FORMALIZED MATHEMATICS
Vol.1, No.4, September–October 1990
Université Catholique de Louvain

A Classical First Order Language

Czesław Byliński[1]
Warsaw University
Białystok

Summary. The aim is to construct a language for the classical predicate calculus. The language is defined as a subset of the language constructed in [8]. Well-formed formulas of this language are defined and some usual connectives and quantifiers of [8,1] are accordingly. We prove inductive and definitional schemes for formulas of our language. Substitution for individual variables in formulas of the introduced language is defined. This definition is borrowed from [7]. For such purpose some auxiliary notation and propositions are introduced.

MML Identifier: `CQC_LANG`.

The articles [10], [3], [4], [5], [9], [2], [8], [1], and [6] provide the notation and terminology for this paper. In the sequel i, j, k will denote natural numbers. One can prove the following proposition

(1) For every non-empty set D and for every finite sequence l of elements of D such that $k \in \mathrm{Seg}(\mathrm{len}\, l)$ holds $l(k) \in D$.

Let x, y, a, b be arbitrary. The functor $(x = y \to a, b)$ is defined as follows: $(x = y \to a, b) = a$ if $x = y$, $(x = y \to a, b) = b$, otherwise.

One can prove the following propositions:

(2) For arbitrary x, y, a, b such that $x = y$ holds $(x = y \to a, b) = a$.

(3) For arbitrary x, y, a, b such that $x \neq y$ holds $(x = y \to a, b) = b$.

Let x, y be arbitrary. The functor $x \dot{\longmapsto} y$ yields a function and is defined as follows:

$x \dot{\longmapsto} y = \{x\} \longmapsto y$.

One can prove the following three propositions:

(4) For arbitrary x, y holds $x \dot{\longmapsto} y = \{x\} \longmapsto y$.

(5) For arbitrary x, y holds $\mathrm{dom}(x \dot{\longmapsto} y) = \{x\}$ and $\mathrm{rng}(x \dot{\longmapsto} y) = \{y\}$.

(6) For arbitrary x, y holds $(x \dot{\longmapsto} y)(x) = y$.

[1]Supported by RPBP.III-24.C1

ISSN 0777–4028

For simplicity we follow the rules: x, y are bound variables, a is a free variable, p, q are elements of WFF, l, ll are finite sequences of elements of Var, and P is a predicate symbol. Let F be a function from WFF into WFF, and let us consider p. Then $F(p)$ is an element of WFF.

One can prove the following proposition

(7) For an arbitrary x holds $x \in \mathrm{Var}$ if and only if $x \in \mathrm{FixedVar}$ or $x \in \mathrm{FreeVar}$ or $x \in \mathrm{BoundVar}$.

A substitution is a partial function from FreeVar to Var.

In the sequel f will be a substitution. Let us consider l, f. The functor $l[f]$ yielding a finite sequence of elements of Var is defined as follows:

$\mathrm{len}(l[f]) = \mathrm{len}\, l$ and for every k such that $1 \leq k$ and $k \leq \mathrm{len}\, l$ holds if $l(k) \in \mathrm{dom}\, f$, then $(l[f])(k) = f(l(k))$ but if $l(k) \notin \mathrm{dom}\, f$, then $(l[f])(k) = l(k)$.

The following proposition is true

(9)[2] $ll = l[f]$ if and only if the following conditions are satisfied:

(i) $\mathrm{len}\, ll = \mathrm{len}\, l$,

(ii) for every k such that $1 \leq k$ and $k \leq \mathrm{len}\, l$ holds if $l(k) \in \mathrm{dom}\, f$, then $ll(k) = f(l(k))$ but if $l(k) \notin \mathrm{dom}\, f$, then $ll(k) = l(k)$.

Let us consider k, and let l be a list of variables of the length k, and let us consider f. Then $l[f]$ is a list of variables of the length k.

One can prove the following proposition

(10) $a \dot{\longmapsto} x$ is a substitution.

Let us consider a, x. Then $a \dot{\longmapsto} x$ is a substitution.

We now state the proposition

(11) If $f = a \dot{\longmapsto} x$ and $ll = l[f]$ and $1 \leq k$ and $k \leq \mathrm{len}\, l$, then if $l(k) = a$, then $ll(k) = x$ but if $l(k) \neq a$, then $ll(k) = l(k)$.

Let A be a non-empty subset of WFF. We see that it makes sense to consider the following mode for restricted scopes of arguments. Then all the objests of the mode element of A are a formula.

