Yaroslav Kokhan

H. S. Skovoroda Institute of Philosophy of the NAS of Ukraine
Department of Logic and Science Methodology
Kyiv

FUNCTIONAL LOGIC as a generalization of PREDICATE LOGIC

FUNCTIONAL LOGIC as a generalization of PREDICATE LOGIC

Predicate logic is a central and main structural part of the whole logic

Actual and outdated structural parts of logic:

- ~~Syllogistics~~ [outated]
- Propositional logic
- ~~Logic of classes~~ [outated]
- Predicate logic

Alternatives to predicate logic:

- Leśniewski's Ontology
- Church's λ-theory and Curry's combinatory logic
- Uyomov's system of «properties, things, and relations»

FUNCTIONAL LOGIC as a generalization of PREDICATE LOGIC

Main and basic notions of predicate logic: a ***particular*** (an ***individual***), ***equality***, a ***sequence***, a ***predicate***.

- **Particulars** (**individuals**): atomic entities, which have no inner structure.

Notions: free individual variables a_i, b_i, c_i ($i \geqslant 0$).

- **Equality** and nonequality: logical relations.

The primitive theory of particulars and equality: atomic formulas are sequences of the form $v = u$; axioms:

$(=_1)$ $a = a$;

$(=_2)$ $a = b \longrightarrow b = a$;

$(=_3)$ $a = b \longrightarrow (b = c \longrightarrow a = c)$.

- **Sequences**: empty and finite progressions of any objects. Every well-ordered expression is a sequence.
- **Predicates**: two notions.

1. Predicates as truth functions: the predicate $F^{(n)}$, or $F(x_1, \ldots, x_n)$, has the type $\mathrm{A}_1 \times \ldots \times \mathrm{A}_n \mapsto \{\mathbf{t}, \mathbf{f}\}$, where $\mathrm{A}_i = \{x_i\}$, $\mathbf{t}$ = truth, $\mathbf{f}$ = falsehood.

2. Predicates as relations: the predicate $R^{(n)}$, or $R(x_1, \ldots, x_n)$, has the type $\mathrm{A}_1 \times \ldots \times \mathrm{A}_n$, where $\mathrm{A}_i = \{x_i\}$. This means that for every $R^{(n)}$ there exists a unique $F^{(n)}$ such that

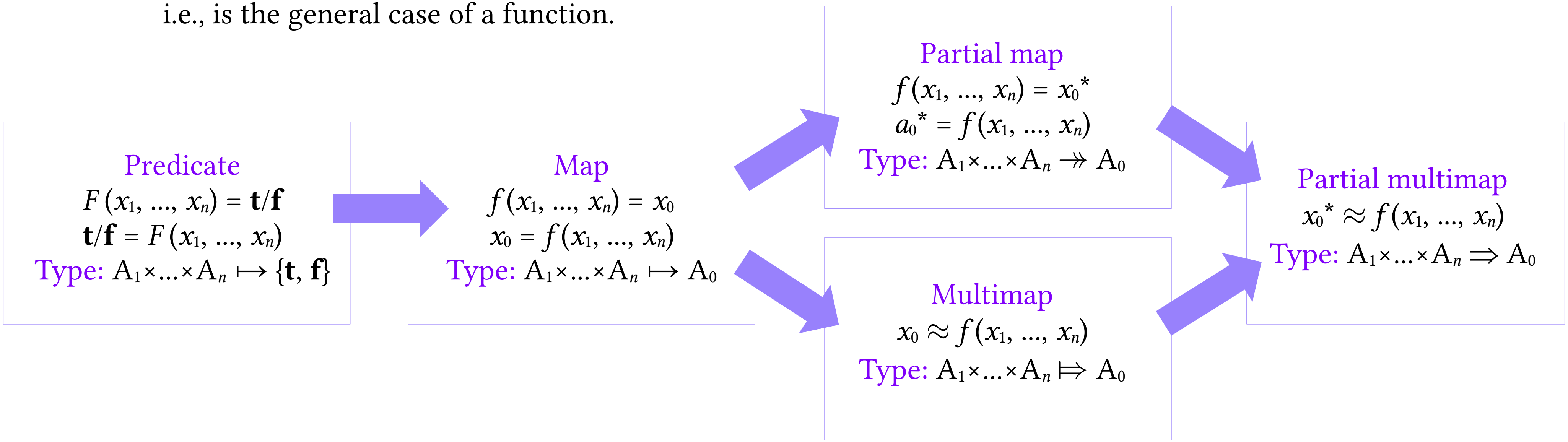
$$R(a_1, \ldots, a_n) \leftrightarrow F(a_1, \ldots, a_n) = \mathbf{t}$$

for every $a_i \in \mathrm{A}_i$.

