $\tau(\alpha_1, \ldots, \alpha_n)$ with the indicated type variables induces a functor $\|\tau\| : (\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$. Furthermore, any legal typing judgement*

$$x_1 : \sigma_1, \ldots, x_k : \sigma_k \rhd t : \tau$$

induces a family of lambda terms $\|t\| = \{\|t\|_{\mathbf{A}} | \mathbf{A} \in \mathcal{C}^n\}$ *which is actually a dinatural transformation* $\|t\| : \|\sigma_1\| \times \cdots \times \|\sigma_k\| \longrightarrow \|\tau\|$. *Beta-eta convertible lambda terms give the same dinatural family.*

We shall call the multivariant functors and syntactic families in the above theorem *definable* functors and dinatural transformations, respectively.

Theorem 2.3 *Let* $\mathcal{C}$ *be any ccc. Then for each* n*, the definable functors* $(\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$ *and definable dinatural transformations between them form a ccc. The cartesian closed structure on definable functors is given by products and twisted exponentials, as in the discussion after Definition* **2.1**.[2]

The nontrivial parts of the last two theorems are that syntactic families induced by terms actually yield dinatural transformations, and moreover that these definable dinatural transformations compose, to yield a category (cf. [2]). The proof proceeds through a rather subtle detour involving Gentzen's sequents and cut-elimination, which we begin in the next section. We also remark that a related semantical theorem (Theorem 2) in [8] is now only known to apply to the core-ML-fragment of polymorphic lambda calculus. The full second order version claimed in that paper is still open.

There is an immediate corollary of the above theorems. If we specialize $\mathcal{C}$ to be the ccc freely generated by $\mathcal{L}$, that is, the syntactical term model of typed lambda calculus with type variables, qua ccc (cf. [19],p. 77) then "equality" means "provable equality" in $\mathcal{L}$. In this setting, it follows that the syntactic families $\|t\|$ above are provably dinatural. Thus, at the linguisitic level, *arbitrary terms* t*, when validly typed, provably satisfy appropriate dinaturality equations in addition to any equations of the typed lambda calculus.*

For the remainder of this section, we reexamine and generalize Example 1.2 from the Introduction, in the light of the discussion above. Recall, from the typing of m we obtain the typing judgement

$$x : \alpha \Rightarrow \beta \ \triangleright \ m(x) : \alpha \times \alpha \Rightarrow \beta \times \beta \ .$$

Thus, in any ccc $\mathcal{C}$, the term m is interpreted as a definable dinatural transformation $F \longrightarrow G$ between certain functors $F, G : (\mathcal{C}^\circ)^2 \times \mathcal{C}^2 \longrightarrow \mathcal{C}$ determined by the types. Indeed, from the discussion after **2.1**, F and G are given by $F = \|\alpha \Rightarrow \beta\|$ and $G = \|\alpha \times \alpha \Rightarrow \beta \times \beta\|$. It may be shown [3] that $F(A, B; A', B') = A \Rightarrow B'$, while $G(A, B; A', B') = A \times A \Rightarrow B' \times B'$,

[2]Indeed, we actually obtain a fibred ccc or hyperdoctrine, ignoring the indexed adjoints [25, 28]

[3]The type variables α and β correspond to covariant projections $P_i : (\mathcal{C}^\circ)^2 \times \mathcal{C}^2 \longrightarrow \mathcal{C}$ where $P_i(\mathbf{C};\mathbf{D}) = D_i, i = 1, 2\,(resp.)$. Thus $F(\mathbf{C};\mathbf{D}) = P_1(\mathbf{D};\mathbf{C}) \Rightarrow P_2(\mathbf{C};\mathbf{D})$, and $G(\mathbf{C};\mathbf{D}) = (P_1 \times P_1)(\mathbf{D};\mathbf{C}) \Rightarrow (P_2 \times P_2)(\mathbf{C};\mathbf{D})$.

where $(A, B; A', B')$ denotes an object in $(\mathcal{C}^\circ)^2 \times \mathcal{C}^2$. The dinaturality of $m(= m(x))$ says that for arbitrary morphisms $a : A \longrightarrow A'$ and $b : B \longrightarrow B'$ in $\mathcal{C}$, the following hexagon commutes:

$$\begin{tikzcd}
 & A \Rightarrow B \arrow[r, "m_{AB}"] & A \times A \Rightarrow B \times B \arrow[dr, "(b \times b)\circ_"] & \\
A' \Rightarrow B \arrow[ur, "_\circ a"] \arrow[dr, "b\circ_"'] & & & A \times A \Rightarrow B' \times B' \\
 & A' \Rightarrow B' \arrow[r, "m_{A'B'}"'] & A' \times A' \Rightarrow B' \times B' \arrow[ur, "_\circ(a \times a)"'] &
\end{tikzcd}$$

Figure 1

where $f\circ\_$ denotes the map $h \mapsto f\circ h$.

Letting $A' = B' = B, b = 1_B$, and $a = f$ in Figure 1, we obtain

$$\begin{tikzcd}
 & A \Rightarrow B \arrow[r, "m_{AB}"] & A \times A \Rightarrow B \times B \arrow[dr, "(1_B \times 1_B)\circ_"] & \\
B \Rightarrow B \arrow[ur, "_\circ f"] \arrow[dr, "1_B\circ_"'] & & & A \times A \Rightarrow B \times B \\
 & B \Rightarrow B \arrow[r, "m_{BB}"'] & B \times B \Rightarrow B \times B \arrow[ur, "_\circ(f \times f)"'] &
\end{tikzcd}$$

Equivalently, we obtain the following commutative square:

$$\begin{tikzcd}
A \Rightarrow B \arrow[r, "m_{AB}"] & A \times A \Rightarrow B \times B \\
B \Rightarrow B \arrow[u, "_\circ f"] \arrow[r, "m_{BB}"'] & B \times B \Rightarrow B \times B \arrow[u, "_\circ(f \times f)"']
\end{tikzcd}$$

This says that for any $e : B \longrightarrow B$ it is the case that

(†) $$m_{AB}(e\circ f) = m_{BB}(e)\circ(f \times f) \ ,$$

and so in particular for $e = 1_B$:

(2a) $$m_{AB}(f) = m_{BB}(1_B)\circ(f \times f)$$

for any $f : A \longrightarrow B$.

