Robert Constable

Per Martin-Löf Informal Notes on Foundations (date uncertain) (estimate 1969)

About the simplest mathematical language for which a non-trivial theory of meaning can be formulated, is the language in which the only objects and functions dealt with are the natural numbers and the primitive recursive functions. Trivial as this language may seem, it is nevertheless rich enough to allow us to frame answers to such questions as

What is a natural number?

What is zero?

What is the successor of a natural number?

Moreover, the answers that we shall give to these questions will serve as paradigms for our answers to the more general questions

What is a type?

What is a mathematical object of a given type?

which we shall pose later for the theory of types.

Characteristic of many investigations of the concept of natural number, is that they attempt to define natural number in terms of other, more primitive, concepts. Thus Frege defined a number as an extension of a property, Russell as a class of similar classes, and von Neumann as a set in the cumulative hierarchy. According to the latter, now orthodox, definition, the natural numbers 0, 1, 2, ... are defined to be the sets $\emptyset$, $\{\emptyset\}$, $\{\emptyset, \{\emptyset\}\}$, ... However, these definitions only reduce the question what a natural number is, to the more refractory questions

What is an extension of a concept?

What is a class?

What is a set in the cumulative hierarchy?

For Dedekind, on the other hand, the natural numbers were the elements of a simply infinite system (that is, a system satisfying the Peano axioms) whose particular nature we have disregarded by our faculty of abstraction. And he believed he was able to prove the existence of infinite systems and therefore also of simply infinite ones. Of yet another kind is Wittgenstein's answer in the Tractatus: A number is the exponent of an operation. It contains in embryo the analysis that we shall give, not only for natural numbers, but for mathematical objects of any type.

The mathematical symbols that we shall use, will be divided into function constants

0, s, +, ·, .!, ...

and numerical variables

x, y, ...

The function constants are further divided into primitive and defined ones. Only 0 and s are primitive. All the rest are defined. An expression (meaningless, in general) is simply a string of symbols, like

0!, x·y+x, x!+s(0), ...

To talk about the language, as we shall do when describing its syntax and semantics, we also need syntactical variables. These

are

f, g, ...

which stand for function constants,

v, w, ...

which stand for numerical variables, and

a, b, ...

which stand for expressions. When we want to indicate that an expression may contain the variables v_1, ..., v_k and no others, we shall denote it by $a(v_1,\ldots,v_k)$. This allows us to denote the result of substituting the expressions a_1, ..., a_k for v_1, ..., v_k in $a(v_1,\ldots,v_k)$ by $a(a_1,\ldots,a_k)$. Linguistic expressions, considered in the ordinary way, are not mathematical objects. Only in metamathematics are they treated as such.* But, of course, when talking about expressions as metamathematical objects, we use linguistic expressions in the ordinary sense. They can never be dispensed with.** Likewise, the substitution operation, which takes a_1, ..., a_k and $a(v_1,\ldots,v_k)$ into $a(a_1,\ldots,a_k)$, is not a function in the mathematical sense. Only when the expressions are treated as metamathematical objects, does substitution become a function, which is defined by recursion on the expression as metamathematical object $a(v_1,\ldots,v_k)$.

The rules of the language of primitive recursive functions produce statements of the five forms

* RC: In computer science, e.g. studying programming languages, they are mathematical objects. So metamath and CS share methodology.

** RC: Brouwer thought they could be.

$f(x_1,\ldots,x_k,x) = y$ (y is the value of the recursively defined function f for the subordinate arguments $x_1, \ldots, x_k$ and the principal argument x)

$\bar{a} = x$ (a denotes x, or x is the value of a)

x is a natural number

a is a numerical function

$\bar{a} = \bar{b}$ or $a \equiv b$ (the functions a and b are definitionally equal, or have the same value)

The phrases following $f(x_1,\ldots,x_k,x) = y$, $\bar{a} = x$ and $\bar{a} = \bar{b}$ in parantheses only show how to read these statements in English. They must in no way be regarded as explanations of the meanings of these statements. As such, they would have to be understood already, and it would be absurd of us to assume this, because the task that we have set ourselves is precisely to explain the meanings of these statements, it being totally irrelevant whether they are expressed in mathematical notation or in English. However, since we shall formulate our theory of meaning in English, it will be convenient to have the option of expressing the statements whose meanings we are to explain in English as well.

When the expression occurring in the left hand member of a denotation statement $\overline{a(v_1,\ldots,v_k)} = x$ contains variables, it will always be derived from certain <u>assignments</u> $\bar{v}_1 = x_1, \ldots, \bar{v}_k = x_k$ of values to the variables. We shall indicate this by a figure of the form

$$\frac{\overline{v}_1 = x_1 \quad \ldots \quad \overline{v}_k = x_k}{\overline{a(v_1,\ldots,v_k)} = x}$$

and say that $\overline{a(v_1,\ldots,v_k)} = x$ <u>for</u> $\overline{v}_1 = x_1$, ..., $\overline{v}_k = x_k$, or, alternatively, that x is the value (denotation) of $a(v_1,\ldots,v_k)$ for the arguments x_1, ..., x_k.

