Barnhusgatan 4
111 23 Stockholm
5 March 1976

Dear Michael,

Here are the notes that Peter Hancock wrote on the first four of the lectures that I gave at All Souls. I am well aware that the general philosophical introduction, covering the first ten pages, needs much further elaboration to be worth anything at all. But the ensuing description of the syntax is in a form which I feel I can stand for. It looks more horrible though than it need to since Peter shows parameters explicitly in all the schemes and rules which I did not in the seminar.

As you see, the notes do not cover the semantics of the language, so they may not be of much interest to you except as a preliminary. For the case that you will discuss it in your Harvard lectures, here are the principal semantical explanations.

The meaning of a sentence of the form

X is a type (proposition)

is the purpose (function, role) which an object (a canonical proof) of the type (proposition) X serves (fulfils, plays). Thus to understand (know the meaning of) a sentence of this form is to know the purpose of an object (a canonical proof) of the type (proposition) X, that is, to know what it is for, or briefly what it is. This, in turn, is the same as knowing what you must be able to do in order to understand a sentence of the form $x \in X$. Instead of speaking of the meaning of the complete sentence X is a type (proposition), it is sometimes convenient to speak of the meaning of X as a type (proposition) or simply the meaning of X, it being clear from the context that it is as a type (proposition) that it should be interpreted (understood)

This is our answer to the question

What is a type (proposition)?

Suppose now that X is a type. Then the meaning of a sentence of the form

$x \in X$ (x is an object of the type X, x is a canonical proof of the proposition X)

is the way in which x fulfils the purpose expressed by X. Thus to understand a sentence of this form is to know how (be able) to use x for the purpose expressed by X, or to fulfil this purpose by means of x. It is clear from this that the explanation of the meaning of a sentence of the form $x \in X$ must be preceded by the explanation of the meaning of the sentence X is a type. An object (a canonical proof) is a tool or instrument towards the fulfilment of a certain purpose, and to know this purpose is to know the type of the object (proposition that it proves). This is our answer to the question

What is an object (a canonical proof)?

The meaning of a sentence of the form

$\overline{A}$ is a type (A is a type-valued or propositional function)

is the way in which a type X is determined such that $\overline{A} = X$. Thus to understand (know the meaning of) a sentence of this form you must know how (be able) to determine a type X, that is, a type expression X for which you have understood the sentence X is a type, such that $\overline{A} = X$. When variables are present, the meaning of a sentence of the form

$\overline{A(u_1, \ldots, u_k)}$ is a type for $\overline{u}_1 \in \overline{A}_1, \ldots, \overline{u}_k \in \overline{A_k(u_1, \ldots, u_{k-1})}$

is the way in which a type X is determined such that

$\overline{A(u_1, \ldots, u_k)} = X$ for $\overline{u}_1 = x_1, \ldots, \overline{u}_k = x_k$, given objects x_1 of type $X_1 = \overline{A}_1, \ldots,$

of type $X_k = \overline{A(u_1, \ldots, u_{k-1})}$ for $\overline{u_1} = x_1, \ldots, \overline{u_k} = x_k$. This is our answer to the question

What is a type-valued (propositional) function?

Suppose that $\overline{A}$ is a type. Then the meaning of a sentence of the form

> $\overline{a} \in \overline{A}$ (a is an $\overline{A}$-valued function, a is a demonstration as opposed to canonical proof of A)

is the way in which an object x of the type $X = \overline{A}$ is determined such that $\overline{a} = x$. Thus to understand (know the meaning of) a sentence of this form you must know how (be able) to determine an object expression x for which you understand $x \in X$ where $\overline{A} = X$. Variables are handled as in the case of type-valued (propositional) functions. This is our answer to the question

What is a function (demonstration as opposed to canonical proof in your terminology)?

The meaning of a sentence of the form

$$\overline{A} = \overline{B}$$

is the way in which a type X is determined such that $\overline{A} = X = \overline{B}$. Thus to understand (know the meaning of) a sentence of this form you must know how (be able) to determine a type, that is, a type expression which you understand as a type, as the common value of A and B.

Finally, suppose that $\overline{A}$ is a type. Then the meaning of a sentence of the form

$$\overline{a} = \overline{b} \in \overline{A}$$

(which I now prefer to the unnecessarily cumbersome form $\overline{A} \ni \overline{a} = \overline{b} \in \overline{B}$) is the way in which an object x of the type $X = \overline{A}$ is 'etermined such that $\bar{a} = x = \bar{b}$. Thus to understand a sentence of this form you must know how (be able) to determine an object x of the type $X = \bar{A}$ (that is, an object expression x for which you know $x \in X$ where $\bar{A} = X$) such that $\bar{a} = x = \bar{b}$.