A "dual" instance of the above equation is obtained in a similar manner by specializing in figure 1: $A' = B = A$ and $a = 1_A$, and then switching notation, by letting $B' = B$ and $b = f$. We obtain the following commutative square:

$$\begin{tikzcd}
A \Rightarrow A \arrow[r, "m_{AA}"] \arrow[d, "f\circ_"'] & A \times A \Rightarrow A \times A \arrow[d, "(f\times f)\circ_"] \\
A \Rightarrow B \arrow[r, "m_{AB}"'] & A \times A \Rightarrow B \times B
\end{tikzcd}$$

This says that for any $u : A \longrightarrow A$ it is the case that

$$(\dagger\dagger) \qquad m_{AB}(f\circ u) = (f \times f)\circ m_{AA}(u) \ ,$$

In particular, letting $u = 1_A$, we obtain

$$(2b) \qquad m_{AB}(f) = (f \times f)\circ m_{AA}(1_A)$$

The equations (2a) and (2b), which together express equation 2 of Section 1, were first noted by Wadler [32].

Remark 2.4 The equations ($\dagger$) and ($\dagger\dagger$) are instances of natural transformations, as follows. From general considerations, dinatural transformations $m : F \longrightarrow G$, where $F, G : (\mathcal{C}^\circ) \times \mathcal{C} \longrightarrow \mathcal{C}$ essentially reduce to natural transformations if either the co- or the contravariant variable is dummy or a fixed constant. In the above discussion, consider the bifunctors $\| \alpha \Rightarrow \beta \|$ and $\| \alpha \times \alpha \Rightarrow \beta \times \beta \|$: $(\mathcal{C}^\circ)^2 \times \mathcal{C}^2 \longrightarrow \mathcal{C}$. Fix the covariant variable $\beta := B$ as a constant (so $\|\beta\| = K_B$). Then the induced composite functors $\|\alpha \Rightarrow B\|$ and $\|\alpha \times \alpha \Rightarrow B \times B\| : (\mathcal{C}^\circ) \times \mathcal{C} \longrightarrow \mathcal{C}$ may actually be considered as (inducing) contravariant functors $\mathcal{C}^\circ \longrightarrow \mathcal{C}$; the dinatural m then becomes a natural transformation $m_{\_B}$ between them. The commutative square above equation ($\dagger$) expresses the special case of naturality along morphisms $f : A \longrightarrow B$.

The dual equation ($\dagger\dagger$) is a special case of the fact that m also induces a natural transformation $m_{A\_}$ between $\|A \Rightarrow \beta\|$ and $\|A \times A \Rightarrow \beta \times \beta\|$, qua covariant functors $\mathcal{C} \longrightarrow \mathcal{C}$.

The above discussion centered on the type constructor $T(\alpha) = \alpha \ \times \ \alpha$, qua covariant definable functor $\mathcal{C} \longrightarrow \mathcal{C}$. For example, for this T, the just-mentioned dual equation (††) refers to a natural transformation $m_{A\_} : \|A \Rightarrow \beta\| \longrightarrow \ \| \, T(A) \ \Rightarrow \ T(\beta) \, \|$. We may generalize this discussion as follows.

Example 2.5 Consider m a closed simply typed lambda term, say

$$m : (\alpha \Rightarrow \beta) \Rightarrow (T(\alpha) \Rightarrow T(\beta)),$$

where T is any definable "pure" covariant functor (i.e. ignoring dummy arguments, consider T as simply a functor $\mathcal{C} \longrightarrow \mathcal{C}$). Then, according to Theorem 2.2 above, m yields a dinatural transformation from $(\,) \Rightarrow (\,)$ to $T(\,) \Rightarrow T(\,)$, considered as functors $(\mathcal{C}^{\circ})^2 \ \times \ \mathcal{C}^2 \ \longrightarrow \ \mathcal{C}$ as in the previous discussion. Following the Remark above, if we fix the contravariant argument of these bifunctors to be $A \in \mathcal{C}$, m induces a natural transformation $m_{A\_}$ from the usual covariant functor $A \Rightarrow (\,)$ to $T(A) \ \Rightarrow \ T\,(\quad)$, thought-of as functors $\mathcal{C} \longrightarrow \mathcal{C}$. Similarly, fixing the covariant argument to be B , m induces a natural transformation $m_{\_B}$ from the contravariant functor $(\,) \Rightarrow B$ to $T(\,) \Rightarrow T(B)$.

Thus, for any instantiations A of type variable α, B of type variable β, etc., and for all $f : B' \longrightarrow B$, the following square involving $m_{A\_}$ commutes :

$$\begin{tikzcd}
A \Rightarrow B' \arrow[r, "m_{AB'}"] \arrow[d, "f\,\circ_"'] & T(A) \Rightarrow T(B') \arrow[d, "T(f)\circ_"] \\
A \Rightarrow B \arrow[r, "m_{AB}"'] & T(A) \Rightarrow T(B)
\end{tikzcd}$$

i.e. for all $g : A \Rightarrow B'$, we have $T(f) \circ m_{AB'}(g) = m_{AB}(f \circ g)$. Specializing to the case where $B' = A$ and $g = 1_A$, we thus have that m satisfies the equation

$$m_{AB}(f) = T(f) \circ m_{AA}(1_A).$$

for all $f : A \longrightarrow B$.

It is left to the reader to dualize the above discussion, using $m_{\_B}$ to obtain that m also satisfies

$$m_{AB}(f) = m_{BB}(1_B) \circ T(f).$$

for any $f : A \longrightarrow B$.

The last example, which includes all the previous ones for appropriate choices of T, may also be applied to other useful functorial type constructors, provided the ccc's $\mathcal{C}$ have the appropriate structure. As an example, we could introduce (as a definable functor) the covariant functor $T(\alpha) = \alpha - list$, and $T(f) = map(f)$ (as first described by Strachey, cf. [27]).

