1 Fregean Function Levels in Formal Languages

YAROSLAV KOKHAN
INSTITUTE OF PHILOSOPHY, NATIONAL ACADEMY OF SCIENCES OF UKRAINE, KYIV, UKRAINE
yarkaen@gmail.com

Modern predicate logic realizes in its formal languages Rasselian predicate orders but not Fregean function levels. Moreover, a predicate is just a partial case of a function, so predicate logic is not extremely wide and can be generalized to function logic. A function in this latter logic is no longer a map but a partial multimap, i. e., it can assign to any list of its arguments any number of values (including 0) and can have no arguments at all (i. e., can be a 0-ary function). If object $s$ is a value of function $t$, we write down this fact by representation formula

$$s \approx t,$$

where representation $\approx$ is a generalization of equality. For instance, if individual $a_0$ is a value of function $f$ at arguments $a_1, \ldots, a_n$, we write down this fact as

$$a_0 \approx f(a_1, \ldots, a_n).$$

If there are functions among values of function $t$, and $s$ is a value of one of such functions, then $s$ is a value of a value of $t$; we write down this fact as

$$s \approx (t);$$

hence, a value of a value of ... a value of $t$ ($n$ times) has been written as '$(\ldots(t)\ldots)' ($n-1$ pairs of parentheses around of '$t$'). This is concerned with values of functions; we sign the very function $f$ with arguments $a_1, \ldots, a_n$ (with argument places $x_1, \ldots, x_n$, with no argument) by

$$f^{\overline{a_1, \ldots, a_n}}$$

(by

$$f^{\overline{x_1, \ldots, x_n}}$$

and

$$f^{\lambda}$$

respectively). So we obtain tautology

$$s \approx t \rightarrow (t^{\lambda} \approx r \rightarrow s \approx (r)).$$

The above expressions belong to untyped function logic (an outline of first-order function logic see in [1]).

References