FUNCTIONAL LOGIC as a generalization of PREDICATE LOGIC

Predicates as functions. Generalization of the notion of a predicate

1. • A ***predicate***, being a truth function, is a partial case of
2. • a ***map***, i.e., a one-valued function.
3. • A ***partial map*** is the case of an at most one-valued function.
   • A ***multimap*** is a many-valued function.
4. • A ***partial multimap*** is a non-negative-valued function,
   i.e., is the general case of a function.



Representation as the ultimate generalization of equality

Equality '$t = s$' presupposes that both equated objects t and s are unambiguously represented (specified). But in general, this is not the case.

Examples:

1. The usual inscription '$\pm 2 = \sqrt{4}$' is not correct because the expression '$\sqrt{4}$' is two-valued so is umbiguous and is not a term. The correct inscriptions: '$2 \approx \sqrt{4}$' and '$-2 \approx \sqrt{4}$'.

2. The correct specification of a circle with the center in the origin is $y \approx \sqrt{r^2 - x^2}$, but not $y = \sqrt{r^2 - x^2}$, because this "equation" has the single value only in two points: $x = -r$ and $x = r$, and has no value for $x < -r$ and $x > r$, but in the open interval $(-r; r)$ has two values y and $-y$ for every argument x.

So, if the object s is represented (specified, described) ambiguously, we can write only '$t \approx s$', but not '$t = s$'. In any case, the object t must be specified unambiguously.

Definition of equality and nonequality

There is a common opinion, that equality can be defined in predicate logic of the 2nd order by the *Leibniz criterion*: $a = b \leftrightarrow \forall P(P(a) \leftrightarrow P(b))$. But this "definition" is refuted by any model M = <M; Ω>, where $a, b \in$ M, $a \neq b$, and Ω contsists of the 1-ary predicates such that every of them either is satisfied by both a and b, or is not satisfied by both a and b.

In fact, equality and nonequality can be defined via representation as follows:

$$t = s \equiv_{\mathrm{Df}} t \approx s \wedge \forall r \forall o (r \approx s \longrightarrow (o \approx s \longrightarrow r \approx o)),$$

$$t \neq s \equiv_{\mathrm{Df}} t \not\approx s \wedge \forall r \forall o (r \approx s \longrightarrow (o \approx s \longrightarrow r \approx o)),$$

where t, r, o are *terms*, i.e., they are represented unambiguously, s is a *quasiterm*, i.e., it can be represented either ambiguously or unambiguously, and logical relation of non-representing $\not\approx$ is obviously defined as follows:

$$t \not\approx s \equiv_{\mathrm{Df}} \neg(t \approx s).$$

Hence equality and non-equality are not complements (negations) to each other, and are not, contrary to common opinion, simply identity and nonidentity. *Uniformity* $\eqcirc$, defined as follows:

$$t \eqcirc s \equiv_{\mathrm{Df}} t \approx s \wedge s \approx t$$

is also can be candidate to identity relation. Non-uniformity (non-identity?) is defined as follows:

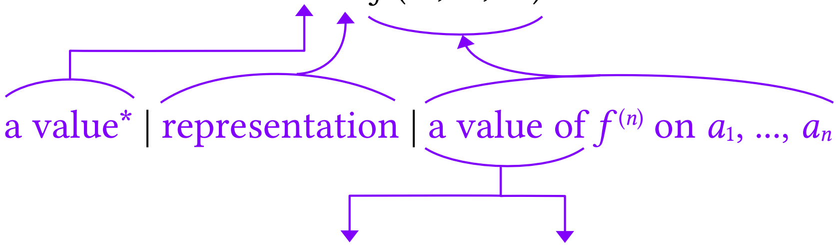
$$t \not\eqcirc s \equiv_{\mathrm{Df}} t \not\approx s \vee s \not\approx t.$$

Function and functional values. Churchian symbolism

Structure of representation formulas (in particular, equations and identity formulas):

«a_0 is a value of the function $f^{(n)}$ on arguments $a_1, \ldots, a_n$»

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



$$a_0 \approx (f^{(n)}(a_1, \ldots, a_n))$$

Hence, ‘$a \approx f^{(0)}$’ means: «a is the function $f^{(0)}$», but ‘$a \approx (f^{(0)})$’ means: «a is a value of the function $f^{(0)}$».