Remark 2.6 In the examples above, the two equations for $m_{AB}(f)$ are special cases of dinaturality, which is an equational condition. They were first observed by Wadler [32] from other considerations based on a universal Horn condition (see below) . An interesting observation, due to Val Breazu-Tannen, is that these two special cases *actually imply* dinaturality. That is, following the discussion in the above example, if we have a family of morphisms

$$m = \{m_{AB} : F(AB; AB) \longrightarrow G(AB; AB) \quad | \ AB \in \mathcal{C}^2\}$$

where $F, G : (\mathcal{C}^\circ)^2 \times \mathcal{C}^2 \longrightarrow \mathcal{C}$ are functors given by $F(AB; A'B') = A \Rightarrow B'$, while $G(AB; A'B') = T(A) \Rightarrow T(B')$, and if $m_{AB}(f)$ satisfies the following two equations (for any f of type $F(AB; AB)$):

$$m_{AB}(f) = T(f) \circ m_{AA}(1_A) = m_{BB}(1_B) \circ T(f)$$

then m actually is a dinatural transformation; i.e. for all arrows $a : A \longrightarrow A'$ and $b : B \longrightarrow B'$ the following diagram commutes:

$$\begin{tikzcd}
 & A \Rightarrow B \arrow[r, "m_{AB}"] & T(A) \Rightarrow T(B) \arrow[dr, "T(b)\circ_"] & \\
A' \Rightarrow B \arrow[ur, "_\circ a"] \arrow[dr, "b\circ_"'] & & & T(A) \Rightarrow T(B') \\
 & A' \Rightarrow B' \arrow[r, "m_{A'B'}"'] & T(A') \Rightarrow T(B') \arrow[ur, "_\circ T(a)"'] &
\end{tikzcd}$$

Proof: $T(b)\circ m_{AB}(g\circ a) = T(b)\circ T(g\circ a)\circ m_{AA}(1_A) = T(b\circ g)\circ T(a)\circ m_{AA}(1_A) = T(b\circ g)\circ m_{AA'}(a) = T(b\circ g)\circ m_{A'A'}(1_{A'})\circ T(a) = m_{A'B'}(b\circ g)\circ T(a)$.

This exercise is just a special case of a theorem that the universal Horn condition considered in Wadler [32] implies dinaturality, for definable functors. This general result, along with the categorical framework for this Horn condition (given by the notion of relator) is discussed in Abramsky, et al [1].

3 Gentzen Sequents and Natural Deduction

In what follows, we sketch the translation of Gentzen sequents into natural deduction calculi [12, 31, 33, 6, 22] which motivate our presentation.

3.1 Natural Deduction and Sequent Calculus

A deduction $\mathcal{D}$ of formula A (to be defined below) is a certain type of finite tree whose vertices are labelled by formulae and whose root is labelled by A. We denote such deductions by

$$\begin{array}{c}\mathcal{D}\\ A\end{array}$$

Certain occurrences of formulae among the leaves of $\mathcal{D}$ will be called *open assumptions* ; other leaves will be called *discharged* or *cancelled* assumptions.

We partition occurrences of assumption formulas into "assumption classes" ([31], [33]) or "parcels of hypotheses" ([12]) , each class consisting of occurrences of a single formula. Note that a given formula may be in more than one class. We write

$$\begin{array}{c}[A_1]\cdots[A_k]\\ \mathcal{D}\\ B\end{array}$$

for a deduction $\mathcal{D}$ of formula B with certain distinguished assumption classes $[A_1]\cdots[A_k]$. We omit specifically denoting such classes where they are not relevant to the discussion.

Deductions are generated as follows:

1. Basis. If A is a formula, the single node tree with label A is a deduction of A. A is an open assumption of this deduction; there are no cancelled assumptions, and there is one assumption class consisting of A itself.

2. Inductive Step. New deductions are built from old ones by the following introduction and elimination rules.

$$\wedge\mathrm{I}\quad \frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ A & B\end{array}}{A\wedge B}\qquad\qquad \wedge\mathrm{E}\quad \frac{\begin{array}{c}\mathcal{D}_1\\ A\wedge B\end{array}}{A}\qquad \frac{\begin{array}{c}\mathcal{D}_1\\ A\wedge B\end{array}}{B}$$

$$\Rightarrow\mathrm{I}\quad \frac{\begin{array}{c}[A]\\ \mathcal{D}_1\\ B\end{array}}{A\Rightarrow B}\qquad\qquad \Rightarrow\mathrm{E}\quad \frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ A\Rightarrow B & A\end{array}}{B}$$

In $\Rightarrow$I , $[A]$ is a particular open assumption class, all of whose members (viz, occurrences of A) are simultaneously discharged. There may be other occurrences of A not discharged.

Now consider the following Gentzen rules for intuitionistic logic based on the connectives $\wedge, \Rightarrow$. Recall [12] sequents have the form $\Gamma \vdash A$, where Γ is a finite list of formulae and A is a single formula.These sequents satisfy the following axioms and rules (here Γ and Δ are finite lists of formulae) :

Axiom: $A \vdash A$

Cut:
$$\frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}$$

Structural:

Exchange
$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}$$

Contraction
$$\frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C}$$

Weakening
$$\frac{\Gamma \vdash C}{\Gamma, A \vdash C}$$

Logical:

$$\wedge\mathrm{L}_i \quad \frac{\Gamma, A_i \vdash C}{\Gamma, A_1 \wedge A_2 \vdash C}\ , i = 1, 2. \qquad \wedge\mathrm{R} \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B}$$

$$\Rightarrow\mathrm{L} \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma\ , \Delta, A \Rightarrow B \vdash C} \qquad \Rightarrow\mathrm{R} \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

The fundamental theorem of Gentzen (Cut-Elimination) states that the cut rule in the above formalism is redundant. Gentzen's theorem is closely related to normalization of lambda terms ([33, 12, 31]). One way to see this is to translate from sequent calculus to natural deduction (cf. Section 3.3 below), then use the so-called Curry-Howard isomorphism [12] to associate lambda terms to natural deduction proofs. This amounts to interpreting the sequent calculus directly into the typed lambda calculus, which we now describe.

3.2 From Sequents to Lambda Terms

Although the Curry-Howard procedure of associating typed lambda terms to natural deduction proofs (cf [12]) is by now quite familiar, a similar process applied to Gentzen sequent calculus appears less so, despite the related work of Lambek connected to categorical coherence theorems [17, 18].

One motivation of this term assignment is to think of an intuitionist sequent $\Gamma \vdash B$ (where $\Gamma = \{A_1, \ldots, A_k\}$) as an input/output device, accepting k inputs of types $A_1, \ldots, A_k$ and returning an output of type B. To this end, recall that our language of typed lambda calculus has types freely generated from type variables and constants under the usual type-forming operations $\times$ and $\Rightarrow$. We inductively define the assignment of lambda terms to proofs of sequents , using the special notation $x_1 : A_1, \ldots, x_k : A_k \rhd t[x_1, \ldots x_k] : B$, or sometimes for brevity the vector notation $\vec{x} : \Gamma \rhd t[\vec{x}] : B$. In this assignment, we assume *the variables x_i to the left of $\rhd$ are distinct* , even if some of the formulae A_i are equal; moreover, the free variables of the term $t[x_1, \ldots, x_k]$ are contained in $\{x_1, \ldots, x_k\}$. At the same time, we shall also give the informal conversion equations that these terms satisfy, when appropriate. Notation: we write $t[a_1/x_1, \ldots, a_k/x_k]$ to denote the simultaneous substitution of a_i for x_i in term t.

