NORMAL FORMS AND CUT-FREE PROOFS AS NATURAL TRANSFORMATIONS

JEAN-YVES GIRARD, ANDRE SCEDROV AND 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:

• 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

\[
\Gamma, \Delta, A \vdash B \\
\Gamma \vdash A, \Delta \vdash B
\]

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 \textit{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 \textit{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 \( \alpha \) 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:

\[
(f \times f) \circ r_A = r_B \circ (f \times f)
\]

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 \( A \rightarrow B \) uniquely correspond to morphisms \( 1 \rightarrow A \Rightarrow B \) [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 \( f : A \rightarrow B \) the following diagram commutes.