Parentheses to denote values of functions was introduced by Alonzo Church in:

Church, A. (1941). The Calculi of Lambda-Conversion. *Annals of Mathematics Studies, 6.* Princeton: Princeton University Press, London: Humphrey Milford Oxford University Press (reprinted, New York: Kraus Reprint Corp., 1965). P. 1.

Choice functions

If the function $f^{(n)}$ is ambiguous, the expression '$(f^{(n)}(a_1, \ldots, a_n))$' is ambiguous too for all $a_1, \ldots, a_n$. A *choice function* is every function such that it selects one of values of every given ambiguous constant function $f^{(n)}(a_1, \ldots, a_n)$, in particular $f^{(0)}$, if this constant function have values, and denotes nothing in the other case:

$$\boldsymbol{ch}_i^{(2)}(f^{(n)}(a_1, \ldots, a_n), j) \mapsto (f^{(n)}(a_1, \ldots, a_n))^j$$

some i^{th} choice function — choice index — a value* of $f^{(n)}(a_1, \ldots, a_n)$

For brevity, we can shorten '$(f^{(n)}(a_1, \ldots, a_n))$' to '$f(a_1, \ldots, a_n)$' and shorten '$(f^{(n)}(a_1, \ldots, a_n))^j$' to '$f^j(a_1, \ldots, a_n)$'.
We call terms of the form t^i, where i is a choice index, *marked*. The expression 't^i' means: «the i^{th} t-particular (t-individual), if it exists». We can quantify marked expressions in two following ways:

$$\forall f^2\,(f^2 \approx g) \rightarrow f^1 \approx g$$
$$\forall f^{1*}\,(f^{1*} \approx g) \rightarrow f^1 \approx g$$

FUNCTIONAL LOGIC as a generalization of PREDICATE LOGIC

Predicates as relations. Resolution of a relation

Every relation in any atomic sentence

$$R\,(a_1, \ldots, a_n)$$

has the single logically selected argument (and hence the selected argument place, or *valency*). In many natural languages, this selected argument is denoted by the grammar subject or the whole subject group. If we change the subject, the relation that the sentence is talking about will also change.

The relation without one its argument is no longer a relation:

$$a_1 \quad \cancel{R}(\ \ , \ldots, a_n)$$

but is the representation of the selected argument by some function (with or without arguments) as its value:

$$a_1 \ ? \ (f_R^{(n)}\,(a_2, \ldots, a_n)).$$

And since there can be more than one value of the function under question, then the connetction between every value a_1 and the function with arguments $f_R^{(n)}\,(a_2, \ldots, a_n)$ cannot be equality, but can be only representation:

$$a_1 \approx (f_R^{(n)}\,(a_2, \ldots, a_n)).$$

So, we can *resolve* any relation with respect to some its argument, obtaining the functional representation:

$$R\,(a_1, a_2, \ldots, a_n) \leftrightarrow a_1 \approx (f_R^{(n)}\,(a_2, \ldots, a_n)).$$

The structure of a relation. Two kinds of relations

We should divide relations on *logical* and *extralogical* (*descriptive*, *contensive*, *contentfull*).
Logical relations are representation, non-representation, and their partial cases: uniformity (identity), non-uniformity (non-identity), equality and nonequality.
The structure of a relation:

	logical relation	descriptive relation
general case:	$s_0 \odot < s_1, \ldots, s_n >$	$s_0 \odot (f^{(n)}(s_1, \ldots, s_n))$
…	…	…
partial case:	$s_0 \odot s_1$ / $s_0 \odot < s_1 >$	$s_0 \odot (f^{(1)}(s_1))$
limit case:	$s_0 \odot \star$	$s_0 \odot (f^{(0)})$

where $s_0, \ldots, s_n$ are terms, $\odot$ can be $\approx$, $\not\approx$, $=$, $\neq$, $\eqcirc$, $\not\eqcirc$, and '$\star$' is the *empty letter*, which points the absence of any object in its position. '$s_0 \approx \star$' means: «There is s_0», so can be treated as «s_0 exists». A sentences of the forms '$\star \approx s_0$' and '$\star \approx (s_0)$' are subjectless (like Czech «Prší» ≡ Ukrainian «Дощить»).
Note that '$f^{(n)} \approx g^{(m)}$' («the function $f^{(n)}$ is the function $g^{(m)}$») is a partial case of '$s_0 \odot s_1$', thus is a logical relation.

