

Implementing and repairing Frege in the style of Quine

M. Randall Holmes

August 5, 2018

In this document we will discuss the repair and implementation of the logic of Frege. This system is famously inconsistent, but as we shall see its mathematically essential features can be replicated and the entire mathematical program of Frege carried out by imposing restrictions on the formation of concepts analogous to the stratification criterion for comprehension axioms found in Quine's set theory New Foundations. New Foundations is not known to be consistent, but New Foundations with urelements (NFU) was shown to be consistent by Jensen. Details of the repairs to Frege's argument suggest use of Rosser's Axiom of Counting, an additional axiom introduced by Rosser for NF and shown by Jensen to be consistent with NFU (because Jensen showed that NFU has omega-models, and the Axiom of Counting obviously holds in an omega-model). Another significant repair is that Frege's definition of double value ranges and the ordered pair must be modified, as the original definitions were not stratified.

A further question which suggests itself is whether the stratification criteria for concept formation that we introduce can be justified in philosophical terms which Frege could have liked. We will address this point at the end of the paper.

We first introduce linear notation for the notions of Frege's logic to replace his interesting but difficult two dimensional notation, and state his axioms and rules of inference in terms of our modified notation.

We then present a semantics for this notation in terms of NFU with the Axiom of Counting (or a nonextensional λ -calculus equivalent to this theory), and verify that the axioms and rules of inference are correct over this semantics, with a restriction on the formation of concepts.

We then present a formulation of axioms and methods of reasoning as closely parallel to Frege's as possible, using a native notion of types of expressions to articulate the stratification restrictions on concept formation, and preserving in our notation, if not in the intended semantics, distinctions which Frege regarded as important.

Finally, we discuss machine implementation of the Grundgesetze using our theorem prover Marcel, which implements a version of NFU suitable for expressing Frege's theory.

We provide certain series of variables. $a, b, c \dots$ are free object variables. $x, y, z \dots$ are bound object variables (we use the same bound variables in both

quantified propositions and courses-of-values; Frege uses different series of variables, but there is no formal need to do this). $f, g, h \dots$ are free unary function variables (functions with one argument). $F, G, H \dots$ are bound function variables (functions with one argument). $r, s, t \dots$ are free binary function variables (functions with two arguments). $R, S, T \dots$ are bound binary function variables (of which I believe he makes no use). The special symbols ξ and ζ may be viewed as special bound object variables.

The first primitive of Frege's system is the notion of function. A function of one argument is an expression in which the variable ξ appears (so we will need to understand what the entire system of expressions is before we fully understand what functions there are. If f is a function and a is an expression, $f(a)$ denotes $f[a/\xi]$, the result of replacing ξ with a in f . There is some indication that Frege may want to have symbols ϕ and ψ representing functions analogous to ξ and ζ , though nothing in the main work requires this.

We will make our own modifications to this: we require that a function of one variable be enclosed in brackets, to indicate the scope, and so we write $[f](a) = f[a/\xi]$, for each function of one argument f . Notice that ξ must actually occur in any function $[f]$. Notice also that $[f][a/\xi]$ is simply $[f]$: a function name is a constant.

Frege also provides for functions of two variables. An expression $[f]$ is the name of a function of two variables if it contains both of the symbols ξ and ζ , and in this case $[f](a, b) = f[a/\xi, b/\zeta]$. We can tell effectively which of $[f](a)$ and $[f](a, b)$ is well-formed.

Frege makes no use of functions of three or more variables.

Frege distinguishes functions from objects. Numbers, courses-of-values, and propositions (!) are all objects.

He uses the notation $\neg x$ to mean the True if x is the True, and the False otherwise: \neg converts general objects to propositions. For Frege, sentences are names of truth values. We change this to $!x$.

We replace his notation for negation: $\neg x$ is the False where x is the True, and otherwise is the True. Note that $!x$ is definable as $\neg\neg x$: only negation is needed as a primitive.

$x = y$ refers to the True when x and y are the same object, and otherwise to the False.

We define $(\forall x : \phi)$ as the True if $\phi[a/x]$ is the True for every a , and the False otherwise. x must occur in ϕ . Moreover, bound variables are typographically distinct from free variables. We define $(\forall_2 F : \phi)$ as the True if $\phi[g/x]$ is the True for every function g of one argument, and the False otherwise. F must occur in ϕ . Moreover, bound function variables are typographically distinct from free function variables (they are capitalized). Bracketed function names can of course also be used to replace bound function variables. Frege does not quantify over functions of two arguments.

For each function $[f]$ we have a value-range $(\lambda x : f[x/\xi])$. A value-range is an object, and $(\lambda x : f[x/\xi]) = (\lambda x : g[x/\xi])$ iff $(\forall x : [f](x) = [g](x))$.

