A system of natural deduction for categories

We will present a system of natural deduction for categories, i.e., a system that includes discharge rules, and whose semantics (via a kind of Curry-Howard isomorphism) interprets terms as objects, morphisms, functors and natural transformations, plus the usual points and functions. The main novelty of that system is that it has discharge rules for forming functors and natural transformations — actually *protofunctors* and *protonatural transformations*, entities that behave only syntactically like functors and natural transformations.

The system is expressive enough to let us represent as relatively short terms things like the Yoneda Lemma and the fact that a category of the form $\mathbf{Set}^{\mathbf{C}}$ has exponentials and a subobject classifier; the categorical constructions that prove those things can be read from the trees that deduce those terms, and it is easy to draw side to side to them other trees that show how each term would be written in the usual notation.

Usually we would choose "x" as the default name for a variable ranging over X, "y" for a variable ranging over Y, etc; we will allow "composite variable names", so "(a,b)" will be our default name for a variable ranging over $A \times B$, $c \to d$ our default name for one ranging over D^C, etc. We call this notation the *notation of typical points*, and the idea is that the name of a variable is the "name of the typical point" of the space in which that variable can take values. Generally — but now always — we will want to attributed related values to variables with related named — for example a being the first coordinate of (a,b).

The symbol $\mathbf{E}$ is used to mean "the space of", in the sense of "the set of all possible values for": we have $x \in X$, and X is considered as the "space of 'x's": $X = \mathbf{E}_x$. Following this idea, $\mathbf{E}_{(a,b)} = \mathbf{E}_a \times \mathbf{E}_b$, $\mathbf{E}_{c \to d} = (\mathbf{E}_d)^{(\mathbf{E}_c)}$, etc. In this way the uppercase letters become free for other uses, and we use fewer symbols. Obviously $\mathbf{E}_x, \mathbf{E}_a, \ldots$ can be seen as variables ranging over the class of all sets, whose values bear some relation with the values of $x, a, \ldots$ — namely that $x \in \mathbf{E}_x$, $a \in \mathbf{E}_a$, etc.

This notation for variables was chosen to make it convenient to "read aloud" construction trees. For example,

$$\dfrac{\dfrac{\dfrac{[a]^1 \quad b}{(a,b)} \qquad (a,b) \to c}{c}}{a \to c}\;1$$

can be read aloud as: "from a value for a and a value for b we have a natural way to choose a value for (a,b)"... — note that currently we are only talking in terms of "a value", not of "the value"; also, we will often say just "an a", "a b", etc, instead of the longer "a value for a", etc. — ..."and given an (a,b) and a function $(a,b) \to c$ we have a natural choice for a c. So, the tree over the bar marked with the '1' can be seen as an operation that from an a, a b and an $(a,b) \to c$ builds a c; for each b and $(a,b) \to c$ fixed this gives a function that takes an a and returns a c, that is, an $a \to c$. So the big tree, with the top a "discharged", denotes a natural way to build an $a \to c$ from a b and an $(a, b) \to c$."

The tree above is a "natural construction" for $a \to c$ from a b and an $(a, b) \to c$; the formal definition is that a natural construction for β from $\alpha_1, \ldots, \alpha_n$ in a certain system is a tree with β in its root where only members of the list $\alpha_1, \ldots, \alpha_n$ can appear as "leaves" (i.e., as undischarged hypotheses), and where each bar is an application of a rule of the system.

Generally the list $\alpha_1, \ldots, \alpha_n$ and the rules of the system will be clear from the context, and then we say that a β *has natural constructions* if there is a natural construction for β from $\alpha_1, \ldots, \alpha_n$ in that system, and we say that β is *well-defined* (in symbols: $\mathrm{wd}(\beta)$) if it has natural constructions and furthermore all its natural constructions give the same result, using the semantical interpretation of the rules and the values for $\alpha_1, \ldots, \alpha_n$. The result of a natural construction for β is called a "natural value" for β; if β and γ are well-defined and have related names then their natural values will be related.

After introducing just a few other auxiliary symbols we will be ready to show a sample construction in the system of natural deduction for categories ("system NDC", from now on). Just like a $x \to y$ is a function from $\mathbf{E}_x$ to $\mathbf{E}_y$, an $a \to b$ is a morphism from $\mathbf{O}_a$ to $\mathbf{O}_b$; if we have an $\mathbf{O}_a$ instead of an $\mathbf{E}_a$ it would generally make no sense to speak of "an a". $\mathbf{Cat}_a$ is the category where $\mathbf{O}_a$ lives; on the presence of an $a \to b$ we must have $\mathbf{Cat}_a = \mathbf{Cat}_b$. An $\mathbf{o}_\mathbf{A}$ is an object of the category $\mathbf{A}$; a $\mathbf{m}_\mathbf{A}$ is a morphism of the category $\mathbf{A}$. $\mathbf{A} \times \mathbf{B}$ is the product of the categories $\mathbf{A}$ and $\mathbf{B}$; an object of $\mathbf{A} \times \mathbf{B}$ is written like $\mathbf{O}[(a; b)]$ and corresponds to a pair $(\mathbf{O}_a, \mathbf{O}_b)$. $\mathbf{A} \Rightarrow \mathbf{B}$ is a category of functors; an $\mathbf{o}[\mathbf{A} \Rightarrow \mathbf{B}]$ is a functor from $\mathbf{A}$ to $\mathbf{B}$, and a $\mathbf{m}[\mathbf{A} \Rightarrow \mathbf{B}]$ is a natural transformation between two such functors.