FUNCTIONAL LOGIC as a generalization of PREDICATE LOGIC

The difference between predicate and functional logic

Functional logic and predicate logic differ from each other linguistically, deductively, and semantically.

1. Formal languages.

Predicate logic: predicate variables, individual (particular) variables

predicate variables
individual (particular) variables
propositional variables
functional variables

Functional logic: individual (particular) variables, propositional variables, functional variables

The notion of a variable is sufficient for predicate logic, but functional logic requires
1) more general notions of a term and a quasiterm as a structured expressions and
2) the distinction between formulas and quasiformulas.

A *free variable* is a limit case of a *term*, a *bound variable* is a limit case of a *quasiterm*. A *term* is a formal equivalent of a name (denoting prase) and contains no bound variables, a *quasiterm* either contains bound variables or, in other case, is a term. The difference between formulas and quasiformulas are analogous.

a_i, $(f_i)^j$, $(f_i^{(n)}(a_1, ..., a_n))^j$, are quasiterms and terms,
x_i, $(f_i)^j$, $(f_i^{(n)}(a_1, ..., a_n))^j$, are quasiterms and terms.

The difference between predicate and functional logic

2. Deduction.

Axioms

0) Propositional axioms:

(1) $A \longrightarrow (B \longrightarrow A)$.
(2) $(A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow B) \longrightarrow (A \longrightarrow C))$.
(3) $(\neg A \longrightarrow \neg B) \longrightarrow ((\neg A \longrightarrow B) \longrightarrow A)$.

1) Core functional axioms for logics of the 1st order (in simplified symbolism):

(a) $\forall f^{1*} (f^{1*} \approx g) \longrightarrow f^{1} \approx g$.
(a$_{1\text{-}2}$) $\neg \forall g^{1*} (f^{1*} \not\approx g^{1*}) \leftrightarrow f^{1} \approx g$.
(b) $f^{1} \approx f^{1}$.
(c) $f^{1} \approx g^{1} \longrightarrow (f^{1} \approx h \longrightarrow g^{1} \approx h)$.
(c1) $f^{1} \approx g \longrightarrow (\forall g^{1*} (g^{1} \approx h) \longrightarrow f^{1} \approx h)$.
(d) $\forall h^{1*} (h^{1} \approx f \longrightarrow h^{1} \approx f^{1}) \longrightarrow (f^{1} \approx g \longrightarrow \forall f^{1*} (f^{1*} \approx g))$.

The difference between predicate and functional logic

2. Deduction.

Axioms

2) Axioms for descriptions (in simplified symbolism):

(e) $\forall f^{1*} (f^{1*} \approx g) \rightarrow f^{1}(f^{1} \approx h) \approx g.$

(e_1) $\neg \forall f^{1*} (f^{1*} \not\approx g) \leftrightarrow f^{1}(f^{1} \approx g) \approx g.$

(e_2) $f^{1}(f^{1} \approx g) \approx h \vee f^{1}(f^{1} \approx g) \not\approx h \rightarrow f^{1} \approx g.$

3) Axioms of functional levels (in simplified symbolism):

($a_{1\text{-}2}\lambda$) $\neg \forall g^{1*} (f^{1*} \not\approx (g^{1*})) \leftrightarrow f^{1} \approx (g).$

(λ) $f^{\lambda} \approx g^{\lambda} \rightarrow (h^{1*} \approx f \rightarrow h^{1*} \approx g).$

(λ_1) $f^{1} \approx g \rightarrow (g^{\lambda} \approx h \rightarrow f^{1} \approx (h)).$

(λ_2) $f^{1} \approx (h^{1}) \rightarrow (g^{\lambda} \approx h^{1} \rightarrow f^{1} \approx g).$

(λ_3) $f^{1} \approx h \rightarrow (g^{1} \approx h \rightarrow ((h^{1} \approx f \leftrightarrow h^{1} \approx g) \rightarrow f^{\lambda} \approx g^{\lambda})).$

Thanks for the attention!