The non-empty subset $\mathrm{WFF}_{\mathrm{CQC}}$ of WFF is defined as follows:

$\mathrm{WFF}_{\mathrm{CQC}} = \{s : \mathrm{Fixed}\, s = \emptyset \wedge \mathrm{Free}\, s = \emptyset\}$.

The following propositions are true:

(12) $\mathrm{WFF}_{\mathrm{CQC}} = \{s : \mathrm{Fixed}\, s = \emptyset \wedge \mathrm{Free}\, s = \emptyset\}$.

(13) p is an element of $\mathrm{WFF}_{\mathrm{CQC}}$ if and only if $\mathrm{Fixed}\, p = \emptyset$ and $\mathrm{Free}\, p = \emptyset$.

Let us consider k. A list of variables of the length k is said to be a variables list of k if:

$\{\mathrm{it}(i) : 1 \leq i \wedge i \leq \mathrm{len}\, \mathrm{it}\} \subseteq \mathrm{BoundVar}$.

One can prove the following propositions:

(14) For every list of variables l of the length k holds l is a variables list of k if and only if $\{l(i) : 1 \leq i \wedge i \leq \mathrm{len}\, l\} \subseteq \mathrm{BoundVar}$.

[2]The proposition (8) became obvious.

(15) Let l be a list of variables of the length k. Then l is a variables list of k if and only if $\{l(i) : 1 \leq i \wedge i \leq \text{len}\, l \wedge l(i) \in \text{FreeVar}\} = \emptyset$ and $\{l(j) : 1 \leq j \wedge j \leq \text{len}\, l \wedge l(j) \in \text{FixedVar}\} = \emptyset$.

In the sequel r, s denote elements of WFF_{CQC}. Next we state two propositions:

(16) VERUM is an element of WFF_{CQC}.

(17) Let P be a k-ary predicate symbol. Let l be a list of variables of the length k. Then $P[l]$ is an element of WFF_{CQC} if and only if $\{l(i) : 1 \leq i \wedge i \leq \text{len}\, l \wedge l(i) \in \text{FreeVar}\} = \emptyset$ and $\{l(j) : 1 \leq j \wedge j \leq \text{len}\, l \wedge l(j) \in \text{FixedVar}\} = \emptyset$.

Let us consider k, and let P be a k-ary predicate symbol, and let l be a variables list of k. Then $P[l]$ is an element of WFF_{CQC}.

We now state two propositions:

(18) $\neg p$ is an element of WFF_{CQC} if and only if p is an element of WFF_{CQC}.

(19) $p \wedge q$ is an element of WFF_{CQC} if and only if p is an element of WFF_{CQC} and q is an element of WFF_{CQC}.

Let us note that it makes sense to consider the following constant. Then VERUM is an element of WFF_{CQC}. Let us consider r. Then $\neg r$ is an element of WFF_{CQC}. Let us consider s. Then $r \wedge s$ is an element of WFF_{CQC}.

One can prove the following three propositions:

(20) $r \Rightarrow s$ is an element of WFF_{CQC}.

(21) $r \vee s$ is an element of WFF_{CQC}.

(22) $r \Leftrightarrow s$ is an element of WFF_{CQC}.

Let us consider r, s. Then $r \Rightarrow s$ is an element of WFF_{CQC}. Then $r \vee s$ is an element of WFF_{CQC}. Then $r \Leftrightarrow s$ is an element of WFF_{CQC}.

We now state the proposition

(23) $\forall_x p$ is an element of WFF_{CQC} if and only if p is an element of WFF_{CQC}.

Let us consider x, r. Then $\forall_x r$ is an element of WFF_{CQC}.

We now state the proposition

(24) $\exists_x r$ is an element of WFF_{CQC}.

Let us consider x, r. Then $\exists_x r$ is an element of WFF_{CQC}.

Let D be a non-empty set, and let F be a function from WFF_{CQC} into D, and let us consider r. Then $F(r)$ is an element of D.

In this article we present several logical schemes. The scheme *CQC_Ind* concerns a unary predicate $\mathcal{P}$, and states that:

for every r holds $\mathcal{P}[r]$

provided the parameter satisfies the following condition:

- for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P holds $\mathcal{P}[\text{VERUM}]$ and $\mathcal{P}[P[l]]$ but if $\mathcal{P}[r]$, then $\mathcal{P}[\neg r]$ but if $\mathcal{P}[r]$ and $\mathcal{P}[s]$, then $\mathcal{P}[r \wedge s]$ but if $\mathcal{P}[r]$, then $\mathcal{P}[\forall_x r]$.