Axiom : $$x : A \rhd x : A$$

Cut : $$\frac{\vec{x} : \Gamma \rhd t[\vec{x}] : A \qquad \vec{y} : \Delta, z : A \rhd f[\vec{y}, z] : B}{\vec{x} : \Gamma, \vec{y} : \Delta \rhd f[\vec{y}, t[\vec{x}]/z] : B}$$

Structural :

Exchange $$\frac{\vec{x} : \Gamma, y : A, z : B, \vec{w} : \Delta \rhd t[\vec{x}, y, z, \vec{w}] : C}{\vec{x} : \Gamma, z : B, y : A, \vec{w} : \Delta \rhd t^e[\vec{x}, z, y, \vec{w}] : C}$$
where $t^e[\vec{x}, z, y, \vec{w}] = t[\vec{x}, y, z, \vec{w}]$

Contraction $$\frac{\vec{x} : \Gamma, y : A, z : A \rhd t[\vec{x}, y, z] : C}{\vec{x} : \Gamma, y : A \rhd t^c[\vec{x}, y] : C}$$
where $t^c[\vec{x}, y] = t[\vec{x}, y, y/z]$.

Weakening $$\frac{\vec{x} : \Gamma \rhd t[\vec{x}] : C}{\vec{x} : \Gamma, y : A \rhd t^w[\vec{x}, y] : C}$$
where $t^w[\vec{x}, y] = t[\vec{x}]$.

Logical :

$\wedge L_i, \; i=1,2$ $$\frac{\vec{x} : \Gamma, y_i : A_i \rhd t[\vec{x}, y_i] : C}{\vec{x} : \Gamma, z : A_1 \wedge A_2 \rhd t^{\wedge}[\vec{x}, z] : C}$$
where $t^{\wedge}[\vec{x}, z] = t[\vec{x}, \pi_i(z)/y_i]$

$\wedge$R $$\frac{\vec{x} : \Gamma \rhd s[\vec{x}] : A \qquad \vec{y} : \Delta \rhd t[\vec{y}] : B}{\vec{x} : \Gamma, \vec{y} : \Delta \rhd < s[\vec{x}], t[\vec{y}] >: A \wedge B}$$

$\Rightarrow$L $$\frac{\vec{x} : \Gamma \rhd f[\vec{x}] : A \qquad \vec{y} : \Delta, z : B \rhd g[\vec{y}, z] : C}{\vec{x} : \Gamma \, , \vec{y} : \Delta, u : A \Rightarrow B \rhd g[\vec{y}, uf[\vec{x}]/z] : C}$$
where , if $u : A \Rightarrow B$ and $a : A$ then $ua : B$ denotes application of u to argument a.

$$\Rightarrow\mathrm{R} \qquad \frac{\vec{x} : \Gamma, y : A \rhd t[\vec{x}, y] : B}{\vec{x} : \Gamma \rhd \lambda y : A.t[\vec{x}, y] : A \Rightarrow B}$$

The inductive clauses in the term assignment above are *not* the same as the inductive clauses used in term judgements and term assignments of ordinary typed lambda calculus; however, the rules above are derivable from the usual term-formation rules. A cut-free version of the above calculus is obtained by omitting the cut rule.

From the point of view of the typed lambda calculus, the cut rule is a general substitution process obtained by plugging one term into another. In fact, the cut-elimination theorem can be restated as:

> *All general instances of term substitution arising from the cut rule are already derivable (up to equality of terms) from the special instances of substitution used in the other rules.*

Cut elimination permits "asymmetrical" interpretations of sequent calculi ([12], p.34). More precisely, suppose we assign to occurrences of atomic predicates a *signature* $+1$ or -1 depending upon whether they appear positively (resp. negatively) in a given sequent . (Obviously, this is closely related to representing type expressions as multivariant functors $(\mathcal{C}^{\circ})^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$, up to rearrangement of factors). The central point is that the rules of the sequent calculus, *except for the identity and cut* , preserve the signature. What about identity and cut? If we think of term assignments $\vec{x} : \Gamma \rhd t(\vec{x}) : B$ as the functional process $\vec{x} \mapsto t(\vec{x})$ assigning "proofs" to formulae, the identity sequent maps (proofs of) negative occurrences, those with signature -1, to (proofs of) positive occurrences of the same formula. Conversely, cut substitutes a proof term whose type has positive signature for one of the same type, of negative signature; viz. in the cut rule above, in $f[\vec{y}, z] : B$ the term $t(\vec{x}) : A$ where A has positive signature is substituted into the variable slot $z : A$, where A has negative signature. In the absence of cut, the symmetry of the situation is broken: we may obtain asymmetrical interpretations of the sequents [12].

Let us give an example to illustrate the power of the method. In the paper [9] one associates to any lambda term an infinite series $\sum a_n \vec{e_n}$ where the coefficients a_n are cardinal numbers. The task is to prove that any normalisable lambda term is *weakly finite* , i.e. all a_n in its series are integers. To prove the statement (since the coefficients are invariant modulo β-conversion) one restricts to normal lambda terms. One introduces the auxiliary notion of being *strongly finite* (i.e. weakly finite plus almost all a_n null), and one checks that

- If $t[x]$ is weakly finite for any strongly finite x, then $\lambda x.t[x]$ is weakly finite.

- If t is strongly finite and u is weakly finite, then tu is strongly finite.

From this one easily proves that, for t normal

- If all variables in t are interpreted by strongly finite series then t becomes weakly finite.

- Furthermore, when t does not start with a λ, it is indeed strongly finite.

The method breaks down on cut, i.e. on general lambda terms, for which we would need the wrong symmetrical statement: strongly finite = weakly finite.

It is precisely the problem of linking positive and negative slots in type expressions which prevents, in general, the composing of dinaturals. Kelly [13], in his abstract treatment of coherence in categories, discusses this very situation in detail for a general calculus of transformations, while special cases of the general problem were already resolved in Eilenberg & Kelly [5]. Cut elimination theorems were successfully applied to coherence questions by Lambek [15, 16, 17, 18] and Mints [22]. Of course, for the simple types of this paper, normalization or cut elimination poses no problems. But even for these types, we shall obtain more: cut elimination implies that the internal language supports a compositional dinatural interpretation (between definable functors). In the language of Kelly-Mac Lane (applied to the case of ccc's), we shall show that every constructible arrow is dinatural.