Frege provides that $!x = (\lambda y : y = !x)$, so the True is identified with the extension of the concept under which only the True falls, and the False is iden-

tified with the extension of the concept under which only the False falls. The advantage is that the truth values are then value-ranges.

Frege provides the operation θ such that if x is $(\lambda x : x = a)$, $\theta(x) = a$, and otherwise $\theta(x) = x$. We have to modify this when we correct the system to a stratified system, so we make this correction now: when x is not of the form $(\lambda y : y = a)$, $\theta(x)$ will be $(\lambda x : \neg x = x)$, in effect the empty set. The specific object used as the default does not matter.

We then introduce $x \rightarrow y$, which is the True unless x is the True and y is not the True, in which case it is the False. We provide that $x \rightarrow y \rightarrow z$ is read $x \rightarrow (y \rightarrow z)$ (generally allowing grouping to the right of implications as long as no parentheses are present).

When we write $\vdash p$, we assert that p is the True for any values we may assign to any free variables appearing in p .

Frege wants to say that the quantifier $(\forall x : \phi)$ and the course of values constructor $(\lambda x : \phi)$ are functions of a second order variable ϕ , thus second level functions (taking a second order argument (the function $[f[\xi/x]]$) to an object), and that $(\forall F : \phi)$ is a third level function, taking the second order function which might naughtily be written as $\phi[\phi/F]$ to an object in each case. We do not feel the need to assign referents to our variable binding operators that Frege feels, though we can in fact assign such referents in our semantics. Notice that the types which Frege assigns to the variable binding operators are not types of objects he ever talks about in general terms. There might seem to be some general consideration of functions from functions to propositions in his account of second-order quantification, but notice that we handle this in our account above entirely in terms of syntax.

We summarize the rules in our notation.

1. $!x$ may be replaced by $!x$ for any term x and vice versa. $\neg x$ may be replaced by $!\neg!x$ for any term x and vice versa. $x \rightarrow y$ may be replaced by $!(x \rightarrow !y)$ and vice versa, for any terms x and y . $(\forall x : \phi)$ may be replaced by $!(\forall x : !\phi)$ and vice versa, for any suitable term ϕ . $(\forall_2 F : \phi)$ may be replaced by $!(\forall_2 F : !\phi)$ and vice versa, for any suitable term ϕ .
2. From a finite sequence s of terms (with domain $[0, n]$), we define s^- as the sequence with domain $[0, n - 1]$ such that $s_n^- = s_{n+1}$, where $n \geq 1$. We define $\text{imp}(s)$ as the term s_0 if the domain of S is $[0, 0]$, and otherwise as $s_0 \rightarrow \text{imp}(s^-)$. Where i, j are in the domain of $[0, n]$ of s , we define $s(ij)$ as the sequence t such that $t_i = s_j$, $t_j = s_i$, and for all k with $k \in [0, n]$ and $k \notin \{i, j\}$ we have $s_k = t_k$. The term $\text{imp}(s(ij))$ can freely replace $\text{imp}(s)$ and vice versa if neither i nor j is the largest element of the domain of s . $\text{imp}(s(in))$, where the domain of s is $[0, n]$ and $i < n$, can freely replace and be replaced by $\text{imp}(c)$, where $c_i = \neg s_n$, $c_n = \neg s_i$, and $c_k = s_k$ for all $k \in [0, n]$ not equal to i or n . Note that all this sequence notation is metatheoretical.
3. $x \rightarrow x \rightarrow y$ can replace and be replaced by $x \rightarrow y$ (the more general form in which Frege states it would require a horrid variation of the awful thing

we did in the previous paragraph, and in any case is redundant).

4. Where s is a suitable sequence, the supercomponents of $\text{imp}(s)$ are s itself, $\text{imp}(s^-)$, and the supercomponents of $\text{imp}(s^-)$ if it has supercomponents. If all occurrences of a free variable a or f occur in a supercomponent ϕ of p , then ϕ may equivalently be replaced in p with $(\forall x : \phi[x/a])$ or $(\forall F : \phi[F/f])$ as appropriate, the variables x and F not occurring in p . This move may also be reversed (introducing a free variable not occurring in the initial assertion).
5. Where s is a suitable sequence, the subcomponents of $\text{imp}(s)$ are the terms s_i for i not maximum in the domain of s . If a first asserted proposition contains a subcomponent which also appears as an asserted proposition, then we can further assert a proposition which is obtained from the first proposition by suppressing that subcomponent. If s is a sequence with domain $[0, n]$ and $0 \leq i < n$, the proposition obtained from $\text{imp}(s)$ by suppressing the i th subcomponent is $\text{imp}(s^{-i})$, where s^{-i} has domain $[0, n-1]$ and has $s_k^{-i} = s_k$ for $k < i$ and $s_k^{-i} = s_{k+1}$ for $k \geq i$. This is a generalization of modus ponens.