<u>Rules of computation</u>. The recursion scheme stipulates that, given numerical functions $a(v_1,\ldots,v_k)$ and $b(v_1,\ldots,v_k,v,w)$, we may define a (k+1)-place function f by the computation rules

$$\frac{\overline{v}_1 = x_1 \quad \ldots \quad \overline{v}_k = x_k \qquad \overline{a(v_1,\ldots,v_k)} = x}{f(x_1,\ldots,x_k,0) = x}$$

$$\frac{f(x_1,\ldots,x_k,x) = y \qquad \overline{v}_1 = x_1 \quad \ldots \quad \overline{v}_k = x_k \quad \overline{v} = x \quad \overline{w} = y \qquad \overline{b(v_1,\ldots,v_k,v,w)} = z}{f(x_1,\ldots,x_k,s(x)) = z}$$

The symbol f is to be uniquely associated with the expressions $a(v_1,\ldots,v_k)$ and $b(v_1,\ldots,v_k,v,w)$, disregarding the choice of the variables that v_1, ..., v_k, v and w stand for (naturally, since a change of those variables does not affect the way in which f is computed). As already indicated, in a computation statement

$$f(x_1,\ldots,x_k,x) = y$$

x_1, ..., x_k are called the <u>subordinate</u> arguments, x the <u>principal</u> argument, and y the value of the function f for these arguments.

<u>Rules of denotation</u>. The computation rules refer back to statements of the form $\overline{a(v_1,\ldots,v_k)} = x$, and hence they do not

allow us to determine the value of a recursively defined function until the denotation rules have been given. They read

$\overline{v} = x$ (assignment)

$\overline{0} = 0$

$$\frac{\overline{a} = x}{\overline{s(a)} = s(x)}$$

$$\frac{\overline{a}_1 = x_1 \quad \ldots \quad \overline{a}_k = x_k \quad \overline{a} = x \quad f(x_1,\ldots,x_k,x) = y}{\overline{f(a_1,\ldots,a_k,a)} = y}$$

In the last rule, f is a (k+1)-place function defined by recursion.

First and second Peano axioms. When formulated as rules, they read

0 is a natural number

$$\frac{x \text{ is a natural number}}{s(x) \text{ is a natural number}}$$

Rules of function formation. These are the rules producing statements saying of an expression that it is a numerical function, namely

v is a numerical function

0 is a numerical function

$$\frac{a \text{ is a numerical function}}{s(a) \text{ is a numerical function}}$$

$$\frac{a_1, \ldots, a_k, a \text{ are numerical functions}}{f(a_1,\ldots,a_k,a) \text{ is a numerical function}}$$

Here, of course, v is a numerical variable, and f is a (k+1)-place function defined by recursion.

Rules of definitional equality. Using the notation $\overline{a} = \overline{b}$ rather than $a \equiv b$, they read

why want these computation rules?

$$\frac{a_1, \ldots, a_k \text{ are numerical functions}}{\overline{f(a_1,\ldots,a_k,0)} = \overline{a(a_1,\ldots,a_k)}}$$

$$\frac{a_1, \ldots, a_k, a \text{ are numerical functions}}{\overline{f(a_1,\ldots,a_k,s(a))} = \overline{b(a_1,\ldots,a_k,a,f(a_1,\ldots,a_k,a))}}$$

$$\frac{\overline{a}_1 = \overline{c}_1 \quad \ldots \quad \overline{a}_k = \overline{c}_k}{\overline{b(a_1,\ldots,a_k)} = \overline{b(c_1,\ldots,c_k)}}$$

$$\frac{a \text{ is a numerical function}}{\overline{a} = \overline{a}} \qquad \frac{\overline{a} = \overline{b}}{\overline{b} = \overline{a}} \qquad \frac{\overline{a} = \overline{b} \quad \overline{b} = \overline{c}}{\overline{a} = \overline{c}}$$

In the first two of these rules, f is a function defined by recursion from the numerical functions $a(v_1,\ldots,v_k)$ and $b(v_1,\ldots,v_k,v,w)$, and, in the third rule, $b(v_1,\ldots,v_k)$ is assumed to be a numerical function.

With this, the formulation of the syntax of the language of primitive recursive functions is complete. Note that each of its rules refers solely to the syntactical forms of the statements occurring as premises and conclusion. This is what makes it into a formal language. It remains for us to explain the meanings of the statements that can be derived by means of the formal rules (or, what amounts to the same, how they are understood) because understanding a language, even a formal one, is not merely to understand its rules as rules of symbol manipulation.* Believing

*RC: What might it mean for a "machine to understand."

this is the mistake of formalism. (hmm.)

To understand a language is to understand its rules, and to understand a rule is to understand the conclusion under the assumption that the premises have been understood.