As we have seen in the previous discussion, asymmetrical interpretation is one of the major approaches to cut elimination, closely related to dinaturality. For instance, dinaturals are like cut-free proofs and their compositionality is problematic just as cut-elimination is a highly non-trivial statement. But in some ways, until we can fully understand the second order (polymorphic) case from this viewpoint, we must honestly admit we are still far from understanding the main features of these fascinating questions.

3.3 From Sequents to Natural Deduction

The lambda term assignment occurs in two stages: the first stage is the translation of sequent calculus to natural deduction (see [12]); the second stage is the Curry-Howard isomorphism. Let us describe the first part. To a cut-free proof of a sequent $\Gamma \vdash A$ in sequent calculus one associates a natural deduction proof in normal form

$$\begin{array}{c}[\Gamma]\\ \mathcal{D}\\ A\end{array}$$

where $[\Gamma]$ denotes a collection of assumption classes built from formulas in Γ. The translation is reasonably well-known in proof theory ([12, 26, 31, 33]); we highlight the main points.

1. To the axiom $A \vdash A$ associate the deduction A.

2. The Exchange Rule is the identity on deductions (because the assumptions are not considered ordered).

3. The Weakening Rule $\dfrac{\Gamma \vdash B}{\Gamma, A \vdash B}$ is interpreted as follows. Given the interpretation of the premiss $\begin{array}{c}[\Gamma]\\ \mathcal{D}\\ B\end{array}$ we obtain $\begin{array}{c}[\Gamma],[A]'\\ \mathcal{D}'\\ B\end{array}$ where $[A]'$ is an assumption class formed by *zero* occurrences of A , and $\mathcal{D}'$ is $\mathcal{D}$ with $[\Gamma]$ replaced by $[\Gamma],[A]'$.

4. Contraction translates as follows: assuming we have interpreted the hypotheses, the conclusion is interpreted by forming the union of the assumption classes corresponding to the contraction formula A.

As for the logical rules,

5. $\wedge\mathrm{L}_i$ $\quad \dfrac{\Gamma, A_i \vdash C}{\Gamma, A_1 \wedge A_2 \vdash C}$ translates to $\begin{array}{cc} & \dfrac{[A_1 \wedge A_2]}{A_i} \\ [\Gamma] & \\ & \vdots \\ & C \end{array}$.

That is, there is an operation ("projection") which allows us to go from $A_1 \wedge A_2$-assumption classes to A_i-assumption classes, $i = 1, 2$.

6. $\wedge\mathrm{R}$ $\quad \dfrac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B}$ translates as $\dfrac{\begin{array}{cc}[\Gamma] & [\Delta]\\ \vdots & \vdots \\ A & B\end{array}}{A \wedge B}$. In this case, we form the union of the two assumption classes.

7. $\Rightarrow\mathrm{L}$ $\quad \dfrac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma\ , \Delta, A \Rightarrow B \vdash C}$ translates as follows: suppose

$$\begin{array}{c}[\Gamma]\\ \mathcal{D}\\ A\end{array} \quad \text{and} \quad \begin{array}{c}[\Delta],[B]\\ \mathcal{D}'\\ C\end{array}$$

interpret the premisses. Then

$$\begin{array}{cc} & \begin{array}{c}[\Gamma] \\ \mathcal{D} \\ A \quad [A \Rightarrow B] \end{array} \\ [\Delta] & \overline{\qquad\qquad B \qquad\qquad} \\ & \mathcal{D}' \\ & C \end{array}$$

interprets the conclusion, where in $\mathcal{D}'$ we replace each occurrence of B in the distinguished assumption class $[B]$ by the indicated proof with assumption class $[A \Rightarrow B]$. Proliferation of occurrences of formulas in Γ are taken into account by making a *new* assumption class, consisting of all the indicated occurrences of Γ so duplicated.

8. $\Rightarrow$ R $\quad \dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$ translates as $\dfrac{\begin{array}{c}[\Gamma],[A] \\ \mathcal{D} \\ B\end{array}}{A \Rightarrow B}$ where all formula occurrences in the assumption class $[A]$ of $\mathcal{D}$ are simultaneously discharged.

One subtlety of the natural deduction calculus is that natural deduction trees are closed under very general substitutions. For example, the Cut Rule $\dfrac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}$ translates as an operation taking proofs of the

premisses $\begin{array}{c}[\Gamma] \\ \vdots \\ A\end{array}$ and $\begin{array}{c}[\Delta],[A] \\ \vdots \\ B\end{array}$ and obtaining a proof of the conclusion

$\begin{array}{c}[\Gamma],[\Delta] \\ \vdots \\ B\end{array}$ by making (possibly many) copies of the tree $\begin{array}{c}[\Gamma] \\ \vdots \\ A\end{array}$, and grafting a copy onto each occurrence of formula A in the given proof of B. But in fact, only very special substitutions of natural deduction trees are required in this translation. We note that left sequent calculus rules translate as certain basic computations on assumption classes induced by these special substitutions .

4 Normal Forms and Cut-Free Proofs

In this section we shall prove the main theorems mentioned above. We shall also observe:

- How the process of cut-elimination, when interpreted by dinaturals, unlocks the clashes (i.e. dependencies) between positive and negative occurrences of formulas.

- How the kind of actions on assumption classes that arise in translating left sequent calculus rules into natural deduction are closely related to Reynolds' relational interpretation of parametric polymorphism [27].

The first observation above permits separate computations in the negative and positive parts of sequents, is related to the asymmetrical interpretation of cut-free proofs ([12]), and to the problem of composing dinaturals.

We shall show that the cut-free Gentzen sequent rules for the $\wedge, \Rightarrow$-fragment of intuitionist logic are interpretable as composable syntactic dinatural transformations between definable functors, as mentioned in Section 2 above. It follows that these transformations form a category. The fact that we use cut-free proofs obviates the need to consider the general problem of compositionality of dinaturals [2]. That is, we start with the translation of sequent calculus into natural deduction, from Section 3 above, then follow by the so-called Curry-Howard isomorphism [12] to obtain the associated lambda terms.

More precisely, lambda terms in normal form (arising from a valid typing judgement) correspond to normal natural deduction proofs of the formula represented by the types. Such normal deductions arise from (not necessarily unique) cut-free Gentzen sequent proofs. Cut-free Gentzen proofs without assumptions translate to certain closed normal lambda terms; these latter terms yield syntactic families which will be shown to be dinatural. Hence we prove the main result by induction on cut-free Gentzen sequent proofs (which is a kind of induction on head normal form [12], p.19).