The scheme *CQC_Func_Ex* concerns a non-empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a ternary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, and a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$ and states that:

there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{B}$ and $F(P[l]) = \mathcal{F}(k, P, l)$ and $F(\neg r) = \mathcal{G}(r')$ and $F(r \wedge s) = \mathcal{H}(r', s')$ and $F(\forall_x r) = \mathcal{I}(x, r')$

for all values of the parameters.

The scheme *CQC_Func_Uniq* concerns a non-empty set $\mathcal{A}$, a function $\mathcal{B}$ from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$, a function $\mathcal{C}$ from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$, an element $\mathcal{D}$ of $\mathcal{A}$, a ternary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, and a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$ and states that:

$\mathcal{B} = \mathcal{C}$

provided the parameters satisfy the following conditions:

- Given r, s, x, k. Let l be a variables list of k. Let P be a k-ary predicate symbol. Let r', s' be elements of $\mathcal{A}$. Suppose $r' = \mathcal{B}(r)$ and $s' = \mathcal{B}(s)$. Then $\mathcal{B}(\mathrm{VERUM}) = \mathcal{D}$ and $\mathcal{B}(P[l]) = \mathcal{F}(k, P, l)$ and $\mathcal{B}(\neg r) = \mathcal{G}(r')$ and $\mathcal{B}(r \wedge s) = \mathcal{H}(r', s')$ and $\mathcal{B}(\forall_x r) = \mathcal{I}(x, r')$,
- Given r, s, x, k. Let l be a variables list of k. Let P be a k-ary predicate symbol. Let r', s' be elements of $\mathcal{A}$. Suppose $r' = \mathcal{C}(r)$ and $s' = \mathcal{C}(s)$. Then $\mathcal{C}(\mathrm{VERUM}) = \mathcal{D}$ and $\mathcal{C}(P[l]) = \mathcal{F}(k, P, l)$ and $\mathcal{C}(\neg r) = \mathcal{G}(r')$ and $\mathcal{C}(r \wedge s) = \mathcal{H}(r', s')$ and $\mathcal{C}(\forall_x r) = \mathcal{I}(x, r')$.

The scheme *CQC_Def_correctn* concerns a non-empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathrm{WFF}_{\mathrm{CQC}}$, an element $\mathcal{C}$ of $\mathcal{A}$, a ternary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, and a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$ and states that:

(i) there exists an element d of $\mathcal{A}$ and there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d = F(\mathcal{B})$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{C}$ and $F(P[l]) = \mathcal{F}(k, P, l)$ and $F(\neg r) = \mathcal{G}(r')$ and $F(r \wedge s) = \mathcal{H}(r', s')$ and $F(\forall_x r) = \mathcal{I}(x, r')$,

(ii) for all elements d_1, d_2 of $\mathcal{A}$ such that there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d_1 = F(\mathcal{B})$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{C}$ and $F(P[l]) = \mathcal{F}(k, P, l)$ and $F(\neg r) = \mathcal{G}(r')$ and $F(r \wedge s) = \mathcal{H}(r', s')$ and $F(\forall_x r) = \mathcal{I}(x, r')$ and there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d_2 = F(\mathcal{B})$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{C}$ and $F(P[l]) = \mathcal{F}(k, P, l)$ and $F(\neg r) = \mathcal{G}(r')$ and $F(r \wedge s) = \mathcal{H}(r', s')$ and $F(\forall_x r) = \mathcal{I}(x, r')$ holds $d_1 = d_2$

for all values of the parameters.

The scheme *CQC_Def_VERUM* concerns a non-empty set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a ternary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$, and a binary functor $\mathcal{J}$ yielding an element of $\mathcal{A}$ and states that:

$\mathcal{F}(\mathrm{VERUM}) = \mathcal{B}$

provided the parameters satisfy the following condition:

- Let p be an element of $\mathrm{WFF}_{\mathrm{CQC}}$. Let d be an element of $\mathcal{A}$. Then $d = \mathcal{F}(p)$ if and only if there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d = F(p)$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{B}$ and $F(P[l]) = \mathcal{G}(k, P, l)$ and $F(\neg r) = \mathcal{H}(r')$ and $F(r \wedge s) = \mathcal{I}(r', s')$ and $F(\forall_x r) = \mathcal{J}(x, r')$.