The meaning of a computation statement $f(x_1,\ldots,x_k,x) = y$ is the way in which it is derived. Similarly, the meaning of a denotation statement $\overline{a(v_1,\ldots,v_k)} = x$ is the way in which it is derived from the assignments $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$. Thus, to understand a computation or denotation statement, we must know nothing beyond how it is derived. This means that, for the computation and denotation statements, the formalist interpretation is correct. For example, we understand that $0 + s(s(0)) = s(s(0))$ from the derivation

$$\dfrac{\dfrac{\dfrac{\overline{x} = 0}{0 + 0 = 0} \quad \dfrac{\overline{z} = 0}{\overline{s(z)} = s(0)}}{0 + s(0) = s(0)} \quad \dfrac{\overline{z} = s(0)}{\overline{s(z)} = s(s(0))}}{0 + s(s(0)) = s(s(0))}$$

Since to understand a rule is to understand the conclusion under the assumption that the premises have been understood, there is nothing to understand about the rules of computation and denotation. If we know how to derive the premises, we know of course how to derive the conclusion, namely, by applying the rule in question. This is the ground for calling them mere stipulations.

The meaning of a statement of the form

x is a natural number

is the way in which a natural number is determined as the value

of a recursively defined function for x as principal and natural numbers as subordinate arguments. Thus, to understand that x is a natural number, we must know how to determine a natural number y such that $f(x_1,\ldots,x_k,x) = y$ under the assumption that f is a (k+1)-place function defined by recursion and $x_1, \ldots, x_k$ are natural numbers. What determines the value of a recursively defined function for given arguments is the scheme of recursion, and hence we cannot understand a particular statement of the form that we are considering without looking back on this scheme. As is implicit in what we have just said, a natural number is an expression for which we have understood the statement above. Or,. as we may say, a natural number is an expression which we have understood (interpreted) as a natural number. This is our answer to the question

What is a natural number?

posed in the beginning. Its relation to Wittgenstein's answer: A natural number is the exponent of an operation, becomes clear if we reformulate it thus: A natural number is the principal argument of functions defined by recursion. Indeed, the iteration scheme

$$f^0(x) = x \qquad \frac{f^n(x) = y \quad f(y) = z}{f^{s(n)}(x) = z}$$

is nothing but a special form of the recursion scheme, the exponent n being the principal and x the subordinate argument of the binary function $f^n(x)$. So our formulation is obtained from Wittgenstein's by considering the recursion scheme in its general

* RC: I did not understand this idea until we understood LTT and extended it and implemented it. Now it seems deep.

form, rather than in the special form, called iteration. And this generalization is necessary since not every recursion (induction) has the form of a simple iteration. One can arrive at Wittgenstein's answer by making the futile attempt to define a natural number as what we get to from zero by iterating the successor operation a finite number of times. This is circular, because the definition uses the words finite number which are synonymous to the words natural number whose meaning it attempts to explain. There is no way out of this circle but to say that a natural number is what we iterate up to, and this is essentially Wittgenstein's explanation.

The meaning of a statement of the form

$a(v_1,\ldots,v_k)$ is a numerical function

is the way in which a natural number is determined as the value (denotation) of $a(v_1,\ldots,v_k)$ when natural numbers are assigned as values to the variables $v_1, \ldots, v_k$. Thus, to understand that $a(v_1,\ldots,v_k)$ is a numerical function, we must know, given natural numbers $x_1, \ldots, x_k$, how to determine a natural number x such that $\overline{a(v_1,\ldots,v_k)} = x$ for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$. What determines the value of $a(v_1,\ldots,v_k)$ for given arguments are the rules of denotation, and hence we cannot understand a statement of this form without looking back on these rules.

We can now also explain the meaning of a statement of the form

f is a (k+1)-place function defined by recursion

Its meaning, like that of any statement, is what we know when we

have understood it. And, to understand it, we merely have to understand that the expressions $a(v_1,\ldots,v_k)$ and $b(v_1,\ldots,v_k,v,w)$, which in a purely formal way are associated with the symbol f, are numerical functions.

Given these explanations, it is clear that what allows us to understand (determines the meaning of) the first Peano axiom

0 is a natural number

is the first clause

$$\frac{\overline{v}_1 = x_1 \quad \ldots \quad \overline{v}_k = x_k \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \overline{a(v_1,\ldots,v_k)} = x}{f(x_1,\ldots,x_k,0) = x}$$

of the recursion scheme. Indeed, that f is a (k+1)-place function defined by recursion means that the expressions $a(v_1,\ldots,v_k)$ and $b(v_1,\ldots,v_k,v,w)$ in terms of which it is defined are numerical functions. In particular, so is $a(v_1,\ldots,v_k)$. Hence, given natural numbers $x_1, \ldots, x_k$, we know how to determine a natural number x such that $\overline{a(v_1,\ldots,v_k)} = x$ for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$. The first clause of the recursion scheme stipulates that this number x is to be the value of f for the subordinate arguments $x_1, \ldots, x_k$ and the principal argument 0. And, of course, there is no other rule which allows us to derive a statement of the form $f(x_1,\ldots,x_k,0) = x$. This is the meaning of the first Peano axiom, that is, the way in which a natural number is determined as the value of a recursively defined function for the principal argument 0 and natural numbers as subordinate arguments. It is

also our answer to the question

What is zero?

because, as stated in the opening sentence of Frege's Grundlagen, the question what the number zero is, comes to the same as the question what the symbol 0 means