To understand (know) a language is to understand its rules, and to understand a rule is to know how the meaning of the conclusion depends on the meanings of the premises.

The rest of the meaning theory consists of explanations of (the meanings of) all the rules of the language. These explanations are fairly obvious except for the rules of type formation, where by a rule of type formation I mean a rule whose conclusion is of the form X is a type. So let me concentrate on a few of these rules.

According to the explanation of what a type is, to explain the meaning of the zero premise rule (axiom)

$$N \text{ is a type}$$

we must explain what an object of type N (a natural number) is, that is, what you must be able to do in order to understand a sentence of the form $x \in N$. The explanation is this. To understand a sentence of the form

$$x \in N$$

you must be able to determine an object y of type $Y = \overline{A(u)}$ for $\bar{u} = x$ such that $f(x) = y$ provided that you have already understood

$$\bar{a} \in \overline{A(0)}$$

$$\overline{b(u,v)} \in \overline{A(s(u))} \text{ for } \bar{u} \in \overline{N},\ \bar{v} \in \overline{A(u)}$$

and $f(u)$ is defined by recursion from a and $b(u,v)$. Thus our answer to the question

What is a natural number?

is that (the purpose of) a natural number is (to serve as) the principal argument of functions defined by recursion. Observe that the first two Peano axioms

$$0 \in N \qquad \frac{x \in N}{s(x) \in N}$$

play no role whatever in our explanation of what a natural number is. In particular, they do <u>not</u> stipulate what a natural number is. On the contrary, their meaning has to be explained, and this can be done only <u>after</u> it has been explained what a natural number is.

Let me also explain the rule of type (proposition) formation

> If $\overline{A}$ is a type and $\overline{B(u)}$ is a type for $\overline{u \in A}$, then $(\Pi u \in A)B(u)$ is a type

Because of the correspondence between propositions and types, this includes as particular cases our explanations of the meanings of the logical operations of implication and universal quantification. Now, to explain this rule, we must explain what an object (a canonical proof) of the type $(\Pi u \in A)B(u)$ (proposition $(\forall u \in A)B(u)$) is, that is, what you must be able to do in order to understand a sentence of the form $z \in (\Pi u \in A)B(u)$, provided $\overline{A}$ is a type and $\overline{B(u)}$ is a type (proposition) for $\overline{u} \in \overline{A}$. The answer is this. To understand a sentence of the form

$$z \in (\Pi u \in A)B(u)$$

you must be able to determine an object y of the type $Y = \overline{B(u)}$ for $\overline{u} = x$ such that $ap(x, z) = y$ when you are given an object x of the type $X = \overline{A}$. Again, note that it is <u>not</u> the rule of functional abstraction

$$\frac{\overline{b(u)} \in \overline{B(u)} \text{ for } \overline{u} \in \overline{A}}{(\lambda u)b(u) \in (\Pi u \in A)B(u)}$$

which determines the meaning of $(\Pi u \in A)B(u)$ ($A \to B$, $(\forall u \in A)B(u)$ and $A \supset B$).

On the contrary, this meaning has to be explained *before* the meaning of the rule of functional abstraction can be explained.

In particular, to explain the meaning of an implication $A \supset B$, we must explain what is the purpose (function, role) of a canonical proof of $A \supset B$. And, specializing the explanation given above, this purpose is to be applied to a canonical proof of the proposition denoted by A, thereby yielding a canonical proof of the proposition denoted by B. In no way is it correct to say that the meaning of $A \supset B$ is determined by the introduction rule

$$\frac{\begin{matrix}[A]\\ B\end{matrix}}{A \supset B}$$

The explanation of the meaning of universal quantification can be extracted in a similar way from the explanation of the meaning of $(\Pi x \in A)B(x)$ given above. And the meaning explanations for $(\Sigma u \in A)B(u)$, $A+B$, $I_A(x,y)$ and N_n (which include as special cases existential quantification, conjunction, disjunction, identity and absurdity) follow the same pattern as the explanations for N and $(\Pi u \in A)B(u)$. Thus, very briefly, the purpose (function, role) of an object (a canonical proof) of a certain type (proposition) is to serve as principal argument for the functions defined by the scheme associated with the type (proposition) in question. And it is this purpose which is its meaning.

Hope that this outline of the meaning theory will be of some use to you, and that you do not find life in U.S.A. too unbearable. Your kindness meant a lot to us while we were in Oxford.

Yours,

Per

P.S. If you have a draft of the first of your Harvard lectures that you could send me,