TYPE THEORY AND ITS MEANING EXPLANATIONS

JONATHAN STERLING

To start, we will consider the notion of a *logical theory*; in my mind, it starts with a species (or set) of judgements that can be proposed, asserted, and (if they are evident) known.

1. Judgements of a Logical Theory

The basic forms of judgement for a logical theory will be $\boxed{P\ \mathit{prop}}$ and $\boxed{P\ \mathit{true}}$; and what is P? It is a member of the species of terms, which are made meaningful in the course of making the judgement $P\ \mathit{prop}$ evident for a proposition P.

To each judgement is assigned a *meaning explanation*, which explicates the knowledge-theoretic content of the judgement. For a judgement $\mathcal{J}$, a meaning explanation should be in the form:

> To know $\mathcal{J}$ is to know...

The meaning of the judgement $P\ \mathit{prop}$ is, then, as follows:

> To know $P\ \mathit{prop}$ is to know that P is a proposition, which is to know what counts as a direct verification of P.

So if a symbol P is taken to denote a proposition, we must know *what sort of thing* is to be taken as a direct verification of P, and this is by definition. A "direct verification" is understood in constrast with an "indirect verification", which is to be thought of as a means or plan for verifying the proposition. Now, the judgement $P\ \mathit{true}$ is only meaningful in case we know $P\ \mathit{prop}$ (this is called a presupposition). Then the meaning of $P\ \mathit{true}$ is as follows:

> To know $P\ \mathit{true}$ is to have a verification of P.

From the (implicit) presupposition $P\ \mathit{prop}$, we already know what counts as a verification, so the meaning explanation is well-defined. Note that it is the same to have a means or plan for verifying P as to have a (direct) verification; this follows from the fact that one may put into action a plan for verifying P and get out such a verification, and likewise, it is possible to propound a plan of verification by appealing to an existing verification.

The judgements we have described so far are "categorical" in the sense that they are made without assumption. We will need to define a further form of judgement, which is called "hypothetical", and this is the judgement under hypothesis $\boxed{\mathcal{J}\ [\mathcal{J}_1 \dots \mathcal{J}_n]}$. And its meaning explanation is as follows:

> To know the judgement $\mathcal{J}\ [\mathcal{J}_1 \dots \mathcal{J}_n]$ is to know the categorical judgement $\mathcal{J}$ when you know the judgements $\mathcal{J}_1 \dots \mathcal{J}_n$.

2. Propositions

Now that we have propounded and explained the minimal system of judgements for a logical theory, let us populate it with propositions. First, we have falsity $\bot$, and we wish to make $\bot$ *prop* evident; to do this, we simply state what counts as a direct verification of $\bot$, which is that there can be no direct verification of $\bot$.

The next basic proposition is trivially true $\top$, and to make $\top$ *prop* evident, we state that a direct verification of $\top$ is trivial. The meaning of $\top$ thus validates the judgement $\top$ *true*.

Next, let us define conjunction; in doing so, we will make evident the hypothetical judgement $P \wedge Q$ *prop* $[P$ *prop*$, Q$ *prop*$]$; equivalently, we can display this as a rule of inference:

$$\frac{P\ \textit{prop} \quad Q\ \textit{prop}}{P \wedge Q\ \textit{prop}}$$

A direct verification of $P \wedge Q$ consists in a verification of P and a verification of Q; this validates the assertion of the judgement $P \wedge Q$ *true* $[P$ *true*$, Q$ *true*$]$. Because it is a valid inference, we can write it as an inference rule:

$$\frac{P\ \textit{true} \quad Q\ \textit{true}}{P \wedge Q\ \textit{true}}$$

A direct verification of $P \vee Q$ may be got either from a verification of P or one of Q. From this definition we know $P \vee Q$ *prop* $[P$ *prop*$, Q$ *prop*$]$, or

$$\frac{P\ \textit{prop} \quad Q\ \textit{prop}}{P \vee Q\ \textit{prop}}$$

The verification conditions of disjunction give rise to two evident judgements $P \vee Q$ *true* $[P$ *true*$]$ and $P \vee Q$ *true* $[Q$ *true*$]$, which we can write as inference rules:

$$\frac{P\ \textit{true}}{P \vee Q\ \textit{true}} \qquad \frac{Q\ \textit{true}}{P \vee Q\ \textit{true}}$$

Finally, we must define the circumstances under which $P \supset Q$ is a proposition (i.e. when $P \supset Q$ *prop* is evident). And we intend this to be under the circumstances that P is a proposition, and also that Q is a proposition assuming that P is true.