Our explanation of the meaning of the symbol 0 is typical of how substance can be given to Wittgenstein's slogan that meaning is use, or, more explicitly, that the meaning of an expression is determined by the rules that govern its use in the language of which it forms a part. It would be absurd to interpret that slogan as saying that meaning is conferred automatically upon the expressions of a language by its rules. There would then be no need for a theory of meaning. If we lay down any old set of rules, like those of set theory with the unrestricted (inconsistent) comprehension axiom or those of the type-free calculus of lambda-conversion, the expressions derivable by means of those rules allow in general no other than the formalist interpretation, according to which the meaning of an expression is the way in which it is derived. The difficulty in explaining the meaning of an expression of a language is to discern what are the rules of the language that determine its meaning. For instance, in the language of primitive recursive functions, the symbol 0 occurs explicitly, not only in the first clause of the recursion scheme, but also (twice!) in the denotation rule $\overline{0} = 0$, in the first Peano axiom, in one of the rules of function formation, and in one of the rules of definitional equality. It is by no means automatic that it is the first clause of the recursion

?

Re: How are these rules isolated from the ones that don't?

scheme and none of the other rules that determines the meaning of the symbol 0 as a natural number.

If somebody suggests that the function of the spark plugs in an internal-combustion engine is to be connected by cables via the distributor to the battery, we must explain to him how the engine works, thereby convincing him that the function of the spark plugs is to ignite the mixture of petrol and air which is sucked into the cylinders from the carburettor. It is not that it is wrong to say that the spark plugs are connected by cables via the distributor to the battery: they certainly are. But that is not what we should primarily pay our attention to in order to understand the function of the spark plugs in the running of the engine. Similarly, in arithmetic, if somebody suggests that the meaning of the symbol 0 is determined by the first Peano axiom, he is wrong, not because 0 is not a natural number: it certainly is, but because the first Peano axiom is not the rule from which the meaning of the symbol 0 is learnt. Instead, we must direct his attention to the first clause of the recursion scheme, because that is the rule which tells him how a natural number is determined as the value of a recursively defined function when 0 is inserted into its principal argument place together with natural numbers as subordinate arguments.

Recall that, to understand a rule, we must understand the conclusion under the assumption that the premises have been understood. Hence, to understand the second Peano axiom

$$\frac{x \text{ is a natural number}}{s(x) \text{ is a natural number}}$$

we must know how to evaluate a recursively defined function for the principal argument $s(x)$, given that we know how to evaluate it for the principal argument x. So let f be a $(k+1)$-place function defined by recursion and $x_1, \ldots, x_k$ natural numbers. By assumption, we know how to determine a natural number y such that $f(x_1,\ldots,x_k,x) = y$. Since f is a $(k+1)$-place function defined by recursion, the second of the expressions associated with the symbol f, call it $b(v_1,\ldots,v_k,v,w)$, is a numerical function. Hence, $x_1, \ldots, x_k$, x and y being natural numbers, we know how to determine a natural number z such that $\overline{b(v_1,\ldots,v_k,v,w)} = z$ for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$, $\overline{v} = x$ and $\overline{w} = y$. The second clause of the recursion scheme

$$\frac{f(x_1,\ldots,x_k,x) = y \qquad \overline{v}_1 = x_1 \quad \ldots \quad \overline{v}_k = x_k \quad \overline{v} = x \quad \overline{w} = y \quad \ldots\ldots\ldots\ldots \quad \overline{b(v_1,\ldots,v_k,v,w)} = z}{f(x_1,\ldots,x_k,s(x)) = z}$$

stipulates that this number z is to be the value of f for the subordinate arguments $x_1, \ldots, x_k$ and the principal argument $s(x)$. This is the meaning of the second Peano axiom, or, what amounts to the same, the meaning of the symbol s. Our explanation of the meaning of $s(x)$ in terms of the meaning of x, is at the same time our answer to the question

What is the successor of a natural number?

because, to ask for the meaning of the expression $s(x)$, is the same as to ask what $s(x)$ is.

It is of utmost importance to realize that the first two

Peano axioms do not stipulate what a natural number is. That would be to say that a natural number is an expression for which we have derived (instead of understood) the statement

x is a natural number

This is to confuse natural numbers with numerals, which is the formalist misinterpretation of the concept of natural number. What the first two Peano axioms show, is the form of a natural number. But an expression is a natural number, not in virtue of its form, but in virtue of its function. And this function is to serve as the principal argument of functions defined by recursion.

Having explained the meanings of the first two Peano axioms, we now turn to the rules of function formation. Remember that, to understand a statement of the form

$a(v_1,\ldots,v_k)$ is a numerical function

we must know, given natural numbers x_1, ..., x_k, how to determine a natural number x such that $\overline{a(v_1,\ldots,v_k)} = x$ for $\overline{v}_1 = x_1$, ..., $\overline{v}_k = x_k$. In particular, we understand the statement

v is a numerical function

for a numerical variable v, by looking back on the denotation rule which says that

$\overline{v} = x$

provided x is assigned as value to the variable v. It is this stipulation which allows us to interpret the symbol v as a numerical function. The statement