The scheme *CQC_Def_atomic* concerns a non-empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, a ternary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a natural number $\mathcal{C}$, a $\mathcal{C}$-ary predicate symbol $\mathcal{D}$, a variables list $\mathcal{E}$ of $\mathcal{C}$, a unary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$, and a binary functor $\mathcal{J}$ yielding an element of $\mathcal{A}$ and states that:

$\mathcal{F}(\mathcal{D}[\mathcal{E}]) = \mathcal{G}(\mathcal{C}, \mathcal{D}, \mathcal{E})$

provided the following requirement is met:

- Let p be an element of $\mathrm{WFF}_{\mathrm{CQC}}$. Let d be an element of $\mathcal{A}$. Then $d = \mathcal{F}(p)$ if and only if there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d = F(p)$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{B}$ and $F(P[l]) = \mathcal{G}(k, P, l)$ and $F(\neg r) = \mathcal{H}(r')$ and $F(r \wedge s) = \mathcal{I}(r', s')$ and $F(\forall_x r) = \mathcal{J}(x, r')$.

The scheme *CQC_Def_negative* deals with a non-empty set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a ternary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, an element $\mathcal{C}$ of $\mathrm{WFF}_{\mathrm{CQC}}$, a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$, and a binary functor $\mathcal{J}$ yielding an element of $\mathcal{A}$ and states that:

$\mathcal{F}(\neg\mathcal{C}) = \mathcal{H}(\mathcal{F}(\mathcal{C}))$

provided the parameters satisfy the following condition:

- Let p be an element of $\mathrm{WFF}_{\mathrm{CQC}}$. Let d be an element of $\mathcal{A}$. Then $d = \mathcal{F}(p)$ if and only if there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d = F(p)$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{B}$ and $F(P[l]) = \mathcal{G}(k, P, l)$ and $F(\neg r) = \mathcal{H}(r')$ and $F(r \wedge s) = \mathcal{I}(r', s')$ and $F(\forall_x r) = \mathcal{J}(x, r')$.

The scheme *QC_Def_conjuncti* concerns a non-empty set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a ternary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$, an element $\mathcal{C}$ of $\mathrm{WFF}_{\mathrm{CQC}}$, an element $\mathcal{D}$ of $\mathrm{WFF}_{\mathrm{CQC}}$, and a binary functor $\mathcal{J}$ yielding an element of $\mathcal{A}$ and states that:

$\mathcal{F}(\mathcal{C} \wedge \mathcal{D}) = \mathcal{I}(\mathcal{F}(\mathcal{C}), \mathcal{F}(\mathcal{D}))$

provided the following condition is satisfied:

- Let p be an element of $\mathrm{WFF}_{\mathrm{CQC}}$. Let d be an element of $\mathcal{A}$. Then $d = \mathcal{F}(p)$ if and only if there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d = F(p)$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{B}$ and $F(P[l]) = \mathcal{G}(k, P, l)$ and $F(\neg r) = \mathcal{H}(r')$ and $F(r \wedge s) = \mathcal{I}(r', s')$ and $F(\forall_x r) = \mathcal{J}(x, r')$.

The scheme *QC_Def_universal* concerns a non-empty set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a ternary functor $\mathcal{G}$ yielding an element of $\mathcal{A}$, a unary functor $\mathcal{H}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{I}$ yielding an element of $\mathcal{A}$, a binary functor $\mathcal{J}$ yielding an element of $\mathcal{A}$, a bound variable $\mathcal{C}$, and an element $\mathcal{D}$ of $\mathrm{WFF}_{\mathrm{CQC}}$ and states that:

$\mathcal{F}(\forall_{\mathcal{C}} \mathcal{D}) = \mathcal{J}(\mathcal{C}, \mathcal{F}(\mathcal{D}))$

provided the following condition is satisfied:

- Let p be an element of $\mathrm{WFF}_{\mathrm{CQC}}$. Let d be an element of $\mathcal{A}$. Then $d = \mathcal{F}(p)$ if and only if there exists a function F from $\mathrm{WFF}_{\mathrm{CQC}}$ into $\mathcal{A}$ such that $d = F(p)$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P and for all elements r', s' of $\mathcal{A}$ such that $r' = F(r)$ and $s' = F(s)$ holds $F(\mathrm{VERUM}) = \mathcal{B}$ and $F(P[l]) = \mathcal{G}(k, P, l)$ and $F(\neg r) = \mathcal{H}(r')$ and $F(r \wedge s) = \mathcal{I}(r', s')$ and $F(\forall_x r) = \mathcal{J}(x, r')$.