To this end, we assume we already know how to interpret the formulas in a sequent as definable functors $(\mathcal{C}^\circ)^n \times \mathcal{C}^n \longrightarrow \mathcal{C}$ as earlier. We shall interpret a sequent $\Gamma \vdash B$ as a dinatural transformation $\prod_{i \leq k} F_i(\mathbf{AA}) \longrightarrow G\mathbf{AA}$, where Γ is a finite list of formulae A_i, $F_i = \|A_i\|$ and $G = \|B\|$, by associating appropriate families of lambda terms to proofs.

The axiom is trivial to interpret: it is the identity transformation between functors. This is clearly a definable family, and moreover is dinatural over any ccc . For the remainder of the rules, we assume as inductive hypothesis that *we already know how to interpret the premisses of a rule as dinatural transformations* . We show how to interpret the conclusion of each rule. We look at the important cases:

1. Contraction: We ignore the passive context Γ (which only adds free variable parameters to all terms). Suppose $F = \|A\|$ and $G = \|C\|$ are definable functors. Suppose the family of terms $d = \{d_{\mathbf{A}} : (F \times F)\mathbf{AA} \longrightarrow G\mathbf{AA} | \mathbf{A} \in \mathcal{C}^n\}$ interprets the premisses of Contraction. Recall that $(F \times$

$F)\mathbf{AA} = F\mathbf{AA} \times F\mathbf{AA}$. Consider the family $d{\circ}\Delta = \{d_{\mathbf{A}}{\circ}\Delta_{F\mathbf{AA}} : \mathbf{A} \in \mathcal{C}^n\}$, given by the following diagram:

$$\begin{tikzcd}
 & F\mathbf{AA} \arrow[r, "\Delta_{F\mathbf{AA}}"] & (F\times F)\mathbf{AA} \arrow[r, "d_{\mathbf{A}}"] & G\mathbf{AA} \arrow[dr, "G\mathbf{Af}"] & \\
F\mathbf{BA} \arrow[ur, "F\mathbf{fA}"] \arrow[dr, "F\mathbf{Bf}"'] \arrow[r, "\Delta_{F\mathbf{BA}}"] & (F\times F)\mathbf{BA} \arrow[ur, "(F\times F)\mathbf{fA}"] \arrow[dr, "(F\times F)\mathbf{Bf}"'] & & & G\mathbf{AB} \\
 & F\mathbf{BB} \arrow[r, "\Delta_{F\mathbf{BB}}"'] & (F\times F)\mathbf{BB} \arrow[r, "d_{\mathbf{B}}"'] & G\mathbf{BB} \arrow[ur, "G\mathbf{fB}"'] &
\end{tikzcd}$$

It is easily verified that the outer hexagon commutes. Thus the family $d{\circ}\Delta$ interprets the conclusion of contraction.

2. Weakening: We may suppose without loss that the passive context Γ has only one formula. Suppose $F = \|\Gamma\|$, $G = \|A\|$ and $H = \|C\|$ are definable functors. Suppose the family of terms $d = \{d_{\mathbf{A}} : F\mathbf{AA} \longrightarrow H\mathbf{AA} | \mathbf{A} \in \mathcal{C}^n\}$ interprets the hypothesis of weakening. Consider the family $d{\circ}\Pi = \{d_{\mathbf{A}}{\circ}\Pi_{F\mathbf{AA}} | \mathbf{A} \in \mathcal{C}^n\}$, given by the following diagram:

$$\begin{tikzcd}
 & (F\times G)\mathbf{AA} \arrow[r, "\Pi_{F\mathbf{AA}}"] & F\mathbf{AA} \arrow[r, "d_{\mathbf{A}}"] & H\mathbf{AA} \arrow[dr, "H\mathbf{Af}"] & \\
(F\times G)\mathbf{BA} \arrow[ur, "(F\times G)\mathbf{fA}"] \arrow[dr, "(F\times G)\mathbf{Bf}"'] \arrow[r, "\Pi_{F\mathbf{BA}}"] & F\mathbf{BA} \arrow[ur, "F\mathbf{fA}"] \arrow[dr, "F\mathbf{Bf}"'] & & & H\mathbf{AB} \\
 & (F\times G)\mathbf{BB} \arrow[r, "\Pi_{F\mathbf{BB}}"'] & F\mathbf{BB} \arrow[r, "d_{\mathbf{B}}"'] & H\mathbf{BB} \arrow[ur, "H\mathbf{fB}"'] &
\end{tikzcd}$$

It is easily verified that the outer hexagon commutes. Thus the family $d{\circ}\Pi$ interprets the conclusion of weakening.

Although the left rules turn out to be more profound from the viewpoint of dinaturality, we shall first examine the right rules.

3. $\wedge$R is interpreted essentially by taking the pointwise product of the two hexagons interpreting its premisses. That is, suppose (wolog) that Γ and Δ consist of only one formula, and are interpreted as definable functors $F = \|\Gamma\|$, $G = \|\Delta\|$, $H = \|C\|$, and $K = \|D\|$,resp. Suppose $s : F \longrightarrow H$ and $t : G \longrightarrow K$ interpret the premisses of $\wedge$R. Then the conclusion is interpreted by the family $d = \{d_{\mathbf{A}} \mid \mathbf{A} \in \mathcal{C}^n\} : F \times G \longrightarrow H \times K$, which is defined as

follows: $d_{\mathbf{A}} = s_{\mathbf{A}} \times t_{\mathbf{A}} =< s_{\mathbf{A}} \circ \pi_1, t_{\mathbf{A}} \circ \pi_2 >: (\ F \times G\ )\ \mathbf{AA} \ \longrightarrow\ (H \times K)\ \mathbf{AA}$. It is easily verified that the associated hexagon below commutes (where $\mathbf{f}$: $\mathbf{A} \longrightarrow \mathbf{B}$ and some arrows are omitted for clarity):

$$
\begin{tikzcd}
 & (F \times G)\mathbf{AA} \arrow[r, "d_{\mathbf{A}}"] & (H \times K)\mathbf{AA} \arrow[dr] & \\
(F \times G)\mathbf{BA} \arrow[ur, "(F \times G)\mathbf{fA}"] \arrow[dr] & & & (H \times K)\mathbf{AB} \\
 & (F \times G)\mathbf{BB} \arrow[r, "d_{\mathbf{B}}"'] & (H \times K)\mathbf{BB} \arrow[ur, "(H \times K)\mathbf{fB}"'] &
\end{tikzcd}
$$

Note that each component of $d_{\mathbf{A}} =< s_{\mathbf{A}} \circ \pi_1, t_{\mathbf{A}} \circ \pi_2 >$ acts separately—there is no communication (cf. [11]).