0 is a numerical function

is understood from the stipulation

$$\overline{0} = 0$$

and the first Peano axiom. Thus we cannot interpret the symbol 0 as a numerical function until <u>after</u> we have interpreted it as a natural number. The rule

$$\frac{a \text{ is a numerical function}}{s(a) \text{ is a numerical function}}$$

is understood from the stipulation

$$\frac{\overline{a} = x}{\overline{s(a)} = s(x)}$$

and the second Peano axiom. Suppose namely that a is a numerical function, that is, that we know how to determine a natural number x such that $\overline{a} = x$. Then s(x) is a natural number by the second Peano axiom, and the denotation rule above stipulates that $\overline{s(a)} = s(x)$. This is the meaning of the above rule of function formation, that is, the way in which a natural number (namely, s(x)) is determined as the value of s(a) under the assumption that a is a numerical function. Note that we can understand this rule only <u>after</u> we have understood the second Peano axiom. The final rule of function formation is

$$\frac{a_1, \ldots, a_k, a \text{ are numerical functions}}{f(a_1,\ldots,a_k,a) \text{ is a numerical function}}$$

where f is a (k+1)-place function defined by recursion. To understand it, we must know how to evaluate $f(a_1,\ldots,a_k,a)$ under the assumption that we know how to evaluate a_1, ..., a_k and a, that is, that we know how to determine natural numbers x_1, ..., x_k and x such that $\overline{a}_1 = x_1$, ..., $\overline{a}_k = x_k$ and $\overline{a} = x$. But this is precisely what the denotation rule

$$\frac{\overline{a}_1 = x_1 \quad \ldots \quad \overline{a}_k = x_k \quad \overline{a} = x \quad f(x_1,\ldots,x_k,x) = y}{\overline{f(a_1,\ldots,a_k,a)} = y}$$

tells us. Indeed, since f is a (k+1)-place function defined by recursion and the subordinate arguments x_1, ..., x_k are natural numbers, we know that a natural number is determined as its value whenever a natural number is inserted into its principal argument place. In particular, this is so for x, that is, we know how to determine a natural number y such that $f(x_1,\ldots,x_k,x) = y$. The denotation rule stipulates that this number y is to be the value (denotation) of $f(a_1,\ldots,a_k,a)$. This is how the meaning of the expression $f(a_1,\ldots,a_k,a)$ as a numerical function is determined in terms of the meanings of the expressions a_1, ..., a_k and a.

Composition of functions. If a_1, ..., a_k and $b(v_1,\ldots,v_k)$ are numerical functions, then so is $b(a_1,\ldots,a_k)$, and $\overline{b(a_1,\ldots,a_k)} = y$ provided $\overline{a}_1 = x_1$, ..., $\overline{a}_k = x_k$ and $\overline{b(v_1,\ldots,v_k)} = y$ for $\overline{v}_1 = x_1$, ..., $\overline{v}_k = x_k$.

This cannot be proved, it has to be understood. And what has to be understood is the way in which $b(a_1,\ldots,a_k)$ is evaluated, and that its value is the same as is obtained by first evaluating a_1, ..., a_k and then evaluating $b(v_1,\ldots,v_k)$ for these values as arguments. So suppose that a_1, ..., a_k are numerical

functions, that is, that we know how natural numbers $x_1, \ldots, x_k$ are determined as their values

$$\begin{array}{ccc} \vdots & \ldots & \vdots \\ \bar{a}_1 = x_1 & & \bar{a}_k = x_k \end{array}$$

Suppose further that $b(v_1,\ldots,v_k)$ is a numerical function. Then, $x_1, \ldots, x_k$ being natural numbers, we know how to determine a natural number y as the value of $b(v_1,\ldots,v_k)$ for these arguments

$$\begin{array}{c} \bar{v}_1 = x_1 \quad \ldots \quad \bar{v}_k = x_k \\ \hline b(v_1,\ldots,v_k) = y \end{array}$$

Replacing the variables $v_1, \ldots, v_k$ in this evaluation by the expressions $a_1, \ldots, a_k$ and attaching to it the evaluations of these

$$\begin{array}{c} \begin{array}{ccc} \vdots & & \vdots \\ \bar{a}_1 = x_1 & \ldots & \bar{a}_k = x_k \end{array} \\ \hline b(a_1,\ldots,a_k) = y \end{array}$$

we see that the natural number y is determined as the value of $b(a_1,\ldots,a_k)$. This is how we understand that $b(a_1,\ldots,a_k)$ is a numerical function.

The last part of the statement about composition of functions states that the value of a composite function is determined by the values of its components. Reading Bedeutung for value, we recognize this as one of the theses that was set forth by Frege in Über Sinn und Bedeutung. Remember that it is not something which can be proved, but must be understood. Hence the word

thesis rather than theorem.

The meaning of a definitional equality, that is, a statement of the form

$$\overline{a(v_1,\ldots,v_k)} = \overline{b(v_1,\ldots,v_k)}$$

is the way in which a natural number is determined as the common value of $a(v_1,\ldots,v_k)$ and $b(v_1,\ldots,v_k)$ when natural numbers are assigned as values to the variables $v_1, \ldots, v_k$. Thus, to understand a statement of this form, we must know, for arbitrarily given natural numbers $x_1, \ldots, x_k$, how to determine a natural number x such that

$$\overline{a(v_1,\ldots,v_k)} = x = \overline{b(v_1,\ldots,v_k)}$$

for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$. Alternatively, we might have said that two expressions are definitionally equal if they are both numerical functions, and, moreover, they take the same value for arbitrarily given natural numbers as arguments.