We now state the proposition

(25) If $\mathrm{Arity}(P) = \mathrm{len}\, l$, then $P[l] = \langle P \rangle \frown l$.

Let us consider x, y, p, q. Then $(x = y \to p, q)$ is an element of WFF.

Let us consider p, x. The functor $p(x)$ yields an element of WFF and is defined as follows:

there exists a function F from WFF into WFF such that $p(x) = F(p)$ and for every q holds $F(\mathrm{VERUM}) = \mathrm{VERUM}$ but if q is atomic, then $F(q) = \mathrm{PredSym}(q)[\mathrm{Args}(q)[\mathbf{a}_0 \dot{\longmapsto} x]]$ but if q is negative, then $F(q) = \neg(F(\mathrm{Arg}(q)))$ but if q is conjunctive, then $F(q) = (F(\mathrm{LeftArg}(q))) \wedge (F(\mathrm{RightArg}(q)))$ but if q is universal, then $F(q) = (\mathrm{Bound}(q) = x \to q, \forall_{\mathrm{Bound}(q)}(F(\mathrm{Scope}(q))))$.

We now state a number of propositions:

(27)[3] Let r be an element of WFF. Then $r = p(x)$ if and only if there exists a function F from WFF into WFF such that $r = F(p)$ and for every q holds $F(\mathrm{VERUM}) = \mathrm{VERUM}$ but if q is atomic, then $F(q) = \mathrm{PredSym}(q)[\mathrm{Args}(q)[\mathbf{a}_0 \dot{\longmapsto} x]]$ but if q is negative, then $F(q) = \neg(F(\mathrm{Arg}(q)))$ but if q is conjunctive, then $F(q) = (F(\mathrm{LeftArg}(q))) \wedge (F(\mathrm{RightArg}(q)))$ but if q is universal, then

[3]The proposition (26) became obvious.

$F(q) = (\text{Bound}(q) = x \rightarrow q, \forall_{\text{Bound}(q)}(F(\text{Scope}(q))))$.

(28) $\text{VERUM}(x) = \text{VERUM}$.

(29) If p is atomic, then $p(x) = \text{PredSym}(p)[\text{Args}(p)[\mathbf{a}_0 \overset{\cdot}{\longmapsto} x]]$.

(30) For every k-ary predicate symbol P and for every list of variables l of the length k holds $(P[l])(x) = P[l[\mathbf{a}_0 \overset{\cdot}{\longmapsto} x]]$.

(31) If p is negative, then $p(x) = \neg(\text{Arg}(p)(x))$.

(32) $\neg p(x) = \neg(p(x))$.

(33) If p is conjunctive, then $p(x) = (\text{LeftArg}(p)(x)) \wedge (\text{RightArg}(p)(x))$.

(34) $p \wedge q(x) = (p(x)) \wedge (q(x))$.

(35) If p is universal and $\text{Bound}(p) = x$, then $p(x) = p$.

(36) If p is universal and $\text{Bound}(p) \neq x$, then $p(x) = \forall_{\text{Bound}(p)}(\text{Scope}(p)(x))$.

(37) $\forall_x p(x) = \forall_x p$.

(38) If $x \neq y$, then $\forall_x p(y) = \forall_x (p(y))$.

(39) If $\text{Free}\, p = \emptyset$, then $p(x) = p$.

(40) $r(x) = r$.

(41) $\text{Fixed}(p(x)) = \text{Fixed}\, p$.

References

[1] Grzegorz Bancerek. Connectives and subformulae of the first order language. *Formalized Mathematics*, 1(**3**):451–458, 1990.

[2] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.

[3] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.

[4] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.

[5] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

[6] Czesław Byliński and Grzegorz Bancerek. Variables in formulae of the first order language. *Formalized Mathematics*, 1(**3**):459–469, 1990.

[7] Witold A. Pogorzelski and Tadeusz Prucnal. The substitution rule for predicate letters in the first-order predicate calculus. *Reports on Mathematical Logic*, (5), 1975.

[8] Piotr Rudnicki and Andrzej Trybulec. A first order language. *Formalized Mathematics*, 1(**2**):303–311, 1990.

[9] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

[10] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.

Received May 11, 1990