4. $\Rightarrow$R. Suppose $t : K \times F \longrightarrow G$ is a dinatural transformation interpreting the hypothesis of $\Rightarrow$R. Then the family $t^* = \{t^*_{\mathbf{A}} | t^*_{\mathbf{A}} : K\mathbf{AA} \longrightarrow G\mathbf{AA}^{F\mathbf{AA}}\}$ given by currying $t_{\mathbf{A}} : K\mathbf{AA} \times F\mathbf{AA} \longrightarrow G\mathbf{AA}$ is a dinatural transformation $t^* : K \longrightarrow G^F$. We leave the diagram and verification to the reader.

We now examine the left rules.

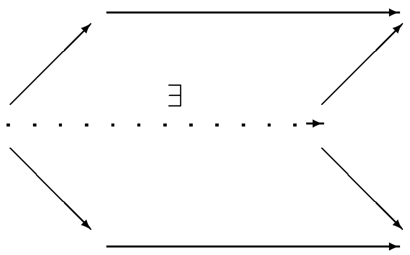
5. $\wedge$L is interpreted almost identically to the case of Weakening above.

6. $\Rightarrow$L. For ease of exposition, suppose the context Δ is empty. For notational convenience, we denote the functorial interpretations $\|\Gamma\|$, $\|C\|$,$\|D\|$, and $\|E\|$ by K,F, G, and H, resp. We are assuming dinatural transformations $w : K \longrightarrow F$ and $t : G \longrightarrow H$ which interpret the premisses. We wish to interpret the conclusion of $\Rightarrow$L ; i.e. (modulo exchange) we wish to find a dinatural transformation $d : G^F \times K \longrightarrow H$. We remind the reader that in functorial semantics, twisted exponentiation is defined by: $G^F\mathbf{BA} = G\mathbf{BA}^{F\mathbf{AB}}$. Behold the following diagram, where $\mathbf{f} : \mathbf{A} \longrightarrow \mathbf{B}$, $\theta_{\mathbf{A}} = 1 \times w_{\mathbf{A}}$ and $\theta_{\mathbf{B}} = 1 \times w_{\mathbf{B}}$.

$$
\begin{tikzcd}
 & & (G^F \times K)\mathbf{AA} \arrow[r, "1 \times w_{\mathbf{A}}"] \arrow[d, phantom, "\text{(I)}"] & & (G^F \times F)\mathbf{AA} \arrow[r, "ev_{\mathbf{AA}}"] & G\mathbf{AA} \arrow[r, "t_{\mathbf{A}}"] & H\mathbf{AA} \arrow[dd, "H\mathbf{Af}"] \\
 & (G^F)\mathbf{BA} \times K\mathbf{AA} \arrow[ur, "k_1"] \arrow[rr, "\theta\mathbf{A}"] & & (G^F)\mathbf{BA} \times F\mathbf{AA} \arrow[ur, "k_2"] \arrow[dr, "1 \times F\mathbf{Af}"] \arrow[rr, phantom, "\text{(II)}"] & & & \\
(G^F \times K)\mathbf{BA} \arrow[ur, "1 \times K\mathbf{fA}"] \arrow[dr, "1 \times K\mathbf{Bf}"'] \arrow[rrrr, phantom, "1 \times \text{Assump.}"] & & & & (G^F)\mathbf{BA} \times F\mathbf{AB} \arrow[r, "ev_{\mathbf{BA}}"'] & G\mathbf{BA} \arrow[uu, "G\mathbf{fA}"] \arrow[dd, "G\mathbf{Bf}"'] \arrow[r, phantom, "\text{Assump.}"] & H\mathbf{AB} \\
 & (G^F)\mathbf{BA} \times K\mathbf{BB} \arrow[rr, "\theta\mathbf{B}"'] \arrow[dr, "l_1"'] & & (G^F)\mathbf{BA} \times F\mathbf{BB} \arrow[ur, "1 \times F\mathbf{fB}"] \arrow[dr, "l_2"'] \arrow[rr, phantom, "\text{(IV)}"] & & & \\
 & & (G^F \times K)\mathbf{BB} \arrow[r, "1 \times w_{\mathbf{B}}"'] \arrow[u, phantom, "\text{(III)}"] & & (G^F \times F)\mathbf{BB} \arrow[r, "ev_{\mathbf{BB}}"'] & G\mathbf{BB} \arrow[r, "t_{\mathbf{B}}"'] & H\mathbf{BB} \arrow[uu, "H\mathbf{fB}"']
\end{tikzcd}
$$

In this diagram, observe how the definition of twisted exponentials fits perfectly with the source and target of the middle horizontal evaluation map. Here, $k_1 = (G^F)\mathbf{fA} \times 1_{K\mathbf{AA}}$ and $k_2 = (G^F)\mathbf{fA} \times 1_{F\mathbf{AA}}$; Similarly for $l_i, i = 1, 2$ with appropriate $\mathbf{A}$ replaced by $\mathbf{B}$. The commutativity of (I) and (III) is trivial; the commutativity of (II) and (IV) comes from a direct calculation. The inductive assumption guarantees the commutativity of the indicated hexagons. Thus we obtain a dinatural transformation $G^F \times K \longrightarrow H$.

Note that in the figure above, (I) and (III) correspond to splitting the preparation for evaluation into " contravariant" and "covariant" channels, with the action separate in each part: there is no communication. This is precisely directed at the question of how positive and negative parts of formulas occur in cut-free proofs. Moreover, we emphasize again that in the case of the structural rules, as well as the left rules, the translation from Gentzen sequents to natural deduction translates to actions on assumption classes. These actions are given by special substitutions of lambda terms mentioned in Section **3** above. In our ccc context, these special substitutions are treated as precomposing by special families of morphisms , as in case 2 and the evaluation morphisms in case 6 above. The crucial point is that these special families satisfy strong naturality properties which have the shape



where the broken arrow indicates the existence of a factorization. Diagrams of this shape arise as Reynolds' invariance conditions in his approach to parametricity [27], [2], pp.53-55, as well as in Statman's theory of logical relations [30], and in Wadler's universal Horn conditions which motivated this discussion. We emphasize, however, that the invariance conditions given by diagrams of this shape have deep roots in the left rules of Gentzen sequent calculus for propositional logic –prior to any considerations of second order type abstraction.