Consider now the first rule of definitional equality

$$\frac{a_1, \ldots, a_k \text{ are numerical functions}}{\overline{f(a_1,\ldots,a_k,0)} = \overline{a(a_1,\ldots,a_k)}}$$

To understand it, we must understand the conclusion under the assumption that the premises have been understood. So suppose that we know how to determine natural numbers $x_1, \ldots, x_k$ such that $\overline{a}_1 = x_1, \ldots, \overline{a}_k = x_k$. By composition of functions, $a(a_1,\ldots,a_k)$ is a numerical function, and $\overline{a(a_1,\ldots,a_k)} = x$ where x is the natural number such that $\overline{a(v_1,\ldots,v_k)} = x$ for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$. On the other hand, the first clause of

the recursion scheme and the denotation rules yield

$$\frac{\begin{array}{ccc}\vdots & & \vdots\end{array} \qquad \begin{array}{c}\overline{v}_1 = x_1 \quad \ldots \quad \overline{v}_k = x_k \\ \hline \overline{a(v_1,\ldots,v_k)} = x\end{array}}{\begin{array}{cc}\overline{a}_1 = x_1 \quad \ldots \quad \overline{a}_k = x_k & \overline{0} = 0 \qquad f(x_1,\ldots,x_k,0) = x\end{array}}$$
$$\overline{f(a_1,\ldots,a_k,0)} = x$$

This is how we determine the natural number x such that

$$\overline{f(a_1,\ldots,a_k,0)} = x = \overline{a(a_1,\ldots,a_k)}$$

The second rule of definitional equality

$$\frac{a_1, \ldots, a_k, \text{ a are numerical functions}}{\overline{f(a_1,\ldots,a_k,s(a))} = \overline{b(a_1,\ldots,a_k,a,f(a_1,\ldots,a_k,a))}}$$

is understood in a similar way. Assume that we know how to determine natural numbers $x_1, \ldots, x_k$ and x such that $\overline{a}_1 = x_1, \ldots, \overline{a}_k = x_k$ and $\overline{a} = x$. Then, since f is assumed to be a (k+1)-place function defined by recursion, we can determine, first, a natural number y such that $f(x_1,\ldots,x_k,x) = y$ and, second, a natural number z such that $\overline{b(v_1,\ldots,v_k,v,w)} = z$ for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$, $\overline{v} = x$ and $\overline{w} = y$. By the denotation rule

$$\frac{\overline{a}_1 = x_1 \quad \ldots \quad \overline{a}_k = x_k \quad \overline{a} = x \quad f(x_1,\ldots,x_k,x) = y}{\overline{f(a_1,\ldots,a_k,a)} = y}$$

and composition of functions,

$$\overline{b(a_1,\ldots,a_k,a,f(a_1,\ldots,a_k,a))} = z$$

On the other hand, this number z is determined as the value of

$$\dfrac{\begin{array}{ccc}\vdots & & \vdots \\ \bar{a}_1 = x_1 & \ldots & \bar{a}_k = x_k\end{array} \qquad \dfrac{\begin{array}{c}\vdots \\ \bar{a} = x\end{array}}{\overline{s(a)} = s(x)} \qquad \dfrac{\begin{array}{c}\vdots \\ f(x_1,\ldots,x_k,x) = y\end{array} \qquad \dfrac{\begin{array}{c}\bar{v}_1 = x_1 \quad \ldots \quad \bar{v}_k = x \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ b(v_1,\ldots,v_k,\end{array}}{f(x_1,\ldots,x_k,s(x)) = z}}{\overline{f(a_1,\ldots,a_k,s(a))} = z}$$

Fig. 1

$f(a_1,\ldots,a_k,s(a))$ by the second clause of the recursion scheme and the denotation rules, as shown in Fig. 1. This is how we understand the second rule of definitional equality. The third rule

$$\frac{\overline{a}_1 = \overline{c}_1 \quad \ldots \quad \overline{a}_k = \overline{c}_k}{\overline{b(a_1,\ldots,a_k)} = \overline{b(c_1,\ldots,c_k)}}$$

is the principle that <u>definitional equality is preserved under substitution</u>. Assume that its premises have been understood, that is, that we know how to determine natural numbers $x_1, \ldots, x_k$ such that

$$\overline{a}_1 = x_1 = \overline{c}_1 \quad \ldots \quad \overline{a}_k = x_k = \overline{c}_k$$

Since, by assumption, $b(v_1,\ldots,v_k)$ is a numerical function, we know how to determine a natural number y such that $\overline{b(v_1,\ldots,v_k)} = y$ for $\overline{v}_1 = x_1, \ldots, \overline{v}_k = x_k$. By composition of functions,

$$\overline{b(a_1,\ldots,a_k)} = y = \overline{b(c_1,\ldots,c_k)}$$

which is precisely what we had to see. Of the remaining rules of definitional equality.

$$\frac{a \text{ is a numerical function}}{\overline{a} = \overline{a}} \qquad \frac{\overline{a} = \overline{b}}{\overline{b} = \overline{a}} \qquad \frac{\overline{a} = \overline{b} \quad \overline{b} = \overline{c}}{\overline{a} = \overline{c}}$$

the reflexivity and symmetry are understood immediately, and the transitivity as follows. Suppose that we know how to determine natural numbers x and y such that $\overline{a} = x = \overline{b}$ and $\overline{b} = y = \overline{c}$. Then, since both x and y are <u>determined</u> as the value of b, they must be the same expressions. Therefore the same natural number x is

determined as the value of both a and c, which is precisely what we must know in order to understand that $\overline{a} = \overline{c}$. This finishes our explanations of the meanings of the rules of the language of primitive recursive functions.