In other words, $P \supset Q\ prop\ [P\ prop, Q\ prop\ [P\ true]]$, or

$$\frac{P\ prop \quad Q\ prop\ [P\ true]}{P \supset Q\ prop}$$

Now, to validate this judgement will be a bit more complicated than the previous ones. But by unfolding the meaning explanations for hypothetical judgement, proposition-hood and truth of a proposition, we arrive at the following explanation:

> To know $P \supset Q\ prop\ [P\ prop, Q\ prop\ [P\ true]]$ is to know what counts as a direct verification of $P \supset Q$ when one knows what counts as a direct verification of P, and, when one has such a verification, what counts as a direct verification of Q.

(Note that unless $P\ true$, it need not be evident that $Q\ prop$.) Now, if this judgement is going to be made evident, then we must indeed come up with what should count as a direct verification of $P \supset Q$ under the assumptions described above.

And so to have a direct verification of $P \supset Q$ is to have a verification of Q assuming that one has one of P; this is the meaning of implication, and it validates the judgement $P \supset Q\ true\ [Q\ true\ [P\ true]]$ (elliding the hypothesis for propositionhood), and may be written as an inference rule as follows:

$$\frac{Q\ true\ [P\ true]}{P \supset Q\ true}$$

3. Judgements for Verifications

So far, we have given judgements which circumscribe what it means to be a proposition, and thence for each proposition, we have by definition a notion of what should count as a verification of that proposition. And by definition, to assert the judgement $P\ true$ is to assert that one has a verification of P, but we have not considered any judgements which actually describe such verifications.

It is a hallmark of Martin-Löf's program to resolve the contradiction between syntax and semantics not by choosing symbols over meanings or meanings over symbols, but by endowing symbols with meaning in the course of making evident the true judgements. As such P is a symbol, but when we assert $P\ prop$ we are saying that we know what proposition P denotes.

A similar thing can be done with verifications themselves, by placing them in the syntactic domain together with the propositional symbols. And then, we can consider a judgement such as "M is a verification of P", and in knowing that judgement, we know what verification M is meant to denote. In practice, this

judgement has been written in several ways:

$M \in P$	M is an element of P
$M \Vdash P$	M realizes P
$P \llcorner \mathsf{ext}\ M \lrcorner$	P is witnessed by M

But they all mean the same thing, and so we will tentatively give the following meaning explanation to this new judgement:

> * To know $M \in P$ is to know that M is a verification of P.

But now that we have started to assign expressions to verifications, we must be more careful about differentiating *direct verifications* (which we will call "canonical"') from *indirect verifications* (which we will call "non-canonical"). So the domain of expressions must itself be accorded with a notion of reduction to canonical form, and this corresponds with putting into action a plan of verification in order to get an actual verification.

> To know $M \Rightarrow M'$ is to know that M is an expression which reduces to a canonical form M'.

Now, we can rewrite the previous meaning explanation as follows:

> To know $M \in P$ is to know an M' such that $M \Rightarrow M'$ and M' is a canonical (direct) verification of P.

The meaning explanation for $P\ \mathit{prop}$ must be accordingly modified to take into account the computational behavior of the expression domain:

> To know $P\ \mathit{prop}$ is to know a P' such that $P \Rightarrow P'$ and P' is a canonical proposition, which is to say, that one knows what counts as a canonical verification for P'.

As an example, then, the we will update the evidence of the assertion $P \supset Q\ \mathit{prop}\ [P\ \mathit{prop}, Q\ \mathit{prop}\ [P\ \mathit{prop}]]$. The meaning of this, expanded into spoken language, is as follows:

> To know $P \supset Q\ \mathit{prop}\ [P\ \mathit{prop}, Q\ \mathit{prop}\ [P\ \mathit{prop}]]$ is to know that what counts as a canonical verification of $P \subset Q$ under the circumstances that $P \Rightarrow P'$, such one knows what counts as a canonical verification P', and, if one has such a verification, $Q \Rightarrow Q'$ such that one knows what counts as a canonical verification of Q'.

And this is evident, since we will say that a canonical verification of $P \subset Q$ is an expression $\lambda x.E$ such that we know $E \in Q\ [x \in P]$. This validates the assertion $\lambda x.E \in P \supset Q\ [E \in Q\ [x \in P]]$, written as an inference rule:

$$\frac{E \in Q\ [x \in P]}{\lambda x.E \in P \supset Q}$$

By the addition of this judgement, we have graduated from a logical theory to a type theory, in the sense of *Constructive Mathematics and Computer Programming* (Martin-Löf, 1979). In fact, we may dispense with the original P *true* form of judgement by *defining* it in terms of the new $M \in P$ judgement as follows:

$$\frac{P \text{ } true}{M \in P}$$

Further forms of judgement, such as assertions of equality between propositions and verifications, may be added, as Martin-Löf does. However, this is not strictly necessary as they too may be defined in terms of the $M \in P$ judgement in the presence of an *equality* proposition; this is what is done in Constable et al's Computational Type Theory, which has only one primitive form of judgement.