A functor is denoted by its action on *names* of objects, and its action on "unnamed" arrows (i.e., undecorated arrows — $a \xrightarrow{f} b$ is not undecorated) is inferred from its action on objects. For example, $a \Rightarrow a^F$ takes $\mathbf{O}_a$ to $\mathbf{O}_{a^F}$, $\mathbf{O}_{a'}$ to $\mathbf{O}_{a'^F}$, $a' \to a''$ to $a'^F \to a''^F$, and so on.

An "unnamed" natural transformation from $a \Rightarrow a^F$ to $a \Rightarrow a^G$ is written as $a \xrightarrow{\bullet} (a^F \to a^G)$; if all its image morphisms are isos, monics or epis we may include that information in the name of the natural transformation by replacing the '$\to$' by '$\leftrightarrow$', '$\hookrightarrow$' or '$\twoheadrightarrow$' respectively.

The table below shows the main rules of system NDC. The full system is much larger than that, and most rules have restrictions — in Fmorf, for example, we must have $\mathbf{Cat}_{a'} = \mathbf{Cat}_a$.

$$\frac{a \to b}{\mathbf{O}[a]}\ \mathrm{src} \qquad \frac{a \to b}{\mathbf{O}[b]}\ \mathrm{tgt} \qquad \frac{\mathbf{O}[a]}{a \to a}\ \mathrm{id} \qquad \frac{a \to b \quad b \to c}{a \to c}$$

$$\frac{\mathbf{O}[a] \quad a \Rightarrow a^F}{\mathbf{O}[a^F]}\ \mathrm{Fobj} \qquad \frac{a' \to a'' \quad a \Rightarrow a^F}{a'^F \to a''^F}\ \mathrm{Fmorf}$$

$$\dfrac{\mathbf{O}[a] \quad a \stackrel{\bullet}{\to} (a^F \to a^G)}{a^F \to a^G}\ \text{NTobj}$$

$$\dfrac{\dfrac{[\mathbf{O}[a]]^1 \quad \dots}{\overline{a^F \to a^G}}}{a \stackrel{\bullet}{\to} (a^F \to a^G)}\ 1\ \text{NTdisch} \qquad \dfrac{\dfrac{[a' \to a'']^1 \quad \dots}{\overline{a'^F \to a''^F}}}{a \Rightarrow a^F}\ 1\ \text{Fdisch}$$

The discharging rules Fdisch and NTdisch form protofunctors and proto-NTs instead of functors and NTs; a protofunctor is an operation that behaves syntactically like a functor, and, likewise, a proto-NT behaves syntactically like a natural transformation. A protofunctor that respects identities and composition of morphisms is a functor; a proto-NT $T : F \to G$ that obeys the "square condition" for any arrow $f : A \to B$ (namely: $T_B \circ F(f) = G(f) \circ T_A$) is a natural transformation. Using 'wd' these conditions can be written as:

$$\begin{array}{l} \forall \mathbf{O}_a \ \ \text{wd}(a^F \to a^F) \\ \forall \mathbf{O}_a, \mathbf{O}_b, \mathbf{O}_c, (a \to b), (b \to c) \ \ \text{wd}(a^F \to c^F) \\ \forall \mathbf{O}_a, \mathbf{O}_b, (a \to b) \ \ \text{wd}(a^F \to b^G) \end{array}$$

Note that these conditions must be checked in the semantics.

A simple example of system NDC at work: here's how to "Curry" the identity functor $(a; b) \Rightarrow (a; b)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{[a' \to a'']^2 \quad \dfrac{[\mathbf{O}[b]]^1}{b \to b}}{(a'; b) \to (a''; b)}}{b \stackrel{\bullet}{\to} ((a'; b) \to (a''; b))}\ 1}{(b \Rightarrow (a'; b)) \to (b \Rightarrow (a''; b))}\ \text{ren}}{a \Rightarrow (b \Rightarrow (a; b))}\ 2$$

From the object of 'b's in $\mathbf{B}$ build the identity arrow $b \to b$; with the morphisms $a' \to a''$ of $\mathbf{A}$ and $b \to b$ build the morphism $(a'; b) \to (a''; b)$ of the product category $\mathbf{A} \times \mathbf{B}$. Now regard what is above the bar marked '1' as an operation that takes an arbitrary object $\mathbf{O}[b]$ of $\mathbf{B}$ and produces a morphism $(a'; b) \to (a''; b)$; this is what a natural transformation $b \stackrel{\bullet}{\to} ((a'; b) \to (a''; b))$ would do, so we can discharge the $\mathbf{O}[b]$ and form a (proto-)natural transformation. Apply a renaming rule and that becomes a morphism between the objects $\mathbf{O}[b \Rightarrow (a'; b)]$ and $\mathbf{O}[b \Rightarrow (a''; b)]$ in $\mathbf{A} \Rightarrow \mathbf{A} \times \mathbf{B}$, a category of functors. So we've got an operation that behaves syntactically as a functor $a \Rightarrow (b \Rightarrow (a; b))$, as it takes a generic morphism of $\mathbf{A}$, $a' \to a''$, and returns a morphism with the right name; discharge the $a' \to a''$ and form a (proto-)functor.

Other examples: the Yoneda lemma corresponds to the term $(a; b \Rightarrow b^F) \stackrel{\bullet}{\to} (a^F \leftrightarrow (b \stackrel{\bullet}{\to} ((a \to b) \to b^F)))$, but explaining it would require a few more definitons. The terms for the exponentials and the classifier in a $\mathbf{Set}^{\mathbf{C}}$ would require even more definitions.

Eduardo Ochs
Departamento de Matemática, PUC-Rio
http://www.mat.puc-rio.br/~edrx/math.html
edrx@mat.puc-rio.br