Statements of the five forms that we have been considering cannot be proved, only understood. There is no question of their being true or false, like ordinary mathematical (or metamathematical) propositions. They can only be meaningful or meaningless. For example, the statement

0 is a natural number

is not true, but meaningful, and the statement

§ is a natural number

is not false, but meaningless, because there is no way of evaluating a recursively defined function for the principal argument §. Or, as we may say, we cannot interpret the symbol § as a natural number. It is meaningless (as a natural number).

When we have understood a language, that is, when we have understood its rules, we know that the statements which can be formally derived by means of those rules are meaningful.

For an ordinary mathematical proposition, there remains the question, even after it has been understood, that is, after it has been understood that it is a proposition, whether it is true or false. And this question can only be settled by proving or disproving (that is, proving the negation of) it. However, that a linguistic expression is a proof of a certain mathematical proposition is not something that we can again set about to

prove: it has to be understood. When we reach the words Q.E.D. at the end of a proof of a theorem, we are supposed to have understood that it is a proof of the theorem in question.

The distinction between statements that can be proved and those that can only be understood, is, essentially, Wittgenstein's distinction in the Tractatus between what can be said and what can only be shown. It is also closely related to his distinction between properties and relations proper (external properties and relations) and formal (internal) properties and relations, and, somewhat later, between concepts proper and formal concepts. A property proper is a propositional function which assigns to an object the proposition that the object has the property in question. This is an ordinary mathematical proposition, which we may try to prove or disprove. Being an even number, a prime number, the sum of two prime numbers, and so on, are all properties proper (of natural numbers). That something falls under a formal concept, on the other hand, cannot be proved: it has to be understood. Or, in Wittgenstein's words, it cannot be said: it shows itself. A formal concept cannot be represented by a propositional function. In the language of primitive recursive functions, the concepts of natural number and numerical function are both formal concepts. And the relation of definitional equality is a formal relation.

In the Tractatus, Wittgenstein apparently thought of a formal (internal) property as a property of objects. This explains why he said that a property is internal if it is unthinkable that its object should not possess it. If the statement

0 is a natural number

is construed as saying of the object 0 that it is a natural number, then we may indeed say that it is unthinkable that 0 were not a natural number, because we cannot perceive the object (understand the symbol) 0 without perceiving (understanding) it as a natural number. However, a formal property is <u>not</u> a property of objects. Only properties proper are properties of objects. A formal property is a property of expressions. Likewise, what falls under a formal concept is an expression, not an object. The first Peano axiom says of the symbol 0 that it is a natural number, that is, that it works as the principal argument of functions defined by recursion.

Instead of saying that we understand statements of the forms

x is a natural number

and

$a(v_1,\ldots,v_k)$ is a numerical function

we have sometimes said that we understand (interpret) the expression x as a natural number and the expression $a(v_1,\ldots,v_k)$ as a numerical function. There is no question of our understanding separately the expression x and the concept of natural number and then proving that x is a natural number: when we understand the expression x, we understand it <u>as a natural number</u>. So we may say that, in the statement that x is a natural number, the predicate is contained in the subject (provided, like Wittgenstein, we take the subject to be the object and not the expression x).

This is precisely the characteristic of a statement which is analytic in Kant's terminology. Similarly, we do not understand the expression $a(v_1,\ldots,v_k)$ separately from the concept of numerical function and then prove that it falls under this concept: we understand it as a numerical function. So the statement that $a(v_1,\ldots,v_k)$ is a numerical function is also analytic. On the other hand, all mathematical propositions in the ordinary sense, that is, propositions which we prove, are, according to Kant, synthetic (and a priori since their truth does not depend on experience). The distinction between statements that can be proved and those that can only be understood, may be regarded as a formulation of the distinction between synthetic and analytic judgements which is not limited to statements of subject-predicate form. Another such formulation (used by Quine, for example) is that a statement is analytic if it is true by virtue of its meaning. But that formulation is less fortunate, because an analytic judgement is a statement which has been understood, and for which there is no question of being true or false.

In terms of knowledge, the distinction is between knowledge of truth and knowledge of meaning. What we know when we have proved a statement is its truth, and what we know when we have understood a statement is its meaning. That a person knows the truth of a proposition manifests itself in his ability to prove it. In the terminology used by Dummett in his Bristol paper, knowledge of truth is verbalizable knowledge. Knowledge of the meaning of a linguistic expression, on the other hand, can only manifest itself in the ability to use the expression correctly.