As a corollary, we obtain the following. Let $\mathcal{C}$ be the cartesian closed category generated by a simply typed lambda calculus $\mathcal{L}$ whose types are built inductively from some atomic types $\alpha, \beta, \ldots$, as in the Theorem of Section 2. Then

- Cut-free proofs are represented by closed terms in normal form, thus arrows in $\mathcal{C}$.
- Any arrow $m : A \longrightarrow B$ in $\mathcal{C}$ (qua closed normal form in $\mathcal{L}$ of type $A \Rightarrow B$) induces a dinatural transformation between definable functors $\|A\|$ and $\|B\|$. These normal forms compose by substitution.
- m provably satisfies the dinaturality equations.

We should remark that all this was done in the special case of intuitionist logic and cartesian closed categories. This analysis can be extended to the multiplicative fragment of linear logic (stemming from Barr's *-autonomous categories) and beyond.

References

[1] S. Abramsky, J. C. Mitchell, A. Scedrov, P. Wadler. Relators (to appear.)

[2] E.S. Bainbridge, P. Freyd , A. Scedrov, and P. J. Scott. Functorial Polymorphism, *Theoretical Computer Science* **70** (1990), pp. 35-64.

[3] E.S. Bainbridge, P. Freyd , A. Scedrov, and P. J. Scott. Functorial Polymorphism: Preliminary Report, *Logical Foundations of Functional Programming*, G. Huet, ed., Addison Wesley (1990), pp. 315-327.

[4] M. Barr. **-Autonomous Categories* . Springer LNM 752, 1979.

[5] S. Eilenberg and G. M Kelly. A generalization of the functorial calculus, *J. algebra* **3** (1966), pp. 366-375.

[6] A. Felty. A Logic Program for Transforming Sequent Proofs to Natural Deduction Proofs. In: *Proc. December 1989 Workshop on Extension of Logic Programming* , ed. by P. Schroeder-Heister, Springer LNCS, to appear.

[7] P. Freyd. Structural Polymorphism, Manuscript, Univ. of Pennsylvania (1989), to appear in *Th. Comp. Science.*

[8] P. Freyd, J-Y. Girard, A. Scedrov, and P. J. Scott. Semantic Parametricity in Polymorphic Lambda Calculus, *Proc. 3rd IEEE Symposium on Logic in Computer Science*, Edinburgh, 1988.

[9] J-Y. Girard. Normal functors, power series, and λ-calculus, *Ann. Pure and Applied Logic* **37** (1986) pp. 129-177.

[10] J-Y. Girard. The System **F** of Variables Types, Fifteen Years Later, *Theoretical Computer Science*, **45**, pp. 159-192.

[11] J-Y. Girard. Linear Logic, *Theoretical Computer Science*, **50**, 1987.

[12] J-Y. Girard, Y.Lafont, and P. Taylor. *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science, **7**, Cambridge University Press, 1989.

[13] G. M. Kelly. Many-variable functorial calculus.I, *Coherence in Categories* Springer LNM**281**, pp. 66-105.

[14] G. M. Kelly and S. Mac Lane. Coherence in closed categories, *J. Pure Appl. Alg.* **1** (1971), pp. 97-140.

[15] J. Lambek. Deductive Systems and Categories I, *J. Math. Systems Theory***2** (1968), pp. 278-318.

[16] J. Lambek. Deductive Systems and Categories II, *Springer LNM***86** (1969), pp. 76-122.

[17] J. Lambek. Multicategories Revisited, *Contemp. Math.***92**, pp. 217-239.

[18] J. Lambek. Logic without structural rules, Manuscript. (Mcgill University), 1990.

[19] J.Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*, Cambridge Studies in Advanced Mathematics **7**, Cambridge University Press, 1986.

[20] S. Mac Lane. *Categories for the Working Mathematician*, Springer Graduate Texts in Mathematics, Springer-Verlag, 1971.

[21] S. Mac Lane. Why commutative diagrams coincide with equivalent proofs, *Contemp. Math.***13** (1982), pp. 387-401.

[22] G. E. Mints. Closed Categories and the theory of proofs, *J. Soviet Math* **15** (1981), pp. 45-62.

[23] J. C. Mitchell and P. J. Scott. Typed Lambda Models and Cartesian Closed Categories, *Contemp. Math.* **92**, pp. 301-316.

[24] J.C. Mitchell and R. Harper, The Essence of ML, *Proc. 15th Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)*, San Diego, 1988, pp. 28-46.

[25] A. Pitts. Polymorphism is set-theoretic, constructively, in: *Category Theory and Computer Science* , Springer LNCS 283 (D. H. Pitt, ed.) pp. 12-39.

[26] D. Prawitz. *Natural Deduction* , Almquist & Wilksell, Stockhom, 1965.

[27] J.C.Reynolds. Types, Abstraction, and Parametric Polymorphism, in: *Information Processing '83* , R. E. A. Mason, ed. North-Holland, 1983, pp. 513-523.

[28] R. A. G. Seely. Categorical Semantics for Higher Order Polymorphic Lambda Calculus. *J. Symb. Logic* **52**(1987), pp. 969-989.

[29] R. A. G. Seely. Linear Logic, *-Autonomous Categories, And Cofree Coalgebras. *Contemp. Math.* **92**, pp. 371-382.

[30] R. Statman. Logical Relations and the Typed Lambda Calculus. *Inf. and Control* **65**(1985), pp. 85-97.

[31] A. S. Troelstra and D. van Dalen.*Constructivism in Mathematics*, Vols. I and II. North-Holland, 1988.

[32] P. Wadler. Theorems for Free! 4th International Symposium on Functional Programming Languages and Computer Architecture, Assn. Comp. Machinery, London, Sept. 1989.

[33] J. Zucker. The Correspondence between Cut-Elimination and Normalization, *Ann. Math. Logic* **7** (1974) pp. 1-112.

Jean-Yves Girard
Laboratoire de
Mathématiques Discrètes
UPR 9016
163 avenue de Luminy
Case 930, 13288 Marseille
Cedex 9, France
girard@lmd.univ-mrs.fr

Andre Scedrov
Dept. of Mathematics
Univ. of Pennsylvania
209 S. 33rd St.
Philadelphia, PA
19104-6395 U.S.A.
andre@cis.upenn.edu

Philip J. Scott
Dept. of Mathematics
Univ. of Ottawa
585 King Edward
Ottawa, Ont.
Canada K1N6N5
scpsg@acadvm1.uottawa.ca