Thus there is no one linguistic act, like that of proving a theorem, which conclusively shows that we know the meaning of a linguistic expression. In particular, it is not in our ability to explain its meaning that our understanding of a linguistic expression manifests itself. If that were the case, almost no mathematician could be said to understand the primitive notions of mathematics. In Dummett's words, knowledge of meaning is implicit knowledge.

Our understanding of the free variable statements

$$a(v_1,\ldots,v_k) \text{ is a numerical function}$$

and

$$\overline{a(v_1,\ldots,v_k)} = \overline{b(v_1,\ldots,v_k)}$$

is necessarily uniform in the arguments, because our knowledge how to determine a natural number x such that

$$\overline{a(v_1,\ldots,v_k)} = x$$

respectively

$$\overline{a(v_1,\ldots,v_k)} = x = \overline{b(v_1,\ldots,v_k)}$$

for $\bar{v}_1 = x_1, \ldots, \bar{v}_k = x_k$, comes solely from the knowledge that $x_1, \ldots, x_k$ are natural numbers. There is no question of proving statements of this kind by induction, that is, by separating cases according to the form of one of the arguments. That would be to treat them as metamathematical propositions proper. For example, we cannot understand the free variable statement

$\overline{0 + \bar{v}} = \bar{v}$

(as opposed to $\overline{\bar{v} + 0} = \bar{v}$) because we cannot see from the defining equations of the addition function and the denotation rules how to determine a natural number x such that $\overline{0 + \bar{v}} = x$ for $\bar{v} = x$. On the other hand, we can prove by induction as metamathematical theorems that $0 + x = x$ is derivable, and hence that $\overline{0 + \bar{v}} = x$ is derivable from $\bar{v} = x$, for all numerals x. But that is something entirely different from understanding that $\overline{0 + \bar{v}} = x$ for $\bar{v} = x$.

When the language of primitive recursive functions is treated metamathematically, the expressions become metamathematical objects of an inductively defined type, and the five different forms of statements that we have been considering are turned into inductively defined metamathematical propositions proper, that is, propositions which can be proved and combined by means of the logical connectives and quantifiers. In particular,

x is a natural number

is turned into a property proper, and

$\overline{a(v_1,\ldots,v_k)} = x$ for $\bar{v}_1 = x_1, \ldots, \bar{v}_k = x_k$

into a (k+1)-place relation proper between expressions as metamathematical objects. Therefore we can form the metamathematical proposition

$(\forall x_1)\ldots(\forall x_k)(x_1$ is a natural number & ...
& x_k is a natural number $\rightarrow (\exists x)(x$ is a natural number
& $\overline{a(v_1,\ldots,v_k)} = x$ for $\bar{v}_1 = x_1, \ldots, \bar{v}_k = x_k))$

in which the quantifiers range over expressions as metamathematical objects. It expresses that the open expression as metamathematical object $a(v_1,\ldots,v_k)$ is <u>convertible</u> in the sense of Tait, and must in no way be confused with the statement

$a(v_1,\ldots,v_k)$ is a numerical function

The former is a proper metamathematical proposition which we may try to prove, whereas the latter can only be understood. Now, it follows directly from our understanding of the language of primitive recursive functions and the fact that a natural number must have the form of a numeral, that, if the statements

x_1 is a natural number

$\vdots$

x_k is a natural number

and

$a(v_1,\ldots,v_k)$ is a numerical function

can be derived, then an expression x is determined for which

x is a natural number

and

$\overline{a(v_1,\ldots,v_k)} = x$

can be derived, in the latter case, from the assignments $\overline{v}_1 = x_1$, ..., $\overline{v}_k = x_k$. The metamathematical counterpart of this is the theorem, easily proved by Tait's method of convertibility, that,

if

$a(v_1,\ldots,v_k)$ is a numerical function

can be derived, then $a(v_1,\ldots,v_k)$ is convertible. Two things must be born in mind in connection with this theorem. First, in both the formulation and the proof of the theorem, we use linguistic expressions in the ordinary sense. Second, the proof of the theorem has to be understood. Otherwise, it would have no cognitive value. And understanding the metalanguage in which the proof of convertibility is carried out, is at least as difficult as understanding the object language, in this case, the language of primitive recursive functions, for which convertibility is proved. It would be absurd to maintain, at least in the case of the language of primitive recursive functions, that we do understand the metalanguage but not the object language, and hence that there is a genuine need for the proof of convertibility. But, of course, it may be interesting for other reasons.

Per was quite influenced by thoralf Skolem, a very constructive mathematician. His 1923 paper "the foundations of elementary arithmetic: established by means of the recursive mode of thought without the use of apparent variables ranging over infinite domains," 31 pages in From Frege to Gödel. Holland, Sweden, Russia, Germany, France, very influenced by constructivity – interto America with Church, Kleene, Bishop.

Key ideas

- definition of a natural number – p 9
- knowledge of truth vs knowledge of meanings, p. 26
- understanding rules – p 12-23, esp 12-15.
- sense vs truth – p. 23
- nature of metamathematics (hence computer science) p. 28
- basis of his semantic method, p. 30